Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes[☆]

Jooyoung Hahn [a,*], Karol Mikula [b], Peter Frolkovič [b], Matej Medl'a [b], Branislav Basara [a]

[a] *Advanced Simulation Technologies, AVL List GmbH, Hans-List Platz 1, 8010 Graz, Austria*
[b] *Department of Mathematics and Descriptive Geometry, Slovak University of Technology, Radlinskeho 11, 810 05 Bratislava, Slovakia*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Level set method
Finite volume method
Polyhedron cell

ABSTRACT

In this paper, we propose a cell-centered finite volume method for advective and normal flows on polyhedron meshes which is second-order accurate in space and time for smooth solutions. In order to overcome a time restriction caused by CFL condition, an implicit time discretization of inflow fluxes and an explicit time discretization of outflow fluxes are used in an iterative procedure. For an efficient computation, an 1-ring face neighborhood structure is introduced. Since it is limited to access unknown variables in an 1-ring face neighborhood structure, an iterative procedure is proposed to resolve the limitation of assembled linear system. Two types of gradient approximations, an inflow-based gradient and an average-based gradient, are studied and compared from the point of numerical accuracy. Numerical schemes are tested for an advective and a normal flow of level-set functions illustrating a behavior of the proposed method for an implicit tracking of a smooth and a piecewise smooth interface.

1. Introduction

Level-set methods are widely used as practical numerical tools when a tracking of a dynamic interface is required in engineering applications [1,2]. Such requirements also occur in particular applications based on flow dynamics of several phases or combustion models in engines [3,4], for which several software tools and libraries are developed and used by a large community of researchers and engineers. To use these tools and libraries with a wide range of computational domains, flexible shapes of cell such as a polyhedron shape are preferable in cases of a complex computational boundary [5]. Moreover, one can find various applications of level set methods e.g. in pharmaceutical sciences [6], colloidal transport [7], biofilm formation [8], or geothermal energy exploitation [9].

An appropriate numerical discretization method to work with a polyhedron mesh is a finite volume method (FVM) designed for general shapes of computational cells. An attractive property of FVM is a direct approximation of local conservation property with the models of conservation laws. Therefore, many variations of FVM [10–13] are used with a polyhedron mesh for the problems of fluid dynamics or the combustion modeling which usually need a complicated domain in engineering applications.

[☆] The second and the fourth author were supported by VEGA 1/0808/15 and APVV-0522-15. The third author was supported by VEGA 1/0728/15 and APVV-0522-15.

[*] Corresponding author.

E-mail addresses: Jooyoung.Hahn@avl.com (J. Hahn), karol.mikula@stuba.sk (K. Mikula), peter.frolkovic@stuba.sk (P. Frolkovič), matej.medla@gmail.com (M. Medl'a), branislav.basara@avl.com (B. Basara).

https://doi.org/10.1016/j.camwa.2018.06.033
0898-1221/© 2018 Elsevier Ltd. All rights reserved.

All these factors give a reasonable motivation to develop a FVM also when the level-set equations on polyhedron meshes are necessary to be solved. The aim of this paper is to propose a FVM to find a numerical solution of level-set equations for an advective and a normal flow that is accurate and robust when used with a polyhedron mesh. In particular, we aim to implement it in the `AVL FIRE`®[1] that is used for many real world engineering applications in the fluid dynamics and combustion modeling.

A flux-based level-set method in [14] is available for an unstructured mesh, but it is based on a vertex-centered finite volume method in space and on a fully explicit time discretization, and it is proposed for simpler meshes used with finite element methods. In this paper we propose a semi-implicit cell-centered FVM on polyhedron meshes for an advective and a normal flow that is second-order accurate in space and time for smooth solutions.

For a normal flow, the authors in [15] suggest to use an inflow-based gradient (IBG) with the second order total variation diminishing Runge–Kutta method in an explicit time discretization. There are two drawbacks of the method in [15] that we aim to resolve in this paper. Firstly, the IBG approximation with the explicit time discretization in [15] seems to be suitable for the normal flow due to a fully upwinded form, but it has an insufficient accuracy with the semi-implicit method in the case of an advective flow. Secondly, the time discretization in [15] is fully explicit with a time step restricted by CFL condition in order to avoid instabilities in numerical solutions. The restriction of time step is very impractical in three types of meshes extensively used in real world engineering problems: i) locally adaptive meshes, ii) moving meshes with elongation or compression in a certain direction, and iii) non-uniform meshes with largely varying size of computational cells that occur often in applications with complex boundaries.

Well-known approaches to avoid the restriction of the time step are implicit or semi-implicit time discretization methods. In [16–19], an inflow-implicit outflow-explicit (IIOE) method for the discretization of an advective and a normal flow is proposed and used. The IIOE methods in [16–19] prove to work well for structured grids, but it has been an open issue how to implement them efficiently also for polyhedron meshes. In [20], the IBG IIOE method is derived for polyhedron meshes, and it is developed for normal flows.

In this paper, together with the IBG as in [15,20], we introduce an average-based gradient (ABG) into an iterative IIOE scheme and explain all necessary details of the implementation including a tessellation of nonplanar faces of computational cells. We compare the accuracy of ABG and IBG not only for a normal flow but also for an advective flow. Furthermore, we show an efficiency of an iterative IIOE method to overcome an 1-ring face neighborhood structure.

The paper is organized as follows. In Section 2, we start to briefly introduce the mathematical model. In Sections 2.2–2.5, we explain the steps of iterative IIOE method: the tessellation of nonplanar faces, the ABG and the IBG approximation, the iterative procedure, and a brief 1D study of the method. In Section 3, we present numerical experiments with some discussion. Finally in Section 4 we conclude the results.

2. Iterative IIOE finite volume method

2.1. Mathematical model

An advective and a normal flow equation is numerically solved in level-set method:

$$\frac{\partial}{\partial t}\phi(\mathbf{x},t)+\mathbf{u}(\mathbf{x},t,\phi(\mathbf{x},t))\cdot\nabla\phi(\mathbf{x},t)=0,\quad(\mathbf{x},t)\in\Omega\times[0,T],\tag{1}$$

where the initial and Dirichlet boundary conditions are given:

$$\begin{aligned}&\phi(\mathbf{x},0)=\phi_0(\mathbf{x}),\quad \mathbf{x}\in\Omega,\\&\phi(\mathbf{x},t)=\phi_b(\mathbf{x},t),\quad(\mathbf{x},t)\in\partial\Omega\times(0,T].\end{aligned}\tag{2}$$

A computational domain is $\Omega\subset\mathbb{R}^3$, T is the final time, and we deal with two forms of the velocity functions:

$$\mathbf{u}(\mathbf{x},t,\phi(\mathbf{x},t))=\mathbf{v}(\mathbf{x},t)\quad\text{or}\quad\mathbf{u}(\mathbf{x},t,\phi(\mathbf{x},t))=\pm\frac{\nabla\phi(\mathbf{x},t)}{|\nabla\phi(\mathbf{x},t)|}.\tag{3}$$

The first and second velocity in (3) give an advective flow and normal flow for an evolution of surface described by a zero level set of ϕ, respectively.

In the following subsections, a detailed explanation of proposed method is presented. Firstly, we introduce some notations to explain a numerical scheme on 3D polyhedron cell and explain how to tessellate a face of distorted cell into triangles. Secondly, we explain the average-based gradient and show later the differences with the inflow-based gradient [15] in 1D case. Thirdly, we propose an iterative IIOE method.

[1] https://www.avl.com/fire.

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

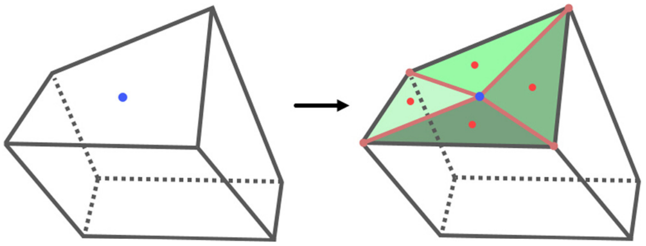

Fig. 1. When a face of cell is nonplanar, the face is tessellated into triangles; from a blue point on the left cell of nonplanar face, two consecutive vertices of the face are selected to construct one triangle. Note that a blue point is usually given by an area average face center (4). A red point on the right cell presents the center of triangle and it is denoted by $\mathbf{x}_f, f \in \mathcal{F}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.2. Distorted cell and subface tessellation

Let $\bar{\Omega} = \bigcup_{p\in\mathcal{I}} \bar{\Omega}_p \subset \mathbb{R}^3$ be a computational domain, where an open set Ω_p with the non-zero volume $|\Omega_p|$ is a cell in discretized domain and $\mathcal{I}$ is the set of cell indices. A distorted cell means that the cell has a nonplanar face, which usually exists when there is a complex computation boundary. It is an unrealistic assumption that all faces of Ω_p for all $p \in \mathcal{I}$ are of a planar shape in a case of industrial problems.

Whenever a face is not triangle, the face is tessellated into triangles; from a given center of the face, two consecutive vertices of the face are selected to construct one triangle. A center $\mathbf{x}^*$ of face whose vertices are $\mathbf{x}_i$, $i = 1, \dots, r$ is usually given by the area average. To make it simple, using a cyclic notation $\mathbf{x}_{r+1} = \mathbf{x}_1$, the area average face center is computed by

$$\mathbf{x}^* = \frac{\sum_{i=1}^{r} |\Delta_i|\, \bar{\mathbf{x}}_i}{\sum_{i=1}^{r} |\Delta_i|}, \tag{4}$$

where $|\Delta_i|$ is the area of Δ_i, $\bar{\mathbf{x}}_i$ is the center of mass on Δ_i, and Δ_i is a triangle of $\mathbf{x}_i$, $\mathbf{x}_{i+1}$, and $\mathbf{x}_0 = \frac{1}{r}\sum_{j=1}^{r} \mathbf{x}_j$ for $i = 1, \dots, r$. Then, the ith tessellated triangle at the face is defined by three points: $\mathbf{x}_i$, $\mathbf{x}_{i+1}$, and $\mathbf{x}^*$; see Fig. 1. The index set $\mathcal{F}$ denotes all triangles $e_f, f \in \mathcal{F}$ tessellated from a face or a triangle face of Ω_p for all $p \in \mathcal{I}$. We denote $\mathbf{x}_f$ as the center of triangle e_f.

To indicate the neighbor cells of Ω_p we consider only the cells whose face is shared:

$$\mathcal{N}_p = \{q \in \mathcal{I} : \text{there exists a face } e_f \subset \partial\Omega_q \cap \partial\Omega_p,\ f \in \mathcal{F}\}.$$

The faces of Ω_p are indicated by two sets:

$$\mathcal{F}_p = \{f \in \mathcal{F} :\ e_f \subset \partial\Omega_p\} \quad \text{and} \quad \mathcal{B}_p = \{f \in \mathcal{F}_p :\ e_f \subset \partial\Omega_p \cap \partial\Omega\}.$$

If Ω_p is a cell whose all faces are not overlapped to $\partial\Omega$, then we call the cell as an internal cell and $\mathcal{B}_p = \emptyset$. Otherwise, we call the cell as a boundary cell. In a similar way, an internal face $e_f, f \in \mathcal{F}$, means that the face is not overlapped to $\partial\Omega$ and a boundary face e_b, $b \in \mathcal{F}$, is a part of $\partial\Omega$. Throughout the rest of paper, the subscript b indicates the face index whose face is a boundary face. When a quantity is defined on a face e_f and it depends on a cell sharing the face, the cell index should be explicitly indicated at the first subscript and the face index at the second subscript. For instance, for a face $e_f, f \in \mathcal{F}_p$, the outward normal vector to the face is indicated by $\mathbf{n}_{pf}$. Note that we also use the length of the normal vector as the area of the face, that is, $|\mathbf{n}_{pf}| = |e_f|$. If a face e_f is an internal face for $f \in \mathcal{F}_p$, there exists a cell Ω_q, $q \in \mathcal{I}$, such that $e_f \subset \partial\Omega_p \cap \partial\Omega_q$. Then, clearly, $\mathbf{n}_{pf} = -\mathbf{n}_{qf}$. Whenever a directional vector is denoted, we use a notation $\mathbf{d}$ with relevant indices:

$$\mathbf{d}_{pq} = \mathbf{x}_q - \mathbf{x}_p, \quad p, q \in \mathcal{I}.$$

2.3. Average-based or inflow-based gradient

In this section, we focus on a spatial discretization of the proposed method and a time discretization is explained in the next section. From the governing equation (1) and using Gauss's theorem, we have

$$\int_{\Omega_p} \frac{\partial\phi}{\partial t} + \int_{\Omega_p} \nabla\cdot(\phi\mathbf{u}) - \int_{\Omega_p} \phi\nabla\cdot\mathbf{u} = 0,$$

and a spatial discretization is written

$$\int_{\Omega_p} \frac{\partial\phi}{\partial t} + \sum_{f\in\mathcal{F}_p} \phi_{pf} \int_{e_f} \mathbf{u}\cdot\mathbf{n}_{pf} - \sum_{f\in\mathcal{F}_p} \phi_p \int_{e_f} \mathbf{u}\cdot\mathbf{n}_{pf} = 0\,. \tag{5}$$

Let us denote the flux of velocity at a face e_f:

$$a_{pf} = \int_{e_f} \mathbf{u} \cdot \mathbf{n}_{pf} \simeq \mathbf{u}(\mathbf{x}_f, t, \phi(\mathbf{x}_f, t)) \cdot \mathbf{n}_{pf} \tag{6}$$

Since e_f is a triangle, $\mathbf{x}_f$ is the center of mass, and the flux is calculated by Gaussian quadrature of degree 1, the integration is exact for polynomials of degree 1. We define disjoint signed index sets

$$\begin{aligned} \mathcal{B}_p^- &= \{b \in \mathcal{B}_p \mid a_{pb} < 0\} \quad \text{and} \quad \mathcal{B}_p^+ = \mathcal{B}_p \setminus \mathcal{B}_p^-, \\ \mathcal{F}_p^- &= \{f \in \mathcal{F}_p \setminus \mathcal{B}_p \mid a_{pf} < 0\} \quad \text{and} \quad \mathcal{F}_p^+ = (\mathcal{F}_p \setminus \mathcal{B}_p) \setminus \mathcal{F}_p^-, \end{aligned} \tag{7}$$

and an inflow or outflow face e_f means that the sign of flux is negative or not, respectively. For instance, the face index of inflow face e_f attached to a cell Ω_p must be $f \in \mathcal{F}_p^- \cup \mathcal{B}_p^-$. Using the inflow and outflow faces, the spatial discretization of (5) is divided into the signed index sets:

$$\begin{aligned} \int_{\Omega_p} \frac{\partial \phi}{\partial t} &+ \sum_{f \in \mathcal{F}_p^-} \left(\phi_{pf} - \phi_p\right) a_{pf} + \sum_{f \in \mathcal{F}_p^+} \left(\phi_{pf} - \phi_p\right) a_{pf} \\ &+ \sum_{b \in \mathcal{B}_p^-} \left(\phi_{pb} - \phi_p\right) a_{pb} + \sum_{b \in \mathcal{B}_p^+} \left(\phi_{pb} - \phi_p\right) a_{pb} = 0. \end{aligned} \tag{8}$$

The rest of section explains how to approximate the value ϕ_{pf} or ϕ_{pb} in (8). Note that the first order upwind scheme is obtained by:

$$\phi_{pf} = \begin{cases} \phi_p, & \text{if } a_{pf} \geq 0, \\ \phi_q, & \text{if } a_{pf} < 0, \end{cases}$$

where there is $q \in \mathcal{N}_p$ such that $e_f \subset \partial\Omega_p \cap \Omega_q \neq \emptyset$, for $f \in \mathcal{F}$.

The authors in [15] use an IBG to approximate the value ϕ_{pf} or ϕ_{pb} for a normal flow and we would like to compare it with an ABG for an advective and a normal flow as explained below. The procedure to compute an IBG or ABG is explained in four steps.

Firstly, we compute a cell-centered gradient $\nabla_p\phi$ as a least square minimizer of a functional:

$$\nabla_p \phi \equiv \underset{\mathbf{y}}{\operatorname{argmin}} \sum_{\mathbf{x} \in \mathcal{S}_p} w_p(\mathbf{x}) \left| \mathbf{y} \cdot (\mathbf{x} - \mathbf{x}_p) - (\phi(\mathbf{x}) - \phi_p) \right|^2, \tag{9}$$

where a weight function is $w_p(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_p|^{-2}$ and a set of points $\mathcal{S}_p$ at the cell $p \in \mathcal{I}$ is defined by:

$$\mathcal{S}_p \equiv \begin{cases} \{\mathbf{x}_q \mid q \in \mathcal{N}_p\} & \text{if } \mathcal{B}_p = \emptyset, \\ \{\mathbf{x}_q \mid q \in \mathcal{N}_p\} \cup \{\mathbf{x}_b \mid b \in \mathcal{B}_p\} & \text{if } \mathcal{B}_p \neq \emptyset. \end{cases} \tag{10}$$

Note that $\nabla_p\phi$ at a boundary cell Ω_p uses a boundary value from Dirichlet condition if it is available, otherwise one can apply a linearly extended value on the boundary, see [15].

With a given Dirichlet boundary value ϕ_b, $b \in \mathcal{B}$, the explicit form of gradient in (9) can be obtained from

$$\mathbf{M}\nabla_p\phi = \sum_{q \in \mathcal{N}_p} \frac{\mathbf{d}_{pq}}{|\mathbf{d}_{pq}|^2} \left(\phi_q - \phi_p\right) + \sum_{b \in \mathcal{B}_p} \frac{\mathbf{d}_{pb}}{|\mathbf{d}_{pb}|^2} \left(\phi_b - \phi_p\right), \tag{11}$$

where a coefficient matrix is obtained by

$$\mathbf{M} = \sum_{q \in \mathcal{N}_p} \frac{\mathbf{d}_{pq} \otimes \mathbf{d}_{pq}}{|\mathbf{d}_{pq}|^2} + \sum_{b \in \mathcal{B}_p} \frac{\mathbf{d}_{pb} \otimes \mathbf{d}_{pb}}{|\mathbf{d}_{pb}|^2}.$$

Note that the matrix is always symmetric and invertible because $|\mathcal{N}_p \cup \mathcal{B}_p| \geq 4$ and a cell in 3D mesh is not flat. Since a multiplication of the inverse of matrix and vectors in the right-hand side of (11) can be pre-computed, computational cost of computing the gradient by the least square method (9) is very low.

Secondly, an internal vertex value $\phi(\mathbf{x}_v)$, $v \in \mathcal{V}$ where the set $\mathcal{V}$ denotes all vertices $\mathbf{x}_v$ of all faces but not on the boundary, is approximated by an inverse distance average from adjacent cells. For a fixed vertex index $v \in \mathcal{V}$, let us denote $\mathcal{N}_v$ as a subset of cell indices where a cell Ω_p contains the internal vertex $\mathbf{x}_v$:

$$\mathcal{N}_v \equiv \left\{p \in \mathcal{I} \mid \mathbf{x}_v \in \partial\Omega_p\right\}, \quad v \in \mathcal{V}.$$

Then the internal vertex value is approximated from the cell-centered values by Newton's approximation and the inverse distance average:

$$\phi(\mathbf{x}_v) = \frac{\sum_{p \in \mathcal{N}_v} \frac{1}{|\mathbf{d}_{pv}|}(\phi(\mathbf{x}_p) + \nabla\phi(\mathbf{x}_p) \cdot \mathbf{d}_{pv})}{\sum_{p \in \mathcal{N}_v} \frac{1}{|\mathbf{d}_{pv}|}}, \quad \mathbf{d}_{pv} = \mathbf{x}_v - \mathbf{x}_p,\ p \in \mathcal{N}_v,\ v \in \mathcal{V}.$$

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

Note that a boundary vertex value is directly calculated from Dirichlet boundary condition if it is available.

Thirdly, a face gradient is computed at the center of a face. For an internal face $e_f, f \in \mathcal{F}_p$, there exists $q \in \mathcal{I}$ such that $e_f \subset \partial\Omega_p \cap \partial\Omega_q$. Then, we define two tetrahedrons whose base is e_f and two opposite apices are $\mathbf{x}_p$ and $\mathbf{x}_q$. We denote all vertices of two tetrahedrons as $\mathcal{P}_f$. Then, a face gradient $\boldsymbol{\beta}_f$ is computed by an weighted minimization:

$$(\alpha_f, \boldsymbol{\beta}_f) = \operatorname*{argmin}_{(a_f, \mathbf{b}_f)\in\mathbb{R}^4} \sum_{\mathbf{x}\in\mathcal{P}_f} w_f(\mathbf{x}) \left| a_f + \mathbf{b}_f \cdot (\mathbf{x} - \mathbf{x}_f) - \phi(\mathbf{x}) \right|^2, \tag{12}$$

where the weight function is defined by $w_f(\mathbf{x}) = |\mathbf{x} - \mathbf{x}_f|^{-2}$. The explicit form of minimizer for (12) can be computed by a similar method to (11). Note that this construction can be seen as a generalization of diamond-cell strategy on a regular structured cubic mesh [16]. Furthermore we note that the face gradient is used to evaluate the fluxes of velocity a_{pf} in (6) in the case of normal flow.

Finally, we define an ABG or IBG as an inverse distance average of some face gradients:

$$(\text{ABG}): \quad \overline{\mathcal{D}}_p\phi = \frac{\sum_{f\in\mathcal{F}_p} \frac{1}{|\mathbf{d}_{pf}|}\boldsymbol{\beta}_f}{\sum_{f\in\mathcal{F}_p} \frac{1}{|\mathbf{d}_{pf}|}}. \tag{13}$$

$$(\text{IBG}): \quad \mathcal{D}_p^-\phi = \frac{\sum_{f\in\mathcal{F}_p^-\cup\mathcal{B}_p^-} \frac{1}{|\mathbf{d}_{pf}|}\boldsymbol{\beta}_f}{\sum_{f\in\mathcal{F}_p^-\cup\mathcal{B}_p^-} \frac{1}{|\mathbf{d}_{pf}|}} \quad \text{if } \mathcal{F}_p^- \cup \mathcal{B}_p^- \neq \emptyset, \tag{14}$$

where $\mathcal{D}_p^- = 0$ if $\mathcal{F}_p^- \cup \mathcal{B}_p^- = \emptyset$.

Now, we compute the face value ϕ_{pf} in (5) from the gradient $\mathcal{D}_p\phi = \overline{\mathcal{D}}_p\phi$ in (13) or $\mathcal{D}_p^-\phi$ in (14). When a face value is computed at an internal face, the value ϕ_{pf} in (5) is computed straightforwardly:

$$\begin{aligned} f \in \mathcal{F}_p \setminus \mathcal{B}_p,\ p \in \mathcal{I} \ &\Rightarrow\ \exists!\, q \in \mathcal{N}_p \ \text{ such that } \ e_f \subset \partial\Omega_p \cap \partial\Omega_q \\ &\Rightarrow\ \phi_{pf} = \begin{cases} \phi_p + \mathcal{D}_p\phi \cdot (\mathbf{x}_f - \mathbf{x}_p) & \text{if } a_{pf} \geq 0, \\ \phi_q + \mathcal{D}_q\phi \cdot (\mathbf{x}_f - \mathbf{x}_q) & \text{if } a_{pf} < 0. \end{cases} \end{aligned} \tag{15}$$

When a face value is computed at a boundary face, the value ϕ_{pf} in (5) is computed by

$$b \in \mathcal{B}_p (\neq \emptyset),\ \ p \in \mathcal{I} \ \Rightarrow\ \phi_{pb} = \begin{cases} \phi_p + \mathcal{D}_p^-\phi \cdot (\mathbf{x}_b - \mathbf{x}_p) & \text{if } a_{pb} \geq 0, \\ \phi_b & \text{if } a_{pb} < 0. \end{cases} \tag{16}$$

Note that we use the Dirichlet boundary condition on an inflow boundary. From (15) and (16), we finally obtain the spatial discretization:

$$\begin{aligned} \int_{\Omega_p} \partial_t\phi = &- \sum_{f\in\mathcal{F}_p^-} \left(\phi_q + \mathcal{D}_q\phi \cdot \mathbf{d}_{qf} - \phi_p\right) a_{pf} - \sum_{f\in\mathcal{F}_p^+} \left(\mathcal{D}_p\phi \cdot \mathbf{d}_{pf}\right) a_{pf} \\ &- \sum_{b\in\mathcal{B}_p^-} \left(\phi_b - \phi_p\right) a_{pb} - \sum_{b\in\mathcal{B}_p^+} \left(\mathcal{D}_p\phi \cdot \mathbf{d}_{pb}\right) a_{pb}, \end{aligned} \tag{17}$$

where the gradient $\mathcal{D}_p\phi = \overline{\mathcal{D}}_p\phi$ in (13) or $\mathcal{D}_p^-\phi$ in (14), $\mathbf{d}_{qf} = \mathbf{x}_f - \mathbf{x}_q$, and for each $f \in \mathcal{F}_p \setminus \mathcal{B}_p, p \in \mathcal{I}$ there exists an index $q \in \mathcal{N}_p$ such that $e_f \subset \partial\Omega_p \cap \partial\Omega_q$.

2.4. Iterative IIOE method

Let us denote an evenly divided time step $\Delta t = T/N$ for a fixed $N \in \mathbb{N}$ and $\phi_p^n = \phi(\mathbf{x}_p, n\Delta t), p \in \mathcal{I}$. An explicit time discretization is used in [15] for a normal flow equation and the time step is restricted by the CFL condition. In case of polyhedron cells with a complex computational boundary, it is very difficult to make an evenly distributed size of cells and it is inevitable to have a small size of cell somewhere in a computation domain. In case of adaptive mesh, a small size of cell is purposely generated in order to obtain a high accuracy. Moreover, for a moving mesh commonly used in a combustion engine simulation, a size of cell is elongated or compressed in a certain direction. Therefore, it is crucial in practice to design a scheme for such cases to overcome a restriction of time step.

Inspired by [16–19], a possible solution is to use an IIOE method, where the time discretization on an outflow and inflow face is treated explicitly and implicitly, respectively:

$$\begin{aligned} &\frac{|\Omega_p|}{\Delta t}\left(\phi_p^n - \phi_p^{n-1}\right) + \sum_{f\in\mathcal{F}_p^-} \left(\phi_q^n + \mathcal{D}_q\phi^n \cdot \mathbf{d}_{qf} - \phi_p^n\right) a_{pf}^{n-1} \\ &+ \sum_{b\in\mathcal{B}_p^-} \left(\phi_b^n - \phi_p^n\right) a_{pb}^{n-1} + \sum_{f\in\mathcal{B}_p^+\cup\mathcal{F}_p^+} \left(\mathcal{D}_p\phi^{n-1} \cdot \mathbf{d}_{pf}\right) a_{pf}^{n-1} = 0. \end{aligned} \tag{18}$$

Note that the nonlinearity in the fluxes a_{pf} of velocity in (6) is treated explicitly in (18).

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

The scheme (18) represents a linear system of algebraic equations. For practical reasons, the simplest overlapping domain decomposition is required under the 1-ring face neighborhood structure in order to simplify the parallel computations. Considering such requirement of the 1-ring face neighborhood structure, there is an obstacle to directly use (18). From a cell Ω_p, it is possible to access all neighbor cells, that is, Ω_q, $q \in \mathcal{N}_p$. However, in order to compute the gradient in the second term on the left-hand side in (18), we need a neighbor of the cell Ω_q which is usually placed at the 2-ring face neighborhood from Ω_p.

Since the 1-ring face neighborhood structure cannot provide the second face neighbor cell from Ω_p, we alternatively propose an iterative IIOE method to complete the time discretization of (17):

$$\begin{aligned} &\frac{|\Omega_p|}{\Delta t}\left(\phi_p^{n,k}-\phi_p^{n-1}\right)+\sum_{f\in\mathcal{F}_p^-}\left(\phi_q^{n,k}+\mathcal{D}_q\phi^{n,k-1}\cdot\mathbf{d}_{qf}-\phi_p^{n,k}\right)a_{pf}^{n-1} \\ &+\sum_{b\in\mathcal{B}_p^-}\left(\phi_b^n-\phi_p^{n,k}\right)a_{pb}^{n-1}+\sum_{f\in\mathcal{B}_p^+\cup\mathcal{F}_p^+}\left(\mathcal{D}_p\phi^{n-1}\cdot\mathbf{d}_{pf}\right)a_{pf}^{n-1}=0, \end{aligned} \tag{19}$$

where $k = 1, \ldots, K$ and $\phi^{n,0} = \phi^{n-1}$. Note that $\mathcal{D}_q\phi^{n,k-1}$ is computed by $\phi_p^{n,k-1}$, $p \in \mathcal{I}$ and ϕ_b^n, $b \in \mathcal{B}$.

A crucial advantage of using the iterative IIOE method (19) is that the matrix of linear system in (19) is an M-matrix and more sparse than the one in (18). A disadvantage is the necessity to use an iterative step which is not the case when solving (18) directly.

Rewriting (19) as a matrix equation

$$\mathbf{M}\phi^{n,k} = \mathbf{F}(\phi^{n-1}, \phi^{n,k-1}), \tag{20}$$

the kth iteration (19) stops at the smallest K_0 when the residual error is less than a threshold, for instance, 10^{-12}:

$$\frac{\sum_{p\in\mathcal{I}}\left|\left(\mathbf{M}\phi^{n,K_0}\right)_p-\mathbf{F}(\phi_p^{n-1},\phi_p^{n,K_0})\right|}{\sum_{p\in\mathcal{I}}|\mathbf{M}_{pp}|} < 10^{-12}, \tag{21}$$

where $\left(\mathbf{M}\phi^{n,K_0}\right)_p$ is the pth component in a multiplication of a matrix $\mathbf{M}$ and a vector ϕ^{n,K_0} and the denominator is the sum of absolute value of diagonal elements $\mathbf{M}_{pp}$ in $\mathbf{M}$. A numerical solution ϕ^n is updated by ϕ^{n,K_0} when the number of iterations K_0 is large enough such that (21) is satisfied.

2.5. 1D comparison

Before the section on numerical experiments in 3D, we would like to show the formulation of (18) in 1D in order to see a difference between ABG and IBG. Let us denote a computational domain:

$$\bar{\Omega} = [0, 1] \subset \bigcup_{i\in\mathcal{I}} I_i, \quad I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right], \quad x_{i-\frac{1}{2}} = \left(i-\frac{1}{2}\right)h,$$

where $\bar{\Omega}$ is evenly divided by a small length $h \ll 1$. For the simplicity, we only consider an advective flow with a positive constant velocity $v > 0$:

$$\frac{\partial\phi}{\partial t} + v\frac{\partial\phi}{\partial x} = 0. \tag{22}$$

Now the particular cases of the IIOE method (19) can be written on an internal cell I_i:

- If an ABG is used, then $\mathcal{D}_p\phi = \overline{\mathcal{D}}_p\phi$ in (13) and

$$\frac{\phi_i^n-\phi_i^{n-1}}{\Delta t} + \frac{v}{2}\left(\frac{3\phi_i^n-4\phi_{i-1}^n+\phi_{i-2}^n}{2h}+\frac{\phi_{i+1}^{n-1}-\phi_{i-1}^{n-1}}{2h}\right) = 0. \tag{23}$$

- If an IBG is used, then $\mathcal{D}_p\phi = \mathcal{D}_p^-\phi$ in (14) and

$$\frac{\phi_i^n-\phi_i^{n-1}}{\Delta t} + \frac{v}{2}\left(\frac{2\phi_i^n-3\phi_{i-1}^n+\phi_{i-2}^n}{2h}+\frac{\phi_i^{n-1}-\phi_{i-1}^{n-1}}{h}\right) = 0. \tag{24}$$

The formulation (23) using ABG shows an average of an implicit second-order backward difference scheme and an explicit central difference scheme in the point x_i for the both schemes. The formulation using IBG (23) shows the average of an implicit second-order backward difference scheme in the point $x_{i+\frac{1}{2}}$ and an explicit central difference scheme in $x_{i-\frac{1}{2}}$. One can also see the stencil of ABG consists of one more point than the one of IBG which may result in higher accuracy in practical computations.

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

Table 1
The number of cells (c) in hexahedron and polyhedron shapes is enumerated. Δt is the time step in (19). The length h is an average of diagonal length of a box enclosing each cell. Note that we generate approximately 8 times smaller volumes between consecutive levels of polyhedron mesh.

		Hexahedrons in a box ($\mathcal{H}$)		Polyhedrons in box ($\mathcal{P}$)	
Level	Δt	c	h	c	h
1	$2.0 \cdot 10^{-2}$	27,000	$3.3 \cdot 10^{-2}$	4,129	$7.5 \cdot 10^{-2}$
2	$1.0 \cdot 10^{-2}$	216,000	$1.7 \cdot 10^{-2}$	32,962	$3.7 \cdot 10^{-2}$
3	$5.0 \cdot 10^{-3}$	1,728,000	$8.3 \cdot 10^{-3}$	262,996	$1.8 \cdot 10^{-2}$
4	$2.5 \cdot 10^{-3}$	13,824,000	$4.2 \cdot 10^{-3}$	2,106,130	$9.8 \cdot 10^{-3}$

Note that the authors in [18] provide an analysis of 1D case for the linear advection equation with a variable velocity where a second order accuracy and a stability for the basic IIOE scheme are proved. Moreover, the authors in [21] perform a numerical von Neumann stability analysis for IBG and ABG schemes on uniform tensor grids and they prove the second order accuracy for both schemes. A similar analysis in a 3D polyhedron mesh is beyond the scope of the current work.

3. Numerical experiments

In following subsections, numerical properties of the proposed method (19) are discussed in various examples. The most examples are computed on two meshes, hexahedrons and polyhedrons, generated by `AVL FIRE`®, see some characteristics in Table 1, where the computational domain is a box:

$$\Omega = [-0.5, 0.5]^3 \subset \mathbb{R}^3. \tag{25}$$

Some examples are presented to check an experimental order of convergence (EOC) of the proposed method. An algebraic multigrid method in `AVL FIRE`® on decomposed computational domains with the 1-ring face neighborhood structure is used to solve the matrix equation (20). Moreover, a time step Δt in (19) for each level from 1 to 4 in Table 1 is fixed for all examples except the cases in Section 3.4.

Let us denote two initial functions whose zero level set is either a sphere or a cube:

$$\phi_0(\mathbf{x}) = \phi_s(\mathbf{x}; \mathbf{a}, r) \quad \text{or} \quad \phi_c(\mathbf{x}; \mathbf{a}, r), \tag{26}$$

where

$$\phi_s(\mathbf{x}; \mathbf{a}, r) = |\mathbf{x} - \mathbf{a}| - r, \tag{27}$$

$$\phi_c(\mathbf{x}; \mathbf{a}, r) = \max_j\{|x_j - a_j|\} - r. \tag{28}$$

The particular test examples are specified by an initial value and velocity in (1):

Test 1. Translating a sphere:

$$\phi_0 = \phi_s\left(\mathbf{x}; \left(-\frac{0.1}{\sqrt{3}}, -\frac{0.1}{\sqrt{3}}, -\frac{0.1}{\sqrt{3}}\right), 0.2\right), \quad \mathbf{u} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right).$$

Test 2. Rotating a sphere (see Fig. 2):

$$\phi_0 = \phi_s\left(\mathbf{x}; (-0.25, 0, 0), 0.2\right), \quad \mathbf{u} = (-\pi x_2, \pi x_1, 0).$$

Test 3. Rotating a cube:

$$\phi_0 = \phi_c\left(\mathbf{x}; (-0.25, 0, 0), 0.2\right), \quad \mathbf{u} = (-\pi x_2, \pi x_1, 0).$$

Test 4. Shrinking a sphere:

$$\phi_0 = \phi_s\left(\mathbf{x}; (-0.25, 0, 0), 0.2\right), \quad \mathbf{u} = -\frac{\nabla\phi}{|\nabla\phi|}.$$

Test 5. Shrinking a cube:

$$\phi_0 = \phi_c\left(\mathbf{x}; (-0.25, 0, 0), 0.2\right), \quad \mathbf{u} = -\frac{\nabla\phi}{|\nabla\phi|}.$$

Test 6. Expanding a sphere:

$$\phi_0 = \phi_s\left(\mathbf{x}; (-0.25, 0, 0), 0.1\right), \quad \mathbf{u} = \frac{\nabla\phi}{|\nabla\phi|}.$$

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

Table 2
An EOC of translating a sphere (Test 1) at $T = 0.1$ on $\mathcal{H}$ and $\mathcal{P}$ in Table 1: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	Test 1: Translating a sphere					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	ABG on $\mathcal{H}$					
1	$7.49 \cdot 10^{-4}$	–	$1.30 \cdot 10^{-4}$	–	$6.63 \cdot 10^{-4}$	–
2	$1.83 \cdot 10^{-5}$	2.04	$3.35 \cdot 10^{-5}$	1.95	$1.47 \cdot 10^{-4}$	2.17
3	$4.37 \cdot 10^{-6}$	2.06	$8.35 \cdot 10^{-6}$	2.00	$3.21 \cdot 10^{-5}$	2.20
4	$1.06 \cdot 10^{-6}$	2.04	$2.09 \cdot 10^{-6}$	2.00	$7.63 \cdot 10^{-6}$	2.07
	IBG on $\mathcal{H}$					
1	$1.78 \cdot 10^{-4}$	–	$4.59 \cdot 10^{-4}$	–	$1.48 \cdot 10^{-3}$	–
2	$4.47 \cdot 10^{-5}$	2.00	$1.20 \cdot 10^{-4}$	1.94	$3.08 \cdot 10^{-4}$	2.27
3	$1.11 \cdot 10^{-5}$	2.00	$2.98 \cdot 10^{-5}$	2.01	$6.56 \cdot 10^{-5}$	2.23
4	$2.78 \cdot 10^{-6}$	2.00	$7.43 \cdot 10^{-6}$	2.00	$1.55 \cdot 10^{-5}$	2.08
	ABG on $\mathcal{P}$					
1	$8.21 \cdot 10^{-4}$	–	$1.54 \cdot 10^{-3}$	–	$5.00 \cdot 10^{-3}$	–
2	$2.13 \cdot 10^{-4}$	1.95	$4.21 \cdot 10^{-4}$	1.87	$1.61 \cdot 10^{-3}$	1.64
3	$4.87 \cdot 10^{-5}$	2.13	$9.24 \cdot 10^{-5}$	2.19	$4.61 \cdot 10^{-4}$	1.80
4	$9.74 \cdot 10^{-6}$	2.32	$1.85 \cdot 10^{-5}$	2.32	$7.86 \cdot 10^{-5}$	2.55
	IBG on $\mathcal{P}$					
1	$1.54 \cdot 10^{-3}$	–	$3.73 \cdot 10^{-3}$	–	$9.56 \cdot 10^{-3}$	–
2	$3.79 \cdot 10^{-4}$	2.02	$1.01 \cdot 10^{-3}$	1.89	$3.44 \cdot 10^{-3}$	1.48
3	$8.86 \cdot 10^{-5}$	2.10	$2.11 \cdot 10^{-4}$	2.25	$6.94 \cdot 10^{-4}$	2.31
4	$1.82 \cdot 10^{-5}$	2.28	$4.38 \cdot 10^{-5}$	2.27	$1.43 \cdot 10^{-4}$	2.28

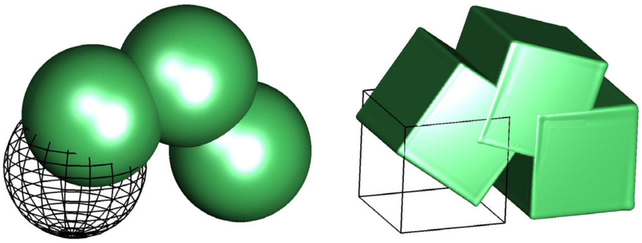

Fig. 2. The figures from left to right are numerical solutions of rotating a sphere and rotating a cube.

Test 7. Expanding a cube:

$$\phi_0 = \phi_c\left(\mathbf{x}; (-0.25, 0, 0), 0.1\right), \qquad \mathbf{u} = \frac{\nabla\phi}{|\nabla\phi|}.$$

Since it is possible to compute exact solutions of Test 1 to Test 7 on $(0, T]$, we use the exact values ϕ_b in Dirichlet boundary conditions and we compute the following errors for all examples:

$$L^1 = \frac{1}{|\Omega|}\int_\Omega \left|\phi(\mathbf{x}, T) - \phi^e(\mathbf{x})\right| \simeq \frac{1}{|\Omega|}\sum_{p\in\mathcal{I}} \left|\phi(\mathbf{x}_p, T) - \phi^e(\mathbf{x}_p)\right| |\Omega_p|, \tag{29}$$

$$L^1_{loc} = \frac{1}{|\Gamma|}\int_\Gamma \left|\phi(\mathbf{x}, T) - \phi^e(\mathbf{x})\right| \simeq \frac{\sum_{p\in\mathcal{J}\subsetneq\mathcal{I}} \left|\phi(\mathbf{x}_p, T) - \phi^e(\mathbf{x}_p)\right| |\Omega_p|}{\sum_{p\in\mathcal{J}\subsetneq\mathcal{I}} |\Omega_p|}, \tag{30}$$

$$L^\infty_{loc} = \max_{\mathbf{x}\in\Gamma}\left\{\left|\phi(\mathbf{x}, T) - \phi^e(\mathbf{x})\right|\right\} \simeq \max_{p\in\mathcal{J}\subsetneq\mathcal{I}}\left\{\left|\phi(\mathbf{x}_p, T) - \phi^e(\mathbf{x}_p)\right|\right\}, \tag{31}$$

where ϕ^e is an exact solution, Γ is a zero level set of ϕ^e, and $\mathcal{J}$ is a set of cell indices whose signs of vertex values of the exact solution ϕ^e are not identical.

In following sections we discuss the results of numerical experiments for the ABG and IBG used in (19).

3.1. Experimental order of convergence

From Test 1 to Test 7, all cases, except Test 3 and Test 7, show that the EOC for the errors (29) and (30) is around 2; see EOCs from tbl8]Tables 2 to 8.

Table 3
An EOC of rotating a sphere (Test 2) at $T = 1.0$ on $\mathcal{H}$ and $\mathcal{P}$ in Table 1: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	Test 2: Rotating a sphere					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	ABG on $\mathcal{H}$					
1	$2.80 \cdot 10^{-4}$	–	$1.02 \cdot 10^{-3}$	–	$4.00 \cdot 10^{-3}$	–
2	$7.20 \cdot 10^{-5}$	1.96	$2.38 \cdot 10^{-4}$	2.10	$6.52 \cdot 10^{-4}$	2.62
3	$1.83 \cdot 10^{-5}$	1.98	$5.93 \cdot 10^{-5}$	2.01	$1.43 \cdot 10^{-4}$	2.19
4	$4.61 \cdot 10^{-6}$	1.99	$1.48 \cdot 10^{-5}$	2.00	$3.38 \cdot 10^{-5}$	2.08
	IBG on $\mathcal{H}$					
1	$7.34 \cdot 10^{-4}$	–	$3.85 \cdot 10^{-3}$	–	$1.76 \cdot 10^{-2}$	–
2	$1.96 \cdot 10^{-4}$	1.91	$8.59 \cdot 10^{-4}$	2.16	$2.39 \cdot 10^{-3}$	2.88
3	$5.08 \cdot 10^{-5}$	1.95	$2.10 \cdot 10^{-4}$	2.03	$4.77 \cdot 10^{-4}$	2.33
4	$1.29 \cdot 10^{-5}$	1.97	$5.23 \cdot 10^{-5}$	2.00	$1.16 \cdot 10^{-4}$	2.04
	ABG on $\mathcal{P}$					
1	$2.16 \cdot 10^{-3}$	–	$7.70 \cdot 10^{-3}$	–	$1.98 \cdot 10^{-2}$	–
2	$5.66 \cdot 10^{-4}$	1.90	$2.77 \cdot 10^{-3}$	1.47	$1.15 \cdot 10^{-2}$	0.79
3	$1.51 \cdot 10^{-4}$	1.93	$5.24 \cdot 10^{-4}$	2.40	$1.31 \cdot 10^{-3}$	3.13
4	$3.25 \cdot 10^{-5}$	2.22	$1.05 \cdot 10^{-4}$	2.32	$2.79 \cdot 10^{-4}$	2.23
	IBG on $\mathcal{P}$					
1	$4.30 \cdot 10^{-3}$	–	$1.59 \cdot 10^{-2}$	–	$3.78 \cdot 10^{-2}$	–
2	$1.28 \cdot 10^{-3}$	1.75	$7.11 \cdot 10^{-3}$	1.16	$2.85 \cdot 10^{-2}$	0.41
3	$3.47 \cdot 10^{-4}$	1.88	$1.55 \cdot 10^{-3}$	2.20	$4.78 \cdot 10^{-3}$	2.58
4	$7.68 \cdot 10^{-5}$	2.17	$3.04 \cdot 10^{-4}$	2.35	$7.61 \cdot 10^{-4}$	2.65

Table 4
An EOC of rotating a cube (Test 3) at $T = 1.0$ on $\mathcal{H}$ and $\mathcal{P}$ in Table 1: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	Test 3: Rotating a cube					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	ABG on $\mathcal{H}$					
1	$2.52 \cdot 10^{-3}$	–	$4.57 \cdot 10^{-3}$	–	$2.16 \cdot 10^{-2}$	–
2	$1.01 \cdot 10^{-3}$	1.33	$2.17 \cdot 10^{-3}$	1.07	$1.76 \cdot 10^{-2}$	0.30
3	$3.79 \cdot 10^{-4}$	1.41	$9.41 \cdot 10^{-4}$	1.21	$1.04 \cdot 10^{-2}$	0.76
4	$1.46 \cdot 10^{-4}$	1.38	$2.94 \cdot 10^{-4}$	1.25	$6.62 \cdot 10^{-3}$	0.65
	IBG on $\mathcal{H}$					
1	$4.98 \cdot 10^{-3}$	–	$8.18 \cdot 10^{-3}$	–	$2.74 \cdot 10^{-2}$	–
2	$2.28 \cdot 10^{-3}$	1.13	$4.57 \cdot 10^{-3}$	0.84	$2.25 \cdot 10^{-2}$	0.28
3	$9.76 \cdot 10^{-4}$	1.22	$2.28 \cdot 10^{-3}$	1.00	$1.31 \cdot 10^{-2}$	0.78
4	$4.14 \cdot 10^{-4}$	1.24	$1.07 \cdot 10^{-3}$	1.10	$8.74 \cdot 10^{-3}$	0.59
	ABG on $\mathcal{P}$					
1	$9.80 \cdot 10^{-3}$	–	$1.46 \cdot 10^{-2}$	–	$4.10 \cdot 10^{-2}$	–
2	$4.25 \cdot 10^{-3}$	1.21	$7.82 \cdot 10^{-3}$	0.90	$2.93 \cdot 10^{-2}$	0.48
3	$1.79 \cdot 10^{-3}$	1.24	$3.69 \cdot 10^{-3}$	1.09	$1.84 \cdot 10^{-2}$	0.67
4	$6.64 \cdot 10^{-4}$	1.43	$1.50 \cdot 10^{-3}$	1.30	$1.22 \cdot 10^{-2}$	0.59
	IBG on $\mathcal{P}$					
1	$1.57 \cdot 10^{-2}$	–	$1.86 \cdot 10^{-2}$	–	$5.59 \cdot 10^{-2}$	–
2	$7.75 \cdot 10^{-3}$	1.02	$1.24 \cdot 10^{-2}$	0.59	$4.15 \cdot 10^{-2}$	0.43
3	$3.54 \cdot 10^{-3}$	1.13	$6.72 \cdot 10^{-3}$	0.88	$2.56 \cdot 10^{-2}$	0.70
4	$1.39 \cdot 10^{-3}$	1.34	$3.00 \cdot 10^{-3}$	1.16	$1.67 \cdot 10^{-2}$	0.62

Concerning the exceptions, we note that in Test 3 singularities in the solution exist on diagonal planes in the box domain that are advected along the rotational flow and that make it difficult to have EOC $\simeq 2$. In Test 7, in addition to the previous singularities, a so-called rarefaction from eight corners of the cubic isosurfaces brings an extra difficulty. In both singular cases we end up having EOC $\simeq 1$ in L^1 and L^1_{loc} norms and less than 1 in L^∞_{loc} norm. Such behavior of higher order numerical methods for the advection is well-known for nonsmooth solutions with singularities, particularly, for the isosurfaces having initially sharp corners. Similar behavior of EOC is reported e.g. in [14,22] for analogous examples in 2D case.

From Figs. 3–6, the yellow surface presents the initial zero level set and the green surface is the zero level set of numerical solutions at $T = 0.02$, $T = 0.06$, and $T = 0.1$.

We emphasize that unlike the linearly extrapolated boundary values used in [15] and [20], the Dirichlet boundary condition is applied to all examples in the proposed method (19). In such a way one can avoid the errors from an inexact treatment of boundary condition and measure the errors coming only from the discretization scheme.

Table 5
An EOC of shrinking a shape (Test 4) at $T = 0.1$ on $\mathcal{H}$ and $\mathcal{P}$ in Table 1: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	Test 4: Shrinking a sphere					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	ABG on $\mathcal{H}$					
1	$1.15 \cdot 10^{-4}$	–	$7.12 \cdot 10^{-4}$	–	$1.99 \cdot 10^{-3}$	–
2	$2.88 \cdot 10^{-5}$	2.00	$1.79 \cdot 10^{-4}$	2.00	$4.99 \cdot 10^{-4}$	1.99
3	$7.12 \cdot 10^{-6}$	2.02	$4.46 \cdot 10^{-5}$	2.00	$1.34 \cdot 10^{-4}$	1.90
4	$1.77 \cdot 10^{-6}$	2.01	$3.11 \cdot 10^{-6}$	2.01	$3.11 \cdot 10^{-5}$	2.10
	IBG on $\mathcal{H}$					
1	$9.74 \cdot 10^{-5}$	–	$4.72 \cdot 10^{-4}$	–	$1.32 \cdot 10^{-3}$	–
2	$2.43 \cdot 10^{-5}$	2.00	$1.19 \cdot 10^{-4}$	1.99	$2.87 \cdot 10^{-4}$	2.21
3	$6.05 \cdot 10^{-6}$	2.01	$3.07 \cdot 10^{-5}$	1.96	$7.29 \cdot 10^{-4}$	1.98
4	$1.51 \cdot 10^{-6}$	2.00	$7.62 \cdot 10^{-6}$	2.01	$1.65 \cdot 10^{-5}$	2.14
	ABG on $\mathcal{P}$					
1	$6.31 \cdot 10^{-4}$	–	$3.62 \cdot 10^{-3}$	–	$1.12 \cdot 10^{-2}$	–
2	$1.61 \cdot 10^{-4}$	1.97	$1.43 \cdot 10^{-3}$	1.33	$3.47 \cdot 10^{-3}$	1.69
3	$3.99 \cdot 10^{-5}$	2.01	$3.97 \cdot 10^{-4}$	1.85	$9.43 \cdot 10^{-4}$	1.88
4	$8.59 \cdot 10^{-6}$	2.22	$7.93 \cdot 10^{-5}$	2.32	$1.90 \cdot 10^{-4}$	2.39
	IBG on $\mathcal{P}$					
1	$6.16 \cdot 10^{-4}$	–	$3.41 \cdot 10^{-3}$	–	$1.08 \cdot 10^{-2}$	–
2	$1.58 \cdot 10^{-4}$	1.97	$1.39 \cdot 10^{-3}$	1.29	$3.46 \cdot 10^{-3}$	1.64
3	$3.90 \cdot 10^{-5}$	2.01	$3.94 \cdot 10^{-4}$	1.82	$8.94 \cdot 10^{-4}$	1.95
4	$8.40 \cdot 10^{-6}$	2.22	$7.78 \cdot 10^{-5}$	2.34	$1.77 \cdot 10^{-4}$	2.34

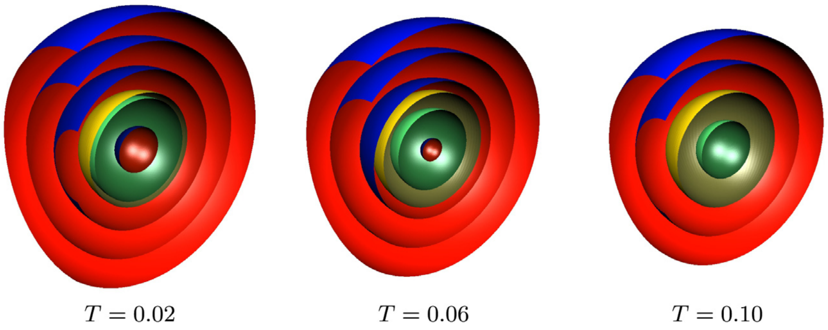

Fig. 3. Isosurfaces of numerical solution computed by ABG for shrinking a sphere (Test 4). The level surfaces are obtained by −0.1, 0, 0.1, 0.2, 0.3 and the green surface is zero level set of numerical solution. The view is only on a side of negative y-axis and the yellow sphere is the initial zero level set of ϕ_0 in (1). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

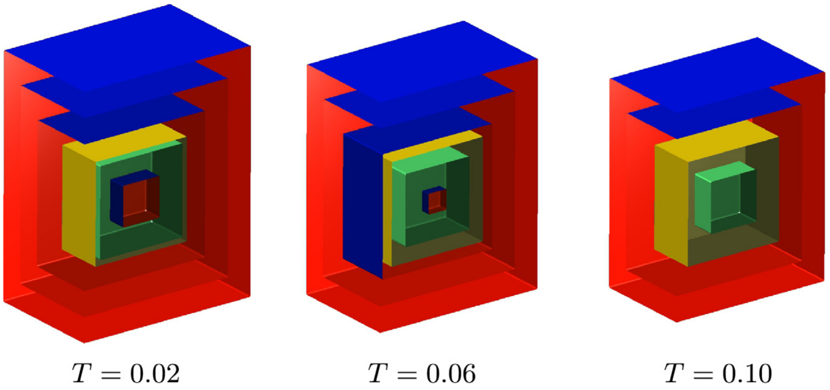

Fig. 4. Isosurfaces of numerical solution computed by ABG for shrinking a cube (Test 5). The level surfaces are obtained by −0.1, 0, 0.1, 0.2, 0.3 and the green surface is zero level set of numerical solution. The view is only on a side of negative y-axis and the yellow cube is the initial zero level set of ϕ_0 in (1). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

Table 6
An EOC of shrinking a cube (Test 5) at $T = 0.1$ on $\mathcal{H}$ and $\mathcal{P}$ in Table 1: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	Test 5: Shrinking a cube					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	ABG on $\mathcal{H}$					
1	$5.82 \cdot 10^{-4}$	–	$2.01 \cdot 10^{-3}$	–	$5.78 \cdot 10^{-3}$	–
2	$4.63 \cdot 10^{-5}$	3.65	$2.38 \cdot 10^{-4}$	3.08	$2.22 \cdot 10^{-3}$	1.38
3	$1.16 \cdot 10^{-5}$	2.00	$6.66 \cdot 10^{-5}$	1.84	$1.10 \cdot 10^{-3}$	1.01
4	$2.89 \cdot 10^{-6}$	2.00	$1.77 \cdot 10^{-5}$	1.91	$5.51 \cdot 10^{-4}$	1.00
	IBG on $\mathcal{H}$					
1	$4.74 \cdot 10^{-4}$	–	$1.65 \cdot 10^{-3}$	–	$5.85 \cdot 10^{-3}$	–
2	$3.03 \cdot 10^{-5}$	3.96	$1.67 \cdot 10^{-4}$	3.30	$2.35 \cdot 10^{-3}$	1.31
3	$7.28 \cdot 10^{-6}$	2.06	$4.20 \cdot 10^{-5}$	1.99	$1.13 \cdot 10^{-3}$	1.06
4	$1.82 \cdot 10^{-6}$	2.00	$1.03 \cdot 10^{-5}$	2.03	$5.66 \cdot 10^{-4}$	1.00
	ABG on $\mathcal{P}$					
1	$2.84 \cdot 10^{-3}$	–	$1.19 \cdot 10^{-2}$	–	$2.38 \cdot 10^{-2}$	–
2	$1.06 \cdot 10^{-3}$	1.42	$4.71 \cdot 10^{-3}$	1.34	$1.31 \cdot 10^{-2}$	0.86
3	$2.91 \cdot 10^{-4}$	1.86	$1.42 \cdot 10^{-3}$	1.73	$6.57 \cdot 10^{-3}$	1.00
4	$6.10 \cdot 10^{-5}$	2.26	$2.86 \cdot 10^{-4}$	2.31	$2.78 \cdot 10^{-3}$	1.24
	IBG on $\mathcal{P}$					
1	$2.70 \cdot 10^{-3}$	–	$1.16 \cdot 10^{-2}$	–	$2.40 \cdot 10^{-2}$	–
2	$9.79 \cdot 10^{-4}$	1.46	$4.43 \cdot 10^{-3}$	1.39	$1.30 \cdot 10^{-2}$	0.89
3	$2.71 \cdot 10^{-4}$	1.85	$1.32 \cdot 10^{-3}$	1.75	$6.52 \cdot 10^{-3}$	0.99
4	$5.73 \cdot 10^{-5}$	2.24	$2.70 \cdot 10^{-4}$	2.29	$2.62 \cdot 10^{-3}$	1.31

Table 7
An EOC of expanding a shape (Test 6) at $T = 0.1$ on $\mathcal{H}$ and $\mathcal{P}$ in Table 1: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	Test 6: Expanding a sphere					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	ABG on $\mathcal{H}$					
1	$2.45 \cdot 10^{-4}$	–	$7.40 \cdot 10^{-4}$	–	$2.09 \cdot 10^{-3}$	–
2	$6.43 \cdot 10^{-5}$	1.93	$1.84 \cdot 10^{-4}$	2.01	$5.73 \cdot 10^{-4}$	1.86
3	$1.71 \cdot 10^{-5}$	1.91	$4.45 \cdot 10^{-5}$	2.05	$1.28 \cdot 10^{-4}$	2.16
4	$5.81 \cdot 10^{-6}$	1.55	$1.10 \cdot 10^{-5}$	2.01	$3.09 \cdot 10^{-5}$	2.05
	IBG on $\mathcal{H}$					
1	$3.52 \cdot 10^{-4}$	–	$1.68 \cdot 10^{-3}$	–	$7.97 \cdot 10^{-3}$	–
2	$1.13 \cdot 10^{-4}$	1.64	$3.37 \cdot 10^{-4}$	2.32	$2.08 \cdot 10^{-3}$	1.94
3	$4.17 \cdot 10^{-5}$	1.44	$2.47 \cdot 10^{-5}$	3.77	$1.09 \cdot 10^{-4}$	4.26
4	$1.65 \cdot 10^{-5}$	1.33	$7.61 \cdot 10^{-6}$	1.70	$1.67 \cdot 10^{-5}$	2.71
	ABG on $\mathcal{P}$					
1	$1.74 \cdot 10^{-3}$	–	$8.73 \cdot 10^{-3}$	–	$3.12 \cdot 10^{-2}$	–
2	$4.27 \cdot 10^{-4}$	2.02	$9.23 \cdot 10^{-4}$	3.24	$4.32 \cdot 10^{-3}$	2.85
3	$1.19 \cdot 10^{-4}$	1.84	$3.87 \cdot 10^{-4}$	1.25	$7.09 \cdot 10^{-4}$	2.61
4	$3.02 \cdot 10^{-5}$	1.98	$7.34 \cdot 10^{-5}$	2.40	$1.30 \cdot 10^{-4}$	2.45
	IBG on $\mathcal{P}$					
1	$2.24 \cdot 10^{-3}$	–	$1.28 \cdot 10^{-2}$	–	$4.24 \cdot 10^{-2}$	–
2	$5.47 \cdot 10^{-4}$	2.03	$1.06 \cdot 10^{-3}$	3.59	$5.78 \cdot 10^{-3}$	2.88
3	$1.74 \cdot 10^{-4}$	1.65	$2.41 \cdot 10^{-4}$	2.14	$7.84 \cdot 10^{-4}$	2.88
4	$5.17 \cdot 10^{-5}$	1.75	$4.52 \cdot 10^{-5}$	2.42	$1.35 \cdot 10^{-4}$	2.54

3.2. *ABG vs IBG*

From the EOC in Section 3.1, the ABG and IBG work very similarly. To see more clearly which one is better, we present a ratio in Table 9 computed by

$$\text{ratio} = \frac{\text{error from IBG}}{\text{error from ABG}}. \tag{32}$$

Note that the ratio computed by Test 1 is very similar to Test 2 or Test 3, so it is omitted in Table 9. The ratio shows that the errors from ABG are similar to or between twice and three times less than the errors from IBG. The case when the errors from IBG are smaller than the errors from ABG is in the example of shrinking a sphere or a cube. In Fig. 7, the numerical results from Test 1 are illustrated that clearly shows the result from ABG is more accurate than the one from IBG.

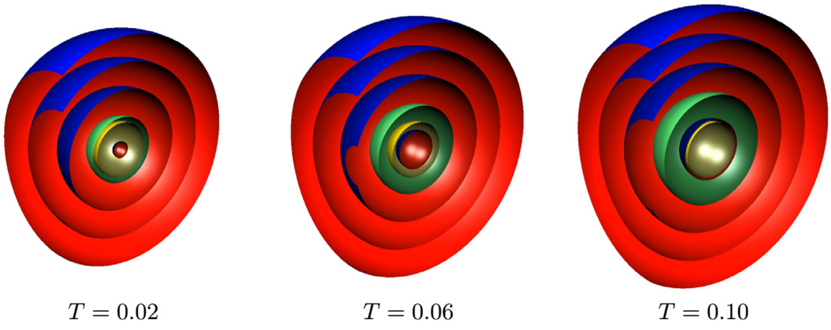

Fig. 5. Isosurfaces of numerical solution computed by ABG for expanding a sphere (Test 6). The level surfaces are obtained by −0.09, 0, 0.1, 0.2, 0.3 and the green surface is zero level set of numerical solution. The view is only on a side of negative y-axis and the yellow sphere is the initial zero level set of ϕ_0 in (1). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 8
Experimental order of convergence (EOC) of expanding a cube (Test 7) at $T = 0.1$ on $\mathcal{H}$ and $\mathcal{P}$ in Table 1: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	Test 7: Expanding a cube					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	ABG on $\mathcal{H}$					
1	$2.86 \cdot 10^{-3}$	–	$4.90 \cdot 10^{-3}$	–	$8.60 \cdot 10^{-3}$	–
2	$1.16 \cdot 10^{-3}$	1.31	$1.97 \cdot 10^{-3}$	1.31	$3.97 \cdot 10^{-3}$	1.11
3	$5.75 \cdot 10^{-3}$	1.01	$1.05 \cdot 10^{-3}$	0.91	$2.70 \cdot 10^{-3}$	0.56
4	$2.90 \cdot 10^{-4}$	0.99	$5.51 \cdot 10^{-4}$	0.93	$2.03 \cdot 10^{-3}$	0.41
	IBG on $\mathcal{H}$					
1	$3.24 \cdot 10^{-3}$	–	$5.79 \cdot 10^{-3}$	–	$1.12 \cdot 10^{-2}$	–
2	$1.36 \cdot 10^{-3}$	1.25	$2.47 \cdot 10^{-3}$	1.23	$5.66 \cdot 10^{-3}$	0.99
3	$6.53 \cdot 10^{-4}$	1.06	$1.28 \cdot 10^{-3}$	0.94	$3.62 \cdot 10^{-3}$	0.65
4	$3.22 \cdot 10^{-4}$	1.02	$6.57 \cdot 10^{-4}$	0.97	$2.45 \cdot 10^{-3}$	0.56
	ABG on $\mathcal{P}$					
1	$7.30 \cdot 10^{-3}$	–	$1.43 \cdot 10^{-2}$	–	$2.92 \cdot 10^{-2}$	–
2	$3.38 \cdot 10^{-3}$	1.11	$6.28 \cdot 10^{-3}$	1.19	$1.44 \cdot 10^{-2}$	1.02
3	$1.58 \cdot 10^{-3}$	1.10	$3.03 \cdot 10^{-3}$	1.05	$7.90 \cdot 10^{-3}$	0.87
4	$6.84 \cdot 10^{-4}$	1.21	$1.33 \cdot 10^{-3}$	1.19	$4.12 \cdot 10^{-3}$	0.94
	IBG on $\mathcal{P}$					
1	$8.64 \cdot 10^{-3}$	–	$1.64 \cdot 10^{-2}$	–	$3.42 \cdot 10^{-2}$	–
2	$4.07 \cdot 10^{-3}$	1.09	$6.89 \cdot 10^{-3}$	1.25	$1.58 \cdot 10^{-2}$	1.12
3	$1.93 \cdot 10^{-3}$	1.08	$3.57 \cdot 10^{-3}$	0.95	$8.58 \cdot 10^{-3}$	0.88
4	$8.45 \cdot 10^{-4}$	1.19	$1.61 \cdot 10^{-3}$	1.15	$4.74 \cdot 10^{-3}$	0.86

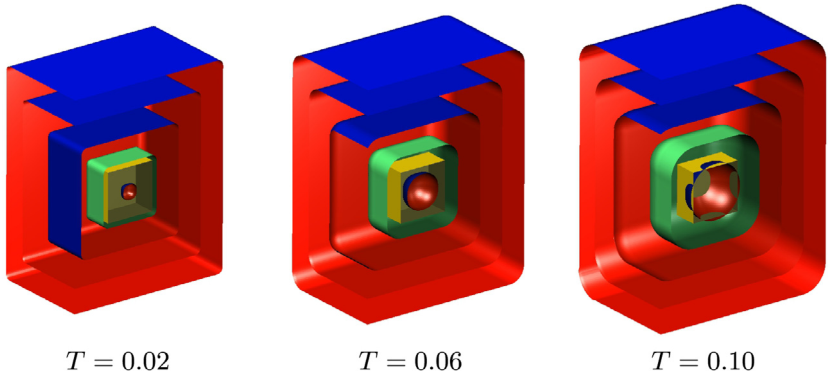

Fig. 6. Isosurfaces of numerical solution computed by ABG for expanding a cube (Test 7). The level surfaces are obtained by −0.09, 0, 0.1, 0.2, 0.3 and the green surface is zero level set of numerical solution. The view is only on a side of negative y-axis and the yellow cube is the initial zero level set of ϕ_0 in (1). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 9
An error ratio in (32) is listed from selected test cases.

Level	Error ratio											
	$\mathcal{H}$			$\mathcal{P}$			$\mathcal{H}$			$\mathcal{P}$		
	L^1	L^1_{loc}	L^∞_{loc}	L^1	L^1_{loc}	L^∞_{loc}	L^1	L^1_{loc}	L^∞_{loc}	L^1	L^1_{loc}	L^∞_{loc}
	Test 2: Rotating a sphere						Test 3: Rotating a cube					
1	2.62	3.78	4.42	1.99	2.07	1.91	1.98	1.79	1.27	1.60	1.28	1.36
2	2.72	3.61	3.67	2.22	2.56	2.48	2.26	2.10	1.28	1.83	1.58	1.42
3	2.78	3.54	3.34	2.29	2.95	3.65	2.57	2.43	1.27	1.97	1.82	1.39
4	2.81	3.54	3.43	2.36	2.90	2.73	2.84	2.79	1.32	2.10	2.00	1.37
	Test 4: Shrinking a sphere						Test 5: Shrinking a cube					
1	0.85	0.66	0.67	0.98	0.94	0.96	0.81	0.82	1.01	0.95	0.97	1.01
2	0.85	0.67	0.57	0.98	0.97	1.00	0.66	0.70	1.06	0.93	0.94	0.99
3	0.85	0.69	0.55	0.98	0.99	0.95	0.63	0.63	1.03	0.93	0.93	0.99
4	0.85	0.69	0.53	0.98	0.98	0.99	0.63	0.58	1.03	0.94	0.94	0.94
	Test 6: Expanding a sphere						Test 7: Expanding a cube					
1	1.44	2.28	3.82	1.29	1.47	1.36	1.13	1.18	1.31	1.18	1.14	1.17
2	1.75	1.83	3.62	1.28	1.15	1.34	1.17	1.25	1.43	1.20	1.10	1.09
3	2.44	0.55	0.85	1.46	0.62	1.11	1.14	1.23	1.34	1.22	1.18	1.09
4	2.85	0.69	0.54	1.71	0.62	1.04	1.11	1.19	1.21	1.24	1.21	1.15

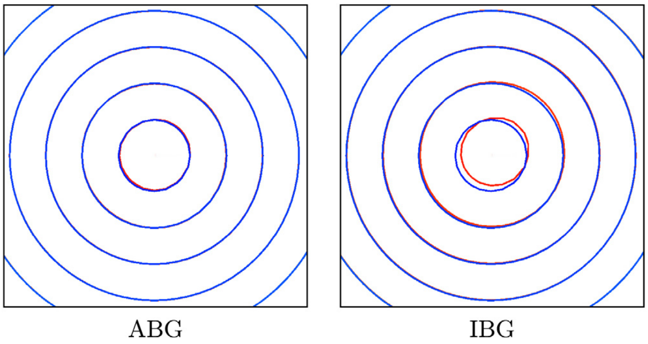

Fig. 7. The numerical results of Test 1 on the z-plane: The blue curve is the exact solution and the red one is the numerical solution at $T = 0.1$ from (19). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 10
For Tests 1 and 4 on $\mathcal{P}$, the error values and corresponding EOC show how many k iterations are practically necessary to obtain the close results from $K = K_0$ in (19).

	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
K	Translating a sphere with ABG on $\mathcal{P}$					
1	$2.48 \cdot 10^{-4}$	0.97	$4.16 \cdot 10^{-4}$	1.01	$6.67 \cdot 10^{-4}$	1.18
2	$1.22 \cdot 10^{-5}$	2.19	$2.02 \cdot 10^{-5}$	2.30	$8.25 \cdot 10^{-5}$	2.53
3	$9.70 \cdot 10^{-6}$	2.33	$1.85 \cdot 10^{-5}$	2.32	$7.85 \cdot 10^{-5}$	2.55
4	$9.73 \cdot 10^{-6}$	2.32	$1.85 \cdot 10^{-5}$	2.32	$7.86 \cdot 10^{-5}$	2.55
K_0	$9.74 \cdot 10^{-6}$	2.32	$1.85 \cdot 10^{-5}$	2.32	$7.86 \cdot 10^{-5}$	2.55
K	Shrinking a sphere with ABG on $\mathcal{P}$					
1	$4.38 \cdot 10^{-4}$	1.05	$7.92 \cdot 10^{-5}$	2.33	$1.94 \cdot 10^{-4}$	2.27
2	$2.36 \cdot 10^{-5}$	1.77	$7.94 \cdot 10^{-5}$	2.31	$1.81 \cdot 10^{-4}$	2.38
3	$9.51 \cdot 10^{-6}$	2.13	$7.93 \cdot 10^{-5}$	2.32	$1.82 \cdot 10^{-4}$	2.37
4	$8.65 \cdot 10^{-6}$	2.21	$7.93 \cdot 10^{-5}$	2.32	$1.80 \cdot 10^{-4}$	2.39
K_0	$8.59 \cdot 10^{-6}$	2.22	$7.93 \cdot 10^{-5}$	2.32	$1.80 \cdot 10^{-4}$	2.39

3.3. *Iterative IIOE method*

The main disadvantage of using the iterative IIOE scheme (19) compared with the noniterative form (18) is the extra k iterations to satisfy the small residual criterion (21). Since the matrix from (19) is an M-matrix and it is more sparse than

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

Table 11
A time stepping used in Tests 1 and 2 on polyhedron meshes in Table 1: CFL_M and CFL_m are the CFL numbers calculated by the minimum and maximum size of cell, respectively. CFL_h is obtained by the average length h in Table 1.

	CFL_1				CFL_2			
Level	Δt	CFL_M	CFL_m	CFL_h	Δt	CFL_M	CFL_m	CFL_h
1	0.15	7.67	0.88	2.00	0.3	15.34	1.76	4.00
2	0.075	6.57	0.97	2.03	0.15	13.13	1.93	4.05
3	0.0375	7.01	0.98	2.08	0.075	14.03	1.96	4.17
4	0.01875	7.65	1.07	1.91	0.0375	15.31	2.13	3.83

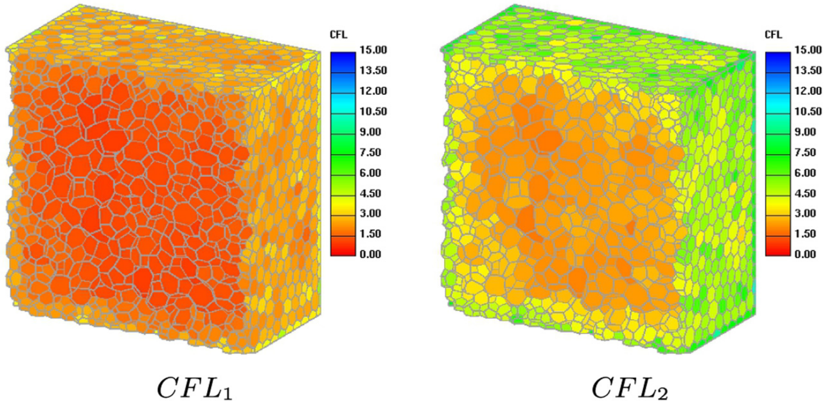

Fig. 8. A polyhedron mesh is from the first level in Table 1 and it only shows a half of computational domain in order to illustrate the CFL number on each cell. The average values of CFL number in CFL_1 and CFL_2 are approximately 2 amd 4 shown in Table 1.

the one in (18), it takes less computational cost to solve each iteration than (18). However, if a lot of iterations are necessary to have the second order convergence, the scheme (19) cannot be practically used.

In Table 10, for Test 1 and Test 4 on the polyhedron mesh $\mathcal{P}$ in Table 1, we use a fixed number of K iterations with $K = 1, \ldots, 4$ and $K = K_0$ means that the result is from satisfying the 10^{-12} residual error in (21). The errors shown in Table 10 are from the level 4 at $T = 0.1$. In the case of advective flow, $K = 2$ is large enough to have very close EOC as computed by the case $K = K_0$, and $K = 3$ shows almost the same error as for the case of $K = K_0$. In the case of normal flow, $K = 1$ is enough for the errors L^1_{loc} and L^∞_{loc} and it is consistent with the observations in [20]. In the case of L^1, the choice $K = 3$ shows almost the same error as for the case $K = K_0$.

3.4. A time stepping

Two examples are tested to show stability and a similar behavior of EOC shown in Tables 2 and 5 in cases of a large time step which violates the CFL condition in advective and normal flows:

Example 1. Translating a sphere:

$$\phi_0 = \phi_s\left(\mathbf{x}; \left(-\frac{0.3}{\sqrt{3}}, -\frac{0.3}{\sqrt{3}}, -\frac{0.3}{\sqrt{3}}\right), 0.2\right), \quad \mathbf{u} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right).$$

Example 2. Shrinking a sphere:

$$\phi_0 = \phi_s\left(\mathbf{x}; (0, 0, 0), 0.4\right), \quad \mathbf{u} = -\frac{\nabla\phi}{|\nabla\phi|}.$$

Since it is more challenging on a polyhedron mesh, the above examples are tested by the proposed method (17) with ABG on all levels of $\mathcal{P}$ in Table 1. The time steps of CFL_1 and CFL_2 cases in Table 11 are 7.5 and 15 times larger than the time step used in Tests 1 and 4 at each level and the corresponding maximal CFL numbers are approximately 7 and 14, respectively, and the average CFL numbers are approximately 2 and 4, respectively. In Fig. 8, the CFL numbers are illustrated on each cell. Example 1. and Example 2. are tested with two CFL numbers shown in Table 11. The EOC of the examples shown in Table 12 is similar in Tables 2 and 5 which are obtained by smaller time step at each level.

Table 12
An EOC of Examples 1. and 2. at $T = 0.3$ on $\mathcal{P}$ in Table 1 with ABG: L^1, L^1_{loc}, L^∞_{loc} are computed by (29), (30), and (31), respectively.

	CFL tests					
	L^1	EOC	L^1_{loc}	EOC	L^∞_{loc}	EOC
Level	Example 1. with CFL_1					
1	$1.87 \cdot 10^{-3}$	–	$6.01 \cdot 10^{-3}$	–	$1.57 \cdot 10^{-2}$	–
2	$5.02 \cdot 10^{-4}$	1.90	$1.72 \cdot 10^{-3}$	1.81	$6.54 \cdot 10^{-3}$	1.27
3	$1.37 \cdot 10^{-4}$	1.88	$4.90 \cdot 10^{-4}$	1.81	$1.26 \cdot 10^{-3}$	2.37
4	$3.61 \cdot 10^{-5}$	1.92	$1.29 \cdot 10^{-4}$	1.93	$3.02 \cdot 10^{-4}$	2.06
	Example 1. with CFL_2					
1	$5.98 \cdot 10^{-3}$	–	$1.69 \cdot 10^{-2}$	–	$4.73 \cdot 10^{-2}$	–
2	$1.96 \cdot 10^{-3}$	1.61	$7.88 \cdot 10^{-3}$	1.10	$3.33 \cdot 10^{-2}$	0.51
3	$5.64 \cdot 10^{-4}$	1.80	$2.41 \cdot 10^{-3}$	1.71	$6.78 \cdot 10^{-3}$	2.30
4	$1.54 \cdot 10^{-4}$	1.87	$6.21 \cdot 10^{-4}$	1.96	$2.19 \cdot 10^{-3}$	1.63
	Example 2. with CFL_1					
1	$1.17 \cdot 10^{-3}$	–	$6.55 \cdot 10^{-3}$	–	$1.02 \cdot 10^{-2}$	–
2	$2.82 \cdot 10^{-4}$	2.05	$2.36 \cdot 10^{-3}$	1.47	$6.86 \cdot 10^{-3}$	0.57
3	$6.82 \cdot 10^{-5}$	2.05	$5.93 \cdot 10^{-4}$	2.00	$1.22 \cdot 10^{-3}$	2.49
4	$1.46 \cdot 10^{-5}$	2.23	$2.49 \cdot 10^{-4}$	2.31	$2.49 \cdot 10^{-4}$	2.49
	Example 2. with CFL_2					
1	$1.26 \cdot 10^{-3}$	–	$6.46 \cdot 10^{-3}$	–	$9.74 \cdot 10^{-3}$	–
2	$2.87 \cdot 10^{-4}$	2.14	$2.37 \cdot 10^{-3}$	1.44	$6.84 \cdot 10^{-3}$	0.51
3	$6.84 \cdot 10^{-5}$	2.07	$5.94 \cdot 10^{-4}$	2.00	$1.22 \cdot 10^{-3}$	2.49
4	$1.46 \cdot 10^{-5}$	2.23	$1.19 \cdot 10^{-4}$	2.32	$2.49 \cdot 10^{-4}$	2.29

4. Conclusions

The iterative IIOE FVM for the level-set equations of advective and normal flows on polyhedron meshes is presented. The method is second-order accurate in space and time for smooth solutions. The crucial step of the tessellation of nonplanar faces in polyhedron cells is required to obtain the second-order accurate schemes.

Two types of the gradient approximations at the cell center are derived and compared for representative examples: ABG and IBG. In this paper, we suggest to use the ABG scheme that has the stencil in the explicit part enlarged additionally when compared with the IBG scheme.

We propose the iterative IIOE to fulfill the 1-ring face neighborhood restriction of assembled matrix, when solving the linear system of equations. In each iteration one has to solve a linear system where the matrix is a M-matrix identical for the IBG and ABG schemes and the off-diagonal components of matrix are only connected by the unknowns of neighboring cells over faces. For the chosen representative numerical examples we can show that it is enough to use two iterations to obtain the second-order accurate results, and that the third iteration gives the results very close to the ones obtained by the fully iterative form with the prescribed norm of the residual.

Furthermore we compare the errors in several norms of the IBG and ABG schemes for the chosen examples. Except for the case of the shrinking sphere we can report better accuracy for the ABG scheme. Therefore we recommend the ABG scheme in general for the level-set equations of advective and normal flows.

References

[1] J.A. Sethian, A fast marching method for monotonically advancing fronts, in: Proc. Nat. Acad. Sci., vol. 93, 1996, pp. 1591–1595.
[2] S. Osher, R. Fedkiw, Level set methods and dynamic implicit surfaces, in: Appl. Math. Sci., vol. 153, Springer, Berlin, 2000.
[3] N. Peters, The turbulent burning velocity for large-scale and small-scale turbulence, J. Fluid Mech. 384 (1999) 107–132.
[4] N. Peters, Turbulent combustion, in: Cambridge Monographs on Mechanics, Cambridge University Press, 2000.
[5] M. Perić, Flow simulation using control volumes of arbitrary polyhedral shape, in: ERCOFTAC Bulletin 62, 2004, pp. 25–29.
[6] N. Ray, T. van Noorden, F. Radu, W. Friess, P. Knabner, Drug release from collagen matrices including an evolving microstructure, ZAMM Z. Angew. Math. Mech. 93 (2013) 811–822.
[7] N. Ray, T. van Noorden, F. Frank, P. Knabner, Multiscale modeling of colloid and fluid dynamics in porous media including an evolving microstructure, Transp. Porous Media 95 (2012) 669–696.
[8] R. Schulz, P. Knabner, Derivation and analysis of an effective model for biofilm growth in evolving porous media, Math. Methods Appl. Sci. (2017) 2930–2948.
[9] C. Bringedal, I. .Berre, I.S. Pop, F.A. Radu, Upscaling of non-isothermal reactive porous media flow with changing porosity, Transp. Porous Media 114 (2016) 371–393.
[10] Y. Coudiére, G. Manzini, The discrete duality finite volume method for convection-diffusion problems, SIAM J. Numer. Anal. 47 (2017) 4163–4192.
[11] B. Andreianov, F. Boyer, F. Hubert, Discrete duality finite volume schemes for Leray-Lions type elliptic problems on general 2D meshes, Numer. Methods PDEs 23 (2007) 145–195.
[12] K. Lipnikov, G. Manzini, Discretization of mixed formulations of elliptic problems on polyhedral meshes, in: Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations, Springer International Publishing, 2016, pp. 311–342.
[13] T.V. Voitovich, S. Vandewalle, Exact integration formulas for the finite volume element method on simplicial meshes, Numer. Methods Partial Differential Equations 23 (2007) 1059–1082.

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.

[14] P. Frolkovič, K. Mikula, High-resolution flux-based level set method, SIAM J. Sci. Comput. 29 (2007) 579–597.

[15] J. Hahn, K. Mikula, P. Frolkovič, B. Basara, Inflow-based gradient finite volume method for a propagation in a normal direction in a polyhedron mesh, J. Sci. Comput. 72 (2017) 442–465.

[16] K. Mikula, M. Ohlberger, A new level set method for motion in normal direction based on a semi-implicit forward-backward diffusion approach, SIAM J. Sci. Comput. 32 (2010) 1527–1544.

[17] K. Mikula, M. Ohlberger, Inflow-implicit/outflow-explicit scheme for solving advection equations, in: J.F. et al. (Ed.), Finite Volumes for Complex Applications VI - Problems & Perspectives, in: Springer Proceedings in Mathematics 4, vol. 1, Springer, 2011, pp. 683–691.

[18] K. Mikula, M. Ohlberger, J. Urbán, Inflow-implicit/outflow-explicit finite volume methods for solving advection equations, Appl. Numer. Math. 85 (2014) 16–37.

[19] P. Frolkovič, K. Mikula, J. Urbán, Semi-implicit finite volume level set method for advective motion of interfaces in normal direction, Appl. Numer. Math. 95 (2015) 214–228.

[20] J. Hahn, K. Mikula, P. Frolkovič, B. Basara, Semi-implicit level set method with inflow-based gradient in a polyhedron mesh, in: C. Cancès, P. Omnes (Eds.), Finite Volumes for Complex Applications VIII - Hyperbolic, Elliptic and Parabolic Problems, Springer International Publishing, 2017, pp. 81–89.

[21] P. Frolkovič, K. Mikula, Semi-implicit second order schemes for numerical solution of level set advection equation on cartesian grids, Appl. Math. Comput. 329 (2018) 129–142.

[22] C. Min, F. Gibou, A second order accurate level set method on non-graded adaptive cartesian grids, J. Comput. Phys. 225 (1) (2007) 300–321.

Please cite this article in press as: J. Hahn, et al., Iterative inflow-implicit outflow-explicit finite volume scheme for level-set equations on polyhedron meshes, Computers and Mathematics with Applications (2018), https://doi.org/10.1016/j.camwa.2018.06.033.