ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2022, Vol. 62, No. 7, pp. 1041–1058. © Pleiades Publishing, Ltd., 2022.
Russian Text © The Author(s), 2022, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2022, Vol. 62, No. 7, pp. 1067–1084.

Nonlinear Finite Volume Method for the Interface Advection-Compression Problem on Unstructured Adaptive Meshes

Yu. V. Vassilevski[a,b,*] **and K. M. Terekhov**[a,c,**]

[a] *Marchuk Institute of Numerical Mathematics, Russian Academy of Sciences, Moscow, 119333 Russia*

[b] *Sechenov University, Moscow, 119991 Russia*

[c] *Moscow Institute of Physics and Technology, Dolgoprudnyi, Moscow oblast, 141701 Russia*

**e-mail: yuri.vassilevski@gmail.com*

***e-mail: terekhov@inm.ras.ru*

Received January 8, 2022; revised January 8, 2022; accepted February 11, 2022

Abstract—The paper is devoted to the nonlinear finite volume method applied for tracking interfaces on unstructured adaptive meshes. The fluid of volume approach is used. The interface location is described by the fraction of fluid in each computational cell. The interface propagation involves the simultaneous solution of the fraction advection and interface compression problems. The compression problem is solved to recover the interface (front) sharpness, which is smeared due to numerical diffusion. The problem discretization is carried out using the nonlinear monotone finite volume method. This method is applied to unstructured meshes with adaptive local refinement.

Keywords: implicit front-tracking, volume of fluid method, interface compression, nonlinear finite volume method, monotone method

DOI: 10.1134/S0965542522060148

1. INTRODUCTION

Interface tracking methods are in demand in many physical applications, such as free-surface fluid flows [1–3], multiphase fluid flows [4], mean curvature flows [5–8], solidification of alloys [9], image enhancement [10], etc. Numerous existing codes use the volume of fluid method to implicitly track the interface [11–15]. Two main approaches to mitigating the interface smearing are described in the literature—geometric reconstruction [16–18] and interface compression methods [19–23]. The second approach requires the solution of a backward parabolic partial differential equation. A typical difficulty in the compression method is the interface distortion, which becomes more pronounced in the case of large time steps [18]. Some works propose adaptation of the interface compression parameters to solve this problem [24–26]. A popular alternative approach is the level set function method [27–30]. This method requires the solution of reinitialization problem, and it does not preserve the fluid mass; a number of works are devoted to these issues [31–34]. In this paper, we consider the approach based on the volume of fluid method in combination with the interface compression method.

A number of high-resolution schemes for solving the volume-of-fluid advection problem were proposed [35–38], in particular, for the advection–compression problem [39, 23, 40–42]. In this paper, we adopt the nonlinear finite volume method that was earlier developed for the anisotropic diffusion problem [43–47], for solving the advection–compression problem. In combination with the backward Euler scheme for the time derivative, the proposed method preserves physical bounds of the calculated fluid fraction and allows for time steps with the Courant number greater than one. The methods applies to general unstrutured meshes, i.e., to consistent meshes with general polyhedral cells.

The paper is organized as follows. In Section 2, we formulate the problem and describe the nonlinear finite volume method. In Section 3, we present our parallel technology for the adaptive refinement of general unstructured meshes. Section 4 describes numerical experiments that demonstrate the capabilities of the method and the technology based on it.

2. NONLINEAR FINITE VOLUME DISCRETIZATION

Let a scalar function ψ represent the fraction of volume occupied by the fluid at each point of the computational domain Ω, and let $\mathbf{u}$ be a given velocity vector field. Then, the advection of the unknown function ψ is governed by the equation

$$\begin{aligned} \frac{\partial\psi}{\partial t} + \operatorname{div}(\psi\mathbf{u}) &= 0 \quad \text{in} \quad \Omega, \\ \mathbf{n}^{\mathrm{T}}\nabla\psi &= 0 \quad \text{on} \quad \partial\Omega, \\ \psi|_{t=0} &= \psi^0 \quad \text{in} \quad \Omega, \end{aligned} \tag{2.1}$$

where $\mathbf{n}$ is the outward normal to $\partial\Omega$. On the boundary, we condsider the homogeneous Neumann boundary condition.

The numerical solution of the advection equation smears the interface due to numerical diffusion. To recover the interface sharpness, the advection equation (2.1) is augmented with the interface compression term [26, 19]

$$\frac{\partial\psi}{\partial t} + \operatorname{div}\left(\psi\mathbf{u} + \alpha\psi(1-\psi)\frac{\nabla\psi}{\|\nabla\psi\|}\right) = 0 \quad \text{in} \quad \Omega, \tag{2.2}$$

subject to the same initial and Neumann boundary conditions. The parameter α is typically defined using $\|\mathbf{u}\|$, and it controls the strength of the interface compression. In this work, we adjust α adaptively following [26]. In the denominator, we use the regularized Frobenius norm.

Remark 2.1. Problem (2.2) is a nonlinear advection–diffusion problem $\frac{\partial\psi}{\partial t} + \operatorname{div}(\psi\mathbf{u} - D\nabla\psi) = 0$ with a negative diffusion coefficient $D = -\alpha\psi(1-\psi)\|\nabla\psi\|^{-1}$, which makes Eq. (2.2) a backward parabolic problem [48].

2.1. Advection Problem

Let Ω be covered by a consistent polyhedral mesh that can be adapted at different time steps. Denote the center of the cell V by $\mathbf{x}_V$ and assign to it a degree of freedom.

First, we consider the advection problem (2.1). Applying the Ostrogradsky–Gauss theorem to the integral over the cell V, we obtain

$$\int_V \left(\frac{\partial\psi}{\partial t} + \operatorname{div}(\psi\mathbf{u})\right) dV = \int_V \frac{\partial\psi}{\partial t} dV + \oint_{\partial V} \psi d\mathbf{S}^{\mathrm{T}}\mathbf{u} = 0, \tag{2.3}$$

which is approximated with the second-order accuracy by

$$|V| \frac{\partial\psi}{\partial t}\bigg|_{\mathbf{x}_V} + \sum_{f\in\partial V} |f|\beta_f\, \psi|_{\mathbf{x}_f} = 0, \tag{2.4}$$

where $|V|$ is the volume of the cell V bounded by the set of faces ∂V, $|f|$ and $\mathbf{x}_f$ are the area and the barycenter of the face f, and β_f is the projection of velocity onto the face normal

$$\beta_f = \mathbf{n}_f^{\mathrm{T}}\mathbf{u}\big|_{\mathbf{x}_f} \approx \frac{1}{|f|}\int_f d\mathbf{S}^{\mathrm{T}}\mathbf{u}. \tag{2.5}$$

The sign of β_f depends on the orientation of the face normal $\mathbf{n}_f$.

To discretize Eq. (2.4), we should approximate the time derivative. To simplify the presentation, we use the backward Euler scheme:

$$\frac{\partial\psi}{\partial t}\bigg|_{\mathbf{x}_V} \approx \frac{\psi^{n+1} - \psi^n}{\Delta t}\bigg|_{\mathbf{x}_V}. \tag{2.6}$$

In practice, we use the Crank–Nicolson scheme.

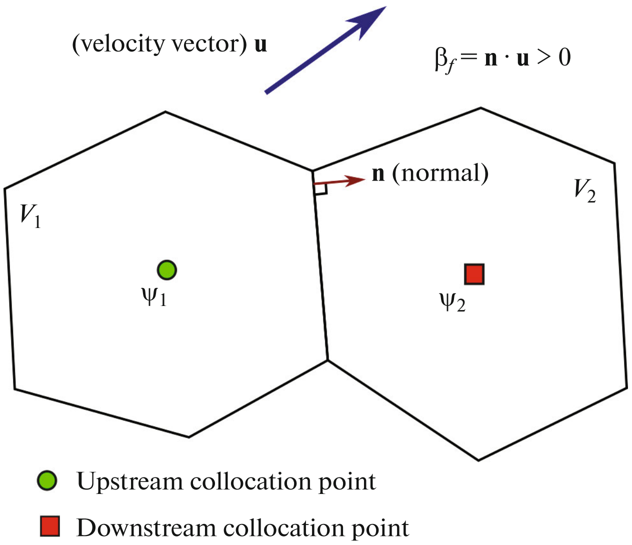
(velocity vector) $\mathbf{u}$
$\beta_f = \mathbf{n}\cdot\mathbf{u} > 0$
$\mathbf{n}$ (normal)
V_1
V_2
ψ_1
ψ_2
● Upstream collocation point
🟥 Downstream collocation point

Fig. 1. In the upstream method, ψ_1 is used for approximating ψ at the face barycenter.

The residual vector $R_a(\psi^{n+1})$ for the backward Euler scheme (2.4) in the cell V is

$$R_a(\psi^{n+1})\Big|_V = |V|\left(\psi^{n+1} - \psi^n\right)\Big|_{\mathbf{x}_V} + \Delta t \sum_{f\in\partial V} |f|\beta_f^{n+1}\psi^{n+1}\Big|_{\mathbf{x}_f}, \tag{2.7}$$

which requires the flux $q = \beta_f^{n+1}\psi^{n+1}\Big|_{\mathbf{x}_f}$ to be approximated on each face f.

The residual vector $R_a(\psi^{n+1})$ is associated with the Jacobian

$$J_a(\psi^{n+1}) = \partial R_a(\psi^{n+1})/\partial(\psi^{n+1})^{\mathrm{T}}.$$

In the following description of the Jacobian properties, we omit the time level index and assume that $\psi \equiv \psi^{n+1}$ and $\beta_f \equiv \beta_f^{n+1}$.

In this work, we assume that a discrete divergent-free velocity field $\mathbf{u}$ is defined on the mesh faces at each time step:

$$0 = \sum_{f\in\partial V} \mathbf{n}_f^{\mathrm{T}}\mathbf{u}\Big|_{\mathbf{x}_f} |f| \approx \sum_{f\in\partial V}\int_f d\mathbf{S}^{\mathrm{T}}\mathbf{u} = \oint_{\partial V} d\mathbf{S}^{\mathrm{T}}\mathbf{u} = \int_V \mathrm{div}(\mathbf{u})dV. \tag{2.8}$$

Since $\beta_f = \mathbf{n}_f^{\mathrm{T}}\mathbf{u}\Big|_{\mathbf{x}_f}$ is assumed to be given, the key point in the approximation of the flux q is the computation of $\psi|_{\mathbf{x}_f}$.

Consider an internal face $f = V_1 \cap V_2$ with the normal $\mathbf{n}_f$ that is outward for the cell V_1 and inward for the cell V_2. Let $\beta_f > 0$; then, the single-point upstream method approximates $\psi|_{\mathbf{x}_f} = \psi_1$ with the first-order accuracy (Fig. 1).

With such an approximation of $\psi|_{\mathbf{x}_f}$, the row of the Jacobian in (2.7) corresponding to the advection in cell V_1 surrounded by its neighbors V_k is assembled as

$$J_a\Big|_{V_1} \leftarrow \Delta t\left(\sum_{f\in\partial V_1}\frac{|\beta_f| + \beta_f}{2}\right)\partial\psi_1 - \Delta t\sum_{f\in\partial V_1\cap V_k}\frac{|\beta_f| - \beta_f}{2}\partial\psi_k, \tag{2.9}$$

which gives a Jacobian that is an M-matrix [49]. Indeed, the diagonal elements of this Jacobian are positive, its off-diagonal elements are negative, and the sum of the elements in each row is zero due to (2.8).

The second-order approximation for q is obtained using the antidiffusive correction

$$\psi|_{\mathbf{x}_f} \approx \psi_1 + (\mathbf{x}_f - \mathbf{x}_1)^{\mathrm{T}} \nabla\psi. \tag{2.10}$$

First, we consider the antidiffusive correction on a one-dimensional mesh with equidistant spacing $h/2$ between the collocation centers and faces. Using the notation of Fig. 2, we introduce two different approximations for the correction $\vartheta = (\mathbf{x}_f - \mathbf{x}_1)^{\mathrm{T}} \nabla\psi$ at the point $\mathbf{x}_f$:

$$\vartheta_1 = \frac{\psi_1 - \psi_0}{2} \quad \text{and} \quad \vartheta_2 = \frac{\psi_3 - \psi_2}{2}, \tag{2.11}$$

where ϑ_1 and ϑ_2 are the approximations on the cells V_1 and V_2, respectively. With these corrections, we obtain two second-order approximations of $\psi|_{\mathbf{x}_f}$:

$$\psi|_{\mathbf{x}_f} \approx \psi_1 + \vartheta_1 = \frac{3\psi_1 - \psi_0}{2} \quad \text{and} \quad \psi|_{\mathbf{x}_f} \approx \psi_1 + \vartheta_2 = \psi_1 + \frac{\psi_3 - \psi_2}{2} \tag{2.12}$$

with the variations

$$\partial\psi_1 + \partial\vartheta_1 = \frac{3}{2}\partial\psi_1 - \frac{1}{2}\partial\psi_0 \quad \text{and} \quad \partial\psi_1 + \partial\vartheta_2 = \partial\psi_1 + \frac{1}{2}\partial\psi_3 - \frac{1}{2}\partial\psi_2. \tag{2.13}$$

The corrections ϑ_1 and ϑ_2 do not affect the row sums in the Jacobian matrix. The second-order approximations of $\psi|_{\mathbf{x}_f} \approx \psi_1 + \vartheta_1$ in the cell V_1 and $\psi|_{\mathbf{x}_f} \approx \psi_1 + \vartheta_2$ in the cell V_2 make the following contributions to the Jacobian matrix:

$$\begin{aligned} J_a|_{V_1} &\leftarrow \Delta t|f|\left(\frac{3\beta_f}{2}\partial\psi_1 - \frac{\beta_f}{2}\partial\psi_0\right), \\ J_a|_{V_2} &\leftarrow \Delta t|f|\left(\frac{\beta_f}{2}\partial\psi_2 - \beta_f\partial\psi_1 - \frac{\beta_f}{2}\partial\psi_3\right), \end{aligned} \tag{2.14}$$

which preserves the M-matrix property. However, the use of different approximations of the flux q on different sides of the face f violates the conservation law. To make the flux unique, we consider a nonlinear convex combination of two approximations with nonnegative coefficients $\mu_1 + \mu_2 = 1$:

$$\psi|_{\mathbf{x}_f} \approx \psi_1 + \mu_1\vartheta_1 + \mu_2\vartheta_2. \tag{2.15}$$

We use the weights known as Van Leer limited averages [50]

$$\mu_1 = \frac{|\vartheta_2|}{|\vartheta_1| + |\vartheta_2|} \quad \text{and} \quad \mu_2 = \frac{|\vartheta_1|}{|\vartheta_1| + |\vartheta_2|}. \tag{2.16}$$

Then, we obtain $\mu_1\vartheta_1 + \mu_2\vartheta_2 = 0$ for $\vartheta_1\vartheta_2 < 0$ and $\mu_1\vartheta_1 + \mu_2\vartheta_2 = 2\mu_1\vartheta_1 = 2\mu_2\vartheta_2$ for $\vartheta_1\vartheta_2 > 0$. To avoid division by zero, we regularize the modulus operation in (2.16) as $|x|_\varepsilon = \sqrt{x^2 + \varepsilon^2}$, where ε is a small constant.

Neglecting the derivatives in μ_1 and μ_2 and considering the approximation $\psi|_{\mathbf{x}_f} \approx \psi_1 + 2\mu_1\vartheta_1$ for V_1 and $\psi|_{\mathbf{x}_f} \approx \psi_1 + 2\mu_2\vartheta_2$ for V_2 in the case $\vartheta_1\vartheta_2 > 0$, we obtain the following contributions of approximations (2.15) to the Jacobian:

$$\begin{aligned} J_a|_{V_1} &\leftarrow \Delta t|f|((\beta_f + \beta_f\mu_1)\partial\psi_1 - \beta_f\mu_1\partial\psi_0), \\ J_a|_{V_2} &\leftarrow \Delta t|f|(\beta_f\mu_2\partial\psi_2 - \beta_f\partial\psi_1 - \beta_f\mu_2\partial\psi_3). \end{aligned} \tag{2.17}$$

This approach is known as the Picard iterative method [51], which preserves the M-matrix property of the Jacobian at each nonlinear iteration and the conservation property when the nonlinear iterations converge. In practice, we use approximation (2.15) within the Newton iterative method.

The antidiffusion correction can be easily extended to unstructured polyhedral meshes in d dimensions. Again, we assume that $\beta_f > 0$ and, therefore, V_1 is an upstream cell. Let us find two corrections ϑ_1

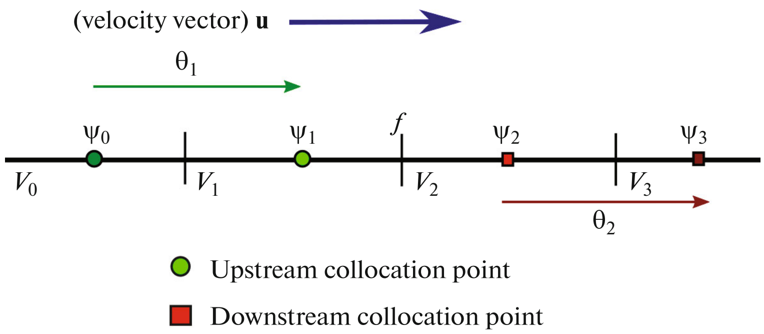

Fig. 2. Notation for the 1D antidiffusive correction.

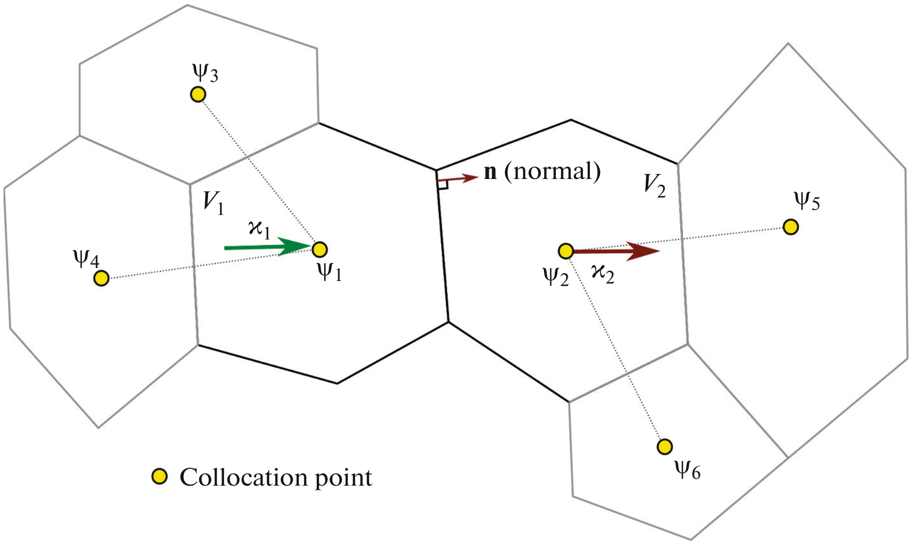

Fig. 3. Notation for the 2D antidiffusive correction on a polygonal mesh.

and ϑ_2 using combinations of adjacent collocation points (see Fig. 3 for the 2D case). We define the antidiffusion correction $\vartheta_1 = (\mathbf{x}_f - \mathbf{x}_1)^{\mathrm{T}}\nabla\psi$ for the face f in the cell V_1 as follows.

Let the set $\boldsymbol{\sigma}_1$ consist of the cells V_k with the collocation $\mathbf{x}_k$ that have at least one common node with V_1 and the boundary faces with the outward normal $\mathbf{n}_{f_k}$ adjacent to V_1. For each element e_k in $\boldsymbol{\sigma}_1$, we define the vector $\mathbf{v}_k$ and the value ψ_k by

$$(\mathbf{v}_k, \psi_k) = \begin{cases} (\mathbf{x}_k - \mathbf{x}_1, \psi_k) & \text{if} \quad e_k \text{ is a cell,} \\ (\mathbf{n}_{f_k}, \psi_1) & \text{if} \quad e_k \text{ is a face.} \end{cases}$$

Note that the scalar product of the vector $\mathbf{v}_k$ with the gradient $\nabla\psi$ is the finite difference in the direction $\mathbf{v}_k$, $\mathbf{v}_k^{\mathrm{T}}\nabla\psi = \psi_k - \psi_1$, and if e_k is a boundary face, then $\mathbf{v}_k^{\mathrm{T}}\nabla\psi = \psi_1 - \psi_1 = 0$ defines the homogeneous Neumann boundary condition. In the set $\boldsymbol{\sigma}_1$, we choose d elements e_{i_k} $(k = 1,\ldots,d)$ that ensure $\mathbf{x}_1 - \mathbf{x}_f = \sum_{k=1}^{d}\gamma_{i_k}\mathbf{v}_{i_k}$ with nonnegative weights $\gamma_{i_k} \geq 0$ minimizing the expression

$$\sum_{k=1}^{d}\gamma_{i_k}\left|\frac{\mathbf{v}_{i_k}}{|\mathbf{v}_{i_k}|} - \frac{\mathbf{x}_1 - \mathbf{x}_f}{|\mathbf{x}_1 - \mathbf{x}_f|}\right|; \tag{2.18}$$

then, the antidiffusion correction ϑ_1 takes the form

$$\vartheta_1 = \left(\sum_{k=1}^{d} \gamma_{i_k}\right)\psi_1 - \sum_{k=1}^{d} \gamma_{i_k}\psi_{i_k}, \tag{2.19}$$

where $\psi_{i_k} = \psi_1$ for the boundary face. The contribution of the correction ϑ_1 preserves the row sums in the Jacobian and its M-matrix property:

$$\partial\vartheta_1 = \left(\sum_{k=1}^{d} \gamma_{i_k}\right)\partial\psi_1 - \sum_{k=1}^{d} \gamma_{i_k}\partial\psi_{i_k}. \tag{2.20}$$

On each boundary face f adjacent to V_1, the unique approximation with the correction $\psi|_{\mathbf{x}_f} \approx \psi_1 + \vartheta_1$ also preserves the M-matrix property of the Jacobian.

Note that changing the direction of the approximated vector $\mathbf{x}_1 - \mathbf{x}_f$ to $\mathbf{x}_f - \mathbf{x}_1$ requires the signs of the coefficients in (2.19)–(2.20) to be reverted.

Similarly, using the set $\boldsymbol{\sigma}_2$ for the cell V_2, we choose in $\boldsymbol{\sigma}_2$ d elements e_{i_k} $(k = 1,\dots,d)$ that ensure $\mathbf{x}_f - \mathbf{x}_1 = \sum_{k=1}^{d} \gamma_{i_k}\mathbf{v}_{i_k}$ with nonnegative weights $\gamma_{i_k} \geq 0$ that minimize the expression

$$\sum_{k=1}^{d} \gamma_{i_k}\left\|\frac{\mathbf{v}_{i_k}}{\|\mathbf{v}_{i_k}\|} - \frac{\mathbf{x}_f - \mathbf{x}_1}{\|\mathbf{x}_f - \mathbf{x}_1\|}\right\|.$$

Then, the antidiffusion correction in V_2 is given by

$$\vartheta_2 = \sum_{k=1}^{d} \gamma_{i_k}\psi_{i_k} - \left(\sum_{k=1}^{d} \gamma_{i_k}\right)\psi_2.$$

The value $\psi|_{\mathbf{x}_f}$ is approximated by the nonlinear combination (2.15) with the coefficients (2.16).

2.2. Interface Compression

Consider the integral over a cell for the interface compression equation

$$\int_V \operatorname{div}\left(\alpha\psi(1-\psi)\frac{\nabla\psi}{\|\nabla\psi\|}\right)dV = \oint_{\partial V} \alpha\psi(1-\psi)d\mathbf{S}^{\mathrm{T}}\frac{\nabla\psi}{\|\nabla\psi\|}. \tag{2.21}$$

The residual of the backward Euler scheme for the finite volume V (2.2) is

$$R(\psi)|_V = R_a(\psi)|_V + \Delta t\sum_{f\in\partial V} |f|\,\alpha\psi(1-\psi)\|\nabla\psi\|^{-1}\mathbf{n}_f^{\mathrm{T}}\nabla\psi\Big|_{\mathbf{x}_f}, \tag{2.22}$$

where $R_a(\psi)|_V$ is defined in (2.7).

In problem (2.22), we should discretize the compressive flux c:

$$c = \alpha\psi(1-\psi)\|\nabla\psi\|^{-1}\mathbf{n}_f^{\mathrm{T}}\nabla\psi\Big|_{\mathbf{x}_f} = \varkappa_f\psi(1-\psi)|_{\mathbf{x}_f}, \tag{2.23}$$

where $\varkappa_f = \alpha\|\nabla\psi\|^{-1}\mathbf{n}_f^{\mathrm{T}}\nabla\psi\Big|_{\mathbf{x}_f}$ is the normal component of the compressive velocity.

Taking into account the normal component β_f of the velocity $\mathbf{u}$ on the face f shared by the cells V_1 and V_2, we define the approximation of the parameter α in the cells V_j $(j = 1,2)$ by

$$\alpha_{f,j} = |\beta_f|\frac{|(\mathbf{x}_2 - \mathbf{x}_1)^{\mathrm{T}}\nabla\psi_j| + \Delta t|\mathbf{u}_{f,j}^{\mathrm{T}}\nabla\psi_j|}{\|\mathbf{x}_2 - \mathbf{x}_1\|\|\nabla\psi_j\| + \Delta t\|\mathbf{u}_{f,j}\|\|\nabla\psi_j\|}. \tag{2.24}$$

This parameter [26] allows us to avoid wrinkles on the interface [18] and thus improves the conventional choice $\alpha = |\beta_f|$ [24, 19].

In formula (2.24), vectors $\mathbf{u}_{f,j}$ and gradients $\nabla\psi_j$ in the cells V_j $(j = 1,2)$ are used. The velocity vector $\mathbf{u}_{f,j}$ in V_j is recovered [52] from the normal face velocities as

$$\mathbf{u}_{f,j} = \beta_f \mathbf{n}_f + \left(\mathbb{I} - \mathbf{n}_f \mathbf{n}_f^{\mathrm{T}}\right) \frac{1}{|V_j|} \sum_{f_k \in \partial V_j} |f_k| \beta_{f_k} \left(\mathbf{x}_{f_k} - \mathbf{x}_j\right). \tag{2.25}$$

The gradient $\nabla\psi_i$ in any cell V_i is recovered by the least-squares method: the set $\boldsymbol{\sigma}_i$ gives the equations $\mathbf{v}_k^{\mathrm{T}} \nabla\psi_i = \psi_k - \psi_i$, which are assembled into a system $A_i \nabla\psi_i = \mathbf{r}_i$. This system is solved by the least-squares method: $\nabla\psi_i = (A_i^{\mathrm{T}} A_i)^{-1} A_i^{\mathrm{T}} \mathbf{r}_i$. Below, we neglect the variation of the gradient in the Jacobian, i.e., $\partial\|\nabla\psi_i\| = 0$.

To recover the normal component of the compressive velocity $\varkappa_{f,1}$ on the face f of the cell V_1, we use the set $\boldsymbol{\sigma}_1$ and define the subset $\tilde{\boldsymbol{\sigma}}_1$ of admissible stencils that ensure $\sum_{k=1}^{d} \gamma_{i_k} \mathbf{v}_{i_k} = -\mathbf{n}_f$ with $\gamma_{i_k} \ge 0$, and compute their weighted average

$$\varkappa_{f,1} = \alpha_{f,1} \|\nabla\psi_1\|^{-1} \sum_{e_{i_k} \in \tilde{\boldsymbol{\sigma}}_1} \sum_{k=1}^{d} \gamma_{i_k} \left(\psi_1 - \psi_{i_k}\right) / \sum_{e_{i_k} \in \tilde{\boldsymbol{\sigma}}_1} \left(\sum_{k=1}^{d} \gamma_{i_k} \right)^{-2}. \tag{2.26}$$

This averaging smoothes the gradient approximation [19]. The computation of $\varkappa_{f,2}$ on the face f of the cell V_2 is similar. Since $\psi_j \in [0,1]$, the expression $\psi_j(1-\psi_j) \in [0, 1/4]$ and, therefore, the contributions $\psi_j(1-\psi_j)\partial\varkappa_{f,j}$ of $\varkappa_{f,j}$ to the Jacobian, yield zero row sum and preserve the M-matrix property, $j = 1,2$.

On the boundary face f adjacent to the cell V_j, the compressive velocity $\varkappa_{f,j}$ is zero due to the Neumann boundary condition.

It remains to discretize $\psi(1-\psi)\big|_{\mathbf{x}_f}$ in the compressive flux $c_{f,j} = \varkappa_{f,j}\psi(1-\psi)\big|_{\mathbf{x}_f}$ on the face $f = V_1 \cap V_2$ for each cell V_j, $j = 1,2$. Once this is done, we use their linear combination for determining the unique compressive flux c_f on f:

$$c_f = \mu_1 c_{f,1} + \mu_2 c_{f,2}, \tag{2.27}$$

where the coefficients are chosen similarly to (2.16):

$$\mu_1 = \frac{|c_{f,2}|}{|c_{f,1}| + |c_{f,2}|}, \quad \mu_2 = \frac{|c_{f,1}|}{|c_{f,1}| + |c_{f,2}|}. \tag{2.28}$$

The approximation $c_{f,j}$ is based on the upwind strategy applied to the nonlinear nonmonotone function $\psi(1-\psi)$:

$$c_{f,j} = \left(\nu_{1,j} c_{1,j} + \nu_{2,j} c_{2,j} + \varepsilon c_{1/2,j}\right) / \left(\nu_{1,j} + \nu_{2,j} + \varepsilon\right), \tag{2.29}$$

where $\varepsilon > 0$ is a small constant and the coefficient $\nu_{i,j} \ge 0$ are

$$\begin{aligned} \nu_{1,j} &= \max\left(\operatorname{sgn}(\varkappa_{f,j})\left(1 - 2\left(\psi_1 + \overline{\vartheta}_{1,j}\right)\right), 0\right), \\ \nu_{2,j} &= -\min\left(\operatorname{sgn}(\varkappa_{f,j})\left(1 - 2\left(\psi_2 + \overline{\vartheta}_{2,j}\right)\right), 0\right); \end{aligned} \tag{2.30}$$

the antidiffusion corrections in the cell V_j are $\overline{\vartheta}_{i,j} = \left(\mathbf{x}_f - \mathbf{x}_i\right)^{\mathrm{T}} \nabla\psi_j$, $i, j = 1,2$.

The upwinding approximation strategy for the nonzero $\varkappa_{f,j}$ gives three possible cases: discretization on both sides of f is admissible (both $\nu_{1,j}$ and $\nu_{2,j}$ are positive), discretization on one side is admissible (either $\nu_{1,j}$ or $\nu_{2,j}$ is positive), and none of discretizations is admissible (both $\nu_{1,j}$ and $\nu_{2,j}$ are zero). The third case is illustrated in Fig. 4. The maximum of the function (red circle) is between two approximations of $\psi\big|_{\mathbf{x}_f}$ (yellow circles). In this case, an admissible discretization is obtained by considering the approximation at the maximum $\psi(1-\psi)\big|_{\mathbf{x}_f} \approx 1/4$, which yields

$$c_{1/2,j} = \frac{\varkappa_{f,j}}{4}, \quad \partial c_{1/2,j} = \frac{\partial\varkappa_{f,j}}{4}.$$

As a result, we add an ε-weighted term to (2.29).

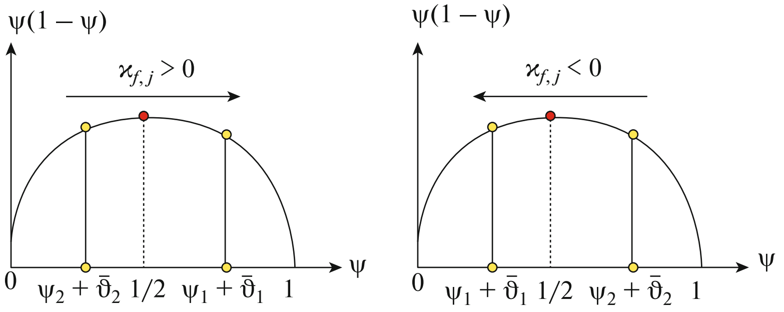
$\psi(1-\psi)$
$\varkappa_{f,j} > 0$
0
$\psi_2 + \bar\vartheta_2$
1/2
$\psi_1 + \bar\vartheta_1$
1
ψ
$\psi(1-\psi)$
$\varkappa_{f,j} < 0$
0
$\psi_1 + \bar\vartheta_1$
1/2
$\psi_2 + \bar\vartheta_2$
1
ψ

Fig. 4. Two cases when both one-sided approximations are not admissible.

Using the second-order expansion in Taylor series in the cell V_j and $\partial[\psi_i(1-\psi_i)] = [1-2\psi_i]\partial\psi_i$, we obtain

$$\psi(1-\psi)\big|_{\mathbf{x}_f} \approx \psi_i(1-\psi_i) + (1-2\psi_i)(\mathbf{x}_f - \mathbf{x}_i)^{\mathrm{T}}\nabla\psi_j = \psi_i(1-\psi_i) + (1-2\psi_i)\bar\vartheta_{i,j}. \tag{2.31}$$

Since $0 \le \psi(1-\psi) \le \frac{1}{4}$, the antidiffusive correction must be bounded by the multiplier $\eta_{i,j} \in [0,1]$ to satisfy

$$0 \le \psi_i(1-\psi_i) + (1-2\psi_i)\eta_{i,j}\bar\vartheta_{i,j} \le \frac{1}{4}. \tag{2.32}$$

From condition (2.32), we derive

$$\eta_{i,j} = \frac{1}{(1-2\psi_i)\bar\vartheta_{i,j}}\begin{cases}\frac{1}{4} - \psi_i(1-\psi_i), & (1-2\psi_i)\bar\vartheta_{i,j} > 0,\\ -\psi_i(1-\psi_i), & (1-2\psi_i)\bar\vartheta_{i,j} < 0.\end{cases} \tag{2.33}$$

Due to (2.23), the one-sided approximations $c_{i,j}$ of the compressive flux on the face f of V_j are

$$c_{i,j} = \boldsymbol{\varkappa}_{f,j}\psi_i(1-\psi_i) + \boldsymbol{\varkappa}_{f,j}(1-2\psi_i)\eta_{i,j}\bar\vartheta_{i,j}. \tag{2.34}$$

The variation of this approximation

$$\partial c_{i,j} = \left(\psi_i(1-\psi_i) + (1-2\psi_i)\eta_{i,j}\bar\vartheta_{i,j}\right)\partial\boldsymbol{\varkappa}_{f,j} + \boldsymbol{\varkappa}_{f,j}\left(1-2\left(\psi_i + \eta_{i,j}\bar\vartheta_i^j\right)\right)\partial\psi_i + \boldsymbol{\varkappa}_{f,j}(1-2\psi_i)\eta_{i,j}\partial\bar\vartheta_{i,j} \tag{2.35}$$

imposes constraints on the choice of the approximation $c_{i,j}$: it is admissible if the coefficient of $\partial\psi_i$ in (2.35) has the sign $(-1)^{i+1}$, $i = 1,2$. This consideration gives the upwind approximation rules (2.29)–(2.30). The corrections $\bar\vartheta_{i,j}$ are recovered in the cell V_j using (2.18)–(2.19) with the account of the sign of $\boldsymbol{\varkappa}_{f,j}(1-2\psi_i)$ at $\partial\bar\vartheta_{i,j}$.

By construction, each flux $c_{f,j}$ preserves the signs of entries in the Jacobian matrix when the contribution of the compressive flux is assembled into the equation of the corresponding cell V_j. However, the zero row sum property of the matrix is violated since the compressive velocity is not divergent-free. As a result, the interface compression method can produce a new extremum of the solution in the interval $[0,1]$ even for the backward Euler scheme.

2.3. Iterative Solution of the Nonlinear Algebraic System

We solve the nonlinear system using the Newton method. Let ξ^l be an approximation of ψ^{n+1} at the iteration l; as the initial approximation, we use the solution obtained at the preceding time step, $\xi^0 = \psi^n$. At each Newton iteration, we solve the linear system

$$J(\xi^l)\Delta\xi = -R(\xi^l), \tag{2.36}$$

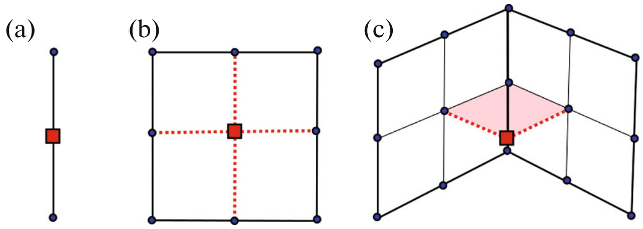

Fig. 5. Splitting steps of an edge (a), face (b), and introduction of faces that split the cell (c). Red indicates new elements at each step.

where $R(\xi^l)$ is the residual in (2.22) and $J(\xi^l) = \partial R(\xi^l)/\partial(\xi^l)^{\mathrm{T}}$ is the corresponding Jacobian. The next approximation is obtained by the rule

$$\xi^{l+1} = \xi^l + \omega\Delta\xi. \tag{2.37}$$

The parameter ω restricts the maximum change of ξ^l in (2.37) in any cell V_k: $\omega \le 0.3/|\Delta\xi_k|$ [53]. Iterations can stagnate; therefore, we apply the following heuristics to decrease ω with increasing number of nonlinear iterations l:

$$\omega = \min\left(\omega, \frac{1+\exp\left(-\frac{5}{2}\right)}{1+\exp\left(\frac{l}{6}-\frac{5}{2}\right)}\right). \tag{2.38}$$

The parameter ω is the only constraint in the computation of the next iterative approximation in the Newton method.

The iterations proceed until $\left\|R(\xi^l)\right\| \le \min\left(\tau_{\mathrm{abs}}, \tau_{\mathrm{rel}}\left\|R(\psi^n)\right\|\right)$, where $\tau_{\mathrm{abs}} = 10^{-9}$, $\tau_{\mathrm{rel}} = 10^{-2}$. The linear system (2.36) is assembled and solved using the capabilities of the INMOST platform [54–56].

3. MESH ADAPTATION

For parallel adaptation of unstructured meshes, we use the tools available on the INMOST platform (www.inmost.org). A detailed description of the algorithms is available in [54–58].

A cell of a polyhedral mesh is locally refined by introducing a new node at its center and hanging nodes at the centers of faces and edges and further splitting of the cell, faces, and edges as shown in Fig. 5. The refinement is gradual—refinement levels of two adjacent cells may differ in no more than one.

The local coarsening uses the hierarchy of sets of mesh elements represented by a tree-like structure (see Fig. 6). As a cell is refined, a new set with a unique name representing this cell is attached to the parent (root) set. All new cells become elements of this set as illustrated in Fig. 6. Cells can be coarsened only on the leaf set of the tree structure that was formed as a result of refinement. The information stored in this set is used to control the local coarsening.

The mesh is refined or coarsened by no more than one level during each sweep over mesh elements. In the parallel environment, the consistency of the mesh distributed among processors, as well as missing and excessive elements of overlapping layers, are automatically recovered [58, 54]. In this work, we used the package Parmetis for mesh partitioning [59, 60].

To transfer data as the mesh is refined or coarsened, we use the mesh modification state. In the modification state, all the old elements are retained up to the end of this state, so that the solution can be interpolated from the old mesh elements to the new ones.

In the preparatory step, we compute the gradients for ψ^{n+1} and ψ^n using the least-squares method. During refinement, the interpolation from an old cell V_0 to the new cells V_i is carried out by the rule

$$\psi_i = \psi_0 + \eta(\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}}\nabla\psi_0, \quad \nabla\psi_i = \nabla\psi_0, \tag{3.1}$$

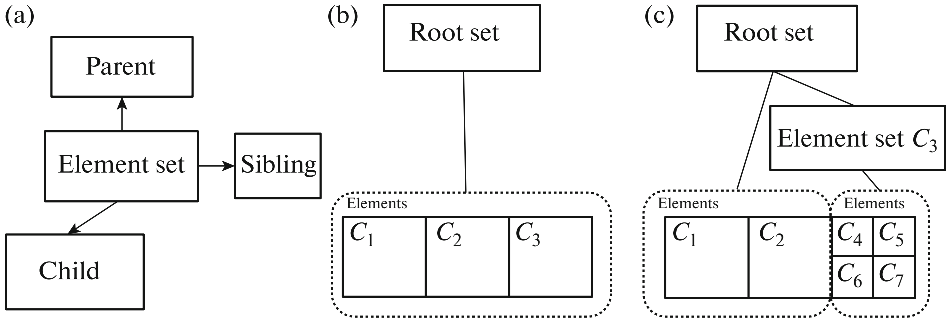

Fig. 6. Organization of sets of elements into a tree structure (a). The original unsplit mesh of three cells (b). After splitting the cell C_3, a new set is attached to the parent (root) set (c). This set remembers all the small cells that form the original cell C_3.

where η is chosen so as to satisfy the following condition for the new cells V_i:

$$\min_{V_j \in \mathcal{V}_n(V_0)} (\psi_j) \le \psi_0 + \eta(\mathbf{x}_i - \mathbf{x}_0)^{\mathrm{T}} \nabla\psi_0 \le \max_{V_j \in \mathcal{V}_n(V_0)} (\psi_j); \tag{3.2}$$

here ψ is either ψ^{n+1} or ψ^n, and $\mathcal{V}_n(V_0)$ is the set of cells that share at least one node with V_0. Such an interpolation is conservative under the condition $\sum_i |V_i|\mathbf{x}_i = |V_0|\mathbf{x}_0$ and is monotone due to (3.2). For coarsening V_i to the cell V_0, we use the simple averaging

$$\psi_0 = |V_0|^{-1} \sum_i |V_i| \psi_i, \quad \nabla\psi_0 = |V_0|^{-1} \sum_i |V_i| \nabla\psi_i, \tag{3.3}$$

which is monotone and conservative.

The refinement criterion is based on estimating the variation of the solution on the mesh faces. Let f be the face dividing the cells V_1 and V_2. Then, both cells are marked to be refined if the maximum refinement level will not be exceeded and if

$$\frac{|\psi_2^{n+1} - \psi_1^{n+1}|}{|\mathbf{x}_2 - \mathbf{x}_1|} > 5. \tag{3.4}$$

A cell is coarsened when none of its faces requires refinement and if the minimum coarsening level is not reached.

The approximate solution of the unsteady problem (2.2) by the finite volume method with mesh adaptation is obtained by the following procedure:

1. Solve the advection equation (2.1) for $\tilde{\psi}^{n+1}$ using the first-order upstream discretization and the backward Euler scheme in one linear iteration to predict the interface location at the next time step.

2. Locally refine the mesh using the variation of $\tilde{\psi}^{n+1}$ on the mesh faces, balance the mesh among processors, and interpolate $\tilde{\psi}^{n+1}$ and ψ^n to the refined mesh.

3. Perform several Newton iterations for the advection problem (2.1) using the nonlinear second-order spatial upstream discretization in combination with the Crank–Nicolson scheme to obtain a better initial guess $\hat{\psi}^{n+1}$ for the next step.

4. Solve the advection–compression equation (2.2) using the nonlinear second-order spatial discretizatiion in combination with the Crank–Nicolson scheme to obtain ψ^{n+1}; here $\hat{\psi}^{n+1}$ is used as the initial approximation for the Newton iterations.

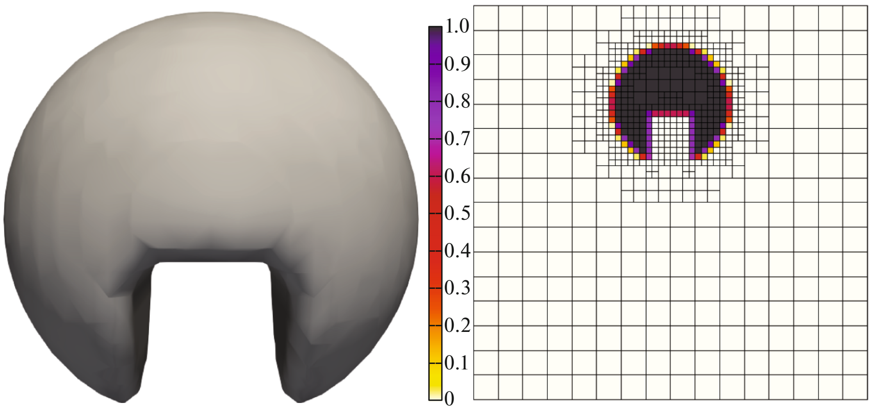

Fig. 7. Isosurface $\psi = 0.5$ and the middle ($z = 0.5$) cutaway of the mesh at $t = 0$ colored with $\psi \in [0,1]$. A similar palette is used in the following figures.

5. Coarsen the mesh using the variation of ψ^{n+1} on the mesh faces, balance the mesh among processors, and interpolate ψ^{n+1} and ψ^{n} to the coarsened mesh.

The functions $\tilde{\psi}^{n+1}$, $\hat{\psi}^{n+1}$, and ψ^{n+1} provide (with different errors) approximations of the true interface location in terms of the isosurfaces for the value $\psi = 0.5$. The best approximation is given by ψ^{n+1} and the worst approximation is given by $\tilde{\psi}^{n+1}$.

4. NUMERICAL EXAMPLES

4.1. Zalesak's Test

Consider Zalesak's benchmark [61] for a sphere rotating in the velocity field $\mathbf{u} = [u, v, w]^{\mathrm{T}}$

$$u = \pi\left(\frac{1}{2} - y\right), \quad v = \pi\left(x - \frac{1}{2}\right), \quad w = 0. \tag{4.1}$$

The velocity field is specified in the unit cube $\Omega = [0,1]^3$.

The rotating object is a sphere of radius $r = 0.15$ centered at the point $\mathbf{x} = [0.5, 0.75, 0.5]^{\mathrm{T}}$ with a cut formed by the box $[0.45, 0.55] \otimes [0.6, 0.725] \otimes [0,1]$. The projection of velocity β_f on the face of the cell and the fraction of fluid ψ in the cells are computed by subdividing polygons and polyhedra into triangles and tetrahedra, respectively, and by integrating the mean values with the 7th order of accuracy. The fluid fraction ψ equals 1 inside the object and equals 0 outside the object. The problem is solved for the time $t \in [0, 2]$ during which the object performs one revolution.

The initial mesh is obtained from the cubic $16 \times 16 \times 16$ mesh with two levels of refinement toward the interface (see Fig. 7). The resulting adaptive mesh with two refinement levels is consistent if its cubic cells are considered as polyhedral. The object is visualized using the program Paraview [62] through the isosurface $\psi = 0.5$.

In Fig. 8, we demonstrate the impact of using the second-order discretization on the solution of the advection problem (2.1) with the time step $\Delta t = 0.01$ and the Courant number $K \approx 1$. The accuracy of discretization both with respect to time and space is extremely important. However, the scheme is fairly dissipative, and severe smearing of the interface causes the mesh coarsening, which further reduces the accuracy of recovering the interface.

In Fig. 9, we compare the nonlinear finite volume method for the advection–compression problem (2.2) with the backward Euler and Crank–Nicolson time discretization schemes with the time step $\Delta t = 0.01$, $K \approx 1$. The compression method significantly improves the interface resolution. The Crank–

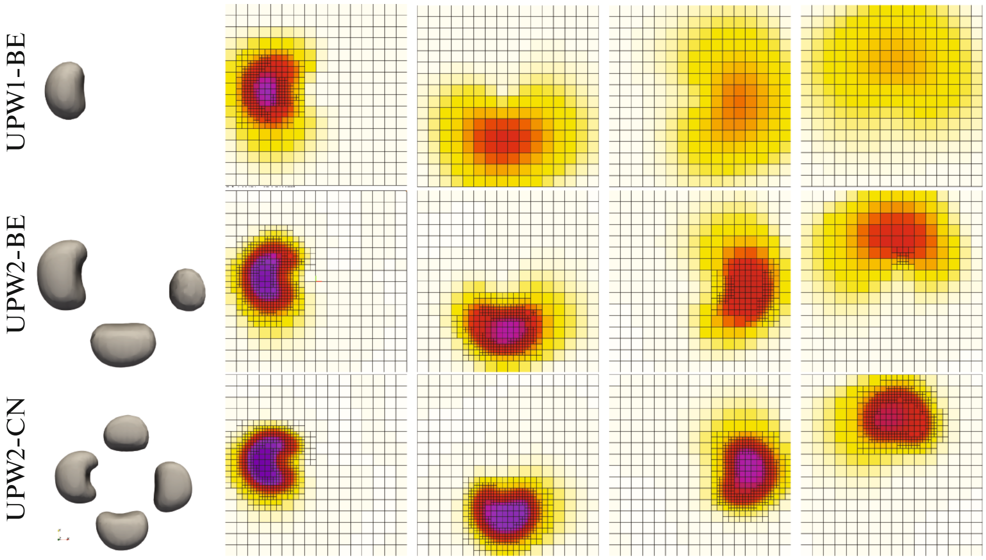

Fig. 8. Isosurface $\psi = 0.5$ and the middle ($z = 0.5$) cutaway of the mesh at $t = \{0.5, 1, 1.5, 2\}$ colored with ψ. Solution of (2.1) by the single-point upstream method with the backward Euler scheme (UPW1-BE), the second-order nonlinear finite volume method with the backward Euler scheme (UPW2-BE), and the second-order nonlinear finite volume method with the Cranc–Nicolson scheme (UPW2-CN). The time step $\Delta t = 0.01$ and $K \approx 1$.

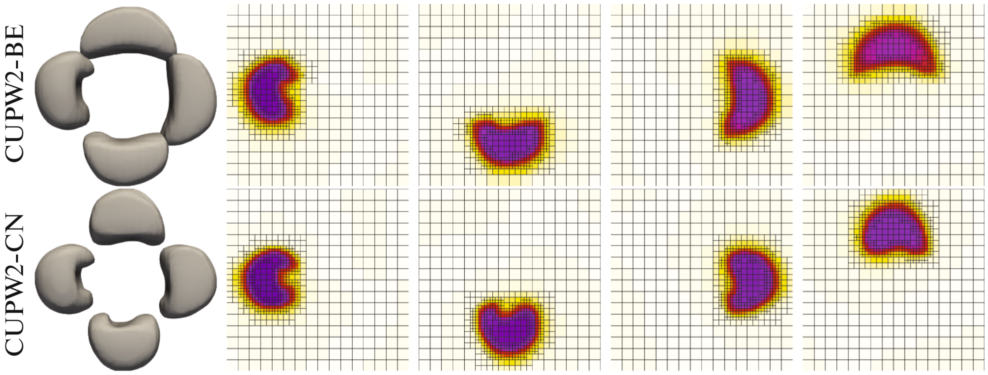

Fig. 9. Isosurface $\psi = 0.5$ and the middle ($z = 0.5$) cutaway of the mesh at $t = \{0.5, 1, 1.5, 2\}$ colored with ψ. Solution of problem (2.2) by the second-order nonlinear finite volume method with the backward Euler scheme (CUPW2-BE) and by the second-order nonlinear finite volume method with the Crank–Nicolson scheme (CUPW2-CN). The time step $\Delta t = 0.01$ and $K \approx 1$.

Nicolson scheme preserves the object's shape better than the backward Euler scheme while maintaining the monotonicity of solution.

Figure 10 illustrates the application of the nonlinear finite volume method for solving problem (2.2) at higher Courant numbers $K \approx 2$ ($\Delta t = 0.02$) and $K \approx 4$ ($\Delta t = 0.04$). For the backward Euler scheme at $K \approx 4$, the interface compression is not sufficiently high to preserve a sharp interface; however, $\psi \in [0,1]$ up to machine precision. The Crank–Nicolson scheme is not monotone at $K > 1$. Therefore, we further use the Crank–Nicolson scheme with $K \approx 1$ because it provides the best interface resolution and still a monotone solution.

The impact of the cubic mesh refinement up to the levels $L = 3$ and $L = 4$ is illustrated in Fig. 11. Even though the interface is captured better with a finer mesh, the interface distortions are more noticeable on finer meshes. In Fig. 12, we compare the effect of the adaptive choice of α (2.24) proposed in [26] with the conventional choice $\alpha = |\beta_f|$. The artifacts on the interface are less pronounced due to the adaptive

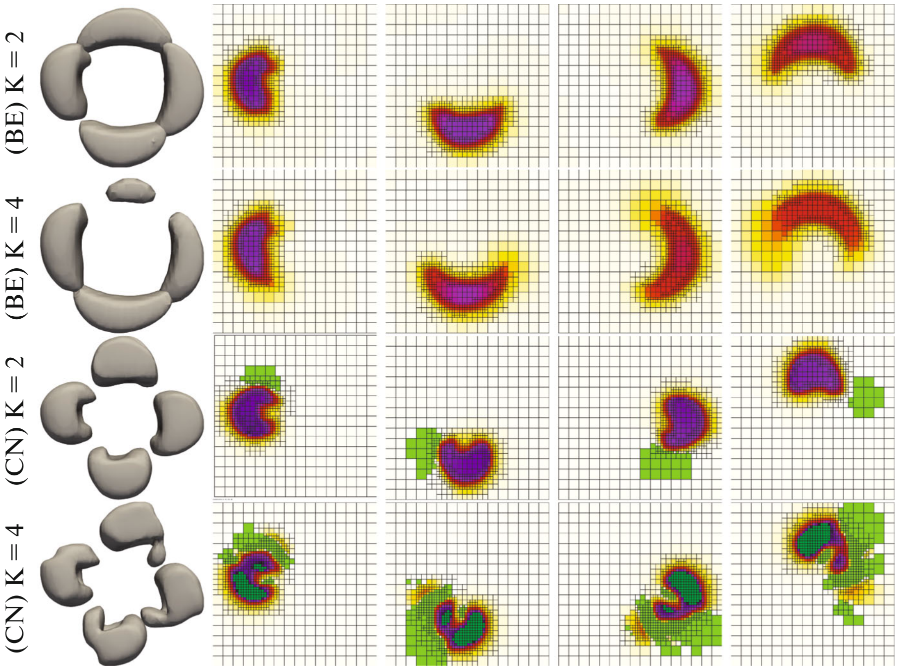

Fig. 10. Isosurface $\psi = 0.5$ and the middle ($z = 0.5$) cutaway of the mesh at $t = \{0.5, 1, 1.5, 2\}$ colored with ψ. Solution of problem (2.1) by the second-order nonlinear finite volume method with the backward Euler scheme (BE) and Crank–Nicolson scheme (CN) with the time step $\Delta t = 0.02$, $K \approx 2$ (top) and $\Delta t = 0.04$, $K \approx 4$ (bottom). Dark green and light green colors correspond to $\psi < 0$ and $\psi > 1$, respectively.

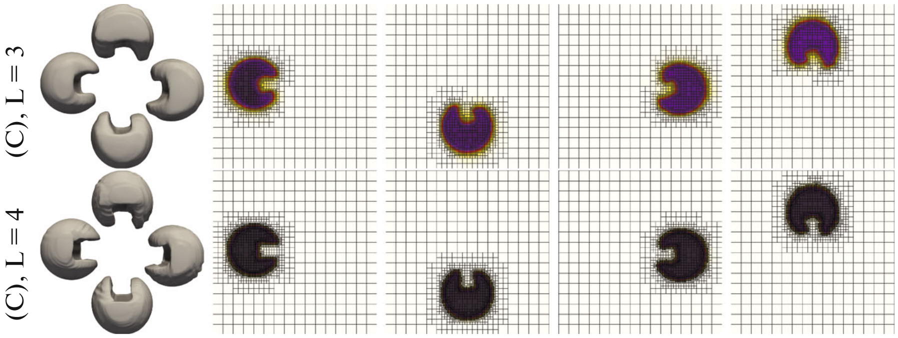

Fig. 11. Isosurface $\psi = 0.5$ and the middle ($z = 0.5$) cutaway of the mesh at $t = \{0.5, 1, 1.5, 2\}$ colored with ψ. Solution of problem (2.2) by the second-order nonlinear finite volume method with the Crank–Nicolson scheme on cubic meshes (C) with $L = 3$ refinement levels with the time step $\Delta t = 0.005$ (top) and $L = 4$ refinement levels with $\Delta t = 0.0025$ (bottom). In both cases, $K \approx 1$.

choice (2.24). This effect is less noticeable on coarser meshes ($L = 2$, $L = 3$). Note that we use a significantly larger time step than the conventional schemes.

The proposed method is implemented in the framework of the INMOST platform and, therefore, it can be used on parallel computers. The distribution of the cubic mesh with $L = 4$ refinement levels among 16 processors is illustrated in Fig. 13.

The dynamics of the number of cells and the number of Newton iteration steps in the simulation on a cubic mesh with $L = 3$ refinement levels is shown in Fig. 14.

Finally, we demonstrate the ability of the method to deal with general polyhedral meshes. We consider hexagonal and triangular prismatic meshes with $L = 2$ and $L = 3$ refinement levels, see Fig. 15. The

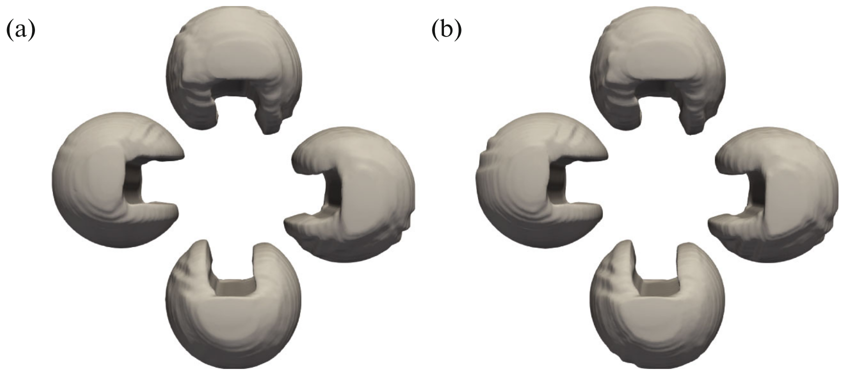

Fig. 12. Isosurface $\psi = 0.5$ at $t = \{0.5, 1, 1.5, 2\}$ calculated on a cubic mesh with $L = 4$ refinement levels. The adaptive choice of α by rule (2.24) (a) and conventional choice $\alpha = |\beta_f|$ (b).

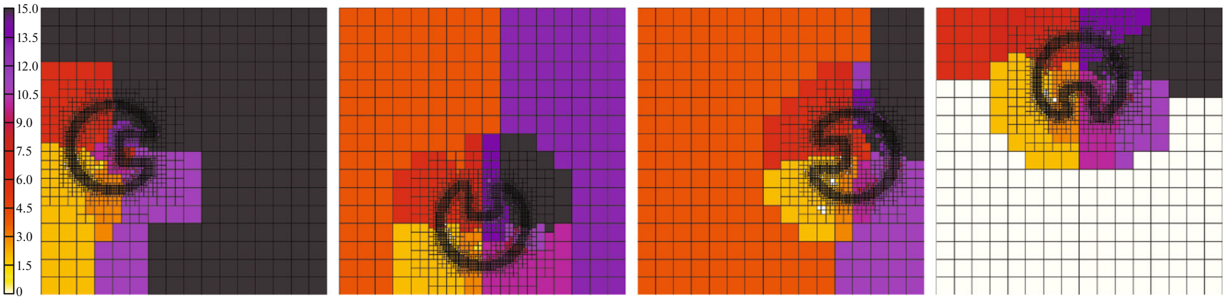

Fig. 13. Middle ($z = 0.5$) cutaway of the cubic mesh with $L = 4$ refinement levels at $t = \{0.5, 1, 1.5, 2\}$ colored with the processor index 0, 1, …, 15.

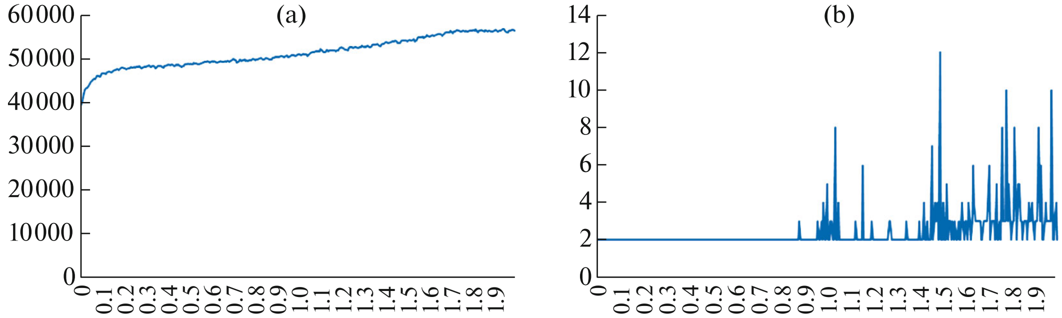

Fig. 14. Evolution of the number of cells (a) and the number of Newton method iterations (b) for solving problem (2.2) by the nonlinear finite volume method with the Crank–Nicolson scheme on a cubic mesh with $L = 3$ refinement levels.

meshes contain 16 prismatic layers in the direction z, and the resolution of the coarse mesh in the directions x and y approximately corresponds to the resolution of the $16 \times 16 \times 16$ cubic mesh.

4.2. Enright's Test

Consider Enright's test [61]. The velocity field $\mathbf{u} = [u, v, w]^{\mathrm{T}}$ is specified in the unit cube $\Omega = [0,1]^3$:

$$\begin{aligned} u &= 2\sin(\pi x)^2 \sin(2\pi y)\sin(2\pi z)\cos(\pi t/3), \\ v &= -\sin(2\pi x)\sin(\pi y)^2 \sin(2\pi z)\cos(\pi t/3), \qquad (4.2) \\ w &= -\sin(2\pi x)\sin(2\pi y)\sin(\pi z)^2 \cos(\pi t/3). \end{aligned}$$

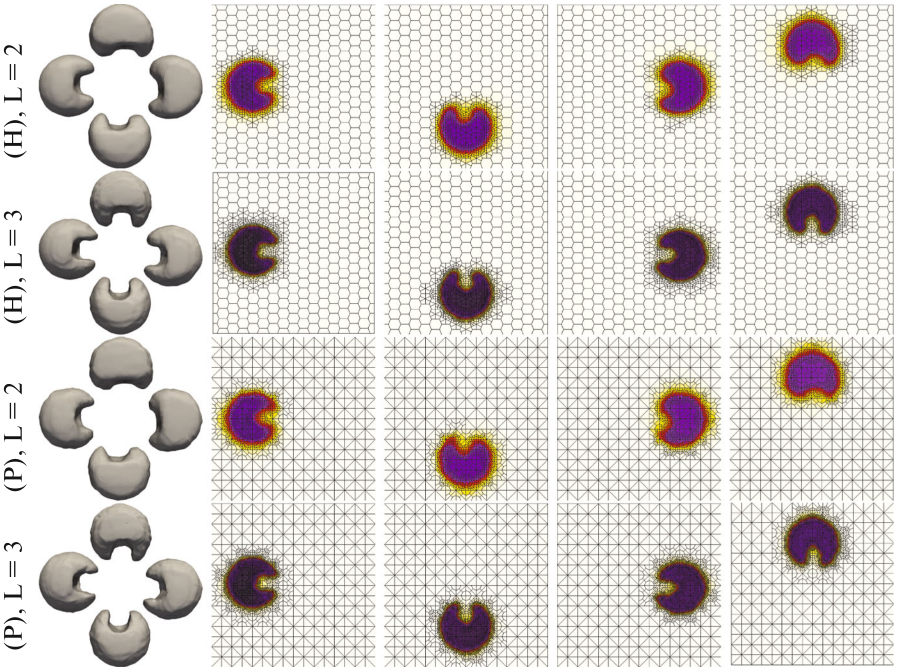

Fig. 15. Isosurface $\psi = 0.5$ and the middle ($z = 0.5$) cutaway of the mesh at $t = \{0.5, 1, 1.5, 2\}$ colored with ψ. Solution of problem (2.2) by the second-order nonlinear finite volume method with the Crank–Nicolson scheme on a hexagonal prismatic mesh (H) and triangular prismatic mesh (P) with $L = 2$ ($\Delta t = 0.01$) and $L = 3$ ($\Delta t = 0.005$) refinement levels. In both cases, $K \approx 1$.

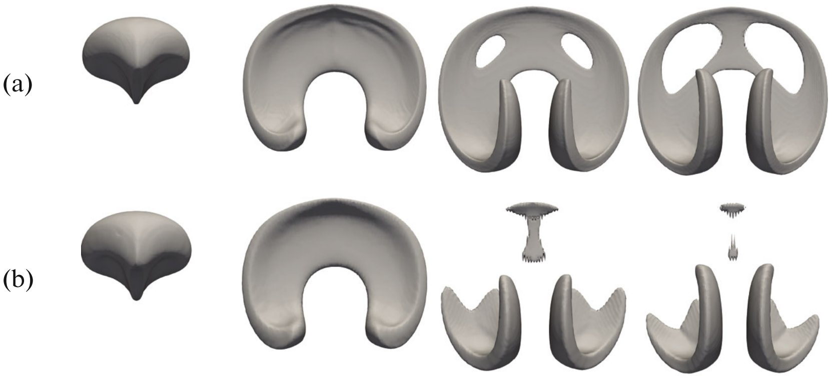

Fig. 16. Enright's test: Isosurface of $\psi = 0.5$ at $T = \{0.3, 0.75, 1.2, 1.5\}$. Cubic mesh with $L = 3$ (b) and $L = 4$ (a) refinement levels.

The problem is solved by the nonlinear finite volume method with the Crank–Nicolson scheme on the time interval $t \in [0,1.5]$. At the initial time $t = 0$, the advected object is defined by the sphere of radius $r = 0.15$ centered at $\mathbf{x} = [0.35, 0.35, 0.35]^T$. The solutions on the cubic meshes with the refinement levels $L = 3$ and $L = 4$ are depicted in Fig. 16. The object surface remains smooth, although the object resolution is much better on the finer mesh. The mesh resolution 16–128 (with $L = 3$ refinement levels) is insufficient for maintaining a simply connected isosurface $\psi = 0.5$.

5. CONCLUSIONS

In this paper, a nonlinear finite volume method for solving the interface advection–compression problem in the volume of fluid framework is proposed. The proposed scheme keeps the solution within the specified bounds [0,1], and it is applicable to adaptive meshes of general shape and admits large time steps. A possible application of this scheme is the fully implicit simulation of free surface flows.

In future, we are going to focus on less dissipative methods for advection, more accurate monotone time integration methods, and consider other adaptive strategies for choosing the interface compression parameter. A promising direction for improving the accuracy of schemes is the use of exponential integrators for flux approximation.

FUNDING

This work was supported by the Moscow Center of Fundamental and Applied Mathematics, project no. 075-15-2019-1624.

ACKNOWLEDGMENTS

Kirill Terekhov is grateful to Mallison and Hamdi Tchelepi for discussions on the application of the nonlinear finite volume method to the advection problem. Yashar Mehmani is acknowledged for the discussion of the interface compression problem.

REFERENCES

1. M. J. Ketabdari, "Free surface flow simulation using VOF method," *Numerical Simulation: From Brain Imaging to Turbulent Flows,* Ed. by Lpez-Ruiz (BoDBooks on Demand, 2016), Vol. 365.
2. K. D. Nikitin, M. A. Olshanskii, K. M. Terekhov, et al. "An adaptive numerical method for free surface flows passing rigidly mounted obstacles," Comput. Fluids **148**, 56–68 (2017).
3. Y. V. Vassilevski, K. Nikitin, M. Olshanskii, and K. Terekhov, "CFD technology for 3D simulation of large-scale hydrodynamic events and disasters," Russ. J. Numer. Anal. Math. Model. **27**, 399–412 (2012).
4. B. Kumar, M. Crane, and Y. Delauré, "On the volume of fluid method for multiphase fluid flow simulation," Int. J. Model. Simul. Sci. Comput. **4** (2), 1350002 (2013).
5. S. J. Ruuth and B. T. Wetton, "A simple scheme for volume-preserving motion by mean curvature," J. Sci. Comput. **19**, 373–384 (2003).
6. Y. Qi, J. Lu., R. Scardovelli, et al. "Computing curvature for volume of fluid methods using machine learning," J. Comput. Phys. **377**, 155–161 (2019).
7. K. D. Nikitin, M. A. Olshanskii, K. M. Terekhov, and Y. V. Vassilevski, "A splitting method for numerical simulation of free surface flows of incompressible fluids with surface tension," Comput. Methods Appl. Math. **15** (1), 59–77 (2015).
8. K. D. Nikitin, K. M. Terekhov, and Y. V. Vassilevski, "Two methods of surface tension treatment in free surface flow simulations," Appl. Math. Lett. **86**, 236–242 (2018).
9. S. McFadden and D. Browne, "A front-tracking model to predict solidification macrostructures and columnar to equiaxed transitions in alloy castings," Appl. Math. Model. **33**, 1397–1416 (2009).
10. R. Malladi and J. A. Sethian, "Level set methods for curvature flow, image enhancement, and shape recovery in medical images," *Visualization and Mathematics* (Springer, 1997), pp. 329–345.
11. S. Popinet, "Gerris: A tree-based adaptive solver for the incompressible Euler equations in complex geometries," J. Comput. Phys. **190**, 572–600 (2003).
12. C. Lalanne, Q. Magdelaine, F. Lequien, and J.-M. Fullana, "Numerical model using a volume-of-fluid method for the study of evaporating sessile droplets in both unpinned and pinned modes," Eur. J. Mech. B Fluids. (2021).
13. C. Kunkelmann and P. Stephan, "CFD simulation of boiling flows using the volume-of-fluid method within OpenFOAM," Numer. Heat Transf. A. **56**, 631–646 (2009).
14. L. Gamet, M. Scala, J. Roenby, et al. "Validation of volume-of-fluid OpenFOAM® isoadvector solvers using single bubble benchmarks," Comput. Fluids **213**, 104722 (2020).
15. A. Albadawi, D. Donoghue, A. Robinson, et al., "On the analysis of bubble growth and detachment at low capillary and bond numbers using volume of fluid and level set methods," Chem. Eng. Sci. **90**, 77–91 (2013).
16. E. G. Puckett, A. S. Almgren, J. B. Bell, et al. "A high-order projection method for tracking fluid interfaces in variable density incompressible flows," J. Comput. Phys. **130**, 269–282 (1997).
17. M. Sussman and E. G. Puckett, "A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows," J. Comput. Phys. **162,** 301–337 (2000).
18. P. Cifani, W. Michalek, G. Priems, et al., "A comparison between the surface compression method and an interface reconstruction method for the VOF approach," Comput. Fluids. **136**, 421–435 (2016).

19. H. Rusche, Computational fluid dynamics of dispersed two-phase flows at high phase fractions, Ph.D. thesis, Imperial College London (University of London), 2003.
20. Y. Okagaki, T. Yonomoto, M. Ishigaki, and Y. Hirose, “Numerical study on an interface compression method for the volume of fluid approach,” Fluids **6** (2), 80 (2021).
21. M. Aboukhedr, A. Georgoulas, M. Marengo, et al., Simulation of micro-flow dynamics at low capillary numbers using adaptive interface compression," Comput. Fluids **165**, 13–32 (2018).
22. J. K. Patel and G. Natarajan, “A generic framework for design of interface capturing schemes for multi-fluid flows,” Comput. Fluids **106**, 108–118 (2015).
23. A. Arote, M. Bade, and J. Banerjee, “An improved compressive volume of fluid scheme for capturing sharp interfaces using hybridization,” Numer. Heat Transf. B: Fundam. **79**, 29–53 (2020).
24. Y. Mehmani, Wrinkle-free interface compression for two-fluid flows, arXiv:1811.09744. 2018.
25. D. J. Piro and K. Maki, An adaptive interface compression method for water entry and exit, Tech. Rep. 2013-350, University of Michigan, Department of Naval Architecture and Marine Engineering, 2013.
26. H. Lee and S. H. Rhee, “A dynamic interface compression method for VOF simulations of high-speed planing watercraft,” J. Mech. Sci. Technol. **29**, 1849–1857 (2015).
27. J. A. Sethian and P. Smereka, “Level set methods for fluid interfaces,” Annu. Rev. Fluid Mech. **35**, 341–372 (2003).
28. D. Adalsteinsson and J. A. Sethian, “The fast construction of extension velocities in level set methods,” J. Comput. Phys. **148**, 2–22 (1999).
29. K. M. Terekhov, K. D. Nikitin, M. A. Olshanskii, and Y. V. Vassilevski, “A semi-Lagrangian method on dynamically adapted octree meshes,” Russ. J. Numer. Anal. Math. Model. **30**, 363–380 (2015).
30. K. Nikitin, M. Olshanskii, K. Terekhov, and Yu. V. Vassilevski, “Preserving distance property of level set function and simulation of free surface flows on adaptive grids,” *Numerical Geometry, Grid Generation and Scientific Computing (NUMGRID-2010),* 2010, pp. 25–32.
31. R. F. Ausas, E. A. Dari, and G. C. Buscaglia, “A geometric mass-preserving redistancing scheme for the level set function,” Int. J. Numer. Methods Fluids **65**, 989–1010 (2011).
32. Z. Ge, J.-C. Loiseau, O. Tammisola, and L. Brandt, “An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces,” J. Comput. Phys. **353**, 435–459 (2018).
33. B. M. Ningegowda, Z. Ge, G. Lupo, et al., “A mass-preserving interface-correction level set/ghost fluid method for modeling of three-dimensional boiling flows,” Int. J. Heat Mass Transf. **162**, 120382 (2020).
34. J.-L. Guermond, M. Q. de Luna, and T. Thompson, “An conservative antidiffusion technique for the level set method,” J. Comput. Appl. Math. **321**, 448–468 (2017).
35. B. Leonard and S. Mokhtari, “Beyond first-order upwinding: The ultra-sharp alternative for non-oscillatory steady-state simulation of convection,” Int. J. Numer. Methods Eng. **30**, 729–766 (1990).
36. L. Silva, C. Fontes, and P. Lage, “Front tracking in recirculating flows: A comparison between the TVD and RCM methods in solving the VOF equation,” Braz. J. Chem. Eng. **22** (1), 105–116 (2005).
37. C.-N. Lu, R.-Y. Wang, and J.-S. Sun, “WENO finite volume method for tracking moving interfaces on unstructured triangle meshes,” J. Hohai Univ. 01 (2009).
38. S. Pirozzoli, S. Di Giorgio, and A. Iafrati, “On algebraic TVD-VOF methods for tracking material interfaces,” Comput. Fluids **189**, 73–81 (2019).
39. M. Darwish and F. Moukalled, “Convective schemes for capturing interfaces of free-surface flows on unstructured grids,” Numer. Heat Transf. B: Fundam. **49**, 19–42 (2006).
40. O. Ubbink and R. Issa, “A method for capturing sharp fluid interfaces on arbitrary meshes,” J. Comput. Phys. **153**, 26–50 (1999).
41. Y.-Y. Tsui, S.-W. Lin, T.-T. Cheng, and T.-C. Wu, “Flux-blending schemes for interface capture in two-fluid flows,” Int. J. Heat Mass Transf. **52**, 5547–5556 (2009).
42. D. Zhang, C. Jiang, D. Liang, et al., “A refined volume-of-fluid algorithm for capturing sharp fluid interfaces on arbitrary meshes,” J. Comput. Phys. **274**, 709–736 (2014).
43. E. Bertolazzi and G. Manzini, “A second-order maximum principle preserving finite volume method for steady convection-diffusion problems,” SIAM J. Numer. Anal. **43**, 2172–2199 (2005).
44. J. Droniou and C. L. Potier, “Construction and convergence study of schemes preserving the elliptic local maximum principle,” SIAM J. Numer. Anal. **49**, 459–490 (2011).
45. K. Lipnikov, D. Svyatskiy, and Y. V. Vassilevski, “Minimal stencil finite volume scheme with the discrete maximum principle,” Russ. J. Numer. Anal. Math. Model. **27**, 369–386 (2012).
46. A. Chernyshenko and Y. Vassilevski, “A finite volume scheme with the discrete maximum principle for diffusion equations on polyhedral meshes,” *Finite Volumes for Complex Applications VII-Methods and Theoretical Aspects* (Springer, 2014), pp. 197–205.
47. K. M. Terekhov, B. T. Mallison, and H. A. Tchelepi, “Cell-centered nonlinear finite-volume methods for the heterogeneous anisotropic diffusion problem,” J. Comput. Phys. **330**, 245–267 (2017).

48. J. Lee and D. A. Sheen, “A parallel method for backward parabolic problems based on the Laplace transformation,” SIAM J. Numer. Anal. **44**, 1466–1486 (2006).

49. R. S. Varga, *Matrix Iterative Analysis, Prentice-Hall Series in Automatic Computation* (Prentice-Hall, Englewood Cliffs, 1962).

50. A. Jameson, “Analysis and design of numerical schemes for gas dynamics, 1: Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence,” Int. J. Comput. Fluid Dyn. **4** (3–4), 171–218 (1995).

51. K. Lipnikov, D. Svyatskiy, and Y. V. Vassilevski, “Anderson acceleration for nonlinear finite volume scheme for advection-diffusion problems,” SIAM J. Sci. Comput. **35** (2), (2013).

52. B. Perot, “Conservation properties of unstructured staggered mesh schemes,” J. Comput. Phys. **159**, 58–89 (2000).

53. R. Younis, H. A. Tchelepi, and K. Aziz, “Adaptively localized continuation—Newton method—nonlinear solvers that converge all the time,” Soc. Pet. Eng. J. **15**, 526–544 (2010).

54. Y. Vassilevski, K. Terekhov, K. Nikitin, and I. Kapyrin, *Parallel Finite Volume Computation on General Meshes* (Springer Nature, 2020).

55. K. Terekhov, “Parallel multilevel linear solver within INMOST platform,” *Russian Supercomputing Days* (Springer, 2020), pp. 297–309.

56. K. Terekhov “Greedy dissection method for shared parallelism in incomplete factorization within INMOST platform,” *Russian Supercomputing Days* (Springer, 2021), pp. 87–101.

57. K. Terekhov and Y. Vassilevski, “Mesh modification and adaptation within INMOST programming platform, *Numerical Geometry, Grid Generation and Scientific Computing* (Springer, 2019), pp. 243–255.

58. K. Terekhov, “Parallel dynamic mesh adaptation within INMOST platform,” *Russian Supercomputing Days* (Springer, 2019), pp. 313–326.

59. G. Karypis, K. Schloegel, and V. Kumar, Parmetis parallel graph partitioning and sparse matrix ordering library, Tech. Rep. 97-060, Univ. of Minnesota, Department of Computer Science and Engineering, 1997.

60. G. Karypis and V. Kumar, MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0, 2009. http://www.cs.umn.edu/ metis.

61. D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, “A hybrid particle level set method for improved interface capturing,” J. Comput. Phys. **183**, 83–116 (2002).

62. J. Ahrens, B. Geveci, and C. Law, “Paraview: An end-user tool for large data visualization,” *The Visualization Handbook,* 2005, Vol. 717, no. 8.

Translated by A. Klimontovich