DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS SERIES S
Volume **14**, Number **3**, March **2021**

doi:10.3934/dcdss.2020350

pp. **865–879**

FLUX BALANCED APPROXIMATION WITH LEAST-SQUARES GRADIENT FOR DIFFUSION EQUATION ON POLYHEDRAL MESH

Peter Frolkovič* and Karol Mikula

Faculty of Civil Engineering, Slovak University of Technology
Department of Mathematics and Descriptive Geometry
Radlinského 11, 810 05 Bratislava, Slovak Republic

Jooyoung Hahn, Dirk Martin and Branislav Basara

Advanced Simulation Technologies, AVL List GmbH
Hans-List Platz 1, 8010 Graz, Austria

Abstract. A numerical method for solving diffusion problems on polyhedral meshes is presented. It is based on a finite volume approximation with the degrees of freedom located in the centers of computational cells. A numerical gradient is defined by a least-squares minimization for each cell, where we suggest a restricted form in the case of discontinuous diffusion coefficient. The flux balanced approximation is proposed without numerically computing the gradient itself at the faces of computational cells in order to find a normal diffusive flux. To apply the method for parallel computations with a 1-ring neighborhood, we use an iterative method to solve the obtained system of algebraic equations. Several numerical examples illustrate some advantages of the proposed method.

1. **Introduction.** In computer-aided engineering used in industry and many related research areas, one has to numerically solve mathematical models that are based on partial differential equations (PDE) involving diffusion terms. The applications in focus include computational fluid dynamics [5, 29, 14], transport and flow in porous media [4, 16], electromagnetism [20] and some others. If such an application is considered for a computational domain of nontrivial shape and form, then one of the most difficult parts is a meshing of the domain.

In several industrial or academic software tools [27, 19, 15, 7, 29], the meshing is resolved by using polyhedral meshes to discretize three-dimensional computational domains in space. In order to approximate a solution to the PDE for such meshes with the degrees of freedom in the centers of computational cells, a reasonable choice of the discretization is finite volume methods (FVM), especially if the considered model is based on conservation laws. In this paper, we are interested in applying the FVM for scalar (stationary or time dependent) diffusion equation that can be seen as a representative model for related terms in more complex nonlinear PDEs.

2020 *Mathematics Subject Classification.* Primary: 65N08; Secondary: 65M08.

Key words and phrases. Finite volume methods, polyhedral mesh, least-squares gradient, deferred correction, cross diffusion.

The first and second author are supported by grants VEGA 1/0709/19 and 1/0436/20 and APVV-15-0522.

* Corresponding author: Peter Frolkovič.

The essential ingredient of FVM is the approximation of fluxes at the shared faces of two neighboring computational cells. Such approximation for simplified geometric settings can be obtained by considering a one-dimensional local description that straightforwardly gives a two-point flux approximation for the diffusion flux [13]. For complex polyhedral meshes that can not be avoided in computational practice used in industry, the two-point flux approximation can not be used in general. This fact was recognized in the scientific community several decades ago. Since then, many results have been published on how to apply the FVM for general polyhedral meshes. For a review and some comparison of different approaches, we refer to [12] and references therein.

We are interested in discretization methods that are based on a least-squares gradient approximation [23, 11, 17, 24] for polyhedral meshes. Such approximation gives the numerical gradient in the cell centers, where the degrees of freedom are located. To compute the numerical gradient in one cell, the degrees of freedom in all neighboring cells with an inverse distance squared weighting are used.

In [23, 11, 17, 24, 22, 10, 3, 28] the cell centered numerical gradients are furthermore used to approximate the gradients at the faces of computational cells by applying some linear combination of the cell centered gradients in the cells sharing the face. The face gradient is then split into two components. The first component is defined along the direction given by two cell centers, and it can be approximated by the two-point flux approximation. The second component of the gradient (that is sometimes called "the cross diffusion term" [11, 22, 3, 28]) is approximated by several variants of the method, see [9] and the references therein.

The aim of this paper is to use an approximation of the normal diffusion flux at the faces, in contrast to previously mentioned methods, without approximating the gradient itself at the faces. The proposed approximation is based on standard assumptions on the continuity of the solution and the diffusive flux at the faces of computational cells, even if the diffusion coefficient is discontinuous across the face. Furthermore, inspired by [30], we apply a restricted least-squares approximation of gradients that respects the discontinuity of the diffusion coefficient and leads to a distance weighted harmonic mean of diffusion coefficient. Finally, we use an iterative method in a deferred correction manner [6, 11, 3] to use a 1-ring neighborhood assumption when a local stiffness matrix is assembled.

The rest of the paper is organized as follows. In Section 2, we introduce all ingredients of the proposed method to approximate an elliptic operator. In Section 3, we compare it with a method based on the face gradient approximations [9] for the Poisson equation with a variable diffusion coefficient. In Section 4, several numerical experiments are shown to illustrate the properties and advantages of the proposed method. Finally, we conclude in Section 5.

2. **Finite volume method on polyhedral mesh.**

Let $\Omega \subset \mathbb{R}^3$ be a bounded Lipschitz domain, and $\partial\Omega$ be its boundary. In this section, we describe a numerical approximation of the linear elliptic operator

$$L\phi = -\nabla \cdot (h(\mathbf{x})\nabla\phi), \quad \mathbf{x} \in \Omega \,, \tag{1}$$

using a Dirichlet boundary condition given by

$$\phi(\gamma) = \phi_D(\gamma), \quad \gamma \in \partial\Omega.$$

The coefficients h and ϕ_D are given piecewise continuous functions. The approximation will be based on the finite volume method [13].

Let $\Omega \subset \mathbb{R}^3$ be discretized by open non-overlapping polyhedral sets $\Omega_p \subset \Omega$, the cells, with $|\Omega_p| \neq 0$ and with $\mathcal{I}$ being the set of cell indices $p \in \mathcal{I}$, i.e., $\bar{\Omega} = \bigcup_{p\in\mathcal{I}} \bar{\Omega}_p$. To indicate the neighbor cells of Ω_p, we consider only the cells Ω_q with intersections having non-zero areas:

$$\mathcal{N}_p = \{q \in \mathcal{I} : |\partial\Omega_q \cap \partial\Omega_p| \neq 0\}.$$

An internal face is denoted by $e_f \subset \partial\Omega_q \cap \partial\Omega_p$, and $|e_f| \neq 0$. The set of indices for all internal faces e_f of the computational domain is denoted by $\mathcal{F}$. Furthermore, we define the set $\mathcal{B}$ of indices for all boundary faces $e_b \subset \partial\Omega_p \cap \partial\Omega$ for $p \in \mathcal{I}$ with $|e_b| \neq 0$. Analogously, the faces of a cell Ω_p, $p \in \mathcal{I}$ are split into two disjoint sets $\mathcal{F}_p$ and $\mathcal{B}_p$. Throughout the rest of paper, the subscript $f \in \mathcal{F}_p$ indicates an index of the internal face e_f, and the subscript $b \in \mathcal{B}_p$ an index of the boundary face e_b if $\mathcal{B}_p \neq \emptyset$.

For a face e_f, $f \in \mathcal{F}_p$, the outward normal vector is denoted by $\mathbf{n}_{pf}$, where we set the length of the normal vector to the area of the face, i.e., $|\mathbf{n}_{pf}| = |e_f|$. Clearly, if $f \in \mathcal{F}_p \cap \mathcal{F}_q$ for $q \in \mathcal{N}_p$ then $\mathbf{n}_{pf} = -\mathbf{n}_{qf}$. Analogously, we define the outward normal vectors $\mathbf{n}_b$ with $|\mathbf{n}_b| = |e_b|$ for $b \in \mathcal{B}_p$.

To apply a finite volume approximation of (1), we integrate it over Ω_p and use the Gauss divergence theorem:

$$\int_{\Omega_p} L\phi = -\int_{\Omega_p} \nabla \cdot (h\nabla\phi) = -\int_{\partial\Omega_p} h\nabla\phi \cdot \mathbf{n}\,.$$

The main ingredient of the proposed method is an approximation of the diffusion flux:

$$\int_{\partial\Omega_p} h\nabla\phi \cdot \mathbf{n} \approx \sum_{f\in\mathcal{F}_p} h\nabla\phi \cdot \mathbf{n}_{pf} + \sum_{b\in\mathcal{B}_p} h\nabla\phi \cdot \mathbf{n}_b\,. \tag{2}$$

Particularly, we have to specify in (2) the approximation of the normal diffusive fluxes $h\nabla\phi \cdot \mathbf{n}_{pf}$ and $h\nabla\phi \cdot \mathbf{n}_b$ through the internal and boundary faces e_f and e_b, respectively. The approximation has to be based on the unknowns ϕ_p that approximate the solution ϕ in the cell centers $\mathbf{x}_p$, i.e., $\phi_p \approx \phi(\mathbf{x}_p)$, $p \in \mathcal{I}$, and on the values $\phi_b = \phi(\gamma_b)$ given by Dirichlet boundary conditions that we apply at the face centers $\gamma_b \in e_b$ for $b \in \mathcal{B}$. Note that in the case of Neumann boundary conditions, defined on a boundary face e_b, one could directly replace the boundary flux in (2) by a value given by this condition.

We begin with the approximation of internal flux at e_f, $f \in \mathcal{F}_p$. To do so, we denote a directional vector $\mathbf{d}_{pf}$ obtained from the cell center $\mathbf{x}_p$, $p \in \mathcal{I}$ and the face center $\mathbf{x}_f$, $f \in \mathcal{F}_p$,

$$\mathbf{d}_{pf} = \mathbf{x}_f - \mathbf{x}_p\,. \tag{3}$$

Analogously to (3), other directional vectors are defined with the indices indicating uniquely the definition of the vector, e.g., $\mathbf{d}_{\alpha\beta} = \mathbf{x}_\beta - \mathbf{x}_\alpha$.

For the cell centers $\mathbf{x}_p$ and $\mathbf{x}_q$ and the face e_f, $f \in \mathcal{F}_p \cap \mathcal{F}_q$, we find points $\mathbf{x}_{p'}$ and $\mathbf{x}_{q'}$ that lie on a line obtained by a prolongation of the normal vector $\mathbf{n}_{pf}$ through the face center $\mathbf{x}_f$ such that the directional vectors $\mathbf{d}_{pp'}$ and $\mathbf{d}_{qq'}$ are perpendicular to $\mathbf{n}_{pf}$, see Figure 1 for an illustration of the notation. Clearly,

$$\mathbf{d}_{pp'} = \mathbf{d}_{pf} - \frac{\mathbf{n}_{pf} \cdot \mathbf{d}_{pf}}{\mathbf{n}_{pf} \cdot \mathbf{n}_{pf}} \mathbf{n}_{pf}\,, \quad \mathbf{d}_{qq'} = \mathbf{d}_{qf} - \frac{\mathbf{n}_{pf} \cdot \mathbf{d}_{qf}}{\mathbf{n}_{pf} \cdot \mathbf{n}_{pf}} \mathbf{n}_{pf}\,.$$

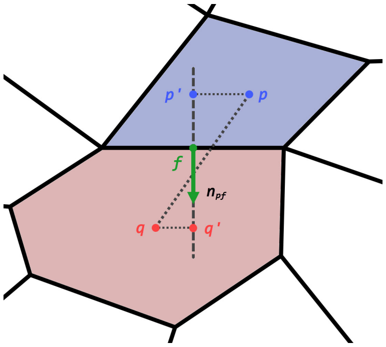

FIGURE 1. An illustration of the basic notation used in the flux approximation.

Note that the position of the point $\mathbf{x}_{p'}$ is different with respect to each face e_f, $f \in \mathcal{F}_p$, that we do not emphasize in the notation.

Analogously, the point $\mathbf{x}_{p'}$ related to the boundary face e_b, $b \in \mathcal{B}_p$ can be determined such that

$$\mathbf{d}_{pp'} = \mathbf{d}_{pb} - \frac{\mathbf{n}_b \cdot \mathbf{d}_{pb}}{\mathbf{n}_b \cdot \mathbf{n}_b} \mathbf{n}_b \, .$$

Let us denote temporarily the auxiliary values of numerical solution at $\mathbf{x_{p'}}$ and $\mathbf{x}_f$ by $\phi_{p'}$ and ϕ_f, respectively. Having such values, we can exploit the two-point flux approximation:

$$\nabla \phi \cdot \mathbf{n}_{pf} \approx \frac{|e_f|}{|\mathbf{d}_{p'f}|} \left(\phi_f - \phi_{p'} \right) \, , \tag{4}$$

and, analogously,

$$\nabla \phi \cdot \mathbf{n}_{qf} \approx \frac{|e_f|}{|\mathbf{d}_{q'f}|} \left(\phi_f - \phi_{q'} \right) .$$

For a boundary face, we obtain

$$\nabla \phi \cdot \mathbf{n}_b \approx \frac{|e_b|}{|\mathbf{d}_{p'f}|} \left(\phi_b - \phi_{p'} \right) . \tag{5}$$

Now, following e.g. [13, 25], we can determine for the internal face $e_f \subset \partial\Omega_p \cap \partial\Omega_q$ the value ϕ_f from the balance of fluxes at the face using the piecewise constant approximation of the diffusion coefficient h, i.e. $h_p = h(\mathbf{x}_p)$, $p \in \mathcal{I}$, by requiring

$$h_p \frac{|e_f|}{|\mathbf{d}_{p'f}|} (\phi_f - \phi_{p'}) = -h_q \frac{|e_f|}{|\mathbf{d}_{q'f}|} (\phi_f - \phi_{q'}).$$

Consequently, we obtain

$$\phi_f = \frac{|\mathbf{d}_{q'f}| h_p \phi_{p'} + |\mathbf{d}_{p'f}| h_q \phi_{q'}}{|\mathbf{d}_{q'f}| h_p + |\mathbf{d}_{p'f}| h_q} . \tag{6}$$

Using (4), (5), and (6), the finite volume approximation of (1) gives:

$$\int_{\Omega_p} L\phi \approx \sum_{f \in \mathcal{F}_p} h_p \frac{|e_f|}{|\mathbf{d}_{p'f}|} \left(\phi_f - \phi_{p'} \right) + \sum_{b \in \mathcal{B}_p} h_p \frac{|e_b|}{|\mathbf{d}_{p'b}|} \left(\phi_b - \phi_{p'} \right) \, . \tag{7}$$

We still have to define the unspecified value $\phi_{p'}$ in (6) and (7). To do so, we define least-squares gradient approximations $\nabla \phi_p$, $p \in \mathcal{I}$ by using the weighted least-squares minimizer that appears robust for many types of polyhedral meshes [26],

$$\nabla\phi_p := \operatorname{argmin}_{\mathbf{y}} \left(\sum_{q\in\mathcal{N}_p} \frac{(\phi_p + \mathbf{y}\cdot\mathbf{d}_{pq} - \phi_q)^2}{|\mathbf{d}_{pq}|^2} + \sum_{b\in\mathcal{B}_p} \frac{(\phi_p + \mathbf{y}\cdot\mathbf{d}_{pb} - \phi_b)^2}{|\mathbf{d}_{pb}|^2} \right). \tag{8}$$

An explicit form of the gradient $\nabla\phi_p$ in (8) can be obtained from

$$\mathbf{M}\nabla\phi_p = \sum_{q\in\mathcal{N}_p} \frac{\mathbf{d}_{pq}}{|\mathbf{d}_{pq}|^2}\left(\phi_q - \phi_p\right) + \sum_{b\in\mathcal{B}_p} \frac{\mathbf{d}_{pb}}{|\mathbf{d}_{pb}|^2}\left(\phi_b - \phi_p\right), \tag{9}$$

where the coefficient matrix $\mathbf{M}$ 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 $\mathbf{M}$ is always a symmetric and invertible matrix because $|\mathcal{N}_p \cup \mathcal{B}_p| \geq 4$. Since a multiplication of the inverse of matrix and vectors in the right-hand side of (9) can be pre-computed, a computational cost of computing the gradient by the least-squares method (8) is low.

Having the least-squares gradients $\nabla\phi_p$, $p\in\mathcal{I}$, we can now use the approximation

$$\phi_{p'} \approx \phi_p + \nabla\phi_p\cdot\mathbf{d}_{pp'}\,,$$

and similarly for $\phi_{q'}$. Using it together with (6), we obtain

$$\begin{aligned}\phi_f - \phi_{p'} &= \frac{|\mathbf{d}_{q'f}|h_p\phi_{p'} + |\mathbf{d}_{p'f}|h_q\phi_{q'}}{|\mathbf{d}_{q'f}|h_p + |\mathbf{d}_{p'f}|h_q} - \frac{|\mathbf{d}_{q'f}|h_p + |\mathbf{d}_{p'f}|h_q}{|\mathbf{d}_{q'f}|h_p + |\mathbf{d}_{p'f}|h_q}\phi_{p'} \\ &= \frac{|\mathbf{d}_{p'f}|h_q}{|\mathbf{d}_{q'f}|h_p + |\mathbf{d}_{p'f}|h_q}(\phi_q + \nabla\phi_q\cdot\mathbf{d}_{qq'} - \phi_p - \nabla\phi_p\cdot\mathbf{d}_{pp'}),\end{aligned}$$

and similarly for boundary faces,

$$\phi_b - \phi_{p'} = \phi_b - \phi_p - \nabla\phi_p\cdot\mathbf{d}_{pp'}\,.$$

Putting it all together, the finite volume approximation (7) takes the form

$$\begin{aligned}\int_{\Omega_p} L\phi \approx &\sum_{q\in\mathcal{N}_p} h_{pq}\frac{|e_f|}{|\mathbf{d}_{p'q'}|}\left(\phi_q - \phi_p + \nabla\phi_q\cdot\mathbf{d}_{qq'} - \nabla\phi_p\cdot\mathbf{d}_{pp'}\right) + \\ &+ \sum_{b\in\mathcal{B}_p} h_p\frac{|e_b|}{|\mathbf{d}_{p'b}|}\left(\phi_b - \phi_p - \nabla\phi_p\cdot\mathbf{d}_{pp'}\right),\end{aligned} \tag{10}$$

where

$$h_{pq} = \frac{|\mathbf{d}_{p'f}| + |\mathbf{d}_{q'f}|}{\dfrac{|\mathbf{d}_{p'f}|}{h_p} + \dfrac{|\mathbf{d}_{q'f}|}{h_q}} \tag{11}$$

is the distance weighted harmonic mean of diffusion coefficient.

Remark 1. We note that the gradient approximations $\nabla\phi_p$ computed by the weighted least-squares minimization in (8) is appropriate if the gradient is continuous in a neighborhood of $\mathbf{x}_p$. This is not the case in general if the diffusion coefficient has a discontinuity in the same neighborhood. Inspired by [30], we use in this case a *restricted least-squares gradient* $\nabla\phi_p$ by restricting the set $\mathcal{N}_p$ in (8) only to the neighborhood where the diffusion coefficient is continuous.

3. **Comparison with other finite volume approximations.** In order to compare the presented approximation (10) with other related methods [9], we apply it to solve the boundary value problem

$$-\nabla \cdot (h(\mathbf{x})\nabla \phi) = g(\mathbf{x})\,,\ \mathbf{x} \in \Omega\,, \quad \phi(\gamma) = \phi_D(\gamma),\ \gamma \in \partial\Omega\,, \tag{12}$$

where the source term g in (12) is an integrable function. Using the approximation (10), we obtain a system of linear algebraic equations for the unknowns ϕ_p, $p \in \mathcal{I}$,

$$\begin{aligned} 0 = &\sum_{q \in \mathcal{N}_p} h_{pq} \frac{|e_f|}{|\mathbf{d}_{p'q'}|} \left(\phi_q - \phi_p + \nabla\phi_q \cdot \mathbf{d}_{qq'} - \nabla\phi_p \cdot \mathbf{d}_{pp'}\right) + \\ &+ \sum_{b \in \mathcal{B}_p} h_p \frac{|e_b|}{|\mathbf{d}_{p'b}|} \left(\phi_b - \phi_p - \nabla\phi_p \cdot \mathbf{d}_{pp'}\right) + |\Omega_p| g_p, \end{aligned} \tag{13}$$

where $g_p = g(\mathbf{x}_p)$.

The system (13) can be solved directly by some linear solvers. Nevertheless, in the case of using a 2-ring neighborhood when a corresponding matrix of the linear system is constructed, the presence of $\nabla\phi_q$ in (13) makes a large number of non-zero elements in the matrix. Such non-zero elements can be widely distributed in the case of a polyhedron mesh that might bring a high computational cost to solve the matrix system. Moreover, the 2-ring neighborhood structure in decomposed domains for parallel computation needs to use much more overlapping layers compared to the 1-ring neighborhood structure, and the number of overlapped cells among the decomposed domains is directly related to the communication cost in parallel computations.

To avoid the aforementioned issues, we use the 1-ring neighborhood with an iterative procedure of a deferred correction approach to solve (13) with the k-th iteration having the form

$$0 = \sum_{q \in \mathcal{N}_p} \left(\zeta_{pq}(\phi_q^k - \phi_p^k) + \eta_{pq}^{k-1}\right) + \sum_{b \in \mathcal{B}_p} \left(\zeta_{pb}(\phi_b - \phi_p^k) + \eta_{pb}^{k-1}\right) + |\Omega_p| g_p, \tag{14}$$

where

$$\zeta_{pq} = \frac{|e_f| h_p h_q}{|\mathbf{d}_{q'f}| h_p + |\mathbf{d}_{p'f}| h_q}, \quad \zeta_{pb} = \frac{|e_b| h_p}{|\mathbf{d}_{p'b}|},$$

and

$$\eta_{pq}^{k-1} = \zeta_{pq}\left(\nabla\phi_q^{k-1} \cdot \mathbf{d}_{qq'} - \nabla\phi_p^{k-1} \cdot \mathbf{d}_{pp'}\right), \quad \eta_{pb}^{k-1} = -\zeta_{pb}\left(\nabla\phi_p^{k-1} \cdot \mathbf{d}_{pp'}\right).$$

The system of linear algebraic equations (14) for the unknowns ϕ_p^k, $k = 1, 2, \ldots$ is now characterized only by 1-ring neighborhood, and it defines the method proposed in this paper. The choice ϕ_p^0 shall be chosen depending on a form of PDE which involves the elliptic operator L. To solve (12), we set simply $\phi_p^0 = 0$.

Rewriting (14) as a matrix equation

$$\mathbf{A}\phi^k = \mathbf{F}(\phi^{k-1}),$$

the k^{th} iteration stops at the smallest $k = K$ such that the residual error is smaller than a chosen threshold ϵ:

$$\frac{\sum_{p\in\mathcal{I}} \left| \left(\mathbf{A}\phi^K - \mathbf{F}(\phi^K)\right)_p \right|}{\sum_{p\in\mathcal{I}} |\mathbf{A}_{pp}|} < \epsilon, \tag{15}$$

where $\left(\mathbf{A}\phi^K - \mathbf{F}(\phi^K)\right)_p$ is the p^{th} component of the vector in the bracket.

3.1. Methods based on face gradient approximations.

We compare the method (14) with a method that directly approximates the gradient at internal faces e_f, $f \in \mathcal{F}$. First, we denote for the face $e_f \subset \partial\Omega_p \cap \partial\Omega_q$,

$$\begin{aligned} \lambda_{pq} &= \frac{\mathbf{n}_{pf}\cdot\mathbf{n}_{pf}}{\mathbf{n}_{pf}\cdot\mathbf{d}_{pq}} = \lambda_{qp}\,, \\ \mathbf{t}_{pq} &= \mathbf{n}_{pf} - \lambda_{pq}\mathbf{d}_{pq}\,, \end{aligned} \tag{16}$$

and, analogously, for the boundary face $e_b \in \partial\Omega_p$, $b \in \mathcal{B}$,

$$\begin{aligned} \lambda_{pb} &= \frac{\mathbf{n}_{b}\cdot\mathbf{n}_{b}}{\mathbf{n}_{b}\cdot\mathbf{d}_{pb}}\,, \\ \mathbf{t}_{pb} &= \mathbf{n}_{b} - \lambda_{pb}\mathbf{d}_{pb}\,. \end{aligned} \tag{17}$$

Clearly, $\mathbf{t}_{pq}\cdot\mathbf{n}_{pf} = 0$ and $\mathbf{t}_{pb}\cdot\mathbf{n}_b = 0$. In the contrast to the approach (4), where the normal diffusive flux in (2) is directly approximated, here the following form is used,

$$\nabla\phi\cdot\mathbf{n}_{pf} = \lambda_{pq}\nabla\phi\cdot\mathbf{d}_{pq} + \nabla\phi\cdot\mathbf{t}_{pq}\,, \tag{18}$$

$$\nabla\phi\cdot\mathbf{n}_{b} = \lambda_{pb}\nabla\phi\cdot\mathbf{d}_{pb} + \nabla\phi\cdot\mathbf{t}_{pb}\,. \tag{19}$$

The first term on the right hand sides of (18) and (19) can be approximated by the two point approximation. To approximate the remaining cross-diffusion term, several approaches are suggested in the literature, see [9] and the references therein, that require an approximation $\nabla\phi_f$, $f \in \mathcal{F}$ of the face gradient. A canonical choice [9, 10, 17, 18, 28] is a linear convex combination:

$$\nabla\phi_f = \omega_{qf}\nabla\phi_p + \omega_{pf}\nabla\phi_q\,, \tag{20}$$

where $f \in \mathcal{F}_p \cap \mathcal{F}_q$, $\omega_{qf} > 0$, $\omega_{pf} > 0$ and $\omega_{qf} + \omega_{pf} = 1$. We adopt the form used in [17]:

$$\omega_{pf} = \frac{|\mathbf{d}_{pf}|}{|\mathbf{d}_{pf}| + |\mathbf{d}_{qf}|}\,.$$

Having such numerical face gradients, one can approximate the normal diffusion flux in (2) using (18),

$$h\nabla\phi\cdot\mathbf{n}_{pq} \approx h_f\lambda_{pq}(\phi_q - \phi_p) + h_f\nabla\phi_f\cdot\mathbf{t}_{pq}\,, \tag{21}$$

where the face diffusion coefficient h_f at $e_f \subset \partial\Omega_p \cap \partial\Omega_q$ can be defined as a distance weighted arithmetic mean of the diffusion coefficients h_p and h_q analogously to (20),

$$h_f = \omega_{qf}h_p + \omega_{pf}h_q\,.$$

For a boundary face e_b, $b \in \mathcal{B}_p$, we use in (2) the approximation

$$h\nabla\phi\cdot\mathbf{n}_b \approx h_p\lambda_{pb}(\phi_b - \phi_p) + h_p\nabla\phi_p\cdot\mathbf{t}_{pb}\,.$$

To sum up, we obtain discrete equations that can be solved by the previously mentioned iterative approach:

$$0 = \sum_{q \in \mathcal{N}_p} \left(h_f \lambda_{pq} (\phi_q^k - \phi_p^k) + \mu_{pq}^{k-1} \right) + \sum_{b \in \mathcal{B}_p} \left(h_p \lambda_{pb} (\phi_b - \phi_p^k) + \mu_{pb}^{k-1} \right) + |\Omega_p| g_p, \tag{22}$$

where

$$\mu_{pq}^{k-1} = h_f \mathbf{t}_{pq} \cdot (\omega_{qf} \nabla \phi_p^{k-1} + \omega_{pf} \nabla \phi_q^{k-1}),$$
$$\mu_{pb}^{k-1} = h_p \mathbf{t}_{pb} \cdot \nabla \phi_p^{k-1}.$$

Remark 2. We note that also other choices of λ_{pq} and $\mathbf{t}_{pq}$ in (16) and (17) can be found in [17, 28, 9]. The variant in (16) and (17) is called the over-relaxation approach in [17, 9], where also other two choices are presented. The over-relaxed approach is recommended in [17, 9] over other variants because of its robustness, especially for highly non-orthogonal meshes. We note that the authors in [10] make a slight modification of (21), when the term $\nabla \phi \cdot \mathbf{t}_{pq}$ in (18) is not approximated by $\nabla \phi_f \cdot \mathbf{t}_{pq}$ as in (21), but

$$\nabla \phi \cdot \mathbf{t}_{pq} = \nabla \phi \cdot (\mathbf{n}_{pf} - \lambda_{pq} \mathbf{d}_{pq}) \approx \nabla \phi_f \cdot \mathbf{n}_{pf} - \lambda_{pq} \left(\frac{\nabla \phi_p + \nabla \phi_q}{2} \right) \cdot \mathbf{d}_{pq}. \tag{23}$$

The meaning of (23) is revisited in [9], where it is claimed to be more robust and efficient method than other cross diffusion methods [9, 18]. Therefore, we compare the proposed method (14) with the method (23) for a representative numerical example, see section 4.2.

4. Numerical experiments.

We begin with straightforward test examples to check the behavior of the proposed numerical method. The examples are computed in different computational domains Ω using polyhedral meshes generated by `AVL FIRE`™. Specific requirements on exact positions of boundary nodes for some domains are met by an application of mesh deformation based on a radial basis function interpolation [8].

First, we solve the Poisson equation that is obtained by choosing some exact solution ϕ and some variable diffusion coefficient h, and then computing a corresponding right hand side g in (12). Furthermore, we compute an example with a piecewise constant diffusion coefficient in a concentric spherical domain with available exact solution to show a comparison between the proposed numerical method (14) and the method in (23), and between the standard least-squares gradient and the restricted one. Finally, a nonlinear time-dependent example, inspired by the Perona-Malik model, is computed.

The examples are considered with Dirichlet boundary conditions that are constructed either from the chosen exact solutions, if available, or by some piecewise constant values. The initial guess for the numerical solution in the iterative procedure is always the zero vector for stationary problems. We stop the iterations according to the criteria (15) with $\epsilon = 1.0 \times 10^{-12}$, and the final index K in (15) is presented in results.

We compute two representative norms of the error. The error E_2 denotes the standard L_2 norm of the difference between the numerical and exact solutions, the error G_1 denotes the standard L_1 norm of the difference between the gradient of numerical and exact solutions.

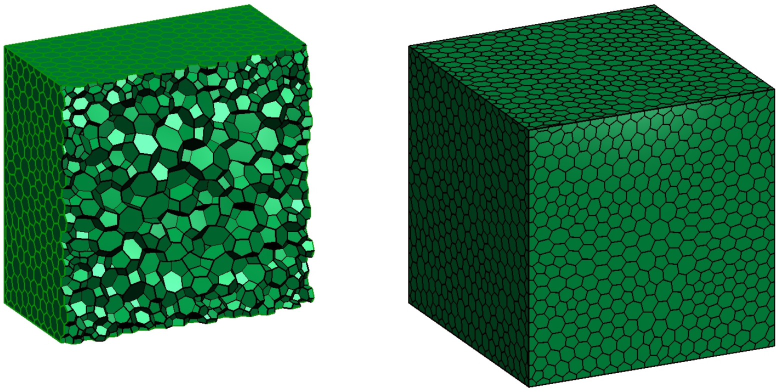

FIGURE 2. The cubic domain and polyhedral mesh with average discretization size $\ell = .19$ used in the Sections 4.1 and 4.4.

ℓ	K	E_2	EOC	G_1	EOC
1.90e-1	10	3.69e-3		7.97e-2	
9.52e-2	10	1.17e-3	1.66	3.22e-2	1.31
4.76e-2	9	3.27e-4	1.85	1.41e-2	1.19
2.48e-2	9	7.53e-5	2.26	6.08e-3	1.30

TABLE 1. The error norms and EOCs for the example in the Section 4.1.

Each computational mesh is characterized by an average discretization size:

$$\ell \equiv \frac{1}{|\mathcal{I}|} \sum_{p \in \mathcal{I}} |\Omega_p|^{\frac{1}{3}} .$$

To compute an Experimental Order of Convergence (EOC), we always create four related meshes by choosing smaller volumes in a certain ratio that give smaller discretization sizes denoted here by ℓ^l, $l = 1, 2, 3, 4$. The corresponding EOC^l for $l = 2, 3, 4$ in the case of L_2 norms E_2^l is then computed by

$$\mathrm{EOC}^l = \frac{\log \frac{E_2^l}{E_2^{l-1}}}{\log \frac{\ell^l}{\ell^{l-1}}}$$

and analogously for G_1^l.

4.1. **Example in a cubic domain.** The first example of (12) is given by

$$h = \frac{x^2 + y^2 + z^2}{2} + 1, \quad \phi = \sin^2(x) + \cos^2(y) + z^3 . \tag{24}$$

The computational domain is a cube of the side length 2.5 that is discretized by polyhedral meshes, see Figure 2. In Table 1, one can observe the expected behavior for the chosen norms of errors. We note that the number of iterations K for which the desired reduction of the residual error is reached, is approximately 10 for each mesh.

4.2. **Example in a concentric spherical domain.** In this section, we present an example of computing an electric potential in a concentric spherical domain Ω that is filled by two different dielectric materials, see Figure 3. The domain is charged 0 on its outer sphere and q_0 on its inner sphere. We solve the stationary diffusion equation with a piecewise constant diffusion coefficient with respect to the three concentric spheres $\mathcal{S}_{r_1}$, $\mathcal{S}_{r_2}$, and $\mathcal{S}_{r_3}$. The center of the spheres is $(0,0,0)$ and $\partial\Omega = \mathcal{S}_{r_1} \cup \mathcal{S}_{r_3}$ and $\mathcal{S}_{r_2}$ is the internal boundary:

$$\begin{aligned} -\nabla \cdot (h \nabla \phi) &= 0, \quad \text{in } \Omega \\ \phi(\mathbf{x}) &= 0, \quad \mathbf{x} \in \mathcal{S}_{r_3} \\ \phi(\mathbf{x}) &= q_0, \quad \mathbf{x} \in \mathcal{S}_{r_1} \end{aligned}$$

with

$$h = \begin{cases} h_1 & r_1 \le r \le r_2 \\ h_2 & r_2 < r \le r_3 \end{cases} \tag{25}$$

and $r = |\mathbf{x}|$, $r_1 = 0.5$, $r_2 = 1.0$, $r_3 = 1.5$, $h_1 = 1$, and $h_2 = 100$. The motivation is to compare the method (14) based on the flux balanced approximation with the method (22) with (23) based on the face gradient approximation that differ mainly in the treatment of piecewise constant diffusion coefficient. Moreover, we show a difference between the application of the full least-squares gradient computations and the restricted one, where the latter one respects the discontinuity of the diffusion coefficient, see Remark 1.

The exact solution is the standard electric potential of concentric spheres:

$$\phi = \begin{cases} \frac{1}{4\pi}\left(\frac{1}{h_1}\left(\frac{1}{r} - \frac{1}{r_2}\right) + \frac{1}{h_2}\left(\frac{1}{r_2} - \frac{1}{r_3}\right)\right), & r_1 \le r \le r_2 \\ \frac{1}{4\pi}\frac{1}{h_2}\left(\frac{1}{r} - \frac{1}{r_3}\right), & r_2 \le r \le r_3 \end{cases}, \tag{26}$$

Note that ϕ is a continuous function, but the gradient $\nabla\phi$ has a discontinuity at $r = 1$. The Dirichlet boundary conditions are used at $\partial\Omega$ given by the exact solution and $q_0 = \phi(r_1)$.

The results for the method (14) based on the flux balanced approximation are presented in Table 2 with two variants; the full and the restricted least-squares gradient. Interestingly, there is practically no difference between the two variants for the error E_2 of the L_2 norm of the difference between the exact and the numerical solution. Nevertheless, the error G_1 for the L_1 norm of the difference of the exact and the numerical gradient is clearly smaller when the restricted least-squares gradient is used which respects the discontinuity of the diffusion coefficient.

Analogous results can be reported for the method (22) based on the face gradient approximation that are given in Table 3. Nevertheless, the results are clearly less precise when compared to the method based on the flux balanced approximation for both variants of the least-squares gradient.

4.3. **Example in a complex domain.** In this section, we present an illustrative example in a complex domain for which some structured mesh is unavailable and where an unstructured mesh might be the only available option. The domain and the corresponding polyhedral mesh are presented in Figure 4, where Ω is a rectangular box with the two missing subvolumes of nontrivial shapes, a cathode and an anode, which are red and blue surfaces in Figure 4, respectively.

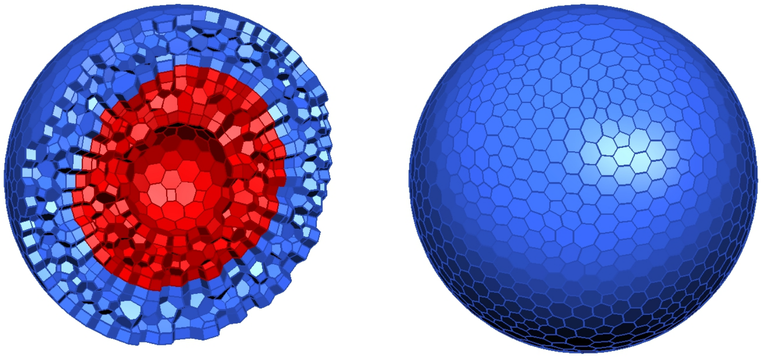

FIGURE 3. The domain used for the examples in the Section 4.2. The left picture shows a cut for which the mesh is visible. The blue and red parts visualize the subdomains with the different constant diffusion coefficients. The right picture shows the outer boundary with the corresponding surface mesh.

ℓ	K	E_2	EOC	G_1	EOC	K	E_2	EOC	G_1	EOC
.121	8	6.43e-5		3.41e-3		8	6.43e-5		1.07e-3	
.090	8	3.08e-5	2.42	2.26e-3	1.36	8	3.08e-5	2.42	5.79e-4	2.03
.072	8	1.80e-5	2.44	1.70e-3	1.29	8	1.80e-5	2.44	3.93e-4	1.75
.060	8	1.15e-5	2.46	1.34e-3	1.27	8	1.15e-5	2.46	2.88e-4	1.71

TABLE 2. The error norms and the EOCs for the example in the concentric spherical domain in the Section 4.2 for the flux balanced method. The full least square gradient approximation is presented in the columns from 2 to 6, and the restricted one in the columns from 7 to 11.

ℓ	K	E_2	EOC	G_1	EOC	K	E_2	EOC	G_1	EOC
.121	10	2.99e-4		6.44e-3		10	2.83e-4		3.49e-3	
.090	9	2.07e-4	1.21	4.53e-3	1.16	9	1.94e-4	1.23	2.49e-3	1.11
.072	9	1.59e-4	1.20	3.48e-3	1.18	9	1.48e-4	1.22	1.92e-3	1.17
.060	9	1.28e-4	1.16	2.81e-3	1.16	9	1.19e-4	1.19	1.56e-3	1.16

TABLE 3. The error norms and the EOCs for the example in the concentric spherical domain in the Section 4.2 using the method based on the face gradient approximation. The full least square gradient approximation is presented in the columns from 2 to 6, and the restricted one in the columns from 7 to 11.

We compute the Laplace equation with Dirichlet boundary conditions given by three constant values defined on the three parts of the boundary $\partial\Omega$:

$$\begin{aligned} -\nabla\cdot(\nabla\phi) &= 0, \quad \text{in } \Omega, \\ \phi(\mathbf{x}) &= 700, \quad \mathbf{x} \text{ on a cathode}, \\ \phi(\mathbf{x}) &= 0, \quad \mathbf{x} \text{ on a anode}, \\ \phi(\mathbf{x}) &= 300, \quad \mathbf{x} \text{ elsewhere}. \end{aligned}$$

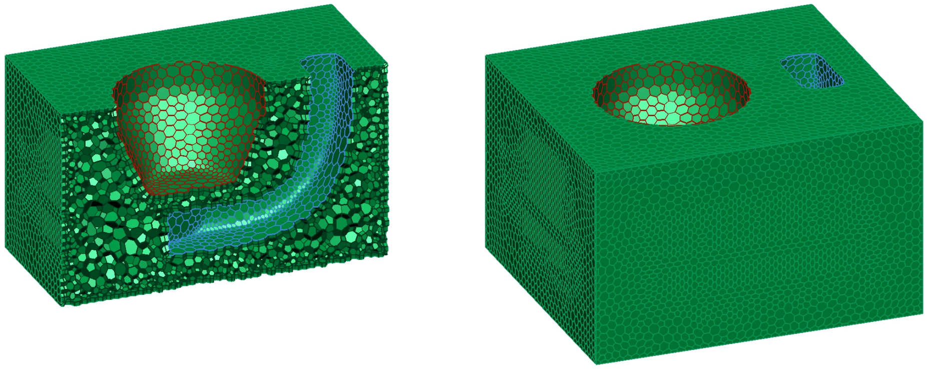

FIGURE 4. The cut (left) and the surface (right) of the complex domain used for the example in the Section 4.3.

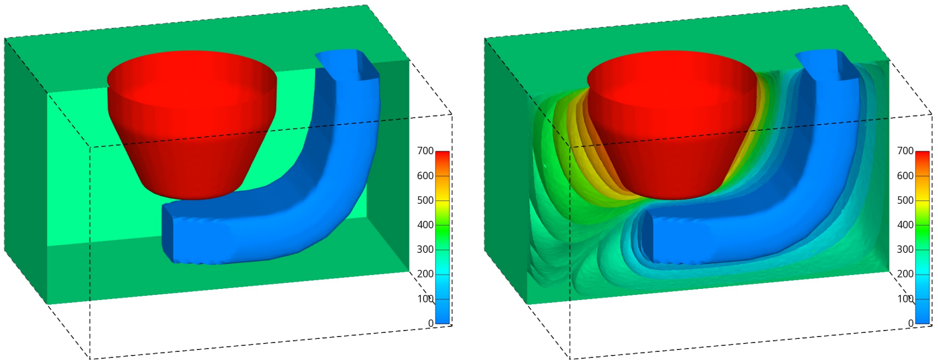

FIGURE 5. The three surfaces of the boundary of computational domain (left) and the isosurfaces of numerical solution (right) for the example in the Section 4.3.

In Figure 5, the isosurfaces of the solution are illustrated. We note that no difficulties occurred to obtain the numerical solution on such mesh, and that the iterative procedure required $K = 17$, 16, 12, 11 iterations for the average discretization size $\ell = .0664$, .0417, .0227, .0129, respectively.

4.4. **Example with Perona-Malik equation.** To test a nontrivial time dependent type of nonlinear diffusion equation, we solve the following problem inspired by a Perona-Malik type equation [1, 21] with a right hand side for which one can easily fabricate an exact solution,

$$\frac{\partial \phi}{\partial t} - \nabla \cdot (h \nabla \phi) = g, \tag{27}$$

where

$$h = \frac{1}{\sqrt{1 + |\nabla \phi|^2}}\,. \tag{28}$$

ℓ	K	E_2	EOC	G_1	EOC
1.90e-1	7	1.84e-4		1.70e-2	
9.52e-2	6	5.31e-5	1.79	6.66e-3	1.36
4.76e-2	6	1.55e-5	1.78	2.96e-3	1.17
2.48e-2	5	4.03e-6	2.07	1.29e-3	1.28

Table 4. The error norms and the EOCs for the example with Perona-Malik equation in the Section 4.4.

The exact solution is chosen in the form

$$\phi = \frac{x^2 + y^2 + z^2}{4} + t^3 \,, \tag{29}$$

and it defines the initial and the Dirichlet boundary conditions used with (27). Note that we solve the equation (27) numerically on polyhedral mesh only to test its behavior for such nontrivial case, although typical applications of the Perona-Malik equation is, e.g., a noise reduction in images represented by regular grids.

The time discretization is realized using a nonlinear Crank-Nicolson scheme, see e.g. [2]. The nonlinear algebraic system of equations is solved using the previously described iterative method, where the nonlinear diffusion coefficient is evaluated using the available values of the numerical solution from a previous iteration. Moreover, the initial guess for $k = 0$ in the iterative method is chosen from the previous time level.

The EOC is presented in the Table 4, where the expected behavior of the EOCs can be observed. It is interesting to note that identical error norms (when rounded to 3 valid digits) as in Table 4 can be obtained when the number of iterations $K = 2$ is fixed.

5. **Conclusions.** In this paper, we present the finite volume method for the diffusion equations that can be used on polyhedral meshes. It is based on the flux balanced approximation using the least-squares gradient with the distance weighted harmonic mean of piecewise continuous diffusion coefficient. We show that in the case of discontinuous diffusion coefficient, it is suitable to use the restricted least-squares gradient that respects the discontinuity in the approximation.

The proposed method that we call the flux balanced approximation with the least-squares gradient, can be used in the iterative way, where each iteration uses a matrix in the 1-ring neighborhood. Such iterative method is convenient, especially for time dependent problems, possibly with nonlinear diffusion coefficient, where the initial guess for each iteration can be used from the previous time step. The method is implemented in the software `AVL FIRE`$^{\text{TM}}$ and tested on several examples. The results of the example with the discontinuous diffusion coefficient show a higher accuracy of the proposed method in comparison with the method based on the explicit face gradient approximation [10].

REFERENCES

[1] L. Alvarez, P.-L. Lions and J.-M. Morel, Image selective smoothing and edge detection by nonlinear diffusion. II, *SIAM J. Numer. Anal.*, **29** (1992), 845–866.

[2] M. Balažovjech and K. Mikula, A higher order scheme for a tangentially stabilized plane curve shortening flow with a driving force, *SIAM J. Sci. Comput.*, **33** (2011), 2277–2294.

[3] B. Basara, Employment of the second-moment turbulence closure on arbitrary unstructured grids, *Int. J. Numer. Methods Fluids*, **44** (2004), 377–407.

[4] P. Bastian, *Numerical Computation of Multiphase Flows in Porous Media*, Ph.D thesis, Habilitationsschrift Univeristät Kiel, 1999.

[5] J. Blazek, *Computational Fluid Dynamics: Principles and Applications*, Elsevier/Butterworth Heinemann, Amsterdam, 2015.

[6] K. Böhmer, P. Hemker and H. J. Stetter, The defect correction approach, in *Defect Correction Methods*, Comput. Suppl., 5, Springer, Vienna, 1984, 1–32.

[7] N. Cinosi, S. Walker, M. Bluck and R. Issa, CFD simulation of turbulent flow in a rod bundle with spacer grids (MATIS-H) using STAR-CCM+, *Nuclear Engrg. Design*, **279** (2014), 37–49.

[8] A. de Boer, M. S. van der Schoot and H. Bijl, Mesh deformation based on radial basis function interpolation, *Comput. Structures*, **85** (2007), 784–795.

[9] I. Demirdžić, On the discretization of the diffusion term in finite-volume continuum mechanics, *Numer. Heat Tr. B-Fund.*, **68** (2015), 1–10.

[10] I. Demirdžić, I. Horman and D. Martinović, Finite volume analysis of stress and deformation in hygro-thermo-elastic orthotropic body, *Comp. Meth. Appl. Mech. Engrg.*, **190** (2000), 1221–1232.

[11] I. Demirdžić and S. Muzaferija, Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology, *Comp. Meth. Appl. Mech. Engrg.*, **125** (1995), 235–255.

[12] J. Droniou, R. Eymard, T. Gallouët and R. Herbin, A unified approach to mimetic finite difference, hybrid finite volume and mixed finite volume methods, *Math. Models Methods Appl. Sci.*, **20** (2010), 265–295.

[13] R. Eymard, T. Gallouët and R. Herbin, Finite volume methods, in *Handbook of Numerical Analysis, Vol. VII*, Handb. Numer. Anal., VII, North-Holland, Amsterdam, 2000, 713–1020.

[14] J. H. Ferziger, M. Perić and R. L. Street, *Computational Methods for Fluid Dynamics*, Springer, Cham, 2020.

[15] A. Fluent, Release 15.0, *Theory Guide, November.*

[16] P. Frolkovič, M. Lampe and G. Wittum, Numerical simulation of contaminant transport in groundwater using software tools of r^3t, *Comput. Vis. Sci.*, **18** (2016), 17–29.

[17] H. Jasak, *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*, Ph.D thesis, Imperial College London (University of London), 1996.

[18] H. Jasak and A. D. Gosman, Automatic resolution control for the finite volume method. Part 1, *Numer. Heat Tr. B-Fund.*, **38** (2000), 237–256.

[19] H. Jasak, A. Jemcov, Z. Tukovic et al., Openfoam: A C++ library for complex physics simulations, in *International Workshop on Coupled Methods in Numerical Dynamics*, 1000, IUC Dubrovnik, Croatia, 2007, 1–20.

[20] J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York, 2012.

[21] J. Kačur and K. Mikula, Solution of nonlinear diffusion appearing in image smoothing and edge detection, *Appl. Numer. Math.*, **17** (1995), 47–59.

[22] S. R. Mathur and J. Y. Murthy, A pressure-based method for unstructured meshes, *Numer. Heat Tr. B-Fund.*, **31** (1997), 195–215.

[23] S. Muzaferija, *Adaptive Finite Volume Method for Flow Prediction Using Unstructured Meshes and Multigrid Approach*, Ph.D thesis, University of London, 1994.

[24] S. Muzaferija and D. Gosman, Finite-volume CFD procedure and adaptive error control strategy for grids of arbitrary topology, *J. Comput. Phys.*, **138** (1997), 766–787.

[25] B. Niceno, A three dimensional finite volume method for incompressible Navier-Stokes equations on unstructured staggered grids, *ECCOMAS CFP*, **2006**.

[26] E. Sozer, C. Brehm and C. C. Kiris, Gradient calculation methods on arbitrary polyhedral unstructured meshes for cell-centered CFD solvers, 52nd Aerospace Sciences Meeting, 2014.

[27] R. Tatschl, B. Basara, J. Schneider, K. Hanjalic, M. Popovac, A. Brohmer and J. Mehring, Advanced turbulent heat transfer modeling for IC-engine applications using AVL FIRE, in *Int. Multidimensional Engine Modelling*, 2, User's Group Meeting, Detroit, MI, 2006, 1–10.

[28] Y.-Y. Tsui and Y.-F. Pan, A pressure-correction method for incompressible flows using unstructured meshes, *Numer. Heat Tr. B-Fund.*, **49** (2006), 43–65.

[29] J. Tu, G.-H. Yeoh and C. Liu, *Computational Fluid Dynamics: A Practical Approach*, Elsevier/Butterworth Heinemann, Amsterdam, 2013.

[30] L. White, R. Panchadhara and D. Trenev, Flow simulation in heterogeneous porous media with the moving least-squares method, *SIAM J. Sci. Comput.*, **39** (2017), B323–B351.

Received January 2019; 1st revision September 2019; 2nd revision November 2019.

E-mail address: branislav.basara@avl.com
E-mail address: peter.frolkovic@stuba.sk
E-mail address: jooyoung.hahn@avl.com
E-mail address: dirk.martin@avl.com
E-mail address: karol.mikula@stuba.sk