Computational Geosciences (2004) 8:301–324
DOI 10.1007/s10596-004-3771-1

The mimetic finite difference method on polygonal meshes for diffusion-type problems [*]

Y. Kuznetsov [a], K. Lipnikov [b,**] and M. Shashkov [b]
[a] University of Houston, Department of Mathematics, Houston, TX 77204, USA
E-mail: kuz@math.uh.edu
[b] Los Alamos National Laboratory, MS B284, Los Alamos, NM 87545, USA
E-mail: {lipnikov;shashkov}@lanl.gov

Received 22 December 2003; accepted 10 September 2004

New mimetic discretizations of diffusion-type equations (for instance, equations modeling single phase Darcy flow in porous media) on unstructured polygonal meshes are derived. The first order convergence rate for the fluid velocity and the second-order convergence rate for the pressure on polygonal, locally refined and non-matching meshes are demonstrated with numerical experiments.

Keywords: diffusion equation, locally conservative method, mimetic discretization

1. Introduction

The determining factor for reliability, accuracy, and efficiency of numerical simulations in porous media applications is accurate locally conservative discretizations. Practical experience shows that the most effective discrete approximations mimic the underlying properties of original continuum differential operators. One such approximations, the mimetic finite difference (MFD) method, has been applied successfully to several applications including diffusion processes on both conformal [7,9,16,21] and locally refined [13] meshes, magnetic diffusion and electromagnetics [8], continuum mechanics [15], and gas dynamics [4]. The convergence of the MFD method on triangular and quadrilateral meshes has been proved in [2,3].

As mathematical modeling of fluid flow in reservoirs becomes more sophisticated, the need for discretization methods that handle meshes with mixed types of elements has become clear. In this paper, we derive new mimetic discretizations on unstructured polygonal meshes. Generally speaking, most meshes used in applications are polygonal

---

[*] The work was partly performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36. The U.S. Government's right to retain a non-exclusive, royalty free license in and to any copyright is acknowledged. The research of the first author was supported by a grant from the Los Alamos Computer Science Institute (LACSI).

[**] Corresponding author.

meshes. For instance, locally adaptive triangular meshes are polygonal meshes consisting of triangular and degenerate quadrilateral elements. Another example comes from applications with nonmatching meshes. Here, we may consider the mesh points on the interface between two nonmatching meshes as vertices of a conformal polygonal mesh.

In this paper, we consider the diffusion problem formulated as a system of the first-order equations for the mass balance and the fluid velocity. The system models single phase Darcy flow in porous medium applications. We begin with considering each mesh polygon as an independent domain and generate an independent discretization for this polygon. The mimetic technique uses the discrete flux and divergence operators which are adjoint to each other. The system of element-based discretizations is closed by imposing continuity conditions for the pressure and normal velocity component on polygon edges.

The earlier MFD method shows reasonable convergence rate for the pressure variable on polygonal meshes consisting of *only* convex elements. Unfortunately, it does not preserve linear solutions, the convergence rate for the velocity variable is below optimal, and it cannot be extended to meshes with degenerate elements. The new MFD method is more accurate than the old one and has optimal convergence rates for both scalar and vector variables. Moreover, the new method supports nonconvex polygons and degenerate polygons, i.e. polygons with s vertices which can be described as polygons with $s-1$ vertices.

The development of the new MFD method on unstructured polygonal meshes became possible after performing a deep analysis of recent theoretical results concerning new finite element prolongation operators in H_{div} on polygonal and polyhedral meshes [12]. These operators produce the finite element vector functions which have constant divergence in each polygon. In this paper, some of the ideas from [12] are used to derive an accurate scalar product in the space of discrete velocities – one of the key elements of the mimetic technique. It is pertinent to note that the new mimetic discretization is the novel discretization. Generally speaking, it cannot be derived from a mimetic discretization on a triangular mesh by eliminating some degrees of freedom (e.g., by computing a Shur complement).

Nowadays, the use of polygonal meshes is limited by a small number of accurate discretization schemes. We mention here the finite volume scheme proposed in [19]. The scheme is exact for linear solutions but results in a nonsymmetric coefficient matrix. Therefore, it requires the use of nontraditional iterative solvers. The new MFD method is also exact for linear solutions but results in an algebraic problem with a symmetric positive definite matrix. Therefore, the problem may be solved with the conjugate gradient method. We believe that the MFD method developed in this paper will make polygonal meshes more attractive for engineering applications.

The paper outline is as follows. In section 2, we formulate a model elliptic boundary value problem and describe the general framework of the MFD method. In section 3, we analyze the new scalar product in the space of discrete velocities. In section 4, we prove that the resulting numerical scheme is exact for uniform flows. An efficient solution technique based on static condensation and computational results are given in section 5.

2. The mimetic finite difference method

2.1. *Formulation of the problem*

Let us consider a model elliptic boundary value problem that in porous medium applications models single phase Darcy flow:

$$\begin{aligned} \operatorname{div} \mathbf{F} &= b, \\ \mathbf{F} &= -\mathbf{K} \operatorname{grad} p. \end{aligned} \tag{1}$$

The problem is posed in a bounded polygonal domain $\Omega \subset \Re^2$ and is subject to appropriate boundary conditions on $\partial\Omega$. Here p denotes a scalar function that we refer to as the pressure, **F** denotes a vector function that we refer to as the velocity, **K** denotes a full tensor representing the rock permeability divided by the fluid viscosity, and b denotes a source function.

Let Ω_h be a non-overlapping conformal partition of Ω onto polygonal elements e_i,

$$\Omega_h = \bigcup_{i=1}^{m} e_i. \tag{2}$$

The elements e_i are closed simply-connected polygons with nonzero angles and edges. The number of polygon edges is equal to the number of polygon vertices. The conformal partition implies that (a) if $e_i \cap e_j$ consists of exactly one point, then it is a common vertex of e_i and e_j and (b) if for $i \neq j$, $e_i \cap e_j$ consists of more than one point, then $e_i \cap e_j$ is the union of common edges of e_i and e_j, i.e.

$$e_i \cap e_j = \bigcup_{k=1}^{s_{ij}} L_{ij}^k,$$

where L_{ij}^k denotes a mesh edge common to e_i and e_j and s_{ij} is the number of those edges (see figure 2). In other words, there are no gaps and overlaps in the mesh.

The class of such partitions is very wide and includes meshes used in many applications. For instance, locally refined (LR) meshes with hanging nodes are among them. If we consider a hanging node as the additional vertex of the corresponding element, we get a polygonal partition (2) containing degenerate polygons, i.e. polygons with angles equal to 180° between some adjacent edges (see section 5.4 for a few examples of such meshes). A nonmatching mesh is another example of a polygonal mesh. If we consider the mesh points on the interface between two nonmatching meshes as vertices of a conformal mesh, we get a polygonal partition (2) containing degenerate (possibly nonconvex) polygons (see section 5.6 for a few examples of such meshes).

2.2. Discretization on element

Let us introduce operators $\mathcal{G}$ and $\mathcal{D}$ for a polygon e by

$$\mathcal{G}\, p = -\mathbf{K}\,\mathrm{grad}\, p, \qquad \mathcal{D}\mathbf{F} = \begin{cases} \mathrm{div}\,\mathbf{F} & \text{on } e, \\ -\mathbf{F}\cdot\mathbf{n} & \text{on } \partial e, \end{cases}$$

where ∂e denotes the boundary of e. We shall refer to $\mathcal{G}$ and $\mathcal{D}$ as the flux and extended divergence operators, respectively. The operator $\mathcal{D}$ acts from the space $H_{\mathrm{div}}(e)$ to the space $L_2(e) \times L_2(\partial e)$. The operators $\mathcal{G}$ and $\mathcal{D}$ are adjoint to each other. To show this property we introduce the following scalar products:

$$(\mathbf{F},\mathbf{G})_{X,e} = \int_e \mathbf{F}\cdot\mathbf{K}^{-1}\mathbf{G}\,\mathrm{d}V \quad \text{and} \quad (p,q)_{Q,e} = \int_e pq\,\mathrm{d}V + \oint_{\partial e} pq\,\mathrm{d}l,$$

in the space X of velocities and in the space Q of pressures, respectively. Using the above notations we may rewrite the Gauss–Green formula

$$\int_e \mathbf{F}\cdot(\mathbf{K}^{-1}\mathcal{G}p)\,\mathrm{d}V - \int_e p\,\mathrm{div}\,\mathbf{F}\,\mathrm{d}V = -\oint_{\partial e} p\,\mathbf{F}\cdot\mathbf{n}\,\mathrm{d}l \tag{3}$$

in the equivalent form:

$$(\mathbf{F},\mathcal{G}p)_{X,e} = (p,\ \mathcal{D}\mathbf{F})_{Q,e}.$$

The last expression clearly states that the flux and extended divergence operators are adjoint to each other, i.e.

$$\mathcal{G} = \mathcal{D}^{*}.$$

The MFD method produces discretizations of these operators which are adjoint to each other with respect to scalar products in the discrete velocity and pressure spaces.

The *first* step of the MFD method is to specify the degrees of freedom for physical variables p and $\mathbf{F}$ and their location. We assume that medium in each mesh element is homogeneous, but material properties may vary between elements. The discrete pressures are defined on an element, p^0, and at mid-points of its edges, $p^1, p^2, \ldots, p^s$, where s is the number of edges of this element (see, e.g., hexagon in figure 1 where $s = 6$). The discrete velocities are defined as edge-based normal components, $f^1, f^2, \ldots, f^s$, located at the mid-points of polygon edges. For instance, f^1 approximates the dot product of $\mathbf{F}$ with the outward unit normal $\mathbf{n}^1$, i.e. $f^1 \approx \mathbf{F}\cdot\mathbf{n}^1$.

The *second* step of the MFD method is to equip the spaces of discrete pressures and velocities with scalar products. We denote the vector space of pressure unknowns (for a polygon e) by Q^d. The dimension of Q^d is equal to $s+1$, where s is the number of polygon edges. Thus, any vector $\vec{p} \in Q^d$, can be written as $\vec{p} = (p^0, p^1, \ldots, p^s)^{\mathrm{T}}$. The scalar product on the vector space Q^d is given by

$$[\vec{p},\vec{q}]_{Q^d,e} = p^0 q^0 V_e + \sum_{k=1}^{s} p^k q^k \ell^k \quad \forall \vec{p},\vec{q} \in Q^d, \tag{4}$$

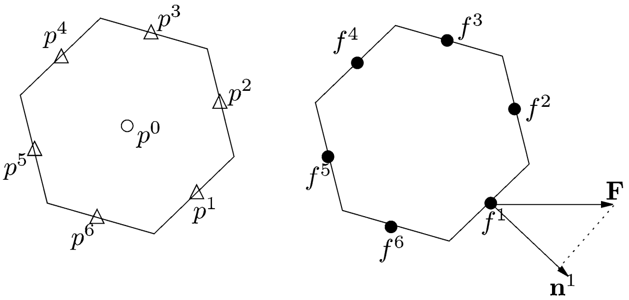

Figure 1. Location of pressure and velocity unknowns for a hexagon.

where V_e denotes the polygon area and ℓ^k denotes the length of its kth edge. Let $\langle\cdot,\cdot\rangle$ denote the conventional inner product in the Euclidean space. Then

$$[\vec{p},\vec{q}]_{Q^d,e} = \langle S\vec{p},\vec{q}\rangle, \qquad S = \mathrm{diag}\{V_e, \ell^1, \dots, \ell^s\}.$$

We denote the vector space of edge-based velocity unknowns by X^d. The dimension of X^d is equal to the number of polygon edges, s. Thus, any vector $\vec{f} \in X^d$ can be written as $\vec{f} = (f^1,\dots,f^s)^{\mathrm{T}}$. The scalar product on X^d is given by

$$[\vec{f},\vec{g}]_{X^d,e} = \langle M\vec{f},\vec{g}\rangle \quad \forall \vec{f},\vec{g} \in X^d, \tag{5}$$

where M is a symmetric positive definite $s \times s$ matrix. It is extremely difficult to find for an arbitrary polygon such a matrix M that the discrete scalar product approximates the continuous one with sufficient accuracy. It was shown in [10] that the accuracy of mimetic discretizations strongly depends on a choice of the matrix M. Note that a similar problem appears in some other mixed discretizations, e.g., in control volume mixed finite elements [17]. We postpone the derivation of matrix M until section 3.

The *third* step of the MFD method is to derive an approximation to the extended divergence operator. The discrete divergence operator, $\mathcal{DIV}^h$, naturally arises from the Gauss divergence theorem as

$$\mathcal{DIV}^h\vec{f} \overset{\mathrm{def}}{=} \frac{1}{V_e}\sum_{k=1}^{s} f^k\ell^k. \tag{6}$$

We emphasize one more time that s is the number of polygon edges which may be bigger than the number of straight line boundary pieces. This happens when some of the polygon angles equal to π.

The discrete extended divergence operator is defined by

$$\mathcal{D}^h\vec{f} = \left(\mathcal{DIV}^h\vec{f}, -f^1,\dots,-f^s\right)^{\mathrm{T}}.$$

The *fourth* step of the MFD method is to define the discrete flux operator $\mathcal{G}^h$, as the adjoint to the discrete extended divergence operator $\mathcal{D}^h$ with respect to scalar products (4) and (5), i.e.

$$[\vec{f},\mathcal{G}^h\vec{p}]_{X^d,e} = [\vec{p},\mathcal{D}^h\vec{f}]_{Q^d,e} \quad \forall\vec{p}\in Q^d,\ \vec{f}\in X^d. \tag{7}$$

Since velocity $\vec{f} \in X^d$ is an arbitrary vector, formula (7) implies that

$$\mathcal{G}^h \vec{p} = M^{-1}(\mathcal{D})^{\mathrm{T}} S \vec{p} = M^{-1} \widetilde{S} \begin{pmatrix} p^1 - p^0 \\ \vdots \\ p^s - p^0 \end{pmatrix}, \tag{8}$$

where $\widetilde{S}$ is the diagonal matrix, $\widetilde{S} = \mathrm{diag}\{\ell^1, \ldots, \ell^s\}$.

Using the discrete flux and divergence operators the continuous problem (1) is discretized as follows:

$$\begin{aligned} \mathcal{DIV}^h \vec{f} &= b_e, \\ \vec{f} &= \mathcal{G}^h \vec{p}, \end{aligned} \tag{9}$$

where b_e denotes the mean value of the source function over a polygon e. We refer to [7,20] for a detailed description of mimetic discretizations.

2.3. Interface conditions

The system of the element-based discretizations (9) is closed by imposing continuity conditions for pressure and velocity on the mesh edges. Hereafter, we shall use a subscript i for vectors, operators and scalar products involved in derivation of (9). For instance, $\vec{p}_i$ denotes a vector from the vector space Q_i^d equipped with the scalar product $[\cdot, \cdot]_{Q_i^d, e_i}$.

Let ℓ_{ij} be the length of a common boundary of polygons e_i and e_j. In general, this interface consists of s_{ij} edges L_{ij}^k, $k = 1, \ldots, s_{ij}$ ($s_{ij} = 3$ in figure 2), i.e.

$$\ell_{ij} \equiv |e_i \cap e_j| = \left| \bigcup_{k=1}^{s_{ij}} L_{ij}^k \right| = \sum_{k=1}^{s_{ij}} \ell_{ij}^k,$$

where $\ell_{ij}^k = |L_{ij}^k|$. It is convenient to use the double subscript for all edge-based unknowns. For instance, p_{ij}^k denotes the pressure unknown associated with the kth edge of interface $e_i \cap e_j$.

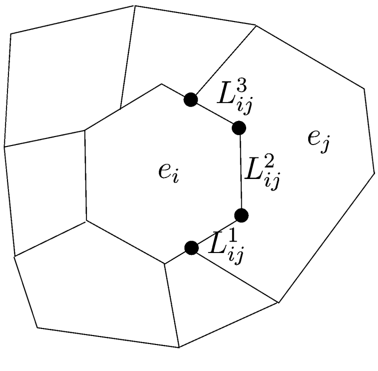

Figure 2. Interface between polygons e_i and e_j consisting of edges L_{ij}^1, L_{ij}^2 and L_{ij}^3.

For the continuous problem (1), we have continuity of the fluid pressure and normal velocity component across mesh edges. For the discrete problem it gives us the following continuity conditions:

$$f_{ij}^k = -f_{ji}^k \quad \text{and} \quad p_{ij}^k = p_{ji}^k, \quad k = 1, \dots, s_{ij}. \tag{10}$$

The system of equations (9), (10) is closed by imposing boundary conditions. For instance, the homogeneous Dirichlet boundary conditions imply that the pressure unknowns associated with boundary edges are equal to zero.

3. Scalar products in space of discrete velocities

In this section, we describe a method for deriving a family of scalar products in the space of discrete velocities (quadrature rules) which are exact for uniform flows. We begin by splitting a polygon into triangles and introducing auxiliary velocities on interior edges. Then, an accurate scalar product over a triangle (see, e.g., [14]) is used to build a scalar product over the polygon. In order to eliminate the auxiliary velocities, we express them via boundary velocities using the assumption that divergence over each triangle is equal to divergence over the polygon. Finally, we consider more general assumptions that may be used to eliminate the auxiliary velocities.

3.1. Derivation of scalar product

Let us first review briefly the scalar product proposed for strictly convex polygons [7,9,16,21] and then develop a new scalar product for general polygons.

Old scalar product. Let e be a strictly convex polygon. Then, the velocity vector can be recovered at each corner of this polygon from two orthogonal projections on edges which share that corner. We denote the recovered vectors by $\mathbf{F}^1, \mathbf{F}^2, \dots, \mathbf{F}^s$, where s is the number of polygon edges. We assume the that $\mathbf{F}^k$ is recovered from f^k and f^{k-1} (see figure 3).

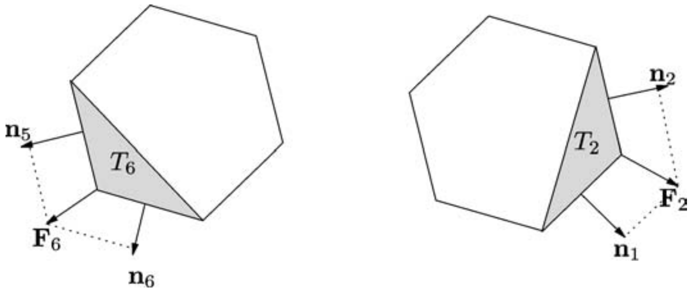

Figure 3. Recovered velocities $\mathbf{F}_2$ and $\mathbf{F}_6$.

Let $\mathbf{n}^k = (n_x^k, n_y^k)^{\mathrm{T}}$ be the unit vector normal to the kth edge of a polygon e. Then,

$$\mathbf{F}^k = \begin{bmatrix} n_x^k & n_y^k \\ n_x^{k-1} & n_y^{k-1} \end{bmatrix}^{-1} \begin{bmatrix} f^k \\ f^{k-1} \end{bmatrix}. \tag{11}$$

Using the recovered velocities, the old scalar product is given by

$$\langle M^{\mathrm{old}} \vec{f}, \vec{g} \rangle = \frac{V_e}{\widetilde{V}_e} \sum_{k=1}^{s} V_{T_k} \mathbf{F}^k \cdot (\mathbf{K}^{-1} \mathbf{G}^k), \quad \widetilde{V}_e = \sum_{k=1}^{s} V_{T_k}, \tag{12}$$

where T_k denotes a triangle uniquely defined by two edges ending at the kth vertex of e and V_{T_k} denotes the area of this triangle. The triangles T_2 and T_6 are shaded triangles in figure 3. It will be shown in section 5.5 that the scalar product (12) results in a numerical scheme with the optimal second-order convergence rate for the pressure (for all types of meshes with convex elements) but non-optimal convergence rate (less than 1) for the velocity (for polygonal meshes with $s \geqslant 5$). For triangular and quadrilateral meshes, scalar product (12) provides a good approximation of the continuous scalar product and the convergence rate for velocities is optimal. However, it cannot be applied for non-convex (see figure 2 and the left quadrilateral in figure 4) and degenerate elements (see the right quadrilateral in figure 4). Later in this section, we propose a new scalar product which recovers the optimal convergence rate for velocities for all types of polygonal meshes.

In the special case when $\mathbf{K} = \mathbf{I}$ and e is a triangle, formulas for entries of matrix M^{old} are quite simple. Let $\mathbf{r}_1, \mathbf{r}_2$ and $\mathbf{r}_3$ be the vertices of this triangle and θ_k be the triangle angle at vertex $\mathbf{r}_k$. Then,

$$M^{\mathrm{old}} = \frac{1}{6} \begin{bmatrix} \dfrac{\ell_1\ell_2}{\sin\theta_2} + \dfrac{\ell_1\ell_3}{\sin\theta_1} & \dfrac{\ell_1\ell_2\cos\theta_2}{\sin\theta_2} & \dfrac{\ell_1\ell_3\cos\theta_1}{\sin\theta_1} \\ \dfrac{\ell_2\ell_1\cos\theta_2}{\sin\theta_2} & \dfrac{\ell_2\ell_3}{\sin\theta_3} + \dfrac{\ell_2\ell_1}{\sin\theta_2} & \dfrac{\ell_2\ell_3\cos\theta_3}{\sin\theta_3} \\ \dfrac{\ell_3\ell_1\cos\theta_1}{\sin\theta_1} & \dfrac{\ell_3\ell_2\cos\theta_3}{\sin\theta_3} & \dfrac{\ell_3\ell_2}{\sin\theta_3} + \dfrac{\ell_3\ell_1}{\sin\theta_1} \end{bmatrix}. \tag{13}$$

New scalar product. Let e be a general polygon. We assume that e is split into t non-overlapping triangles Δ_l, $l = 1, \ldots, t$. As shown in figure 5 there are a few ways to split

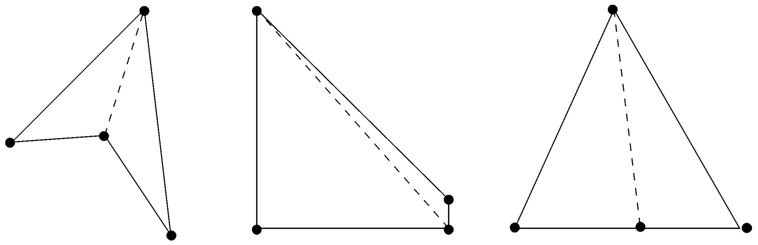

Figure 4. A few cases of irregularly shaped polygons.

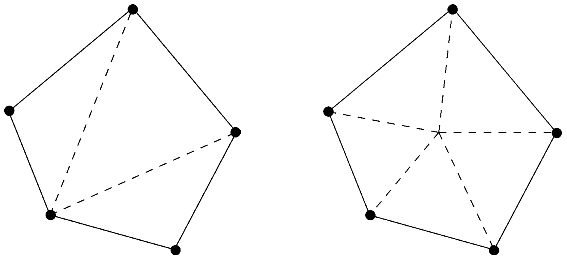

Figure 5. Two ways for splitting a pentagon.

a polygon into triangles. On each interior edge we temporarily introduce an additional unknown, the normal component of the fluid velocity. Let $\vec{f}^{\rm int}$ be the vector of these auxiliary unknowns and $\vec{\hat{f}}$ be the vector of all edge-based normal velocity components such that $\vec{\hat{f}} = (\vec{f}, \vec{f}^{\rm int})^{\rm T}$ where $\vec{f} \in X^d$. For the pentagon shown in figure 5, vector $\vec{f}$ has five components associated with boundary edges and vector $\vec{f}^{\rm int}$ has two and five components for the left and right splittings, respectively. Furthermore, let $\vec{f}_l \in \Re^3$ be a vector of normal velocity components associated with edges of a triangle Δ_l. In this section we shall use subscript l for the vectors associated with a triangle Δ_l and superscript k for their components.

Using the above notations, we write the scalar product over polygon e as the sum of scalar products over triangles:

$$\langle \widehat{M}\vec{\hat{f}}, \vec{\hat{g}} \rangle \equiv \sum_{l=1}^{t} \langle M_l^{\rm old}\vec{f}_l, \vec{g}_l \rangle, \tag{14}$$

where $M_l^{\rm old}$ is 3×3 matrix defined in (12) and $\widehat{M}$ is a symmetric positive definite matrix obtained by the standard assembling of the matrices $M_l^{\rm old}$, $l = 1, \ldots, t$. This statement follows from the fact that $M_l^{\rm old}$ are symmetric positive definite matrices and $\widehat{M}$ is an irreducible matrix.

Now, we eliminate degrees of freedom associated with the interior edges assuming that

$$\mathcal{DIV}_l^h \vec{f}_l = \mathcal{DIV}^h \vec{f}, \tag{15}$$

where the discrete divergence operators are defined in (6). Let a vector $\vec{f}$ be given. The resolvability of system (15) for $\vec{f}^{\rm int}$ is analyzed in the following lemma.

Lemma 3.1. System (15) has a solution $\vec{f}^{\rm int}$ for any vector $\vec{f}$.

Proof. Let r be the number of interior edges of a polygon e. System (15) may be rewritten as follows:

$$B_2\vec{f}^{\rm int} = B_1\vec{f} \tag{16}$$

where $B_2 \in \Re^{t\times r}$ and $B_1 \in \Re^{t\times s}$. System (16) is consistent if and only if the right-hand side, $B_1\vec{f}$, is orthogonal to $\ker B_2^{\mathrm{T}}$. In other words, the residual of system (15) should be orthogonal to $\ker B_2^{\mathrm{T}}$. Let us show that the dimension of $\ker B_2^{\mathrm{T}}$ is equal to one. If $\vec{\psi} \in \ker B_2^{\mathrm{T}} \subset \Re^t$ then

$$\left(B_2^{\mathrm{T}}\vec{\psi}, \vec{\eta}\right) = 0 \quad \forall \vec{\eta} \in \Re^r. \tag{17}$$

Since each interior edge is shared by two triangles, each row of B_2^{T} has exactly two nonzero entries. Let Δ_k and Δ_l be two triangles having a common edge and let $\vec{\eta}$ be a vector in $\Re^r$ whose only nonzero component corresponds to this edge. Then equation (17) results in

$$\frac{1}{V_{\Delta_k}}\psi^k = \frac{1}{V_{\Delta_l}}\psi^l. \tag{18}$$

Since polygon e is simply connected, one component of vector $\vec{\psi}$ defines uniquely the other components. Therefore, the dimension of $\ker B_2^{\mathrm{T}}$ is at most 1. It is obvious that vector $\vec{\psi} = \{V_{\Delta_1}, \ldots, V_{\Delta_t}\}$ is a nontrivial solution of (18). Thus, $\dim(\ker B_2^{\mathrm{T}}) = 1$.

Now, we prove that $\vec{\psi} \perp B_1\vec{f}$. Since vector $\vec{f}$ is arbitrary, the orthogonality condition holds if and only if $B_1^{\mathrm{T}}\vec{\psi} = 0$. The definition of the discrete divergence and system (15) imply that

$$\left(B_1^{\mathrm{T}}\vec{\psi}\right)^k = \sum_{l=1}^{t}\frac{\ell^k}{V_e}\psi^l - \frac{\ell^k}{V_{\Delta_k}}\psi^k = \ell^k\left(\sum_{l=1}^{t}\frac{V_{\Delta_l}}{V_e} - 1\right) = 0.$$

Since k is arbitrary, we get $B_1^{\mathrm{T}}\vec{\psi} = 0$ which proves the assertion of the lemma. □

Corollary 3.1. $\dim(\ker B_2) = r - t + 1$.

Let us consider a couple of typical examples illustrating lemma 3.1 and corollary 3.1. As shown in figure 5, there exist a few ways to split polygon e into triangles either by introducing additional interior points (one or more) or without such points.

In the first case (figure 5(a)) we have $t - 1$ auxiliary interior edges ($t = 3$) and t equations for the corresponding velocities. Thus, $\dim(\ker B_2) = (t-1) - t + 1 = 0$ and system (15) has a unique solution. In the second case (figure (5(b)) we have t auxiliary interior edges and t equations for the corresponding velocities. Thus, $\dim(\ker B_2) = t - t + 1 = 1$ and we get one-parametric family of solutions for (15). In both cases it is possible to construct a matrix $B \in \Re^{r\times s}$ such that

$$\vec{f}^{\,\mathrm{int}} = B\vec{f} \tag{19}$$

is a solution to equations (15). Using these relations, we may rewrite the scalar product (14) as follows:

$$\begin{aligned}\langle M^{\text{new}}\vec{f}, \vec{g}\rangle &\equiv \langle \widehat{M}\vec{\hat{f}}, \vec{\hat{g}}\rangle \\ &= \left\langle \begin{bmatrix} \widehat{M}_{11} & \widehat{M}_{12} \\ \widehat{M}_{21} & \widehat{M}_{22} \end{bmatrix} \begin{bmatrix} \vec{f} \\ B\vec{f} \end{bmatrix}, \begin{bmatrix} \vec{g} \\ B\vec{g} \end{bmatrix} \right\rangle. \end{aligned} \tag{20}$$

It is obvious that M^{new} is a symmetric positive definite matrix. Thus, it may be used to generate a scalar product on X^d. Using (20), we may write down the explicit formula for matrix M^{new}:

$$M^{\text{new}} = \widehat{M}_{11} + B^{\mathrm{T}}\widehat{M}_{22}B + \widehat{M}_{12}B + B^{\mathrm{T}}\widehat{M}_{21}. \tag{21}$$

3.2. *Examples of irregularly shaped polygons*

In this subsection, we consider a few special cases of irregularly shaped polygons. It is sufficient to consider only irregularly shaped quadrilaterals shown in figure 4.

Polygons with relatively small edges. The middle quadrilateral in figure 4 has an edge whose length is relatively much smaller than the length of other edges. The splitting shown in the figure results in a thin triangle. Let us enumerate the edges of this quadrilateral counter-clockwise starting with the smallest one. Thus, ℓ^1 be the length of the smallest edge and ℓ^5 be the length of the interior edge. Using the above notations, the vector of normal velocity components associated with the thin triangle is $\vec{f}_1 = (f^1, f^2, f^5)^{\mathrm{T}}$. Furthermore, let Δ_1 be the thin triangle and Δ_2 be the other triangle. It follows from (15) that

$$f^5 = \frac{V_{\Delta_2}(f^1\ell^1 + f^2\ell^2) - V_{\Delta_1}(f^3\ell^3 + f^4\ell^4)}{\ell^5(V_{\Delta_1} + V_{\Delta_2})}.$$

In the asymptotic case, $\ell^1 \to 0$, (i.e. $V_{\Delta_1} \to 0$ and $\ell^2 \to \ell^5$) the above formula gives the expected result:

$$f^5 = f^2 + \mathcal{O}(\ell^1).$$

Substituting this estimate into the scalar product generated by the matrix M_1^{old}, we get that

$$\langle M_1^{\text{old}}\vec{f}_1, \vec{g}_1\rangle \sim \ell^1\langle \vec{f}_1, \vec{g}_1\rangle.$$

Thus, the contribution of the thin triangle into the scalar product over the quadrilateral is relatively negligible.

Nonconvex polygons. The left polygon e in figure 4 is a nonconvex quadrilateral. The splitting shown in the figure converts e into two triangles. It is possible that the mass center of a nonconvex element lies outside this element. In this case it could be useful to generalize the assumption of constant divergence as follows:

$$\mathcal{DIV}_l^h\vec{f}_l = \alpha_l\mathcal{DIV}^h\vec{f}. \tag{22}$$

The numbers α_l, $l = 1, \ldots, t$, represent relations between values of the discrete divergence over the triangles and the polygon. The analysis presented in the previous subsections can be easily extended to (22). In particular, the following corollary of lemma 3.1 holds.

Corollary 3.2. Let the numbers α_l, $l = 1, \ldots, t$, be such that

$$\sum_{l=1}^{t} \alpha_l V_{\Delta_l} = V_e. \tag{23}$$

Then, system (22) has a solution $\vec{f}^{\text{int}}$ for any vector $\vec{f}$.

The numbers α_l, $l = 1, \ldots, t$, may be used (if necessary) to improve the discretization for nonconvex mesh elements.

The right quadrilateral in figure 4 has two edges which form the angle of 180 degrees. Like in the case of the nonconvex quadrilateral, a proper splitting shown in the figure resolves the problem.

The meshes with irregularly shaped polygons will be further analyzed in sections 5.3 and 5.6.

4. Accuracy considerations

Let us prove that discrete system (9) with the scalar product on X^d generated by the matrix M^{new} is exact for linear solutions.

Let $\vec{f}^*$ and $\vec{f}_l^*$, $l = 1, \ldots, t$, be vectors of normal velocity components for a uniform flow (linear pressure and constant velocity) associated with edges of polygon e and triangles Δ_l, $l = 1, \ldots, t$, respectively. Furthermore, let $\vec{p}^*$ and $\vec{p}_l^*$, $l = 1, \ldots, t$, be vectors of pressure unknowns for the linear solution associated with the polygon and the triangles, respectively. Note, that $p^{*,0}$ and $p_l^{*,0}$, $l = 0, \ldots, t$, are evaluated at mass centers of the polygon and triangles, respectively. Our goal is to show that

$$\vec{f}^* = \mathcal{G}^h \vec{p}^*. \tag{24}$$

It has been shown in [14] that for a triangular element Δ_l, formula (24) is exact, i.e.

$$\vec{f}_l^* = \mathcal{G}_l^h \vec{p}_l^* \quad \text{for } l = 1, \ldots, t.$$

Using the symmetry relation between the discrete flux $\mathcal{G}_l^h$ and extended divergence $\mathcal{D}_l^h$ operators, we rewrite the above identities in an equivalent form:

$$\left[\vec{f}_l^*, \vec{g}_l\right]_{X_l^d,\Delta_l} = \left[\mathcal{G}_l^h \vec{p}_l^*, \vec{g}_l\right]_{X_l^d,\Delta_l} = \left[\vec{p}_l^*, \mathcal{D}_l^h \vec{g}_l\right]_{Q_l^d,\Delta_l} \quad \forall \vec{g}_l \in X_l^d. \tag{25}$$

Since the last formulas hold for arbitrary vectors $\vec{g}_l$, $l = 1, \ldots, t$, they are certainly true for vectors where only components associated with boundary edges of polygon e

are arbitrary and the other components are given by formula (19), i.e. $\vec{g}^{\mathrm{int}} = B\vec{g}$. It is obvious that both sides of (15) equal to zero for a uniform flow. Therefore, formula (19) holds for the uniform flow, i.e. $\vec{f}^{*,\mathrm{int}} = B\vec{f}^*$. Summing up the identities (25), we get

$$\sum_{l=1}^{t} [\vec{f}_l^*, \vec{g}_l]_{X_l^d,\triangle_l} = [\vec{f}^*, \vec{g}]_{X^d,e} = \sum_{l=1}^{t} [\vec{p}_l^*, \mathcal{D}_l^h \vec{g}_l]_{Q_l^d,\triangle_l} \quad \forall \vec{g} \in X^d.$$

The definition of the extended divergence operator, assumption (15), linearity of $p(\mathbf{x})$, and the definition of the scalar product in the space of discrete pressures imply that

$$\begin{aligned}\sum_{l=1}^{t} [\vec{p}_l^*, \mathcal{D}_l^h \vec{g}_l]_{Q_l^d,\triangle_l} &= \sum_{l=1}^{t} p_l^{*,0} \big(\mathcal{DIV}_l^h \vec{g}_l\big) V_{\triangle_l} + \sum_{k=1}^{t} p^{*,k} g^k \ell^k \\ &= \mathcal{DIV}^h \vec{g} \sum_{l=1}^{t} p_l^{*,0} V_{\triangle_l} + \sum_{k=1}^{t} p^{*,k} g^k \ell^k \\ &\equiv [\vec{p}^*, \mathcal{D}^h \vec{g}]_{Q^d,e}.\end{aligned}$$

The symmetry relation between the discrete flux, $\mathcal{G}^h$, and the extended divergence, $\mathcal{D}^h$, operators imply that

$$[\vec{f}^*, \vec{g}]_{X^d,e} = [\vec{p}^*, \mathcal{D}^h \vec{g}]_{Q^d,e} = [\mathcal{G}^h \vec{p}^*, \vec{g}]_{X^d,e} \quad \forall \vec{g} \in X^d.$$

Since $\vec{g}$ is arbitrary, we get (24).

In the case when $\alpha_k \neq 1$, the mimetic discretization will remain exact for linear solutions, i.e. formula (24) holds, if we evaluate the element-based pressure unknown, $p^{*,0}$, at the point

$$\mathbf{c}^0 = \sum_{l=1}^{t} \alpha_l \frac{V_{\triangle_l}}{V_e} \mathbf{c}_l$$

where $\mathbf{c}_l$ is the centroid of triangle $\triangle_l$. Note that for $\alpha_l = 1$, $l = 1, \ldots, t$, $\mathbf{c}^0$ is the mass center of the polygon. A special choice of numbers α_l may move (if necessary) this point inside a nonconvex mesh element.

The convergence of mimetic finite difference, mixed finite element and finite volume discretizations has been studied by many authors. We refer readers to articles [1–3,5,6] and references therein. Note that the majority of publications consider triangular or quadrilateral (usually orthogonal) meshes. The connection between MFD and MFE methods established in [2] for quadrilateral meshes can be revised for polygonal meshes using recently proposed new finite elements [12]. It opens a door for deriving convergence estimates for the MFD method on polygonal meshes.

5. Numerical experiments

5.1. Implementation issues

In all numerical experiments except section 5.3, we split a polygon into triangles by inserting one interior point and connecting it with the polygon vertices. The interior point is placed into the geometric center of this polygon. This splitting means that the matrix B in (19) is not unique. However, if we specify a particular formula for computing one of the interior edge-based velocities, the other velocities will be calculated recursively using equations (15). In order to do that, we find the minimal angle in a polygon e and recover the velocity vector at the corresponding vertex using formula (11). Let $\mathbf{F}^k$ be the recovered vector. Then, we project $\mathbf{F}^k$ onto the unit vector, $\mathbf{n}^{\mathrm{int},k} = (n_x^{\mathrm{int},k}, n_y^{\mathrm{int},k})^{\mathrm{T}}$, normal to the interior edge corresponding to the vertex. It gives us the required expression for the interior velocity, $f^{\mathrm{int},k}$, in terms of boundary velocities:

$$f^{\mathrm{int},k} = \left(n_y^{k-1} n_x^{\mathrm{int},k} - n_x^{k-1}\, n_y^{\mathrm{int},k}\right) f^k - \left(n_y^k n_x^{\mathrm{int},k} - n_x^k n_y^{\mathrm{int},k}\right) f^{k-1}.$$

In section 5.3 we consider meshes consisting of both convex and nonconvex quadrilaterals. We choose $\alpha_l = 1$ and split a quadrilateral into two triangles by one of the two diagonals. We use the shortest diagonal lying inside the quadrilateral.

System (9), (10) is a typical example of a saddle point problem with a symmetric coefficient matrix. Therefore, it can be solved with the preconditioned Lanzcos iterative method. However, a few simple algebraic transformations will reduce the system to a problem with a symmetric positive definite matrix. Note, that some of the discrete unknowns, normal velocity components and element-based pressures are only connected within a single element. Therefore, they may be easily excluded from the system resulting in the new system

$$A\vec{p} = \vec{d}, \tag{26}$$

where $\vec{p}$ denotes the global vector of edge-based pressure unknowns (one unknown for each mesh edge) and $\vec{d}$ is a right-hand side vector.

Lemma 5.1. The matrix A is symmetric and positive definite except the case of the Neumann boundary value problem when A is semi-positive definite.

The proof of this lemma follows the proofs of similar results in the theory of hybrid-mixed finite elements and is omitted here. In addition to the properties mentioned above, the matrix A is sparse. Two examples of the stencil of the matrix A are shown in figure 6. The pressure unknown marked by a black bullet is connected only with pressure unknowns located on edges of two polygons sharing the edge (marked by blank bullets).

Problem (26) can be solved by the preconditioned conjugate gradient method. In the numerical experiments we use the algebraic multigrid preconditioner from [22]. After solving problem (26), we may recover the primary variables element-by-element using the local systems (9).

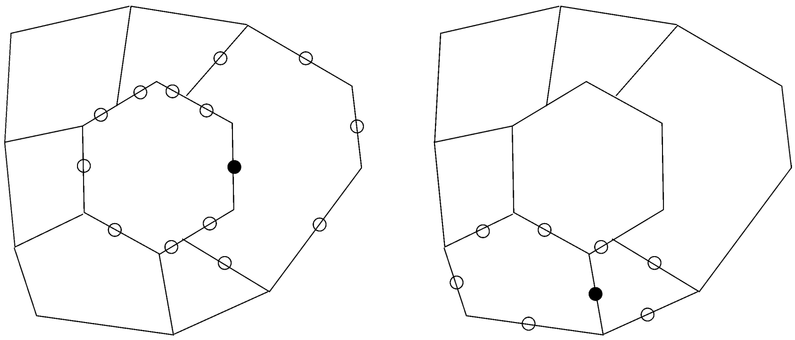

Figure 6. Two examples of the stencil of the matrix A.

In the rest of this section, we present computational results which demonstrate the accuracy of our method, its flexibility and the efficiency of the solution method. The algebraic multigrid method was chosen as an example of a method applicable to arbitrary matrix stencils. However, its theoretical justification is limited to M-matrices. In the case of highly distorted meshes and random isotropic tensors, its performance may degrade. In order to ensure the robustness of the solver, the algebraic multigrid method may be replaced by (or combined with) more robust multigrid methods such as aggregation multigrid methods [23] and algebraic multigrid methods with projectors [11].

5.2. *Uniform flows*

The first set of experiments verifies that our method is exact for linear functions. We consider problem (1) in the unit square $(0, 1)^2$ subject to nonhomogeneous boundary conditions. Let $\mathbf{K} = \mathbf{I}$ and $p(x, y) = x + y$ be the exact solution.

The results of the numerical experiments are shown in the left picture in figure 7. As it was proved in section 4, the discrete solution is exact at both edge mid-points and polygon mass centers.

5.3. *Nonconvex quadrilateral meshes*

The second set of calculations addresses convergence of mimetic discretizations on random quadrilateral meshes with nonconvex elements. We consider problem (1) in the unit square subject to the Dirichlet boundary conditions. Let $\mathbf{K} = \mathbf{I}$ and

$$p(x, y) = \sin(2\pi x)\sin(2\pi y)$$

be the exact solution.

The random mesh is generated by moving each mesh point of a uniform mesh (with mesh step size h) to a random position inside a square centered at the point. The sides of this square are aligned with the coordinate axes and equal to $1.6h$. We explicitly check that the resulting mesh is valid. The computational mesh and isolines of the discrete solution are shown in figure 8. The mesh contains 1.17% of nonconvex quadrilaterals (one of them is marked by the circle). As it was shown in previous sections, a

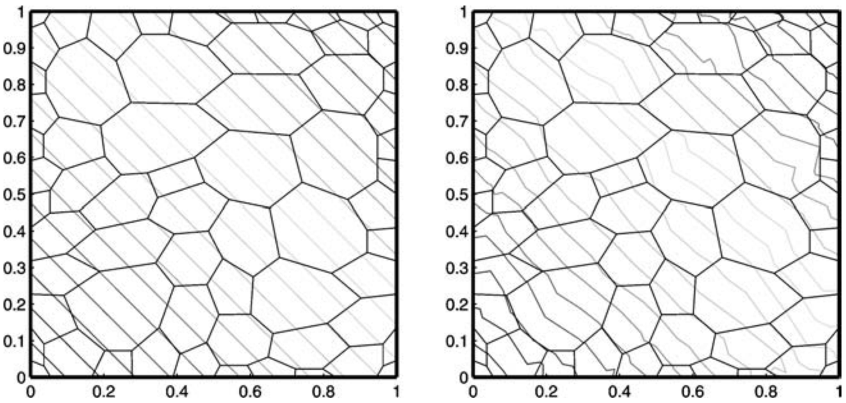

Figure 7. The isolines of discrete solutions for two scalar products generated by matrices M^{new} (left) and M^{old} (right).

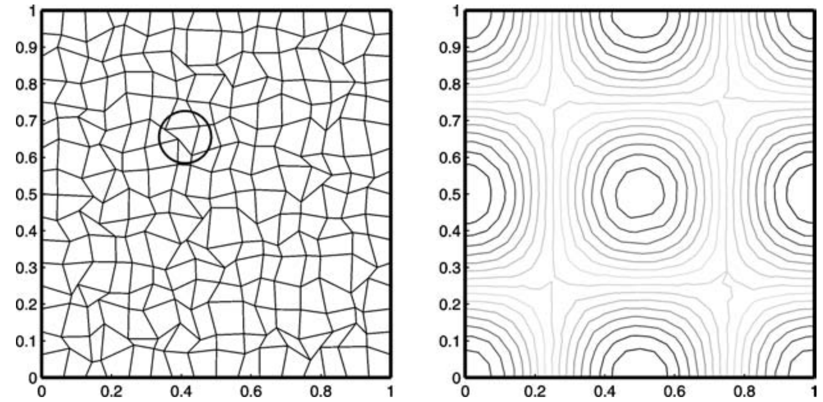

Figure 8. The computational mesh and isolines of the discrete solution.

proper splitting of these elements results in the discrete scheme exact for linear solutions. It is pertinent to note that the previously elaborated mimetic discretizations (see, e.g., [7,13]) either cannot be used on such a mesh or do not preserve linear solutions.

The convergence of new mimetic discretization is demonstrated in table 1. Convergence rates of the relative L_2 errors ε_p and ε_f for the element-centered pressure and edge-based velocity unknowns, respectively, are close to the optimal rates for random meshes. Percentage of irregular elements $\sigma(\Omega_h)$ is shown in the last column of table 1.

5.4. *Locally refined meshes*

The third set of calculations addresses the accuracy of the method on distorted locally refined (LR) meshes (see also [5,13]). We again consider problem (1) in the unit

Table 1
Convergence analysis on meshes with non-convex elements.

$1/h$	ε_p	ε_f	$\sigma(\Omega_h)$, %
16	3.39e−2	3.28e−2	1.17
32	9.00e−3	1.58e−2	2.44
64	2.31e−3	9.78e−3	2.88
128	5.72e−4	4.53e−3	2.87
256	1.44e−4	2.67e−3	2.98
rate	1.97	0.91	

Table 2
Convergence on a sequence of LR meshes.

l	m	LR/polygonal meshes		LR/hanging nodes	
		ε_p	ε_f	ε_p	ε_f
0	256	5.34e−2	8.26e−2	7.11e−2	9.25e−2
1	556	1.01e−2	3.28e−2	1.47e−2	3.25e−2
2	988	2.71e−3	1.16e−2	3.96e−3	1.55e−2
3	3952	6.92e−4	5.56e−3	9.60e−4	7.89e−3
4	15808	1.73e−4	2.84e−3	2.45e−4	3.69e−3
	rate	2.67	1.58	2.66	1.47

square subject to nonhomogeneous Dirichlet boundary conditions. Let $\mathbf{K} = \mathbf{I}$ and

$$p(x, y) = 1 - \tanh\left(\frac{(x - 0.5)^2 + (y - 0.5)^2}{0.01}\right)$$

be the exact solution. This function has a sharp peak in the middle of the domain and close to zero near the domain boundary. Calculations were performed on a sequence of randomly distorted locally refined meshes build in two steps.

The sequence of meshes begin with a 16×16 mesh. We use a simple geometric approach to create a few first locally refined meshes. Each mesh element of the initial mesh is uniquely identified by two indices i_0 and j_0, $1 \leqslant i_0, j_0 \leqslant 16$. On the first refinement level ($l = 1$), we split the mesh elements with indices $4 \leqslant i_0, j_0 \leqslant 13$ into four elements. The new mesh elements are uniquely identified by two indices i_1 and j_1, $1 \leqslant i_1, j_1 \leqslant 20$. On the second refinement level, we split mesh elements with indices $5 \leqslant i_1, j_1 \leqslant 16$ into four elements. On the subsequent levels ($l > 2$) we uniformly refine all mesh elements.

The random mesh is generated by moving each mesh point to a random position inside a square centered at the point. The sides of this square are aligned with the coordinate axes and equal 80% of the size of the smallest element sharing the point. Note that the hanging mesh points are always located at the edge mid-points. The relative L_2 errors ε_p and ε_f are given in table 2. Note that the new method developed in this paper results in more accurate solution than the method proposed in [13]. This is probably

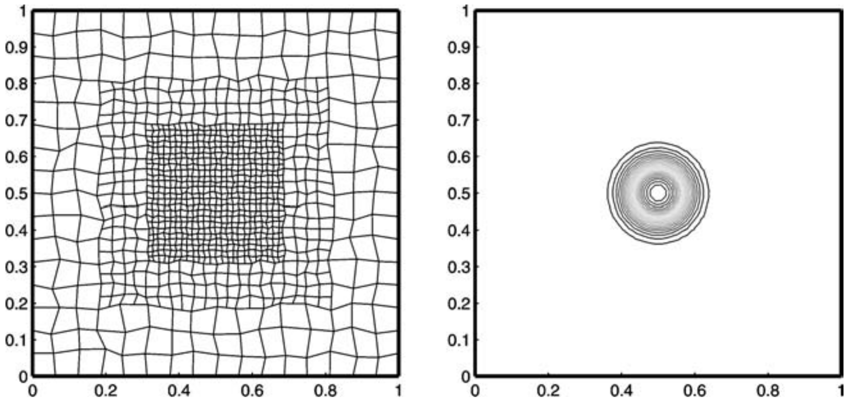

Figure 9. The computational mesh on level $l = 2$ and isolines of the discrete solution.

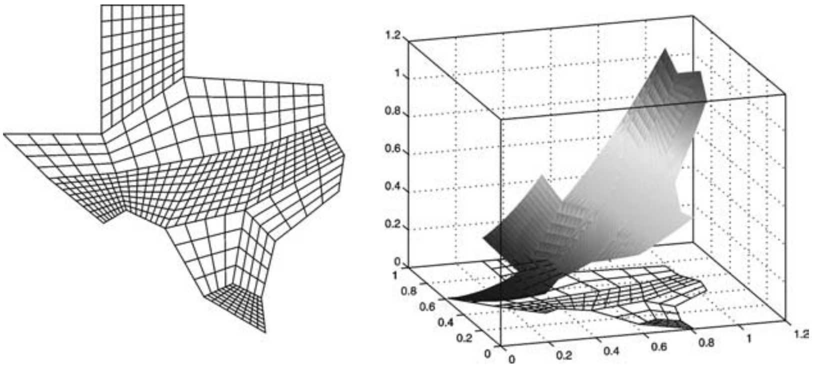

Figure 10. The computational multi-block mesh ($l = 2$) and the exact solution.

due to the fact that the new discretization scheme uses two velocities on the edge with hanging node while the scheme described in [13] uses only one velocity unknown. The computational mesh and isolines of the discrete solution (for the new method) are shown in figure 9.

The fourth set of experiments addresses the convergence rate for the discrete velocities on smooth LR meshes. We consider the Neumann boundary value problem in the domain resembling shape of state Texas (see figure 10). The full tensor **K** and exact solution are given by

$$\mathbf{K}(x, y) = \begin{bmatrix} (x+1)^2 + y^2 & -xy \\ -xy & (x+1)^2 \end{bmatrix}, \qquad p(x, y) = x^3y^2 + x\cos(xy)\sin(x).$$

An example of a computational mesh is shown in figure 10. The mesh consists of 13 quadrilateral blocks with different levels of uniform refinement. The mesh shown in figure 10 corresponds to $l = 2$ where l denotes the refinement level.

Multi-block meshes provide a flexible tool for simulating layered structures occurring in many porous medium applications. The L_2-norms of errors are given in table 3.

Table 3
Convergence analysis for the full tensor.

l	ε_p	ε_f
2	2.97e−4	1.37e−2
3	7.37e−5	4.64e−3
4	1.87e−5	1.68e−3
5	4.72e−6	7.23e−4
6	1.19e−6	2.42e−4
rate	1.99	1.43

Note that the convergence rate for the fluid velocity is close to 1.5 which is considered to be optimal for such types of meshes and low-order discretizations. Note that we have not discussed the optimal splitting of a polygon as well as the optimal choice of the matrix B in (19). Answers to these questions may allow us to improve the accuracy of the discretization.

5.5. Median meshes

In the fifth set of experiments, we compare the scalar products in the space of velocities generated by matrices M^{new} and M^{old} (see formulas (12) and (21), respectively) for polygonal meshes with strictly convex elements. As we mentioned before, the method based on the scalar product generated by matrix M^{old} is not exact for linear solutions. We illustrate that in the right picture in figure 7.

In order to compare two scalar products we consider problem (1) in the unit square subject to nonhomogeneous Dirichlet boundary conditions. Let $\mathbf{K} = \mathbf{I}$ and

$$p(x, y) = \sin(2\pi x)\sin(2\pi y)$$

be the exact solution.

The results of the numerical experiments are shown in table 4. The errors are computed on a sequence of polygonal median meshes. The set of points $\mathbf{x}_{ij} = (x_{ij}, y_{ij})$ for generating the Voronoi tessellation [18] is given by

$$\begin{aligned} x_{ij} &= \xi_i + 0.1\sin(2\pi\xi_i)\sin(2\pi\eta_j), \qquad i = 0, \dots, n_x, \\ y_{ij} &= \eta_j + 0.1\sin(2\pi\xi_i)\sin(2\pi\eta_j), \qquad j = 0, \dots, n_y, \end{aligned}$$

where $\xi_i = i/n_x$ and $\eta_j = j/n_y$. The median mesh is constructed from the Voronoi mesh by moving a mesh vertex to the mass center of a triangle formed by the centers of three Voronoi cells sharing the vertex.

Both scalar products (generated by matrices M_{old} and M_{new}) result in the second order convergence rate for the pressure unknown. However, the new scalar product gives better resolution of both the fluid pressure and the velocity. The computational mesh and the isolines of the discrete solution (for the new method) are shown in figure 11.

Table 4
Convergence analysis for two scalar products.

m	New scalar product		Old scalar product	
	ε_p	ε_f	ε_p	ε_f
166	1.07e−1	3.68e−1	1.81e−1	4.57e−1
598	2.60e−2	1.64e−1	3.39e−2	2.52e−1
2230	5.11e−3	8.28e−2	6.64e−3	1.72e−1
8566	1.05e−3	4.29e−2	1.51e−3	1.20e−1
rate	2.23	1.03	2.31	0.63

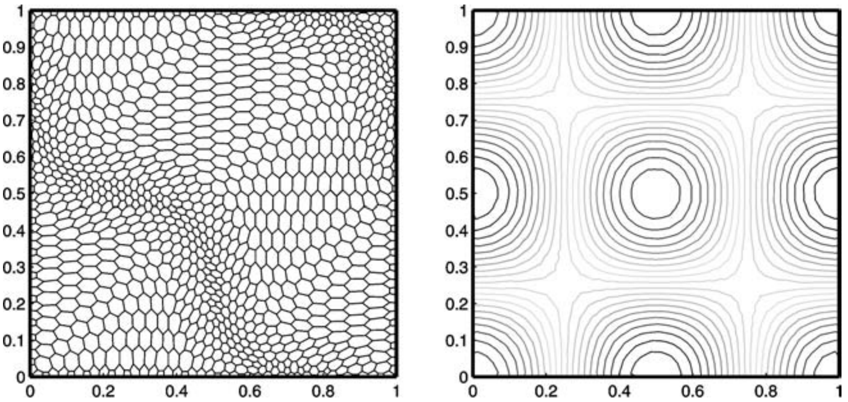

Figure 11. The computational mesh and isolines of the discrete solution.

5.6. Non-matching meshes

In the sixth set of experiments, we consider non-matching quadrilateral meshes and perform numerical analysis of the convergence rate (see also [1]). We consider problem (1) in the unit square with mixed boundary conditions. On the bottom and top boundary parts, we impose the Dirichlet boundary condition. The homogeneous Neumann boundary condition is set on the rest of the boundary. Let tensor $\mathbf{K}$ be the scalar matrix equal to $K_1\mathbf{I}$ in the region defined by $y < 0.5$ and equal to $K_2\mathbf{I}$ in the rest of the domain. The source term is chosen in such a way that the exact solution is given by

$$p(x,y) = \begin{cases} \dfrac{7}{16} - \dfrac{K_2}{12K_1} + \dfrac{2K_2}{3K_1}y^3, & y < 0.5, \\ y - y^4, & y \geqslant 0.5. \end{cases}$$

The computational mesh and the discrete solution for $K_2 = 4$ and $K_1 = 1$ are shown in figure 12. The random meshes below and above interface line $y = 0.5$ were generated using the rules described in section 5.4. The isolines of the discrete solution are straight lines except the isoline located below the mesh interface where the computational mesh is rather coarse. The relative L_2 errors ε_p and ε_f presented in table 5 show the second or-

Table 5
Convergence analysis on non-matching meshes.

m	ε_p	ε_f	$\max_i \rho_i$	$\#itr$	CPU, s
175	4.91e−3	1.39e−2	167.3	11	0.02
780	1.12e−4	6.35e−3	267.1	13	0.11
3286	2.70e−4	2.89e−3	159.5	12	0.61
13482	6.63e−5	1.45e−3	612.1	14	3.14
54610	1.64e−5	7.22e−4	2024	14	13.2

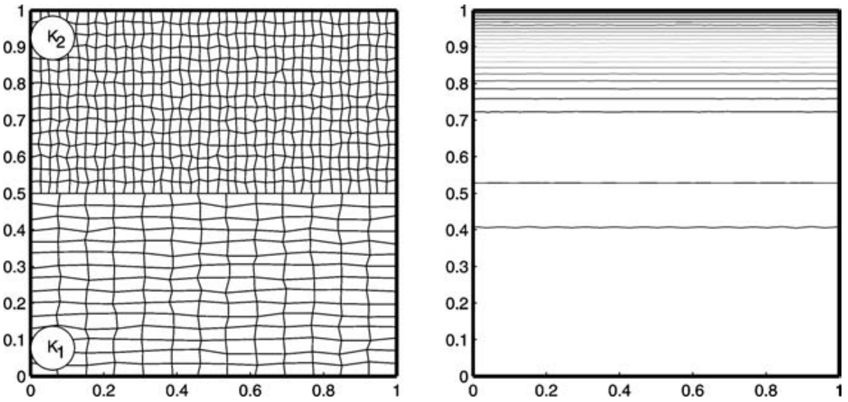

Figure 12. The computational mesh and isolines of the discrete solution.

der convergence rate for the pressure and only the first order for the velocity. It confirms the convergence rates observed in the previous experiments on nonsmooth meshes.

Non-matching quadrilateral meshes may result in very bad interface polygons (see, e.g., the middle polygon in figure 12). We measure the geometric quality of a mesh polygon e_i as ratio ρ_i of the biggest edge length to the smallest one. The middle column in table 5 shows that ρ_i may reach a few thousands. However, it seems that it does not have any noticeable impact on the convergence rate.

A non-matching mesh is a challenging example for algebraic solvers. Therefore, we present here more details about the performance of the solver. The last two columns in table 5 demonstrate excellent performance of the AMG preconditioner. The stopping criterion for the preconditioned conjugate gradient method is the relative decrease in the norm of the residual by factor 10^{-12}. The computational time, denoted by CPU, includes time for initializing the AMG solver and time for solving problem (26). The arithmetical cost per iteration grows almost linearly in the number of elements, m, except for very coarse meshes where cache memory effects play an important role.

Let us keep the setting of the last experiment but replace the source term and boundary conditions such that the exact solution is now $p(x, y) = x - x^4$. The left picture in figure 13 represents the discrete solution corresponding to the mesh shown in figure 12. The breaking of isolines which are supposed to be the straight lines is due to a coarse

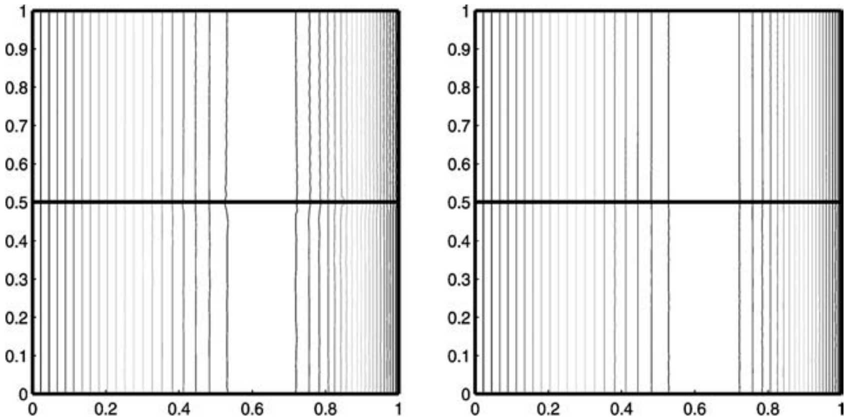

Figure 13. Isolines of the discrete solutions on two consecutive meshes.

mesh in the bottom half of the computational domain. Indeed, the solution isolines on the refined mesh (right picture) are very close to straight lines.

We stress one more time that the previously elaborated mimetic discretizations cannot be used on non-matching meshes. The method described in the review paper [7] uses the scalar product in the velocity space generated by the matrix M^{old}. Therefore, the velocity vector cannot be recovered at polygon vertices corresponding to adjacent edges with angle 180° between them. The method developed in [13] can be applied on meshes with only one hanging node per mesh edge. The method proposed in this paper overcomes all problems of the above methods.

6. Conclusion and future work

We have constructed new discretizations for the diffusion equation on unstructured polygonal meshes. The class of polygonal meshes includes meshes with nonconvex polygons (in particular, nonconvex quadrilaterals are very important for existing codes), locally refined and nonmatching meshes.

The new mimetic discretization has been proved to be exact for uniform flows. We have shown by numerical experiments the second-order convergence rate for the pressure variable and the first order convergence rate for the velocity variable. These convergence rates are considered optimal for sufficiently smooth solutions and lower order discretization methods on unstructured meshes.

In future work we plan a more rigorous analysis of the new discretization. In particular, we shall exploit effects of polygon splittings, other scalar products for triangles, and different expressions for auxiliary velocities on accuracy of mimetic discretizations. In addition, we shall analyze extension of the method to other PDEs, including Maxwell's equations, Navier–Stokes equations, equations of linear elasticity, and diffusion-type problems with a lack of elliptic regularity.

Existence of accurate mimetic discretizations for tetrahedral meshes, enables us to extend the method to unstructured polyhedral meshes.

Acknowledgements

The authors thank Dr. Raphael Loubere (LANL) for his assistance in generating polygonal meshes. We also thank Jim Morel and David Moulton (LANL) for fruitful discussions.

References

[1] T. Arbogast, L. Cowsar, M. Wheeler and I. Yotov, Mixed finite element methods on non-matching multiblock grids, SIAM J. Numer. Anal. 37 (2000) 1295–1315.

[2] M. Berndt, K. Lipnikov, D. Moulton and M. Shashkov, Convergence of mimetic finite difference discretizations of the diffusion equation, J. Numer. Math. 9 (2001) 265–284.

[3] M. Berndt, K. Lipnikov, M. Shashkov, M. Wheeler and I. Yotov, Superconvergence of the velocity in mimetic finite difference methods on quadrilaterals, Los Alamos Report LA-UR-03-7904 (submitted to SIAM J. Numer. Anal.).

[4] J. Campbell and M. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (2001) 739–765.

[5] R.E. Ewing, R.D. Lazarov and P.S. Vassilevski, Local refinement techniques for elliptic problems on cell-centered grids I, error analysis, Math. Comp. 56 (1991) 437–461.

[6] R.E. Ewing, M. Liu and J. Wang, Superconvergence of mixed finite element approximations over quadrilaterals, SIAM J. Numer. Anal. 36(3) (1999) 772–787.

[7] J. Hyman, J. Morel, M. Shashkov and S. Steinberg, Mimetic finite difference methods for diffusion equations, Comput. Geosci. 6(3/4) (2002) 333–352.

[8] J. Hyman and M. Shashkov, Mimetic discretizations for Maxwell's equations and the equations of magnetic diffusion, Progress Electromagnetic Res. 32 (2001) 89–121.

[9] J. Hyman, M. Shashkov and S. Steinberg, The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials, J. Comput. Phys. 132 (1997) 130–148.

[10] J. Hyman, M. Shashkov and S. Steinberg, The effect of inner products for discrete vector fields on the accuracy of mimetic finite difference methods, Internat. J. Comput. Math. Appl. 42 (2001) 1527–1547.

[11] Y. Kuznetsov, Two-level preconditioners with projectors for unstructured grids, Russian J. Numer. Anal. Math. Modelling 15(3/4) (2000) 247–256.

[12] Y. Kuznetsov and S. Repin, New mixed finite element method on polygonal and polyhedral meshes, Russian J. Numer. Anal. Math. Modelling 18(3) (2003) 261–278.

[13] K. Lipnikov, J. Morel and M. Shashkov, Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes, J. Comput. Phys. 199 (2004) 589–597.

[14] R. Liska, M. Shashkov and V. Ganzha, Analysis and optimization of inner products for mimetic finite difference method on triangular grid, Math. Comput. Simulation 67 (2004) 55–66.

[15] L. Margolin, M. Shashkov and P. Smolarkiewicz, A discrete operator calculus for finite difference approximations, Comput. Methods Appl. Mech. Engrg. 187 (2000) 365–383.

[16] J. Morel, R. Roberts and M. Shashkov, A local support-operators diffusion discretization scheme for quadrilateral $r - z$ meshes, J. Comput. Phys. 144 (1998) 17–51.

[17] R. Naff, T. Russell and J. Wilson, Shape functions for velocity interpolation in general hexahedral cells, Comput. Geosci. 6(3/4) (2002) 285–314.

[18] A. Okabe, B. Boots, K. Sugihara and S.N. Chiu, *Concepts and Applications of Voronoi Diagrams* (Wiley, New York, 2000).

[19] T.S. Palmer, Discretizing the diffusion equation on unstructured polygonal meshes in two dimensions, Ann. Nuclear Energy 28 (2001) 1851–1880.

[20] M. Shashkov, *Conservative Finite-Difference Methods on General Grids* (CRC Press, Boca Raton, FL, 1996).

[21] M. Shashkov and S. Steinberg, Solving diffusion equations with rough coefficients in rough grids, J. Comput. Phys. 129 (1996) 383–405.

[22] K. Stüben, Algebraic multigrid (AMG): Experiences and comparisons, Appl. Math. Comput. 13 (1983) 419–452.

[23] P. Vanêk, M. Brezina and J. Mandel, Convergence of algebraic multigrid based on smoothed aggregation, Numer. Math. 88 (2001) 559–579.