LA-UR-03-7608

Mimetic finite difference method on polygonal meshes *

Yuri Kuznetsov† Konstantin Lipnikov‡ Mikhail Shashkov‡

October 28, 2003

Abstract

Mimetic discretizations based on the support-operators methodology are derived for general polygonal meshes. The second-order convergence rate is verified with numerical examples on general polygonal, locally refined and non-matching meshes.

1 Introduction

Discrete approximations used to predict diffusion heat or matter on a mesh covering the domain are often the determining factor for the reliability, accuracy, and efficiency of the simulations. One of the most effective ways to derive such discrete approximations is to preserve and mimic the underlying properties of the simulated systems. The problem becomes more difficult when the mesh is distorted so that it can conform and adapt to the physical domain and problem solution. One of such approaches, the mimetic finite difference technique based on the support-operator (SO) methodology, has been applied successfully to several applications including diffusion processes (on both conformal [13, 4, 9] and locally refined [7] meshes), electromagnetics [3], and gas dynamics [2].

In this paper we extend a class of mimetic discrete approximations for modeling the diffusion equations on general polygonal meshes. The class of polygonal meshes is very wide and includes meshes used in many applications. For instance, adaptive meshes with hanging nodes are among them. If we consider the hanging nodes as the additional vertices, we shall get a conformal polygonal mesh. Another example comes from applications using nonmatching grids. We may again consider all the mesh points on the interface between nonmatching meshes as vertices of a conformal polygonal mesh.

The SO method can be applied to various formulations of the diffusion equation. In this paper we consider the diffusion problem formulated as the system of two first-order equations for the mass conservation and the flux definition. We begin with considering each mesh polygon as an independent domain and generate an independent discretization for each

*The work was performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36.

†University of Houston, Department of Mathematics, Houston, 77204, kuz@math.uh.edu

‡Los Alamos National Laboratory, MS B284, Los Alamos, NM, 87545, lipnikov@lanl.gov, misha@t7.lanl.gov.

polygon. The local discretization mimics essential properties of the divergence and gradient operators; namely, discrete analogs of these operators satisfy the discrete analog of the Gauss-Green formula. The global discretization is achieved by imposing continuity conditions across polygon interfaces.

Error estimates are important to quantify the accuracy of the discrete solutions and to know when to trust them. The convergence of the SO discretization on triangular and quadrilateral meshes has been proved in [1] by developing a connection of the mimetic discretization with the mixed finite element (MFE) method. Unfortunately the relationship between the SO and textbook MFE methods can not be extended even to non-orthogonal hexahedral meshes. As it was shown in [10] the usual Piola transformed shape functions can not match uniform flow on such meshes. On the other hand, the discretization technique presented in this paper can be extended to general polyhedral meshes resulting in discretizations which are exact for uniform flows.

The extension of the SO methodology to general polygonal meshes was possible after deep analysis of recent theoretical results concerning new finite element prolongation operators in H_{div} on polygonal and polyhedral meshes proposed in [6]. These operators produce the finite element functions which have constant divergence in each polygon. In the context of the SO discretization, the assumption of constant divergence can be waived without lost in the convergence rate (see Section 3). This enlarges the class of the SO discretizations accurate for uniform flows and allows potentially to add new features in the discretization. The new SO discretization results in an algebraic problem with a symmetric positive definite matrix. Therefore, the problem can be solved with the conjugate gradient method.

Limited use of polygonal meshes is mainly restricted by a small number of accurate discretization schemes. We mention here a finite volume scheme proposed in [11]. The scheme is exact for homogeneous linear solutions but results in a non-symmetric coefficient matrix. Therefore, it requires the use of non-traditional iterative solvers.

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

2 Mimetic finite difference method

Let us consider a model elliptic boundary value problem

$$\begin{aligned} \text{div } \mathbf{F} &= Q\,, \\ \mathbf{F} &= -\mathbf{K} \text{ grad } p\,, \end{aligned} \tag{2.1}$$

in a bounded polygonal domain $\Omega \in \Re^2$ with the homogeneous Dirichlet boundary conditions, i.e. $p = 0$ on $\partial\Omega$. Here p denotes a scalar function that we refer to as the pressure, $\mathbf{F}$ denotes a vector function that we refer to as the flux, $\mathbf{K}$ denotes a diffusion tensor, and Q denotes a source function.

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

$$\Omega_h = \bigcup_{i=1}^{m} e_i. \tag{2.2}$$

The elements e_i are general simply-connected polygons with non-zero angles and edges. Therefore, the number of polygon edges equals 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 a common edge of e_i and e_j.

The class of such partitions is very wide and includes meshes used in many applications. For instance, adaptive meshes with hanging nodes (AMR meshes) are among them. A hanging node occurs when two (or more) cells share an edge with one cell. If we consider the hanging node as an additional vertex of that cell, we shall get a polygonal partition (2.2). Another example comes from applications using nonmatching grids. The nonmatching grid occurs when two different meshes are used in two adjacent subdomains. If we consider all the mesh points on the interface between the subdomains as vertices of a conformal grid, we get again a polygonal partition (2.2).

The purpose of this paper is to extend the support-operator (SO) method from [13] to generate new discretizations of (2.1) on general polygonal meshes. The SO method is particularly amenable to polygonal meshes, because the discretization process occurs in two steps. On the first step we consider each element of partition Ω_h as an independent domain and generate an independent discretization for each element. On the second step we obtain the global discretization by imposing continuity of the pressure and the normal component of the flux across element interfaces.

The general scheme of the SO method is as follows. We begin by introducing the vector spaces of discrete functions and scalar products on them. Then, we derive a discrete approximation for the divergence operator which we call the *prime* operator. The approximation to the gradient operator refereed to as the *derived* operator will follow naturally from a discrete analog of the Gauss-Green formula.

We assume that media in each mesh element is homogeneous, but material properties (diffusion tensor) may vary between elements. The scalars are defined at both element centers, p_i^0, and edge centers, $p_i^1, p_i^2, \ldots, p_i^{s_i}$, where s_i is the number of edges of element e_i (see Fig. 1). The vectors are defined in terms of edge-normal components, $f_i^1, f_i^2, \ldots, f_i^{s_i}$, located at the midpoints of element edges. For instance, f_i^1 denotes the dot product of $\mathbf{F}$ with the outward unit normal $\mathbf{n}_i^1$, i.e. $f_i^1 = \mathbf{F} \cdot \mathbf{n}_i^1$.

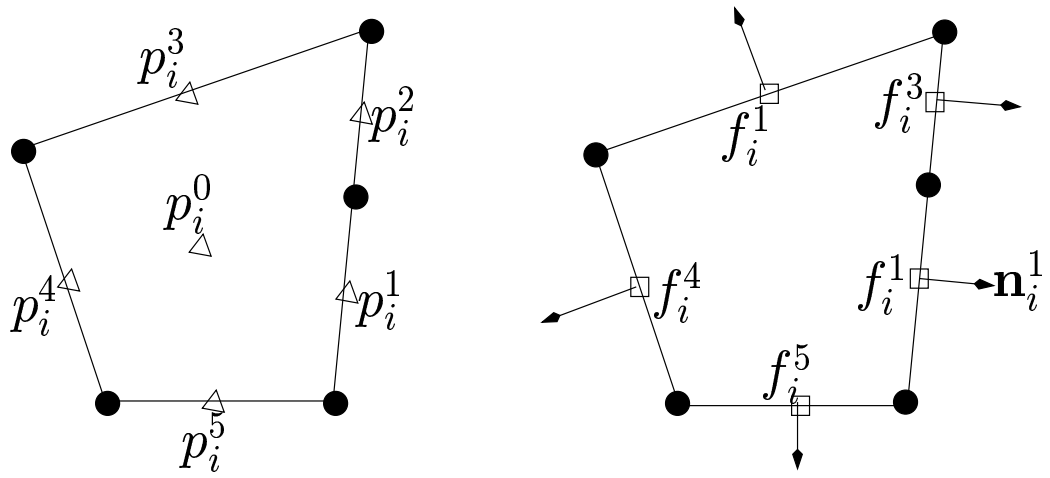

Figure 1: Location of pressure and flux unknowns for a degenerated pentagon.

Let e_i be a convex polygon. Then, the full flux can be recovered at each corner of this cell from two orthogonal projections on edges that share that corner. We denote the recovered vectors by $\mathbf{F}_i^1, \mathbf{F}_i^2, \ldots, \mathbf{F}_i^{s_i}$, and assume that $\mathbf{F}_i^k$ is recovered from f_i^k and f_i^{k+1}. The recovered vectors play important role in the SO methodology (see, e.g. (3.1)). Let $\mathbf{n}_i^k = (n_{ix}, n_{iy})^T$ be the normal vector corresponding to unknown f_i^k. Then,

$$\mathbf{F}_i^k = \begin{bmatrix} n_{ix}^k & n_{iy}^k \\ n_{ix}^{k+1} & n_{iy}^{k+1} \end{bmatrix}^{-1} \begin{bmatrix} f_i^k \\ f_i^{k+1} \end{bmatrix}.$$

The case of a generated polygon is considered in the next section.

The quantities defined above are uniquely associated with a single element. The next step is to use them to derive approximations to the divergence and the gradient operators over the i-element. The discrete divergence operator, DIV , follows naturally from the Gauss divergence theorem as

$$(\mathrm{DIV}\,\mathbf{F})_{e_i} \stackrel{def}{=} \frac{1}{V_i} \sum_{k=1}^{s_i} f_i^k \, \ell_i^k \tag{2.3}$$

where V_i denotes the area of the i-th cell and ℓ_i^k denotes the length of the edge associated with f_i^k. We emphasize one more time that s_i 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 gradient operator, GRAD , is derived from a discrete analog of the Gauss-Green formula

$$\int_{e_i} \mathbf{H} \cdot \mathrm{grad}\, q \, \mathrm{d}V + \int_{e_i} \mathrm{div}\, \mathbf{H}\, q \, \mathrm{d}V = \int_{\partial e_i} \mathbf{H} \cdot \mathbf{n}\, q \; \mathrm{d}l \tag{2.4}$$

where q is an arbitrary scalar function, $\mathbf{H}$ is an arbitrary vector function, and $\mathbf{n}$ denotes the outward unit normal to ∂e_i. The discrete analog of (2.4) is obtained by applying quadrature rules to all three integrals. Let $\mathbf{G} = -\mathbf{K}\,\mathrm{grad}\, q$ be a flux corresponding to the scalar function q. Furthermore, let $h_i = (h_i^1, h_i^2, \ldots, h_i^{s_i})^T$ and $g_i = (g_i^1, g_i^2, \ldots, g_i^{s_i})^T$ be the vectors of edge-normal components of fluxes $\mathbf{H}$ and $\mathbf{G}$, respectively, over element e_i. Then, the first integral in (2.4) is discretized as follows:

$$\int_{e_i} \mathbf{H} \cdot \mathrm{grad}\, q \, \mathrm{d}V = -\int_{e_i} \mathbf{H} \cdot \mathbf{K}^{-1}\mathbf{G} \, \mathrm{d}V \approx -h_i^T \, M_i \, g_i \tag{2.5}$$

where $M_i \in \Re^{s_i} \times \Re^{s_i}$ is a symmetric positive definite matrix representing a quadrature rule. A few methods for deriving accurate quadrature rules will be discussed in the next section.

The second integral in (2.4) is discretized using the definition of the discrete divergence operator and the center point quadrature rule:

$$\int_{e_i} \mathrm{div}\, \mathbf{H}\, q \, \mathrm{d}V \approx (\mathrm{DIV}\,\mathbf{H})_{e_i} \; q_i^0 \, V_i = q_i^0 \sum_{k=1}^{s_i} h_i^k \, \ell_i^k = q_i^0 \left(h_i^T \, L_i \, u_i \right) \tag{2.6}$$

where L_i denotes the diagonal matrix with entries $\ell_i^1, \ \ell_i^2, \ldots, \ell_i^{s_i}$, and $u_i = (1, 1, \ldots, 1)^T$. Finally, the boundary integral in (2.4) is discretized with the central point quadrature rule:

$$\int_{\partial e_i} \mathbf{H} \cdot \mathbf{n}\, q \; \mathrm{d}l \approx \sum_{k=1}^{s_i} h_i^k \, q_i^k \, \ell_i^k = h_i^T \, L_i \, q_i \tag{2.7}$$

where $q_i = (q_i^1, q_i^2, \ldots, q_i^{s_i})^T$ denotes a vector of edge-based scalar unknowns.

Substituting the right hand sides from (2.5), (2.6), (2.7) into the Gauss-Green formula (2.4), we get its discrete analog:

$$h_i^T(-M_i g_i + q_i^0(L_i u_i) - L_i q_i) = 0. \tag{2.8}$$

Components of vector g_i represent the discrete gradient operator GRAD. Note that for the general diffusion tensor, GRAD approximates the continuous operator $-\mathbf{K}\,\mathrm{grad}$. Since flux h_i is an arbitrary vector, formula (2.8) implies that

$$(\mathrm{GRAD}\, q)_{e_i} = M_i^{-1} L_i\,(q_i - q_i^0 u_i). \tag{2.9}$$

The discrete gradient and divergence operators applied to the continuous problem (2.1) result in a local discretization

$$\begin{aligned} (\mathrm{DIV}\ \mathbf{F})_{e_i} &= Q_i, \\ f_i &= (\mathrm{GRAD}\ p)_{e_i} \end{aligned} \tag{2.10}$$

where Q_i denotes the mean value of the source function in the i-th element. The discretization (2.10) is called mimetic because the discrete operators DIV and GRAD *mimic* the essential property of continuous operators div and $-\mathbf{K}\mathrm{grad}$; namely, the Gauss-Green formula. We refer to [12] for a detailed analysis of mimetic discretizations.

The global discretization is achieved by imposing continuity conditions for primary variables $\mathbf{F}$ and p across element interfaces. In the case of continuous problem, the continuity conditions mean that the sum over all cells of (2.4) results in a similar integral identity over the entire domain due to the cancellation of the interior boundary integrals. In the discrete case, the cancellation of the interior boundary contributions happens if

$$f_i^k\, p_i^k\, \ell_i^k + f_j^l\, p_j^l\, \ell_j^l = 0 \tag{2.11}$$

where $\ell_i^k = \ell_j^l$ denotes the length of a common edge of elements e_i and e_j. Note that in the case of a complex interface consisting of a few edges, the assumptions (2.11) are sufficient but not necessary. If we enforce continuity of normal component of fluxes, we end up with a continuity conditions of scalars, and *vice verse*. Thus, the continuity conditions are

$$f_i^k = -f_j^l \quad \text{and} \quad p_i^k = p_j^l. \tag{2.12}$$

Remark 2.1 *It is pertinent to note some advantages of the continuity conditions (2.12) in the case of triangular or quadrilateral AMR meshes. In a particular case when element e_i shares one of its edges with exactly two elements e_j and e_k, the interfaces conditions proposed in [7] enforce the three corresponding fluxes be equal each other. Such continuity condition results in poorer resolution of interface fluxes than (2.12). Moreover, its generalization to AMR meshes with higher level of refinement (three and more) becomes quites complicated. On contrast, the continuity condition (2.12) remains the same for any level of mesh refinement.*

The system of equations (2.10), (2.12) is closed by imposing boundary conditions. For our model problem, the homogeneous Dirichlet boundary conditions imply that the scalar unknowns associated with boundary edges equal to zero.

3 Scalar products in space of fluxes

In this section, we describe a method for deriving a family of scalar products in space of fluxes (quadrature rules) which are exact for uniform flows. The method is based on a quadrature rule for a triangular element proposed and analyzed in [8]. The quadrature rule for a general polygonal element e_i will be given in a form of a symmetric positive definite matrix $M_i \in \Re^{s_i \times s_i}$ where s_i is the number of edges of e_i.

Let Δ be a triangle with vertices $\mathbf{r}_1, \mathbf{r}_2$ and $\mathbf{r}_3$. Let f^k and g^k, be the normal components of fluxes $\mathbf{F}$ and $\mathbf{H}$, respectively, defined at mid points of edges opposite to vertices $\mathbf{r}_k$, $k = 1, 2, 3$. Furthermore, let $\mathbf{H}^k$ and $\mathbf{G}^k$ be the fluxes recovered at vertices $\mathbf{r}_k$, $k = 1, 2, 3$. Then the matrix M_Δ inducing the accurate scalar product of vectors $h_\Delta = (h^1, h^2, h^3)^T$ and $g_\Delta = (g^1, g^2, g^3)^T$ is defined by

$$h_\Delta^T M_\Delta g_\Delta = \frac{|\Delta|}{3} \sum_{k=1}^{3} \mathbf{H}^k \cdot \mathbf{K}_\Delta^{-1} \mathbf{G}^k \tag{3.1}$$

where $|\Delta|$ denotes the area of the triangle and $\mathbf{K}_\Delta$ denotes the value of the diffusion tensor at the mass center of Δ.

In the special case $\mathbf{K}_\Delta = \mathbf{I}$ formulas for entries of the matrix M_Δ are quite simple. Let θ_k be the triangle angle at vertex $\mathbf{r}_k$. Then,

$$M_\Delta = \frac{1}{6} \begin{bmatrix} \dfrac{\ell_1\ell_2}{\sin\theta_3} + \dfrac{\ell_1\ell_3}{\sin\theta_2} & \dfrac{\ell_1\ell_2\cos\theta_3}{\sin\theta_3} & \dfrac{\ell_1\ell_3\cos\theta_2}{\sin\theta_2} \\ \dfrac{\ell_2\ell_1\cos\theta_3}{\sin\theta_3} & \dfrac{\ell_2\ell_1}{\sin\theta_3} + \dfrac{\ell_2\ell_3}{\sin\theta_1} & \dfrac{\ell_2\ell_3\cos\theta_1}{\sin\theta_1} \\ \dfrac{\ell_3\ell_1\cos\theta_2}{\sin\theta_2} & \dfrac{\ell_3\ell_2\cos\theta_1}{\sin\theta_1} & \dfrac{\ell_3\ell_1}{\sin\theta_2} + \dfrac{\ell_3\ell_2}{\sin\theta_1} \end{bmatrix}. \tag{3.2}$$

In the case of an arbitrary polygon, the quadrature rules (3.1) suggests the following strategy. We split the polygon into a few triangles, apply the above quadrature rule, and then "eliminate" the unknowns associated with interior auxiliary edges. The main attributes of the approach are its simplicity and the fact that it can be easily incorporated into existing simulation codes. Moreover, it can be easy shown that thus obtained scalar product of fluxes results in a numerical scheme which is exact for linear solutions.

Let us assume that a polygon e_i is split into p triangles Δ_{il}, $l = 1, \ldots, p$. Let $h_{\Delta_{il}} = (h_{il}^1, h_{il}^2, h_{il}^3)^T$, $l = 1, \ldots, p$, be the vectors of edge-based normal component of fluxes, $q_{\Delta_{il}} = (q_{il}^1, q_{il}^2, q_{il}^3)^T$ be the vectors of edge-based pressures, and q_{il}^0 be the triangle-centered pressures. It is obvious that local mimetic discretizations similar to (2.8) can be derived for each of the triangles. The fact that the mimetic discretization on a triangular mesh is exact for linear solutions means that the discrete Gauss-Green formulas

$$h_{\Delta_{il}}^T M_{\Delta_{il}} g_{\Delta_{il}} + (\mathrm{DIV}\,\mathbf{H})_{\Delta_{il}} q_{il}^0 V_{\Delta_{il}} = \sum_{k=1}^{3} h_{il}^k \, \ell_{il}^k \, q_{il}^k, \qquad l = 1, \ldots, p, \tag{3.3}$$

are identities when q_{il}^0 are evaluated at the mass centers of Δ_{il}, q_{il}^k and g_{il}^k are evaluated at the mid-points of the associated triangle edges, and $h_{\Delta_{il}}$ is an arbitrary vector from $\Re^3$. Here $V_{\Delta_{il}}$ denotes the area of triangle Δ_{il}.

Recall that the original variables associated with the polygon e_i are the normal components of the boundary fluxes and the cell-centered pressure q_i^0. In order to eliminate the other

variables, we make a simple assumption about divergence of vector fields over the polygon e_i which will allow us to combine formulas (3.3) in such a way to derive the discrete Gauss-Green formula (2.8) for the polygon e_i. We rewrite formula (2.8) using notations similar to that in (3.3):

$$h_i^T M_i\, g_i + (\mathrm{DIV}\,\mathbf{H})_{e_i}\, q_i^0 V_{e_i} = \sum_{k=1}^{s_i} h_i^k\, \ell_i^k\, q_i^k. \tag{3.4}$$

The matrix M_i represents a quadrature rule over the polygon e_i which gives us a numerical scheme exact for linear solutions. Let $\tilde{h}_i^T = (\tilde{h}_{i,1}^T, \tilde{h}_{i,2}^T)$ be a vector of the edge-based normal components of fluxes where $\tilde{h}_{i,1}$ corresponds to boundary edges, i.e. $\tilde{h}_{i,1} = h_i$, and $\tilde{h}_{i,2}$ corresponds to interior edges. Furthermore, let α_{il}, $l = 1, \ldots, p$, be *a priori* given non-zero numbers such that

$$(\mathrm{DIV}\,\mathbf{H})_{\Delta_{il}} = \alpha_{il} (\mathrm{DIV}\,\mathbf{H})_{e_i}. \tag{3.5}$$

The numbers α_{il} represent relations between values of the discrete divergence over the polygon and its parts. We assume that the same relations hold for the vector field $\mathbf{G}$. If e_i is a convex regular-shaped polygon, it is physically reasonable to assume that the divergence over each triangle equals to the divergence over the polygon, i.e. $\alpha_{il} = 1$. A particular example of a non-convex polygon and $\alpha_{il} \neq 1$ will be given later.

The standard assembling procedure applied to (3.3) gives

$$\tilde{h}_i^T \tilde{M}_i \tilde{g}_i + (\mathrm{DIV}\,\mathbf{H})_{e_i} \sum_{l=1}^{p} \alpha_{il} V_{\Delta_{il}} q_{il}^0 = \sum_{k=1}^{s_i} h_i^k\, \ell_i^k\, q_i^k \tag{3.6}$$

where

$$\tilde{M}_i = \sum_{s=1}^{p} \mathcal{N}_{i,l} M_{\Delta_{il}} \mathcal{N}_{i,l}^T$$

and $\mathcal{N}_{i,l}$ are the assembling matrices. Note that the terms in the right hand sides of (3.3) corresponding to interior edges cancel one another in (3.6) leaving only the boundary contributors. Assumption (3.5) was used to derive the second term in the left hand side. A comparison of this term with the second term in (3.4) gives us a relation between cell-centered scalar unknowns induced by this assumption:

$$q_i^0 = \sum_{s=1}^{p} \alpha_{il} \frac{V_{\Delta_{il}}}{V_i} q_{il}^0. \tag{3.7}$$

It is pertinent to stress here that formula (3.7) may be used only for a theoretical analysis. In numerical experiments we compute q_i^0 by solving the saddle point problem (2.10), (2.12).

The vectors $\tilde{h}_{i,2}$ and $\tilde{g}_{i,2}$ are involved solely in the first term. They are associated with interior edges and have to be eliminated in order to comply with the original discrete Gauss-Green formula (2.8) (or with its equivalent representation (3.4)). Note that vectors $\tilde{h}_{i,2}$ and $\tilde{g}_{i,2}$ are related to vectors $\tilde{h}_{i,1} = h_i$ and $\tilde{g}_{i,1} = g_i$, respectively, via equations (3.5). The solvalibility of these equations is analyzed in the following lemma.

Lemma 3.1 *Let the numbers α_{il} be such that*

$$\sum_{l=1}^{p} \alpha_{il} V_{\Delta_{il}} = V_{e_i}. \tag{3.8}$$

Then system (3.5) has a solution for any vector h_i.

Proof. Let r be the number of interior edges of polygon e_i. The system (3.5) may be rewritten as follows:

$$B_2\,\tilde{h}_{i,2} = B_1\,h_i \tag{3.9}$$

where $B_2 \in \Re^{p\times r}$ and $B_1 \in \Re^{p\times s_i}$. The system (3.9) is compartible if and only if the right hand side, $B_1\,h_i$, is orthogonal to $\ker B_2^T$. Let us show that the dimension of $\ker B_2^T$ equals to one. If $\psi \in \ker B_2^T \in \Re^p$ then

$$(B_2^T\psi,\,\eta) = 0, \qquad \forall\eta \in \Re^r. \tag{3.10}$$

Since each interior edge is shared by two triangles, each row of B_2^T has exactly two non-zero entries. Let Δ_{ik} and Δ_{il} be two triangle having a common edge. Furthermore, let vector η has only one non-zero component corresponding to this edge. Then equation (3.10) results in

$$\frac{1}{V_k}\psi_k = \frac{1}{V_l}\psi_l. \tag{3.11}$$

Since the polygon e_i is simply connected, one component of vector ψ defines uniquely the other components. Therefore dimension of $\ker B_2^T$ is at most 1. It is obvious that vector $\psi = \{V_1,\ldots,V_p\}$ is a non-trivial solution of (3.11). Thus, $\dim(\ker B_2^T) = 1$.

Now, we prove that $\psi \perp B_1 h_i$. Since the vector h_i of normal components of boundary fluxes is arbitrary, the orthogonality condition holds if and only if $B_1^T\psi = 0$. The definition of the discrete divergence and system (3.5) imply that

$$(B_1^T\psi)_k = \sum_{l=1}^{p}\frac{\ell_{ik}}{V_{e_i}}\alpha_{il}\,\psi_l - \frac{\ell_{ik}}{V_{\Delta_{ik}}}\,\psi_k = \ell_{ik}\left(\sum_{l=1}^{p}\frac{V_{\Delta_{il}}\alpha_{il}}{V_{e_i}} - 1\right).$$

The equation (3.8) imply now that $(B_1^T\psi)_k = 0$. Since the k is arbitrary, we get $B_1^T\psi = 0$ which proves the assertion of the lemma. □

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

Let us consider a couple of typical examples illustrating Lemma 3.1 and Corollary 3.1. As shown on Fig. 2, there exist a few ways to split a polygon e_i into triangles either by introducing additional interior points (one or more) or without such points.

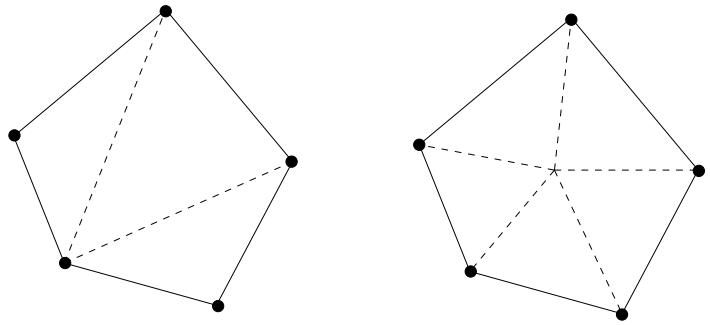

Figure 2: Two ways for splitting a pentagon.

In the first case (left picture in Fig. 2) we have $p-1$ auxiliary interior edges and p equations for the corresponding fluxes. Thus, $\dim(\ker B_2) = (p-1) - p + 1 = 0$ and system (3.5) has a

unique solution. In the second case (right picture in Fig. 2) we have p auxiliary interior edges and p equations for the corresponding fluxes. Thus, $\dim(\ker B_2) = p - p + 1 = 1$ and we get one-parametric family of solutions for (3.5). In both cases, it is possible to resolve equations (3.5) as

$$\tilde{h}_{i,2} = B\, h_i \qquad \text{and} \qquad \tilde{g}_{i,2} = B\, g_i \tag{3.12}$$

where $B \in \Re^{r\times p}$ is a rectangular matrix. Using these relations, we may rewrite the first term in (3.4) as follows:

$$\tilde{h}_i^T \tilde{M}_i\, \tilde{g}_i = \begin{bmatrix} h_{i,1}^T & \tilde{h}_{i,2}^T \end{bmatrix} \begin{bmatrix} \tilde{M}_{i,11} & \tilde{M}_{i,12} \\ \tilde{M}_{i,21} & \tilde{M}_{i,22} \end{bmatrix} \begin{bmatrix} g_{i,1} \\ \tilde{g}_{i,2} \end{bmatrix} = h_i^T M_i\, g_i \tag{3.13}$$

where

$$M_i = \tilde{M}_{i,11} + B^T \tilde{M}_{i,22} B + \tilde{M}_{i,12} B + B^T \tilde{M}_{i,21}.$$

Let us prove that the discrete Green formula (3.4) with the mass matrix M_i derived above is exact for linear solutions. It is proved in [8] that formula (3.3) is exact for triangles and linear solutions. Therefore formula (3.6) is exact for linear solutions if $\mathbf{H}$ satisfies (3.5) (resp., if $\tilde{h}_i$ satisfies (3.12)). Since the boundary fluxes h_i are arbitrary in (3.12), formula (3.4) is exact for linear solutions when q_i^0 is evaluated at the point

$$\mathbf{c}_i^0 = \sum_{s=1}^{p} \alpha_{il} \frac{V_{\Delta_{il}}}{V_i} \mathbf{c}_{il}^0$$

where $\mathbf{c}_{il}^0$ is the centroid of triangle Δ_{il}. The above statement follows from the fact that $q_{il}^0 = q(\mathbf{c}_{il}^0)$ and $q(\mathbf{x})$ is linear in $\mathbf{x}$. In the special case when all α_{il} equal to one, $\mathbf{c}_i^0$ is the mass center of the polygon e_i.

At the end of this section we shall consider a few special cases of polygons which may cause some problems in the existing codes (see Fig. 3).

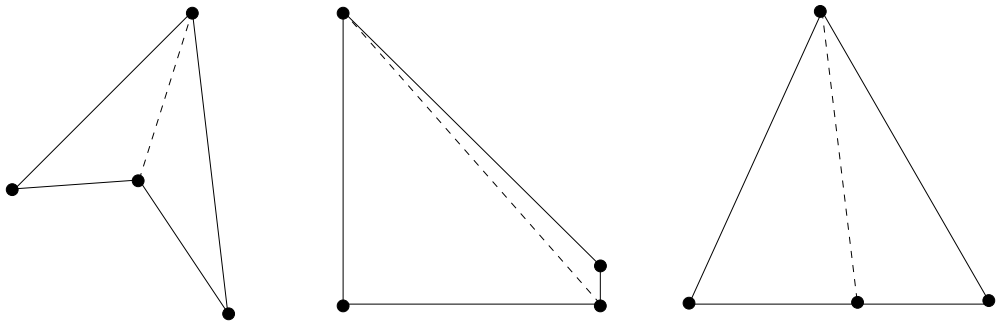

Figure 3: A few cases of bad polygons.

The left polygon in Fig 3 is a non-convex quadrilateral element e_i. A proper splitting shown on the figure converts it into two triangles. It is possible that the mass center of a non-convex element lies outside the element. In this case a special choice of numbers α_{il} and probably another splitting of element e_i may help to move a point where a linear solution has to be evaluated from the mass center of the element to a point inside it.

The middle quadrilateral e_i in Fig 3 has an edge whose length is relatively smaller than length of the other edges. The splitting shown in figure results is a thin triangle. Let us

enumerate the edges of this quadrilateral counter-clockwise starting with the smallest edge. Thus, ℓ_i^1 be the length of the smallest edge and ℓ_i^5 be the length of the auxiliary interior edge. Furthermore, let Δ_1 denote the thin triangle and Δ_2 be the other triangle. Assuming for simplicity that $\alpha_1 = \alpha_2 = 1$ in (3.5), we get

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

In the asymptotic case, $\ell_i^1 \to 0$, the above formula gives the expected result:

$$f_i^5 = f_i^2 + O(l_i^1).$$

Substituting this estimate into scalar product (3.1), (3.2), we get that

$$h_{\Delta_1}^T M_{\Delta_1} g_{\Delta_1} \sim \ell_i^1\, h_{\Delta_1}^T g_{\Delta_1}.$$

Thus, the contribution of the thin triangle into the scalar product over the quadrilateral e_i is negligent.

The right quadrilateral e_i in Fig. 3 has two edges which form an angle of 180 degrees. Like in the case on the non-convex quadrilateral, a proper splitting shown on the picture resolves the problem.

4 Solution method

In this section we shall show that system (2.10), (2.12) can be easily reduced to a system of linear algebraic equations for edge-based pressure unknowns with a symmetric positive definite matrix.

The continuity conditions (2.12) allow us to introduce a single scalar unknown for each interior mesh edge. Let ℓ_{ij} be the length of common edge between elements e_i and e_j. We denote the corresponding pressure unknown as p_{ij} where $i < j$. In a similar way, we denote the flux from i-th element to j-th element as f_{ij}. The new notations allow us to rewrite the flux continuity conditions as

$$\ell_{ij} f_{ij} = -\ell_{ij} f_{ji}, \qquad i < j. \tag{4.1}$$

Using the above notations, we rewrite system (2.10), (2.12) in a symmetric form. Let $C_i = L_i u_i$, i.e. $C_i = (\ell_i^1, \ell_i^2, \ldots, \ell_i^{s_i})^T$. Then,

$$\begin{array}{rcl} M_i f_i + C_i p_i^0 - \sum\limits_{k=1}^{s_i} \ell_{ij} p_{ij} &=& 0, \\ C_i^T f_i &=& Q_i, \\ -\ell_{ij} f_{ij} - \ell_{ij} f_{ji} &=& 0, \end{array} \tag{4.2}$$

where $i = 1, \ldots, m$ and $i < j$.

System (4.2) is a typical example of a saddle point problem. Since it is symmetric, it can be solved by the Lanzcos iterative method. However, a few simple algebraic transformations will reduce (4.2) to a problem with a symmetric positive definite matrix. Note that the primary variables f_i and p_i^0, $i = 1, \ldots, N$, are only connected within a single cell. Therefore, they may be easy excluded from the system with $O(m)$ arithmetical operations resulting in

$$S\,p = b \tag{4.3}$$

where p denotes the global vector of edge-based scalar unknowns p_{ij}, and b is a right-hand side vector.

Lemma 4.1 *The matrix S is symmetric and 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 S is sparse.

Problem (4.3) can be solved by the preconditioned conjugate gradient method. In the numerical experiments we use the algebraic multigrid preconditioner from [14]. After solving problem (4.3), we may recover the primary variables element-by-element using local systems (4.2).

5 Numerical experiments

In 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 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 disturbed meshes and random isotropic diffusion tensors, its performance may degrade. In order to ensure the robustness of the solver, the algebraic multigrid may be replaced by (or combined with) more robust multigrid methods such as aggregation multigrid methods [15] and algebraic multigrid methods with projectors [5].

5.1 Uniform flows

The first set of experiments verifies that our method is exact for linear functions. We consider the diffusion problem (2.1) in the unit square $(0,\,1)^2$ subject to the non-homogeneous boundary conditions. Let $\mathbf{K} = \mathbf{I}$ and $p(x,\,y) = x + y$ be the exact solution.

We split each polygon into triangles by inserting one interior point and connecting it with the polygon vertices. We consider the case when all $\alpha_{il} = 1$. This splitting means that the matrix B from (3.12) is not unique. However, if we specify a particular formula for computing one of the interior edge-based fluxes, the other fluxes will be calculated recursively using equations (3.5). For a given polygon e_i, we define one of the minimal angles and recover flux $\mathbf{F}_i^k$ at the corresponding vertex. Then the required formula defines the orthogonal projection of the recovered flux $\mathbf{F}_i^k$ onto a unit vector normal to the interior edge corresponding to the chosen angle.

The result of numerical experiment are shown in the left picture in Fig. 4. As it was proved in Section 3, the computed solution is exact at both the edge mid points and the polygon mass centers.

5.2 Median meshes

The second set of calculations addresses the accuracy of the method on highly distorted adaptive meshes in the Cartesian coordinate system. We again consider the diffusion problem (2.1) in the unit square subject to the Dirichlet boundary conditions. Let $\mathbf{K} = \mathbf{I}$ and

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

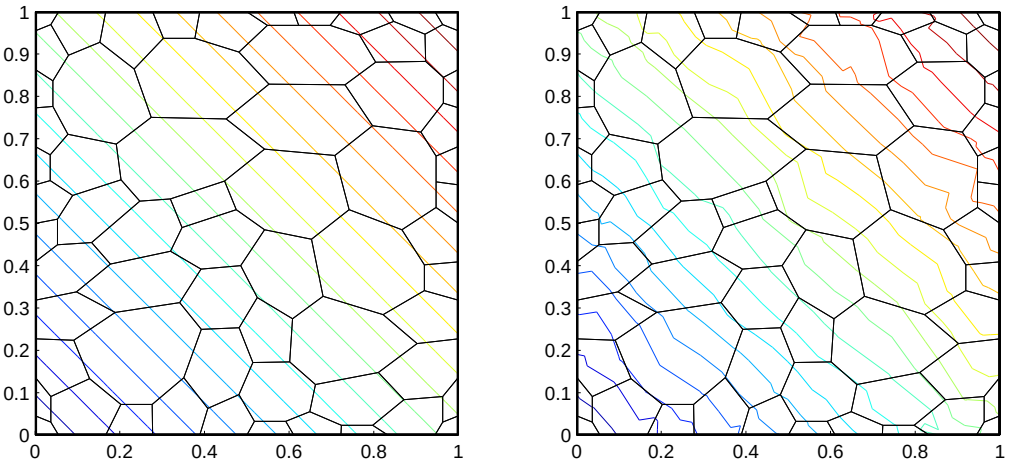

Figure 4: The isolines of discrete linear solutions for two scalar products.

Table 1: Convergence on a sequence of AMR meshes.

l	m	AMR/polygonal grids		AMR/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

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 grids build in two steps.

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

The random grid is generated by moving each mesh point to a random position inside a square centered at the point. The sides of the square are aligned with the coordinate axis and equal 80% of the size of the smallest cell sharing the point. Note that the hanging mesh points are always located in the middle of the associated edges. The relative L_2 errors ε_p and ε_f for cell-centered scalars and edge-based fluxes, respectively, are given in Table 1. The new scalar product results in more accurate solution than the discretization technique proposed in [7]. This is probably due to the fact the new discretization scheme uses two fluxes on the edge with hanging node while the scheme described in [7] uses only one flux. The computational grid and isolines of the computed solution are shown in Fig. 5.

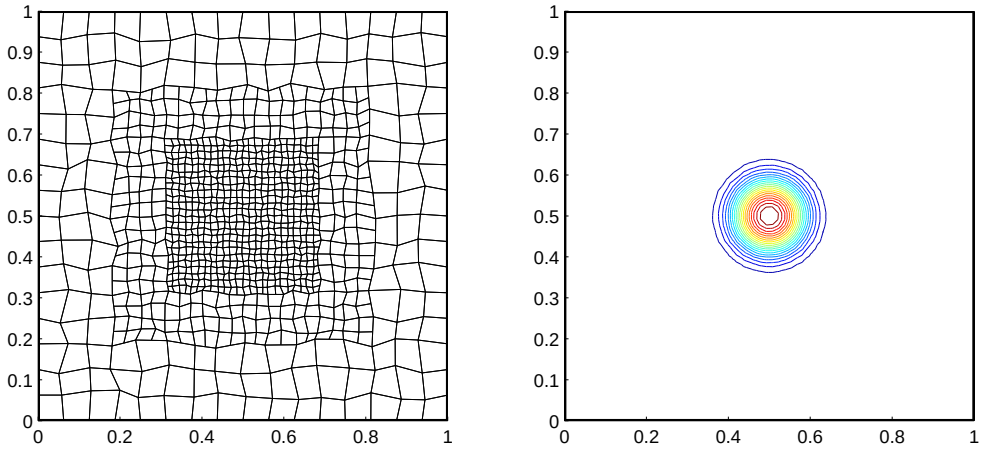

Figure 5: The computational grid on level $l = 2$ and isolines of the discrete solution.

5.3 AMR meshes

In the third set of experiments we compare the new scalar product in the space of fluxes with that analyzed in [1, 7]. The previously used scalar product extends the idea of (3.1) to arbitrary quadrilaterals. This idea can be further extended to a convex polygon e_i:

$$h_i^T M_i^{old} g_i = \frac{V_i}{\tilde{V}_i} \sum_{k=1}^{s_i} \tilde{V}_{\Delta_k} \mathbf{F}_i^k \cdot \mathbf{K}_i \, \mathbf{G}_i^k, \qquad \tilde{V}_i = \sum_{k=1}^{s_i} \tilde{V}_{\Delta_k}, \tag{5.1}$$

where Δ_k denotes a triangle uniquely defined by two edges ending at the k-th vertex of e_i and $\tilde{V}_{\Delta_k}$ denotes the area of this triangle. Note that if e_i is a triangle, then $V_i = V_{\Delta_k}$ and we recover formula (3.1). Unfortunately, scalar product (5.1) results in numerical schemes which are not exact for linear solutions (see right picture in Fig. 4).

In order to compare two scalar product we consider diffusion problem (2.1) on 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 result of numerical experiments are shown on Table 2. The errors are computed on a sequence of polygonal median meshes. The set of points $\mathbf{x}_{ij} = (x_{ij}, y_{ij})$ for generating first the Voronoï tesselation and then the median mesh is given by

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

where $\xi_i = i/n_x$ and $\eta_j = j/n_y$.

Both quadratures result in the second order convergence rate for the pressure unknown. However, the new quadrature gives better resolution of both the pressure and the flux. The computational grid and isolines of the computed solution are shown in Fig. 6.

5.4 Non-matching meshes

In the forth set of experiments we consider non-matching quadrilateral grids and perform numerical analysis of the convergence rates. We consider the elliptic problem (2.1) in the unit square with mixed boundary conditions. On the bottom and top boundary parts, we impose

Table 2: 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

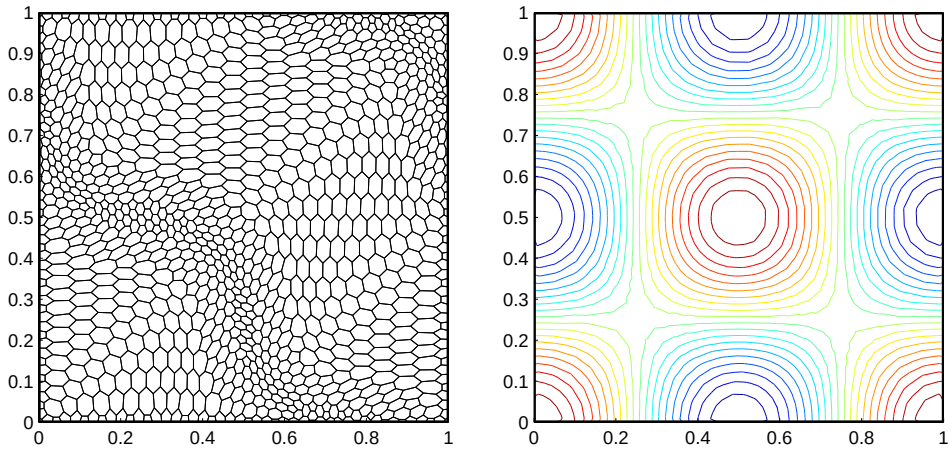

Figure 6: The computational grids and isolines of the discrete solutions on the finer grid.

the Dirichlet boundary condition. The homogeneous Neumann boundary condition is set on the rest of the boundary. Let the diffusion coefficient equals to K_1 in the region defined by $y < 0.5$ and equals to K_2 in the rest of the domain. The source term $Q(x, y)$ is chosen in such a way that the exact solution is given by

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

The computational mesh and the discrete solution for $K_2 = 4$ and $K_1 = 1$ are shown in Fig. 7. The random grids below and above interface line $y = 0.5$ were generated using the rules described above. The isolines of the discrete solution are straight lines except the isoline located below the mesh interface where the computational grid is quite coarse. The relative L_2 errors ε_p and ε_f presented in Table 3 show the second order convergence rates for the pressure and only the first order for the flux. This confirms the convergence rates observed in the previous experiments.

We measure the aspect ratio, denoted by ρ_i, of polygon e_i as the ratio of the biggest edge length to the smallest one. The middle column in Table 3 shows that the non-matching quadrilateral grids may result in very bad interface polygons (see, e.g., the middle polygon in Fig. 3). However, it did not make any noticeable impact on the convergence rate.

The last two columns in Table 3 demonstrate excellent performance of the AMG solver in the case of smooth solutions. The stopping criterion for the preconditioned conjugate

Table 3: Convergence analysis on non-matching grids.

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

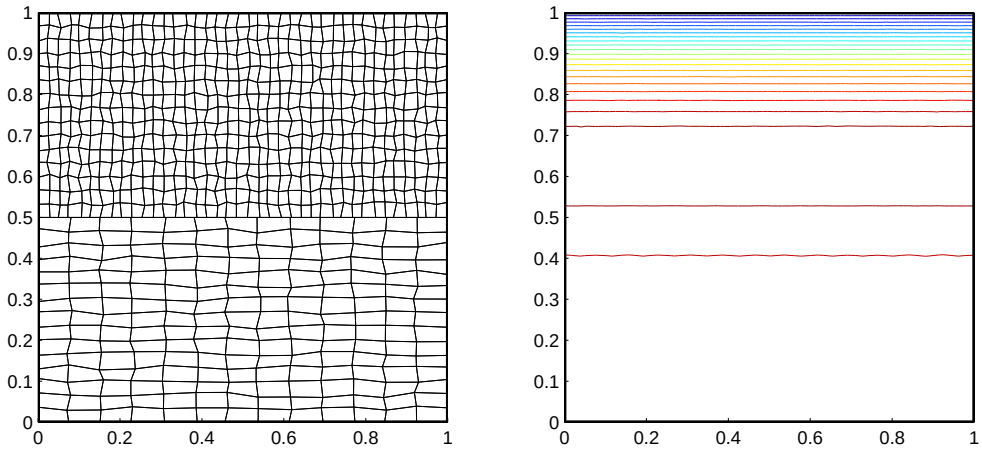

Figure 7: The computational grid and isolines of the discrete solution.

gradient method was the relative decrease in the norm of the residual by factor 10^{-12}. The computational time, denoted by CPU, includes the arithmetical cost for initializing the AMG solver and solving problem (4.3). The arithmetical cost per iteration grows almost linear in the number of cells, m, except for the very coarse grids where cache memory effects play the 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 Fig. 8 represents the discrete solutions corresponding to the grid shown in Fig. 7. The breaking of isolines which are supposed to be the straight lines is due to a coarse grid discretization. Indeed, the solution isolines on the refined grid (right picture) are very close to straight lines.

6 Conclusion

The SO discretization has been shown to be second-order accurate for the pressure variable on general polygonal, AMR and non-matching meshes. Existence of accurate SO discretizations for tetrahedral meshes, enables us an extension of the methodology to general polyhedral meshes. We shall address this issue in future publications.

Our methodology for deriving accurate SO discretizations on polygonal meshes can be repeated for other PDEs including equations of linear elasticity and gas dynamics.

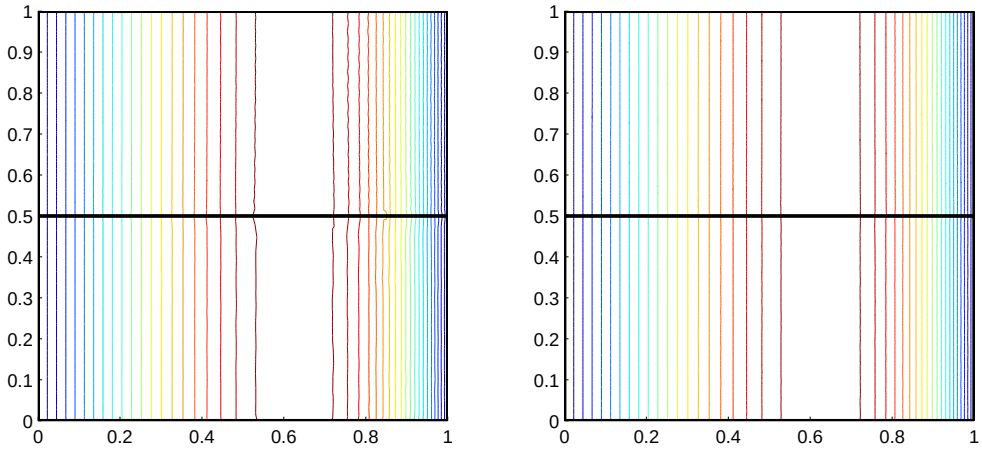

Figure 8: Isolines of the discrete solutions on two consecutive meshes.

Acknowledgments

The authors thank Dr. Raphael Loubere (LANL) for his assistance in generating polygonal meshes.

References

[1] M. Berndt, K. Lipnikov, D. Moulton, and M. Shashkov. Convergence of mimetic finite difference discretizations of the diffusion equation. *J. Numer. Math.*, 9:253–284, 2001.

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

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

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

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

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

[7] K. Lipnikov, J. Morel, and M. Shashkov. Mimetic finite difference methods for diffusion equations on non-orthogonal AMR meshes. *J. Comput. Phys.*, 2003. submitted.

[8] R. Liska, M. Shashkov, and V. Ganzha. Analysis and optimization of inner products for mimetic finite difference method on triangular grid. Technical report, Los Alamos National Laboratory, 2003.

[9] J. Morel, R. Roberts, and M. Shashkov. A local support-operators diffusion disretization scheme for quadrilateral $r-z$ meshes. *J. Comput. Phys.*, 144:17–51, 1998.

[10] R. Naff, T. Russell, and J. Wilson. Shape functions for velocity interpolation in general hexahedral cells. Technical Report Report 178, Center for Computational Mathematics, University of Colorado at Denver, 2001.

[11] T. S. Palmer. Discretizing the diffusion equation on unstructured polygonal meshes in two deminsions. *Annals of Nuclear Energy*, 28:1851–1880, 2001.

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

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

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

[15] P. Vanêk, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik*, 88:559–579, 2001.