A NODE RECONNECTION ALGORITHM FOR MIMETIC FINITE DIFFERENCE DISCRETIZATIONS OF ELLIPTIC EQUATIONS ON TRIANGULAR MESHES*

PAVEL VÁCHAL †, MARKUS BERNDT ‡, KONSTANTIN LIPNIKOV §, AND MIKHAIL SHASHKOV¶

Abstract. Most efficient adaptive mesh methods employ a few strategies, including local mesh refinement (h-adaptation), movement of mesh nodes (r-adaptation), and node reconnection (c-adaptation). Despite its simplicity, node reconnection methods are seldom analyzed apart from the other adaptation methods even in applications where severe restrictions are imposed on topological operations with a mesh. However, using only node reconnection the discretization error can be significantly reduced. In this paper, we develop and numerically analyze a new c-adaptation algorithm for mimetic finite difference discretizations of elliptic equations on triangular meshes. Our algorithm is based on a new error indicator for such discretizations, which can also be used for unstructured general polygonal meshes. We demonstrate the efficiency of our new algorithm with numerical examples.

Key words. adaptive mesh method, node reconnection, mimetic finite difference, tensor coefficient

MSC2000 subject classifications. 65N50, 65N06

1. Introduction

The predictions and insights gained from numerical simulations can only be as good as the underlying physical models and their discrete approximations. One of the approaches that can be used to increase accuracy of discrete approximations is to employ adaptive meshes. The strategies that are used for the generation of adaptive meshes can be divided into three categories: local mesh refinement (h-adaptation; for early work in the context of finite element discretizations, see [2, 21]), movement of mesh nodes (r-adaptation; see, for example, [5, 17, 24]), and node reconnection (c-adaptation; e.g. [23]). Note that r-adaptation does not change the connectivity of the mesh, while both h-adaptation and c-adaptation do, by the introduction of new mesh nodes and edges, and by changing the connectivity of the mesh, respectively. Despite their simplicity, c-adaptation methods are seldom analyzed apart from the other adaptation methods.

In this paper, we study a pure node reconnection strategy for a diffusion problem. For a number of engineering applications, the diffusion problem is only a small part of a complex simulation. The physical processes involved in the simulation may impose severe restrictions on the mesh topological operations. For example, in the Lagrangian methods, the r- and h-adaptation methods are still under development and have many unresolved numerical and theoretical problems. Some of the existing numerical schemes are sensitive to sharp mesh changes as in the h-adaptation methods. The

r-adaptation methods improve solution accuracy during a single time step, but the accumulated effect of errors due to data remapping in a long simulation may overcome the benefits of r-adaptation. To avoid interference of different physical processes, each step in the simulation has to be analyzed separately. The insight we get from this analysis can be used to build a robust and flexible mesh adaptation method which combines more than one adaptation strategy.

In our node reconnection adaptation strategy, we fix the positions of mesh nodes and modify only the topology. For triangular meshes, the topological changes are reduced to swapping of mesh edges. In particular, a given edge has two adjacent triangles that form a quadrilateral for which this edge is a diagonal. Deleting this edge and introducing a new edge that coincides with the other diagonal of the patch is referred to as edge swapping. We show that even with such a simple approach the discretization error can be significantly decreased. It has been shown by many authors (see, e.g., [9, 22, 8, 1] and references therein) that the interpolation error for P_1-finite element approximations is reduced when the mesh is aligned with solutions features. The optimal mesh contains elements which are stretched along the direction of minimal second derivative of a solution. Our numerical experiments show that the same statement may not be true for mixed discretizations (see Sec. 5).

We consider the model elliptic boundary value problem:

$$\text{div } \vec{J} = q \tag{1.1}$$

$$\vec{J} = -K \text{ grad } \phi. \tag{1.2}$$

Here ϕ denotes a scalar function that we refer to as the intensity, $\vec{J}$ denotes a vector function referred to as the current, K denotes a full symmetric diffusion tensor, and q denotes a source function. The problem is posed in a bounded polygonal domain $\Omega \subset \Re^2$, and is subject to appropriate boundary conditions on $\partial\Omega$. For simplicity, we assume that homogeneous Dirichlet boundary conditions are imposed on $\partial\Omega$.

On a particular mesh, the above defined elliptic problem will be discretized using the mimetic finite difference (MFD) method. This method mimics the fundamental properties of the continuous equation; namely, the symmetry of the divergence and the gradient operators ($\text{div}^* = -K \text{ grad}$ in K^{-1} weighted inner product), local mass balance, and the null spaces of the involved operators. The MFD method has been successfully applied to many different systems of first-order PDEs on simplicial [15, 14, 11, 19] and polygonal meshes [16, 7]. The family of MFD methods contains a mixed finite element method with the lowest order Raviart-Thomas elements [7, 19].

A node reconnection strategy (as well as the other adaptation strategies) requires an efficient error indicator, which is used to decide where to modify the mesh. We propose a new error indicator which is well suited for mimetic finite difference discretizations. On each triangular element, a linear function is reconstructed using a least-squares fit to the discrete solution. The error is then estimated by integrating the square of the discontinuity in the linear functions across triangle edges. Note that this error indicator can also be used on unstructured polygonal meshes.

The paper outline is as follows. In Section 2, we describe briefly the MFD method. In Section 3, we develop a new error indicator. The actual mesh adaptation strategy based on a local version of this error indicator is covered in Section 4. The numerical tests in Section 5 demonstrate the basic properties of the proposed method and its efficiency.

2. Mimetic finite difference method

Let Ω_h be a non-overlapping conformal partition of Ω into triangular elements E. For every element E we denote by $|E|$ its area and by h_E its diameter. Similarly, for each edge ℓ we denote by $|\ell|$ its length. For the sake of simplicity of our notation, we denote by ∂E the boundary of E as well as the union of edges of E.

We introduce the current operator $\mathcal{G}$, $\mathcal{G}\phi = -K\operatorname{grad}\phi$. Furthermore, we introduce two inner products

$$(\vec{J}, \vec{I})_X = \int_\Omega \vec{J} \cdot K^{-1} \vec{I} \, dA \qquad \text{and} \qquad (\phi, \psi)_Q = \int_\Omega \phi\psi \, dA \qquad (2.1)$$

in the space X of currents and in the space Q of intensities, respectively.

Using the introduced notation and imposing the homogeneous Dirichlet conditions, the well known identity

$$\int_{\partial\Omega} \phi(\vec{J}, \phi) \, dl = \int_\Omega \phi \operatorname{div} \vec{J} \, dA + \int_\Omega (\vec{J}, \operatorname{grad}\phi) \, dA \qquad (2.2)$$

can be written as

$$(\vec{J}, \mathcal{G}\phi)_X = (\phi, \operatorname{div}\vec{J})_Q.$$

This expression clearly states that the current and divergence operators are adjoint to each other, i.e. $\mathcal{G} = \operatorname{div}^*$. The MFD method produces discretizations of these operators that are adjoint to each other with respect to inner products in discrete spaces. The MFD method can be divided into four major steps.

The first step of the MFD method. The first step of the MFD method is to specify degrees of freedom for the physical variables ϕ and $\vec{J}$ and their location. We consider the space Q^d of discrete intensities that are constant on each triangle E. The dimension N_Q of Q^d is obviously equal to the number of triangles in Ω_h. The component of $\phi \in Q^d$ associated with E will be denoted by ϕ_E. Finally, for every sufficiently smooth function ψ (e.g. $\psi \in L^1(\Omega)$), we define the vector $\boldsymbol{\psi}^I \in Q^d$ by

$$\boldsymbol{\psi}^I_E = \frac{1}{|E|} \int_E \psi \, dA. \qquad (2.3)$$

With every element E in Ω_h and to every edge ℓ of E, we associate a number $\mathbf{J}^\ell_E$ approximating the normal component of the average current through edge ℓ. Here, we make the *continuity* assumption that for each edge ℓ shared by two triangles E_1 and E_2, we have

$$\mathbf{J}^\ell_{E_1} = -\mathbf{J}^\ell_{E_2}. \qquad (2.4)$$

Let X^d be the vector space of edge-based discrete currents. The dimension N_X of X^d is equal to the number of boundary edges plus *twice* the number of interior edges. We take X^d to be the subspace of $\Re^{N_X}$ in which the continuity conditions (2.4) hold. The restriction of $\mathbf{J} \in X^d$ to the boundary of E will be denoted by $\mathbf{J}_E$. Finally, for every sufficiently smooth vector-valued function $\vec{I}$ (e.g., $\vec{I} \in (L_s(\Omega))^2$, $s > 2$, and $\operatorname{div}\vec{I} \in L^2(\Omega)$), we define $\mathbf{I}^I \in X^d$ for an edge $\ell \in \partial E$ by

$$(\mathbf{I}^I)^\ell_E = \frac{1}{|\ell|} \int_\ell \vec{I} \cdot \vec{n}^\ell_E \, dl. \qquad (2.5)$$

It was proved in [6] that this interpolation operator is well defined and uniformly bounded.

The second step of the MFD method. In the second step of the MFD method, we equip the spaces of discrete intensities and currents with inner products. The inner product on Q^d is given by

$$[\phi, \psi]_{Q^d} = \sum_{E \in \Omega_h} \phi_E \psi_E |E|, \qquad (2.6)$$

while the inner product on X^d is given by

$$[\mathbf{J}, \mathbf{I}]_{X^d} = \sum_{E \in \Omega_h} [\mathbf{J}, \mathbf{I}]_E \qquad \text{with} \qquad [\mathbf{J}, \mathbf{I}]_E = \mathbf{J}_E^T M_E \mathbf{I}_E, \qquad (2.7)$$

where M_E is a symmetric positive definite matrix. We refer to [11, 12] for the derivation of matrix M_E (see also [7] for the derivation of M_E in the case of a general polygon).

Some minimal approximation properties for this inner product are required to obtain the MFD method with the optimal convergence rate. The first assumption states that there are two positive constants s_1 and s_2 such that

$$s_1 |E| \sum_{\ell \in \partial E} (\mathbf{I}_E^\ell)^2 \le [\mathbf{I}, \mathbf{I}]_E \le s_2 |E| \sum_{\ell \in \partial E} (\mathbf{I}_E^\ell)^2, \qquad (2.8)$$

for all E. In other words, the matrix M_E is spectrally equivalent to the identity matrix. The constants s_1 and s_2 depend only on the anisotropy of the mesh elements and the tensor K.

The second assumption states that for every element E, every linear function ψ^L on E and every $\mathbf{I} \in X^d$, we have the Green's formula

$$[(K \nabla \psi^L)^I, \mathbf{I}]_E = \sum_{\ell \in \partial E} \int_\ell \psi^L \, \mathbf{I}_E^\ell \, dl - \int_E \psi^L (\mathcal{DIV}^d \, \mathbf{I})_E \, dA, \qquad (2.9)$$

where $(\cdot)^I$ denotes the interpolation operator (2.5). The discrete divergence operator $\mathcal{DIV}^d$ is defined below. In other words, (2.9) states that the MFD method is exact for diffusion problems with piecewise constant tensors and piecewise linear solutions.

The third step of the MFD method. The third step of the MFD method consists of the derivation of an approximation to the divergence operator. The discrete divergence operator, $\mathcal{DIV}^d : X^d \to Q^d$, naturally arises from the Gauss divergence theorem applied on an element E as

$$(\mathcal{DIV}^d \, \mathbf{I})_E \overset{def}{=} \frac{1}{|E|} \sum_{\ell \in \partial E} \mathbf{I}_E^\ell |\ell|. \qquad (2.10)$$

The fourth step of the MFD method. In the fourth step of the MFD method, we define the discrete current operator, $\mathcal{G}^d : Q^d \to X^d$, as the adjoint to the discrete divergence operator, $\mathcal{DIV}^d$ with respect to inner products (2.6) and (2.7), i.e.,

$$[\mathbf{I}, \mathcal{G}^d \, \phi]_{X^d} = [\phi, \mathcal{DIV}^d \, \mathbf{I}]_{Q^d}, \qquad \forall \phi \in Q^d, \forall \mathbf{I} \in X^d. \qquad (2.11)$$

The discrete current operator is uniquely defined by the discrete divergence operator and the two inner products. If we change one of the three, we obtain a new formula for the operator $\mathcal{G}^d$. Definition (2.11) ensures symmetry of the forthcoming algebraic problem (2.14).

Using the discrete current and divergence operators, the continuous problem (1.1), (1.2) is discretized as follows:

$$\mathcal{DIV}^d \ \mathbf{J}_d = \mathbf{q}^I \tag{2.12}$$

$$\mathbf{J}_d = \mathcal{G}^d \boldsymbol{\phi}_d \tag{2.13}$$

where $\mathbf{q}^I$ is the interpolant of the source function q given by (2.3).

The convergence of the discrete solution $(\mathbf{J}_d, \boldsymbol{\phi}_d)$ has been studied in [3, 4, 6]. There it was proved that, on unstructured polygonal meshes, the convergence of the current is of first order, while the convergence of the intensity is of second order.

Solution method. Since the matrix associated with inner product (2.7) is irreducible, the discrete operator $\mathcal{DIV}^d \mathcal{G}^d$ is a dense matrix, that is also symmetric positive definite. However, problem (2.12), (2.13) can be reduced to a problem with a sparse symmetric positive definite matrix using standard hybridization arguments (see e.g. [10] in the context of finite element methods). To that end, we consider continuity condition (2.4) as the constraint condition for the solution of (2.12), (2.13). Then the Karush-Kuhn-Tucker (KKT) theory of constrained optimization [20] implies that the resulting problem can be written as a symmetric system of three equations in the primary variables $(\mathbf{J}_d, \boldsymbol{\phi}_d)$ and a vector of the Lagrange multipliers $\boldsymbol{\lambda}_d$. The size of vector $\boldsymbol{\lambda}_d$ is equal to the number of mesh edges. The primary unknowns can be explicitly eliminated from the KKT system resulting in the algebraic problem

$$A \boldsymbol{\lambda}_d = \mathbf{b}_d \tag{2.14}$$

where A is a sparse symmetric positive definite matrix (see [18] for more detailed exposition). Note that vector $\boldsymbol{\lambda}_d$ approximates the intensity ϕ at mid-points of mesh edges with second order accuracy.

3. Error indicator

The theory of *a priori* and *a posteriori* error estimates is well established for mixed finite element methods on simplicial meshes [25]. On triangular meshes, the MFD method can be formulated as a mixed finite element (MFE) method with a special numerical quadrature rule which corresponds to the inner product of two currents [3]. It was proved in [3] that the additional error induced by this numerical integration is sufficiently small so that both methods have the same convergence order. In [4], finite element analysis has been used to prove superconvergence of currents on smooth meshes. These relations can be used to derive optimal *a posteriori* error estimates for the MFD method on simplicial meshes.

In this section, we propose an error indicator for triangular meshes, which can be easily extended to general polygonal meshes. The indicator contains terms which have analogs in residual based error estimates for MFE methods [13]. The theoretical analysis of this error indicator will be the topic of future research.

We define a function ϕ^L that is linear on each element E, but may be discontinuous across mesh edges. The restriction of ϕ^L to element E, further denoted by ϕ_E^L, is the best least-squares fit to the values of vector $\boldsymbol{\lambda}_d$ at the mid-points of mesh edges and the

value of the intensity at the center of element E. Let $x_{E,1}, x_{E,2}, x_{E,3}$ be the edge midpoints of element E and $\boldsymbol{\lambda}_{E,1}, \boldsymbol{\lambda}_{E,2}, \boldsymbol{\lambda}_{E,3}$ be the corresponding Lagrange multipliers. Further, let $\boldsymbol{\phi}_E$ be the intensity at the element center $x_{E,m}$. Then

$$\phi_E^L = \arg\min_{\phi_E^L} \left[\left(\phi_E^L(x_{E,m}) - \boldsymbol{\phi}_E \right)^2 + \sum_{i=1}^{3} \left(\phi_E^L(x_{E,i}) - \boldsymbol{\lambda}_{E,i} \right)^2 \right]. \tag{3.1}$$

Our error indicator is given by

$$\eta^2 = \sum_{\ell \in \Omega_h} \eta_\ell^2 \qquad \text{with} \qquad \eta_\ell^2 = \frac{1}{h_\ell} \int_\ell \left[\phi^L \right]_\ell^2 \mathrm{d}l \tag{3.2}$$

where $[\cdot]_\ell$ denotes the jump of ϕ^L across mesh edge ℓ and h_ℓ denotes the characteristic size of mesh elements sharing edge ℓ (e.g. the edge length). Since we use homogeneous Dirichlet boundary conditions, jump for boundary edges can be defined as $\left[\phi^L \right]_\ell = \phi^L$. We note that using only the edge-centered values $\boldsymbol{\lambda}_{E,i}$ without taking into account the cell-centered intensity $\boldsymbol{\phi}_E$, that is using

$$\phi_E^L = \arg\min_{\phi_E^L} \sum_{i=1}^{3} \left(\phi_E^L(x_{E,i}) - \boldsymbol{\lambda}_{E,i} \right)^2 \tag{3.3}$$

instead of (3.1), would be insufficient for certain mesh configurations, as will be explained in Section 4 and proved in Appendix A. Note that in this latter case all edge-centered values $\boldsymbol{\lambda}_{E,i}$ are coincident with the plane given by ϕ_E^L.

Some of the residual-based error estimates in MFE methods use the jump of the tangential component of a discrete current across mesh edges, which is equivalent to η for a triangular element (see e.g. [13]).

The motivation for using (3.2) as the error indicator comes from the following observation. The error between the integral average of the continuous current $\mathbf{J}^I$ and the discrete solution $\mathbf{J}_d$ satisfies [6]

$$\mathcal{DIV}^d (\mathbf{J}^I - \mathbf{J}_d) = 0.$$

For a triangular element E, this error is uniquely defined by its values on any two edges. Thus, $(\mathbf{J}^I - \mathbf{J}_d)_E$ is the interpolant of a constant vector or

$$(\mathbf{J}^I - \mathbf{J}_d)_E = (\nabla \psi_E^L)^I$$

where ψ_E^L is a linear function. Now, assumption (2.9) results in the estimate

$$\begin{aligned}
|||\mathbf{J}^I - \mathbf{J}_d|||_{X^d}^2 &= \sum_{\ell \in \Omega_h} \int_\ell \left[\psi^L \right]_\ell (\mathbf{J}^I - \mathbf{J}_d)^\ell \mathrm{d}l \leq \\
&\leq C \left(\sum_{\ell \in \Omega_h} \frac{1}{h_\ell} \int_\ell \left[\psi^L \right]_\ell^2 \mathrm{d}l \right)^{1/2} |||\mathbf{J}^I - \mathbf{J}_d|||_{X^d},
\end{aligned} \tag{3.4}$$

where $|||\cdot|||_{X^d}$ denotes the norm induced by inner product (2.7)

$$|||\mathbf{I}|||_{X^d}^2 := [\mathbf{I}, \mathbf{I}]_{X^d}.$$

The constant C in (3.4) depends only on constant s_1 from (2.8) and the shape regularity of mesh elements.

4. Node reconnection strategy

In this section, we describe our c-adaptation strategy for the solution of an elliptic problem. Recently, similar strategy was suggested for improving the quality of mesh elements in the rezone step of Arbitrary Lagrangian-Eulerian simulations [23].

(a)   (b)

FIG. 4.1. *Definition of edge swap. A pair of original triangular elements O_1 and O_2 (a) and the pair of triangular elements S_1 and S_2 (b) that results from swapping of the common edge.*

As was mentioned in the introduction, on triangular meshes, the c-adaptation strategy can simply be described as a sequence of edge swaps, or, in other words, replacements of a selected edge by another one. In particular, each interior edge is shared by two triangular elements, forming a quadrilateral patch as shown in Figure 4.1(a). Swapping this edge with the other diagonal of the quadrilateral replaces the two elements O_1 and O_2 by the other pair S_1 and S_2 (see Figure 4.1(b)).
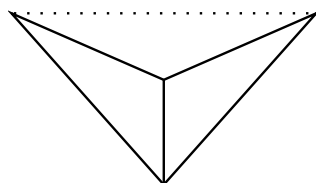

FIG. 4.2. *An example of two edge connected triangular elements that do not allow for edge swapping. The swapped edge (dashed line) would lie outside the union of the two elements.*

In order to compute the error indicator for the pair S_1 and S_2 in Figure 4.1(b), we need a simple estimate for two new element-based intensities ϕ_{S_1} and ϕ_{S_2} and one Lagrange multiplier for the swapped edge (see (3.1)). Recall that the discrete Green's function for the elliptic problem has a fast decrease rate. Therefore, it is possible to estimate the required data by solving a local Dirichlet problem on a mesh patch containing triangles S_1 and S_2. The error due to inaccurate data on boundary edges depends on the size of this patch and its alignment with the principle directions of tensor K. In this paper, we consider the smallest possible patch, namely the patch that consists of only two triangles.

For the reason mentioned above, we may simplify the computation of the error indicator by ignoring contributions from boundary edges of the patch. Thus, we have to compare error indicators associated only with the two diagonals of the quadrilateral patch. The numerical experiments presented later confirm the validity of our assumptions. In principle, we can work with patches consisting of more than two elements to increase robustness and optimality of our error indicator.

We want to emphasize, that it is necessary to use the cell-centered value ϕ_E of the intensity and recompute the edge-based Lagrange multiplier to estimate the error. Indeed, for any rectangular patch, linear reconstructions based on (3.3) result

in equal error estimates for both original and swapped edges and thus no swap would be performed. This is proved in Appendix A.

In Figure 4.3, we introduce our edge swapping algorithm in detail. Note that we assume that the edges can be accessed in some particular order and that some edges might have been flagged for the purpose of excluding them from the edge swapping algorithm. For example, a given mesh might already be constructed in such a way that it is aligned with a material interface. Such edges that lie on this interface should then be excluded from the edge swapping algorithm. It is also possible that a particular edge cannot be swapped. Figure 4.2 illustrates such a case. Here, the geometry of the mesh is such that the swapped edge would fold the mesh.

Algorithm (c-adaptation):
  1. Compute the solution of the discrete problem on the current mesh.
  2. Loop over all interior edges. For each edge do the following:
      (a) If the edge is flagged for exclusion from edge swapping, proceed to the next edge.
      (b) If the geometry does not allow swapping, proceed to the next edge.
      (c) Compute the error indicator $\eta_{\ell,O}$ for the current pair of connected elements (O_1 and O_2 in Figure 4.1(a)).
      (d) Compute the error indicator $\eta_{\ell,S}$ for the pair of elements in the swapped configuration (elements S_1 and S_2 in Figure 4.1(b)). This requires the solution of a local Dirichlet problem.
      (e) If $\eta_{\ell,S} \leq \eta_{\ell,O} - \varepsilon$, add the edge to the list of edges to be processed further (also store $\gamma_\ell = \eta_{\ell,O} - \varepsilon - \eta_{\ell,S}$), and proceed to the next edge.
  3. Loop over the list of edges that have been selected for further processing in Step 2 in the order of their associated error reduction indicator γ_ℓ from the largest to the smallest. For each edge, perform the swap. Then delete this edge from the list, together with all edges, if any, that are on the boundary of the patch.
  4. If no swaps have been performed in Step 3, stop; else proceed to Step 1.

FIG. 4.3. *The c-adaptation algorithm.*

The small positive threshold ε in Step 2(e) of the c-adaptation algorithm is used to avoid infinite alternation between mesh configurations with similar error estimates, and thereby helps to ensure that the algorithm terminates.

Note that in our algorithm, we recompute the solution of the discrete problem after each iteration of swapping (i.e. after each loop over the edges). A cheaper, albeit certainly less accurate, version of the c-adaptation algorithm can be obtained by computing the solution of the discrete problem only in the first iteration. Approximate values for intensities and currents on adjacent-triangle patches that result from edge swapping are computed during the error estimation phase (Steps 2(c) and 2(d) of the algorithm) and can be used to initialize the new mesh. In this paper, we are not concerned with the linear solver that is used in Step one of the c-adaptation algorithm. However, we want to emphasize that in subsequent iterations after the initial iteration, we can obtain a very good starting value for an iterative linear solver, by using the

intensity and current values computed in the error estimation phase. Since the errors in this starting value are of high frequency, a few iterations of a simple stationary method such as Gauss-Seidel relaxation should be enough to obtain the solution of the discrete linear system on the new mesh.

Step three of the c-adaptation algorithm, requires the sorting of the list of edges that have been selected for further processing, either as this list is assembled in Step two, or directly in Step three. A cheaper alternative is to use a binning technique, where edges are categorized into a small number of bins according to the size of their associated error reduction indicator. One first determines the largest error reduction indicator γ_{max} and picks the number of bins to be used, N_{bins}. An edge is assigned to bin k, if its error reduction indicator γ_ℓ satisfies $k/N_{bins} \leq \gamma_\ell/\gamma_{max} < (k-1)/N_{bins}$. Finally, Step three of the c-adaptation algorithm is performed by looping over the bins from largest to smallest and processing the edges in each bin in whichever order they are stored. The removal of edges from the list of edges to be processed is an expensive operation if implemented in a naive fashion. However, if the edge data structure is rich enough, one can delete edges from this list in constant time per edge.

Overall, the c-adaptation algorithm has the potential to be implemented very cheaply. Clearly, the discrete system must be solved in the first iteration. In subsequent iterations, it will be as costly as a fixed number of evaluations of a residual. Each of the iterations of the rest of the algorithm is of complexity $O(N_X)$. It is not clear, however, how fast the c-adaptation algorithm converges. So we cannot estimate the overall complexity of the algorithm.

5. Numerical experiments

In this section, we present numerical results by way of selected examples. The example PDEs are chosen such that there are regions for which the solution has a gradient of a large magnitude. In all following examples, the diffusion tensor is the identity tensor ($K = \text{diag}\{1,1\}$) and the true error is computed in the mesh dependent norm

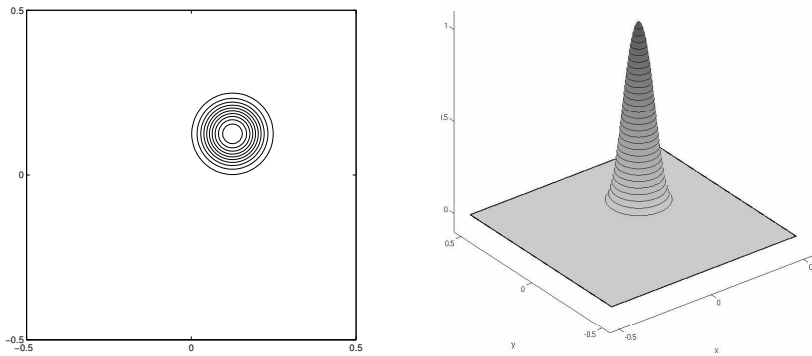$$|||\phi^I - \phi_d|||_Q = [\phi^I - \phi_d, \phi^I - \phi_d]_Q^{1/2}.$$

FIG. 5.1. *Example 1: Analytical solution $\phi(x,y)$, isolines on the left and surface plot on the right.*

In our first example, the exact solution is given by

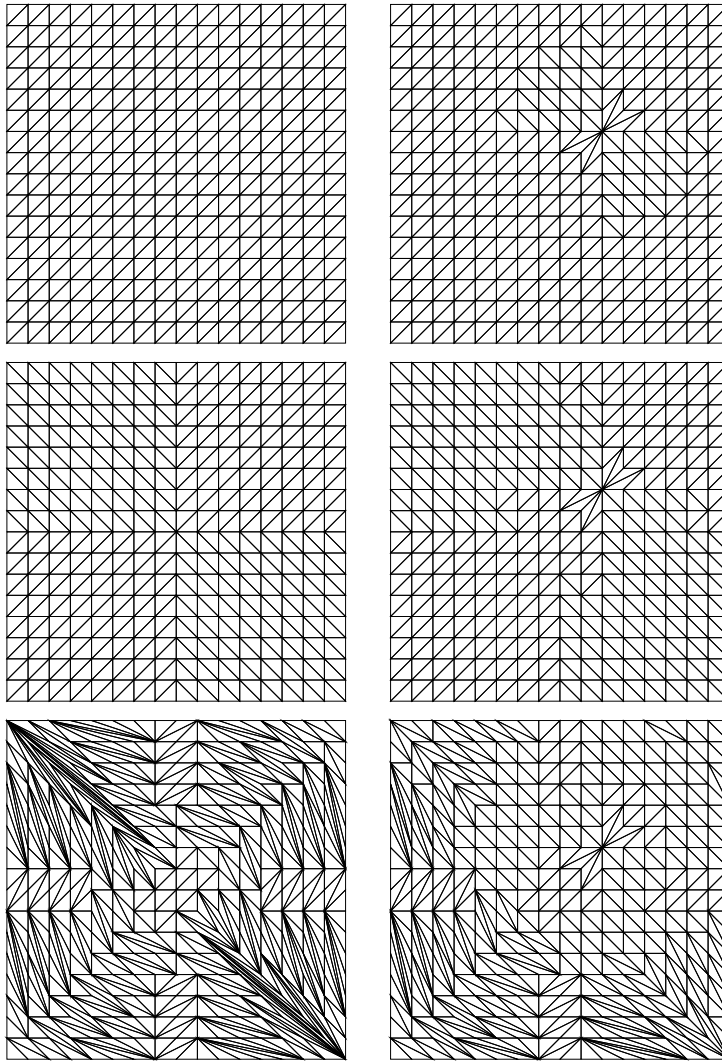$$\phi(x,y) = 1 - \tanh \frac{(x-0.125)^2 + (y-0.125)^2}{e}$$

FIG. 5.2. *Example 1. Original meshes are shown on the left panel. The corresponding adapted meshes are shown on the right panel.*

where $e = 0.01$. The solution is shown is Fig. 5.1, has a shape of a bell centered at point $(0.125, 0.125)$, and is almost zero in the region outside the bell.

We consider three different meshes on the domain $(-0.5, 0.5) \times (-0.5, 0.5)$, each consisting of 512 triangular elements. Results of the c-adaptation algorithm (see Figure 4.3) are shown in Figure 5.2, with original meshes on the left and the corresponding adapted meshes on the right. Note that, except in one circular region, the exact solution is close to constant (see the isolines and the surface plot in the upper part of Figure 5.2). Therefore, swapping edges outside of the circular region with large gradients will not significantly decrease the discretization error. In fact, all meshes that were obtained after edge swapping (on the right in Figure 5.2) are equally suitable for solution of this problem. The discrepancy of the error among

them is less than 0.2%. In the first and second case, only one iteration (i.e. one loop over the edges) was necessary to decrease the error by 15% (from 0.01373 to 0.01166) and 14% (from 0.01353 to 0.01166), respectively. In the last case, the original mesh was of very poor quality, and our c-adaptation procedure took 10 iterations, with an eventual improvement of the solution by 72% (from 0.0411 to 0.01169).

In the second and third examples, the exact solution is given by

$$\phi(x,y) = \exp\left\{-a^2\left((x+by)^2 + c^2(x-y/b)^2\right)\right\}$$

with $a = 1/2$, $b = 1/3$ and $c = 16/3$. A surface plot with isolines of this function is shown on the top-left picture in Fig. 5.3. The solution is anisotropic and varies slowly in the direction given by vector $(1, 1/3)$.

In the second example, the computational domain is the unit square $(-1,1) \times (-1,1)$, while it is the disk of radius 1 in the third example. The initial meshes for these examples are quasi-uniform and consist of 3042 and 2352 triangles, respectively, giving a comparable number of elements per unit area.

In both examples, the adapted meshes have elements aligned perpendicular to solution isolines (see the bottom-left picture in Fig. 5.3 and the right picture in Fig. 5.4). It is pertinent to point out that the major error in the discrete solution is in the vicinity of the line $y = x/3$. Although some elements visually appear to be of poor quality, the error is significantly decreased in both examples. In the second example, the error drop is 49% (from 0.0148 to 0.007516) after the first iteration and 65% (from 0.0148 to 0.005171) after the second one. In the third example, the first iteration swapped just 66 edges to decrease the error by 25% (from 0.008903 to 0.006659). The resulting mesh is shown on the right picture in Fig. 5.4.

The adapted meshes do not comply with the interpolation theory developed for adaptive meshes and P_1 *finite element approximations* (see, e.g. [9, 22, 1]). This theory predicts that the adaptive mesh will contain elements that are stretched in the direction of minimal second derivative of the solution, i.e., in our example, in the direction given by vector $(1, 1/3)$. Indeed, a c-adaptation method that aims to minimize the local interpolation error of the exact solution produces the mesh shown in the bottom-right picture in Fig. 5.3. Using a P_1 finite element method on this grid and comparing the L_2 norm of the discretization error to the L^2 norm of the discretization error obtained from a P_1 finite element method on the regular grid (top-right picture in Fig. 5.3), we observe that it is 3.5 times larger on the regular grid. However, if we use the bottom-right mesh in Fig. 5.3 in the MFD method and compare to an MFD method on the regular grid, the L^2 norm of the discretization error is increased from 0.0148 for the regular grid to 0.06993 on the bottom-right grid in Fig. 5.3. We want to emphasize that the error indicator (3.2) that is used in our node reconnection strategy can be used to bound the discretization error in the current from above (see 3.4). This is in contrast to the case of a P_1 finite element method with focus on only the pressure.

Our experiments indicate that further analysis of optimal meshes for mixed discretizations, in particular for the MFD method, is required. We shall address this problem in future research.

6. Conclusions

We have analyzed numerically a pure c-adaptation strategy for an elliptic problem and have demonstrated that the discretization error can be significantly reduced by edge swapping alone. We have also proposed a new error indicator for this strat-
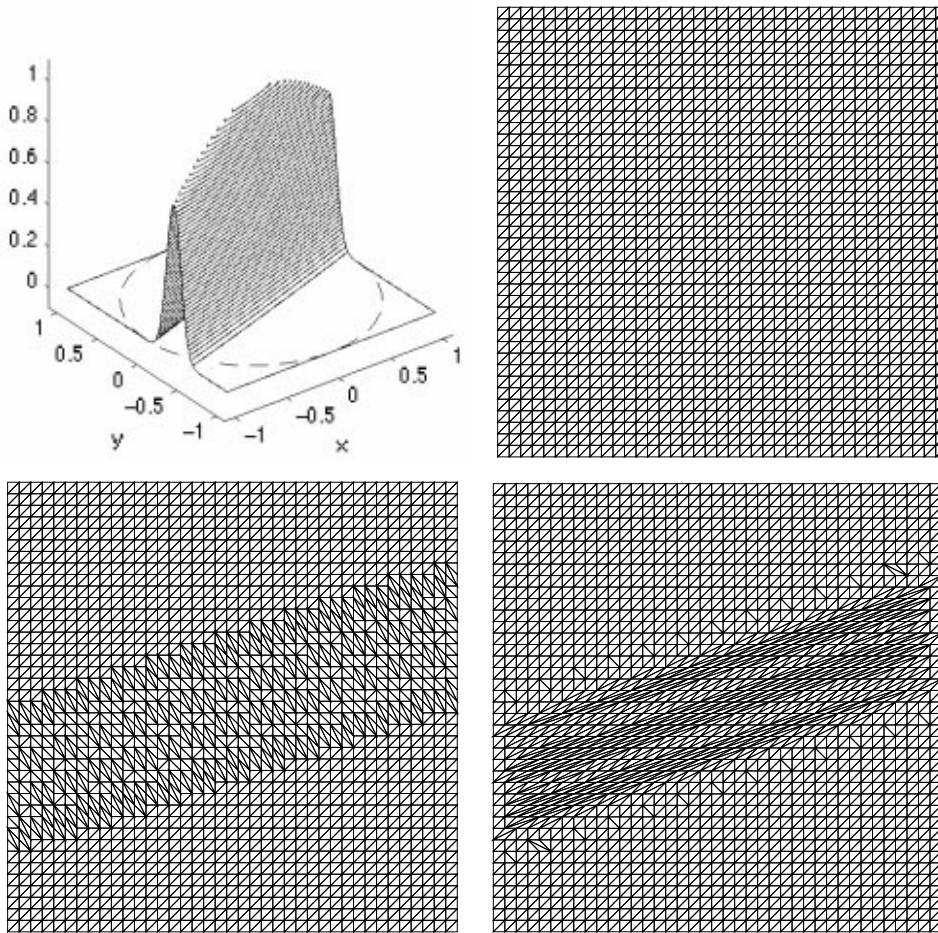
FIG. 5.3. *The top-left picture shows the analytical solution $\phi(x,y)$ used in second and third examples. The mesh in example two fills the whole domain, while the mesh in example three fills only the dashed circle. The top-right picture shows the original mesh in the second example. The bottom-left picture shows the adapted mesh after two iterations. The mesh minimizes the L_2-norm of error for the MFD method. The bottom-right picture shows the mesh minimizing the interpolation error for the P_1 finite element approximation.*

egy. Combination of the c-adaptation strategy with other adaptation strategies is a promising and challenging task for future research.

Appendix A. Necessity of cell-centered intensities.

THEOREM A.1. *For a rectangular patch, the error estimate based only on edge-centered values of pressure (i.e. without considering cell-centered values), will be exactly the same for the swapped edge as for the original one.*

Proof. We consider a rectangle of size $2\Delta x \times 2\Delta y$ centered at point (x_0, y_0) that is given by nodes $(x_0 \pm \Delta x, y_0 \pm \Delta y)$. Without loss of generality, we use the subscripts as in Figure A.1 when referring to particular positions. The rectangle is split into two triangular elements O_1 and O_2 by edge ℓ_O as shown in Figure A.1(a). If we replace
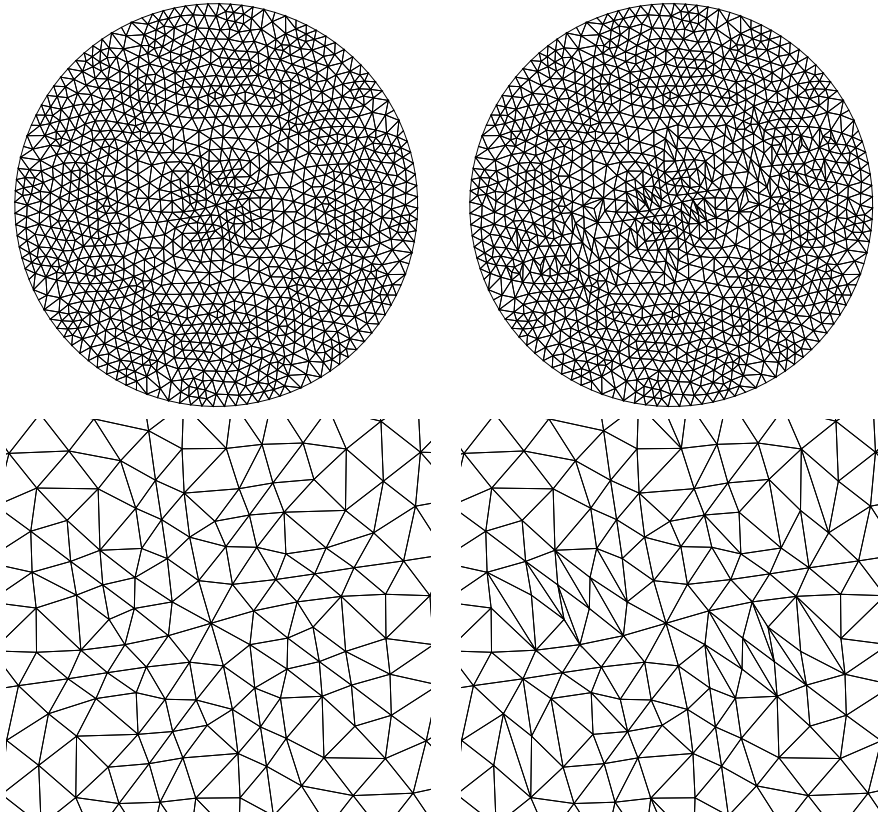
FIG. 5.4. *Example 3. The top-left picture shows the original mesh. the top-right picture shows the adapted mesh after one iteration. The pictures in the bottom line show details of the region centered at point* (0,0).
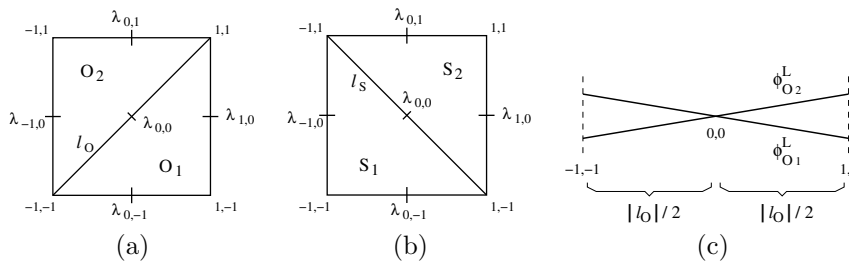

FIG. A.1. *Rectangle split into two triangular elements by the original (a) or swapped (b) edge. View of linear reconstructions along the original edge.*

(swap) the edge by ℓ_S, the new triangles will be denoted S_1 and S_2 as in Figure A.1(b). The intensity is given by discrete values in edge mid-points, that is by $\lambda_{0,\pm1}$, $\lambda_{\pm1,0}$ and $\lambda_{0,0}$ as computed by (2.14). In each of the triangles, we construct a linear function $\phi^L(x,y) = Ax + By + C$, going through the exact edge-centered values. (Since we are working with triangles, such a function always exists.) If we denote the jumps of linear reconstructions for the original and swapped triangles by $\left[\phi^L\right]_{\ell_O} = \phi^L_{O_2} - \phi^L_{O_1}$ and

$\left[\phi^L\right]_{\ell_S} = \phi^L_{S_2} - \phi^L_{S_1}$, then the error indicator (3.2) using the edge-based reconstruction (3.3) for the original patch is

$$\eta_O^2 = \frac{1}{\sqrt{V}} \int_{l_O} \left[\phi^L(x,y)\right]^2_{\ell_O} \mathrm{d}l, \tag{A1}$$

and similarly for the swapped one. The patch volume V is clearly equal for both configurations: $V = |O_1| + |O_2| = |S_1| + |S_2|$. The theorem states that

$$\eta_O^2 = \eta_S^2. \tag{A2}$$

To show this, we first simplify (A1). Since the values λ at edge mid-points are known, values $\left[\phi^L\right]_{\ell_O}$ in nodes can be computed from linear reconstructions and, hence, the integrand is quadratic. We use Simpson's rule and get

$$\eta_O^2 = \frac{1}{\sqrt{V}} \int_{\ell_O} \left[\phi^L(x,y)\right]^2_{\ell_O} \mathrm{d}l = \frac{|\ell_O|}{6\sqrt{V}} \left(\left[\phi^L\right]^2_{\ell_O;-1,-1} + 4\left[\phi^L\right]^2_{\ell_O;0,0} + \left[\phi^L\right]^2_{\ell_O;1,1} \right), \tag{A3}$$

where $|\ell_O|$ is the length of the original edge. This can be further simplified, since the linear functions $\phi^L_{O_1}$ and $\phi^L_{O_2}$ are constructed so, that they intersect at the mid-point of the common edge of their associated triangles. From here it follows (compare Figure A.1(c)) that $\left[\phi^L\right]_{\ell_O;0,0} = 0$ and $\left[\phi^L\right]_{\ell_O;1,1} = -\left[\phi^L\right]_{\ell_O;-1,-1}$, which reduces the error estimate (A3) to

$$\eta_O^2 = k\left[\phi^L\right]^2_{\ell_O;1,1} = k\left[\phi^L\right]^2_{\ell_O;-1,-1},$$

with constant $k = |\ell_O|/(3\sqrt{V})$. This holds for any quadrilateral patch. Moreover, for the rectangle, both the original and the swapped edge have the same length $|\ell_O| = |\ell_S|$, so that

$$\eta_S^2 = k\left[\phi^L\right]^2_{\ell_S;-1,1} = k\left[\phi^L\right]^2_{S;1,-1},$$

with the same constant k. Thus, to prove Theorem A2, we only need to show that

$$\left| \left[\phi^L\right]_{O;1,1} \right| = \left| \left[\phi^L\right]_{S;-1,1} \right|. \tag{A4}$$

To construct the linear function for the triangle with edge mid-points (x_α, y_α), (x_β, y_β) and (x_γ, y_γ), we compute A, B and C such that

$$\begin{pmatrix} x_\alpha & y_\alpha & 1 \\ x_\beta & y_\beta & 1 \\ x_\gamma & y_\gamma & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} \lambda_\alpha \\ \lambda_\beta \\ \lambda_\gamma \end{pmatrix}.$$

If we denote the matrix by $\mathbf{R}$ and the right-hand side by $\mathbf{\Lambda}$, the reconstruction is

$$\phi^L(x,y) = \begin{pmatrix} A \\ B \\ C \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{R}^{-1}\mathbf{\Lambda} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Particularly for the upper right hand and upper left hand corners of the rectangle, we have

$$\left[\phi^L\right]_{O;1,1} = \left(\left(\mathbf{R^{-1}\Lambda}\right)_{O_2} - \left(\mathbf{R^{-1}\Lambda}\right)_{O_1}\right) \begin{pmatrix} x_0 + \Delta x \\ y_0 + \Delta y \\ 1 \end{pmatrix}$$

$$= \lambda_{0,1} + \lambda_{0,-1} - \lambda_{1,0} - \lambda_{-1,0},$$

$$\left[\phi^L\right]_{S;-1,1} = \left(\left(\mathbf{R^{-1}\Lambda}\right)_{S_2} - \left(\mathbf{R^{-1}\Lambda}\right)_{S_1}\right) \begin{pmatrix} x_0 - \Delta x \\ y_0 + \Delta y \\ 1 \end{pmatrix}$$

$$= \lambda_{0,1} + \lambda_{0,-1} - \lambda_{1,0} - \lambda_{-1,0},$$

so that (A4) holds and thus the theorem is proved. □

REFERENCES

[1] A. Agouzal, K. Lipnikov and Yu. Vassilevski, *Adaptive generation of quasi-optimal tetrahedral meshes*, East-West J. Numer. Math., 7(4), 223-244, 1999.
[2] I. Babuška and W. C. Rheinboldt, *A posteriori error estimates for the finite element method*, Internat. J. Numer. Methods Engrg., 12, 1597-1615, 1978.
[3] M. Berndt, K. Lipnikov, J. D. Moulton and M. Shashkov, *Convergence of mimetic finite difference discretizations of the diffusion equation*, East-West J. Numer. Math., 9, 253-284, 2001.
[4] M. Berndt, K. Lipnikov, M. Shashkov, M. Wheeler and I. Yotov, *Superconvergence of the velocity in mimetic finite difference methods on quadrilaterals*, submitted, 2004.
[5] J. U. Brackbill and J. S. Saltzman, *Adaptive zoning for singular problems in two dimensions*, J. Comput. Phys., 46, 342-368, 1982.
[6] F. Brezzi, K. Lipnikov and M. Shashkov, *Convergence of mimetic finite difference method for diffusion problems on polyhedral meshes*, SIAM J. Numer. Anal., to appear, 2005.
[7] F. Brezzi, K. Lipnikov and V. Simoncini, *A family of mimetic finite difference methods on polygonal and polyhedral meshes*, Math. Mod. Meth. Appl. Sci., 15(10), 2005.
[8] G. C. Buscaglia and E. A. Dari, *Anisotropic mesh optimization and its application in adaptivity*, Int. J. Numer. Meth. Engng., 40(22), 4119-4136, 1997.
[9] E. D'Azevedo and R. Simpson, *On optimal triangular meshes for minimizing the gradient error*, Numerische Mathematik, 59, 321-348, 1991.
[10] B. Fraeijs de Veubeke, *Displacement and equilibrium models in the finite element method*, In O. C. Zienkiewicz and G. Holister, editors, Stress Analysis, John Wiley and Sons, New York, 1965.
[11] V. Ganzha, R. Liska, M. Shashkov and C. Zenger, *Support operator method for Laplace equation on unstructured triangular grid*, Selcuk Journal of Applied Mathematics, 3(1), 21-48, 2002.
[12] V. Ganzha, R. Liska, M. Shashkov and C. Zenger, *Mimetic finite difference methods for diffusion equations on unstructured triangular grid*, in M. Feistauer, V. Dolejsi, P. Knobloch, and K. Najzar, editors, Numerical Mathematics and Advanced Applications, Springer-Verlag Berlin Heidelberg New York, ISBN: 3-540-21460-7, 368-377, 2004.
[13] R. Hoppe and B. Woklmuth, *A comparison of a posteriori error estimators for mixed finite element discretizations by raviart-thomas elements*, Mathematics of Computation, 68(228), 1347-1378, 1999.

[14] J. Hyman, J. Morel, M. Shashkov and S. Steinberg, *Mimetic finite difference methods for diffusion equations*, Comp. Geosciences, 6(3-4), 333-352, 2002.

[15] 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.

[16] Y. Kuznetsov, K. Lipnikov and M. Shashkov, *Mimetic finite difference method on polygonal meshes for diffusion-type problems*, Comp. Geosciences, in press, 2004.

[17] J. Lang, W. Cao, W. Huang and R.D. Russell, *A two-dimensional moving finite element method with local refinement based on a posteriori error estimates*, Applied Numer. Math., 46, 75-94, 2003.

[18] K. Lipnikov, M. Shashkov and D. Svyatskiy, *The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes*, J. Comput. Phys., 211, 473-491, 2005.

[19] R. Liska, M. Shashkov and V. Ganza, *Analysis and optimization of inner products for mimetic finite difference methods on triangular grid*, Mathematics and Computers in Simulation, 67, 55-66, 2004.

[20] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, Berlin, Heidelberg, New York, 1999.

[21] W. C. Rheinboldt and C. K. Mesztenyi, *On a data structure for adaptive finite element mesh refinements*, ACM Trans. Math. Softw., 6(2), 166-187, 1980.

[22] S. Rippa, *Long and thin triangles can be good for linear interpolation*, SIAM J. Numer. Anal., 29(1), 257-270, 1992.

[23] P. Vachal and R.V. Garimella, *On quality improvement of triangular meshes using node reconnection*, in A. Handlovicova, Z. Kriva, K. Mikula and D. Sevcovic, editors, Proceedings of Algoritmy 2005, 17th Conference on Scientific Computing, Bratislava, Slovakia, Slovak University of Technology, 93-102, 2005.

[24] P. Vachal, R.V. Garimella and M.J. Shashkov, *Untangling of 2D meshes in ALE simulations*, J. Comput. Phys., 2(196), 627-644, 2004.

[25] R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh Refinement Techniques*, Teubner Verlag and J. Wiley, Stuttgart, 1996.