Mimetic Finite Difference Methods for Elliptic Equations on Unstructured Triangular Grid

Victor Ganzha[1], Richard Liska[2], Mikhail Shashkov[3], Christoph Zenger[1]
[1]*Department of Informatics*
Technical University of Munich
Arcisstrasse 21, 80333 Munich, Germany
ganzha@in.tum.de, zenger@in.tum.de
[2]*Faculty of Nuclear Sciences and Physical Engineering*
Czech Technical University in Prague
Břehová 7, 115 19 Prague 1, Czech Republic
liska@siduri.fjfi.cvut.cz
http://www-troja.fjfi.cvut.cz/~liska
[3]*Group T-7, Los Alamos National Laboratory*
Los Alamos, NM 87544, USA
misha@t7.lanl.gov

December 13, 2001

Abstract

A finite difference algorithm for solution of generalized Laplace equation on unstructured triangular grid is constructed by a support operator method. The support operator method first constructs discrete divergence operator from the divergence theorem and then constructs discrete gradient operator as the adjoint operator of the divergence. The adjointness of the operators is based on the contiuum Green formulas which remain valid also for discrete operators. Developed method is exact for linear solution and has second order convergence rate. It is working well for discontinuous diffusion coefficient and very rough or very distorted grids which appear quite often e.g. in Lagrangian simulations. Being formulated on the unstructured grid the method can be used on the region of arbitrary geometry shape. Numerical results confirm these properties of the developed method.

1 Introduction

Development of high-quality finite-difference methods (FDMs) for generalized Laplace equation is a part of a bigger effort to create a discrete analog of vector and tensor calculus [1], [2], [3], [4] that can be used to accurately approximate continuum models for a wide range of physical processes. These FDMs preserve fundamental properties of the original continuum differential operators and allow the discrete approximations of partial differential equations (PDEs) to mimic critical properties including conservation laws and symmetries in the solution of the underlying physical problem. The discrete analogs of differential operators satisfy the identities and theorems of vector and tensor calculus and provide new reliable algorithms for a wide class of PDEs. This approach has been used to construct high-quality mimetic FDMs approximating the diffusion equation [5], [6], [7], [8], [3], [9] gas dynamics equations [10], equations of continuum mechanics [4], Maxwell's first-order curl equations [11], and the equations of magnetic diffusion.

The goal of this paper is to apply our ideas to construction of mimetic FDMs for solution of generalized Laplace problems in strongly heterogeneous materials on unstructured triangular computational grids in 2D. The paper is arranged as follows. The next section presents basic properties of continuum differential operators which we want to preserve also in the discrete case. The section 3 introduces all the data structures describing unstructured triangular grid which are later used during the discretization. The section 4 dealing with discretization starts by describing the type of discretization of scalar and vector functions on unstructured triangular grid, continues by introducing natural and formal scalar products of grid functions and ends with derivation of discrete approximation of continuum differential operators. The last section 5 then presents numerical examples justifying properties of the developed numerical method.

2 Continuum problem

We are treating generalized Laplace equation with Dirichlet or Neumann boundary conditions

$$-\mathrm{div}\, K\, \mathbf{grad}\, u = f \qquad \text{on} \qquad V \tag{1}$$

$$u = \psi \qquad \text{or} \tag{2}$$

$$(K\, \mathbf{grad}\, u, \mathbf{n}) = \psi \qquad \text{on} \qquad \partial V \tag{3}$$

on the arbitrary 2D region V with the border ∂V. K is the matrix of diffusion coefficients, $\mathbf{n}$ is the outer normal to the boundary, f and ψ are given functions. The problem is solved for unknown function u. In general we have to assume that the diffusion matrix K is invertible with the inverse K^{-1}, while in particular here in the discrete case we consider only diagonal matrix of diffusion coefficients $K = k\mathbf{I}$.

Because our discretizations will be based on using discrete analogs of first order coordinate invariant operators it is natural to write Laplace equation (1) as a system of first order equations and then the problem (1)-(3) can be rewritten as

$$\begin{array}{rl} \mathrm{div}\, \mathbf{w} = f & \text{on} \quad V \\ \mathbf{w} = -K\, \mathbf{grad}\, u & \text{on} \quad V \\ u = \psi \quad \text{or} \quad -(\mathbf{w}, \mathbf{n}) = \psi & \text{on} \quad \partial V, \end{array} \tag{4}$$

where first equation is mass balance equation (conservation law), $\mathbf{w}$ is flux, which has clear physical meaning, and second equation is definition of the flux (Darcy law). This formulation also suggest that we have to use discrete analogs of both u and $\mathbf{w}$ as a primary variables in our FDMs. For a moment we will not consider boundary conditions. We introduce the operator $\mathbf{G}$ as a generalized gradient

$$\mathbf{G}u = -K\, \mathbf{grad}\, u \tag{5}$$

and the operator $\mathbf{D}$ as an operator of the extended divergence

$$\mathbf{D}\, \mathbf{w} = \left\{ \begin{array}{lll} \mathrm{div}\, \mathbf{w} & \text{on} & V \\ -(\mathbf{w}, \mathbf{n}) & \text{on} & \partial V \end{array} \right. . \tag{6}$$

The operator $\mathbf{G}$ operates from the space H of the smooth scalar functions on the region V into the space $\mathbf{H}$ of smooth vector functions on the region V and the operator $\mathbf{D}$ operates the other way from the space $\mathbf{H}$ into the space H. To show an important property connecting these operators we consider these spaces as Hilbert spaces and define on them inner products. On the space H we define the inner product

$$(u,v)_H = \int_V u\ v\ d\ V + \oint_{\partial V} u\ v\ d\ S \tag{7}$$

and on the space $\mathbf{H}$ we define the inner product

$$(\mathbf{A},\mathbf{B})_{\mathbf{H}} = \int_V (K^{-1}\mathbf{A},\mathbf{B}) d\ V. \tag{8}$$

Now our operators are acting between the spaces H and $\mathbf{H}$ as:

$$\mathbf{G} : H \to \mathbf{H}, \mathbf{D} : \mathbf{H} \to H,$$

Our operator $\mathbf{D}$ is the operator of extended divergence. Its basic property is given by the divergence Green formula

$$\int_V \operatorname{div} \mathbf{w}\ d\ V - \oint_{\partial V} (\mathbf{w},\mathbf{n})\ d\ S = 0 \tag{9}$$

which is valid for any region V and later we will use it to define finite difference approximation to the operator $\mathbf{D}$. This property can be expressed in terms of our scalar product as

$$(\mathbf{D}\ \mathbf{w}, 1)_H = 0. \tag{10}$$

The Gauss theorem for any functions $u \in H, \mathbf{w} \in \mathbf{H}$ can be written as

$$\int_V u \operatorname{div} \mathbf{w}\ d\ V - \oint u(\mathbf{w},\mathbf{n})\ d\ S + \int_V (\mathbf{w}, K^{-1} K \mathbf{grad}\, u) d\ V = 0, \tag{11}$$

where we can identify our operators $\mathbf{D}$ and $\mathbf{G}$, and gives us the relation between the operators of extended divergence and gradient. Using the operators $\mathbf{D}$ (6) and $\mathbf{G}$ (5) and inner products (7), (8) on the spaces $H, \mathbf{H}$ the Gauss theorem (11) can be rewritten as

$$(\mathbf{D}\mathbf{w}, u)_H = (\mathbf{w}, \mathbf{G}u)_{\mathbf{H}} \tag{12}$$

so that

$$\mathbf{G} = \mathbf{D}^* \tag{13}$$

where * denotes the adjoint operator. This means that gradient is minus (note (5)) adjoint of extended divergence which is a very important property of these operators which we want to preserve also in the discrete case below.

Now we return to boundary conditions and consider the case of Neumann boundary condition (3). We define the extended right hand side of the Laplace equation (1) with Neumann boundary condition (3) as

$$\mathbf{F} = \begin{cases} f & \text{on} \quad V \\ \psi & \text{on} \quad \partial V \end{cases}. \tag{14}$$

Now the flux form of our problem (4) with Neumann boundary condition can be written as

$$\begin{aligned} \mathbf{D}\mathbf{w} &= \mathbf{F} \\ \mathbf{w} &= \mathbf{G}u \end{aligned}$$

on the whole region V. Eliminating the flux $\mathbf{w}$ from these equation we obtain

$$\mathbf{P}u = \mathbf{DG}u = \mathbf{F} \tag{15}$$

where we have defined the global operator $\mathbf{P} = \mathbf{DG}$. From (13) we get

$$\mathbf{P} = \mathbf{DD}^*. \tag{16}$$

Now it is easy to show important properties of the global operator $\mathbf{P}$. The first one is that the operator $\mathbf{P}$ is positive definite:

$$(\mathbf{P}u, u)_H = (\mathbf{DD}^*u, u)_H = (\mathbf{D}^*u, \mathbf{D}^*u)_{\mathbf{H}} > 0 \tag{17}$$

The second one is that the operator $\mathbf{P}$ is self-adjoint:

$$(\mathbf{P}u, u)_H = (\mathbf{DD}^*u, u)_H = (\mathbf{D}^*u, \mathbf{D}^*u)_{\mathbf{H}} = (u, \mathbf{D}^{**}\mathbf{D}^*)_H = (u, \mathbf{DD}^*u)_H \tag{18}$$

as $\mathbf{D} = \mathbf{D}^{**}$:

$$(\mathbf{Dw}, u)_H = (\mathbf{w}, \mathbf{D}^*u)_{\mathbf{H}} = (\mathbf{D}^*u, \mathbf{w})_{\mathbf{H}} = (u, \mathbf{D}^{**}\mathbf{w})_H. \tag{19}$$

So we have shown that the global operator $\mathbf{P}$ of our problem with Neumann boundary conditions is self-adjoint and positive definite

$$\mathbf{P} = \mathbf{P}^* > 0. \tag{20}$$

By a similar procedure presented in [12] we can also show that the global operator in the case of Dirichlet boundary conditions is also self-adjoint and positive definite.

These properties of the global operator extend also to the discrete case and are crucial for the choice of the numerical method for the solution of the system of linear equations obtained from the finite difference method. We have used conjugate gradient method for which the matrix of the linear system has to be symmetric and positive definite.

3 Unstructured triangular grid

To describe our method we need substantial data structures describing the unstructured triangular grid. To simplify the notation of different data structures which are related to different objects (triangles, vertices, edges) we introduce some uniform index notation to be used in this paper, namely:

- i denotes the index of a triangle
- j denotes the index of a vertex
- k denotes the index of an edge
- m_i denotes the median of the triangle i
- l_k denotes the center of the edge k

Note that we use bold face to denote vector quantities so that e.g. $\mathbf{j}$ denotes two coordinates of vertex j.

All three basic objects of the unstructured triangular grid, i.e. triangles, vertices and edges, are ordered in an abritrary chosen manner (except edges - first are boundary edges):

- triangles are numbered by $1, \dots, N_t$
- vertices are numbered by $1, \dots, N_v$
- boundary edges are numbered by $1, \dots, N_{eb}$
- interior edges are numbered by $N_{eb} + 1, \dots, N_e$

where

- N_t is the total number of triangles
- N_v is the total number of vertices
- N_e is the total number of edges
- N_{eb} is the total number of boundary edges

To distinguish the boundary we use the index zero as a special case, i.e. triangle 0, vertex 0 and edge 0 all do not belong to our region V. In fact as we are not computing out of the region V we do not need to distinguish which particular object it is, we just use the index 0 as flag telling us that we are on the boundary.

For each object of the grid we use several data structures:

for each vertex j

- $\mathbf{j} = (x_j, y_j)$ coordinates of the vertex j
- list L_j of triangles to which the vertex j belongs; for boundary vertices $0 \in L_j$

for each triangle i (see Fig. 1)

- indices of three vertices j_i^1, j_i^2, j_i^3 of the triangle i ordered in the counter-clockwise direction
- indices of three edges k_i^1, k_i^2, k_i^3 making the triangle i; the edge k_i^J has end vertices j_i^J, j_i^{J+1} (with cyclic extension, i.e. $j_i^4 = j_i^1$)
- median $\mathbf{m}_i = (\mathbf{j}_i^1 + \mathbf{j}_i^2 + \mathbf{j}_i^3)/3$ of the triangle i
- the area VC_i of the triangle i
- the weights $V_{k_i^J}^{k_i^{J+1}}$ associated with two edges of the triangle (or one vertex) which sum up to the area VC_i of the triangle i

  $$VC_i = \sum_{J=1}^{3} V_{k_i^J}^{k_i^{J+1}} \tag{21}$$

  for triangle out of boundaries we define $V_k^0 = V_0^k = 0$ for any k; we use the weights $V_{k_i^J}^{k_i^{J+1}} = VC_i/3$

for each edge k (see Fig. 2)

- two indices of edge vertices j_k^1, j_k^2 so that $j_k^1 < j_k^2$

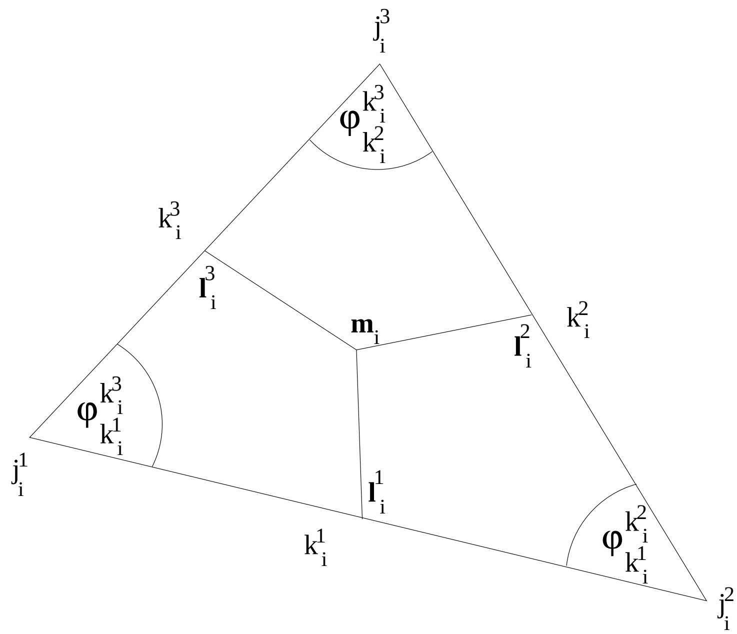


Figure 1: Triangle i and quantities related to it.

- vector defined by the edge $\mathbf{k} = \mathbf{j}_k^2 - \mathbf{j}_k^1$
- length of the edge $S_k = |\mathbf{k}|$
- middle point of the edge $\mathbf{l}_k = (\mathbf{j}_k^1 + \mathbf{j}_k^2)/2$
- two indices of triangles i_k^1, i_k^2 neighboring the edge k so that when we look at the edge in vertical position with vertex j_k^1 down and j_k^2 up as in Fig. 2 then the triangle i_k^1 is on left and the triangle i_k^2 is on right from the edge; for boundary edges either i_k^1 or i_k^2 is zero; note that $\{i_k^1, i_k^2\} = L_{j_k^1} \cap L_{j_k^2}$ with exception of inner edge conecting two boundary vertices when the intersection contains also zero
- two indices of vertices j_k^3, j_k^4 which are the third vertices of two triangles joining at the edge k, so that j_k^3 is the third vertex of the triangle i_k^1 (i.e. on left) and j_k^4 is the third vertex of the triangle i_k^2 (i.e. on right when edge vector $\mathbf{k}$ is pointing up); again for boundary edge one of these indices is zero
- four indices of edges $k_k^{IJ}, I = 1,2, J = 1,2$ which are remaining edges of the triangles $i_k^I, I = 1,2$ where the index I denotes the triangle and the index $J = 1$ denotes lower edge and $J = 2$ upper edge of given triangle when the edge vector $\mathbf{k}$ is pointing up as in Fig. 2; again for boundary edge two of these indices are zero
- four angles $\phi_k^{k_k^{IJ}}, I = 1,2, J = 1,2$ which are the angles between oriented edges $\mathbf{k}$ and $-\mathbf{k}_k^{IJ}$ defined by

$$(\mathbf{k}, \mathbf{k}_k^{IJ}) = -|\mathbf{k}||\mathbf{k}_k^{IJ}| \cos \phi_k^{k_k^{IJ}} \tag{22}$$

note that depending on the orientation of the edges k_k^{IJ} either $\phi_k^{k_k^{IJ}} = \varphi_k^{k_k^{IJ}}$, i.e. they are the angles of the neighboring triangles with φ angles shown in Fig. 2, or $\phi_k^{k_k^{IJ}} = \pi - \varphi_k^{k_k^{IJ}}$

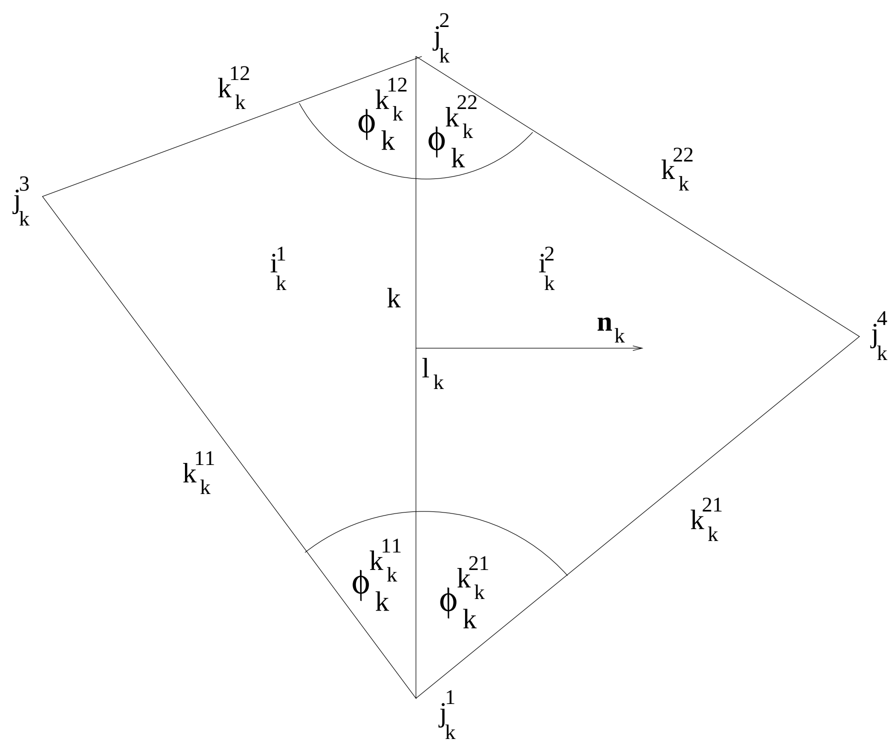


Figure 2: Edge k and quantities related to it.

4 Discretization

Our goal is to construct the discretization scheme for the generalized Laplace equation problem in such a way that the discrete operators approximating divergence and gradient will also posses the properties of the continuum operators **D** and **G** which we have derived in the section 2. Namely we want to mimic in the discrete case the Green formula (10) and Gauss theorem (11) stated in the form of inner products (12). So we need to define the discrete approximation of inner products of scalar functions (7) and of vector functions (8) which we call natural inner products. For deriving the adjoint operator (13) we also need to define so called formal inner products which are just a plain sum of products of values over all the discretization points. For applications usually local conservation is important and natural choice for discretization of scalar function u is so-called *cell-centered* discretization (we call it HC, C stands for cells), where we have one value of discrete function u in the cell. One can think about this value as integral average of the function over the cell, for this reason it is assigned to entire cell and not to particular point in this cell. It suggests that in discrete case forcing function f also has to be defined in cells, as well as range of values of discrete analog of divergence is space HC.

We mention that only normal component of flux is continous on interface between different materials. This suggests to use these normal components to describe flux vector in discrete case and also to define them on the faces of the cells. That is, on the each face of the cell we will have only one unknown which meaning will be dot product of flux with normal to this face. This is not only consistent from physical point of view, but also effectively enforces continuity of normal component of fluxes because it is the same on both sides of interface. We will call such space of discrete vector functions as **HL**. For this discretization of vector functions construction of discrete divergence is trivial. In fact, if we will choose volume of the cell as V in formula (9) then in right-hand side we will have summation of products of areas of faces and our normal components of vectors.

Such definition of discrete divergence (which we will denote by D) also perfectly fits more general consideration of our discrete vector analysis [1]. By construction $D : \mathbf{HL} \to HC$. Because discrete flux operator (we will call it G) will be adjoint to divergence we have $G : HC \to \mathbf{HL}$. Now when discrete analogs of divergence and flux operator are constructed they are used to construct our finite difference method for Laplace equation in form (4) by substituting differential operators by discrete ones.

4.1 Scalar and vector functions on triangular grid

Of course we have to start with discretization of functions on the unstructured triangular grid. Scalar function u is on the triangular grid represented by its value U_i inside the triangle i for all triangles of the grid. Further it is represented on the boundaries, for each boundary edge $k, k = 1, \cdots, N_{eb}$ it is represened by the value U_k in the middle of the edge.

Vector function $\mathbf{w}$ is represented at the middle point of the edge k by the projection W_k of the vector $\mathbf{w}$ on the normal to the edge (see Fig. 3). The direction of the normal to the edge is given by the left hand rule, i.e. when edge $\mathbf{k}$ points up, vertex j_k^1 is down and j_k^2 is up ($j_k^1 < j_k^2$), then the normal is pointing right (see Fig. 2).

The diffusion coefficient matrix K is assumed to be a multiple of unit matrix $K = k\mathbf{I}$ and the diffusion coefficient k_i is constant inside each triangle i. The function f on the right hand side of the Laplace equation (1) is discretized inside each triangle by f_i. The function ψ on the right hand side of the boundary conditions (2), (3) is discretized at the center of boundary edges by ψ_k. The outer normal vector $\mathbf{n}$ is needed also in the centers of boundary edges and is defined by the scalar $n_k = \pm 1$ according to the direction of the normal to the edge defined above. When the edge points up, i.e. vertex j_k^1 is down and j_k^2 is up ($j_k^1 < j_k^2$ as on Fig. 2) then if the inner triangle is on the left (i.e. $i_k^2 = 0$) then $n_k = 1$ and if the inner triangle is on the right (i.e. $i_k^1 = 0$) then $n_k = -1$. In both cases the other triangle does not exist (i.e. $i_k^J = 0$) and that area is out of our computational region.

4.2 Natural inner products

Natural discrete inner products are approximation to the continuum inner products (7), (8). On the space HC of scalar grid functions U discretized inside the triangles by U_i and in the center of boundary edges by U_k we approximate the continuum inner product (7) by

$$(U, V)_{HC} = \sum_{i=1}^{N_t} U_i \; V_i \; VC_i + \sum_{k=1}^{N_{eb}} U_k \; V_k \; S_k. \tag{23}$$

The situation for the space **HL** of vector grid functions $\mathbf{A}$ discretized in the centers of the triangles edges by the scalar component A_k orthogonal to the edge is more complicated. We

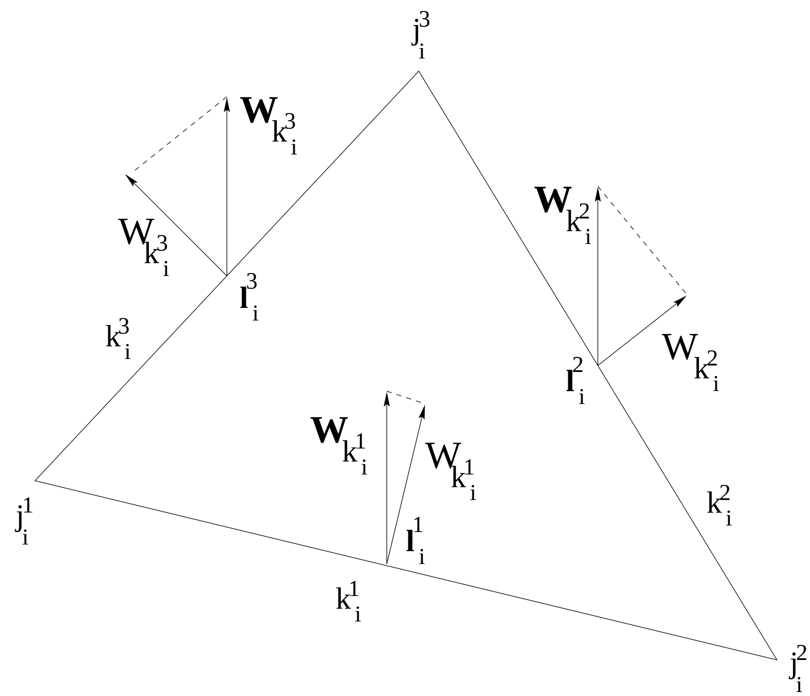


Figure 3: Vector function $\mathbf{w}$ is represented by the projection of the vector on the edge normals at centers of the edges.

cannot approximate the inner product at the center of the edge because we have there only one component of the vector. We first try to approximate the inner product in one vertex j_i^J of a particular triangle i. Note that also other triangles having this vertex will contribute to the inner product. Having one vector grid function $\mathbf{W}$ we first move its discrete components $W_{k_i^J}$ and $W_{k_i^{J-1}}$ from the centres of the edges k_i^J and k_i^{J-1} into the vertex j_i^J as is done at Fig. 4. We do this movement for both vector grid functions $\mathbf{A}$ and $\mathbf{B}$ for which we are computing the inner product. Now we have both vectors at the same point, vertex j_i^J and we can express their scalar product at this point. One can derive that this scalar product in terms of components orthogonal to the edges is

$$(\mathbf{A},\mathbf{B})_{j_i^J} = \frac{A_{k_i^J}B_{k_i^J} + A_{k_i^{J-1}}B_{k_i^{J-1}} + (A_{k_i^J}B_{k_i^{J-1}} + A_{k_i^{J-1}}B_{k_i^J})\cos\phi_{k_i^J}^{k_i^{J-1}}}{\sin^2\phi_{k_i^J}^{k_i^{J-1}}} \tag{24}$$

which is the contribution from the vertex j_i^J to the inner product at the triangle i. The approximation of the inner product (8) at the triangle i is then defined as

$$(\mathbf{A},\mathbf{B})_i = \frac{1}{VC_i}\sum_{J=1}^{3}(\mathbf{A},\mathbf{B})_{j_i^J}V_{k_i^J}^{k_i^{J-1}} \tag{25}$$

where $V_{k_i^J}^{k_i^{J-1}}$ are the weights (21) in the triangle i associated with the vertex j_i^J. Now we define the natural inner product on the space $\mathbf{HL}$ of vector grid functions as

$$(\mathbf{A},\mathbf{B})_{\mathbf{HL}} = \sum_{i=1}^{N_t}(A,B)_i\frac{VC_i}{\mathbf{K}_i}. \tag{26}$$

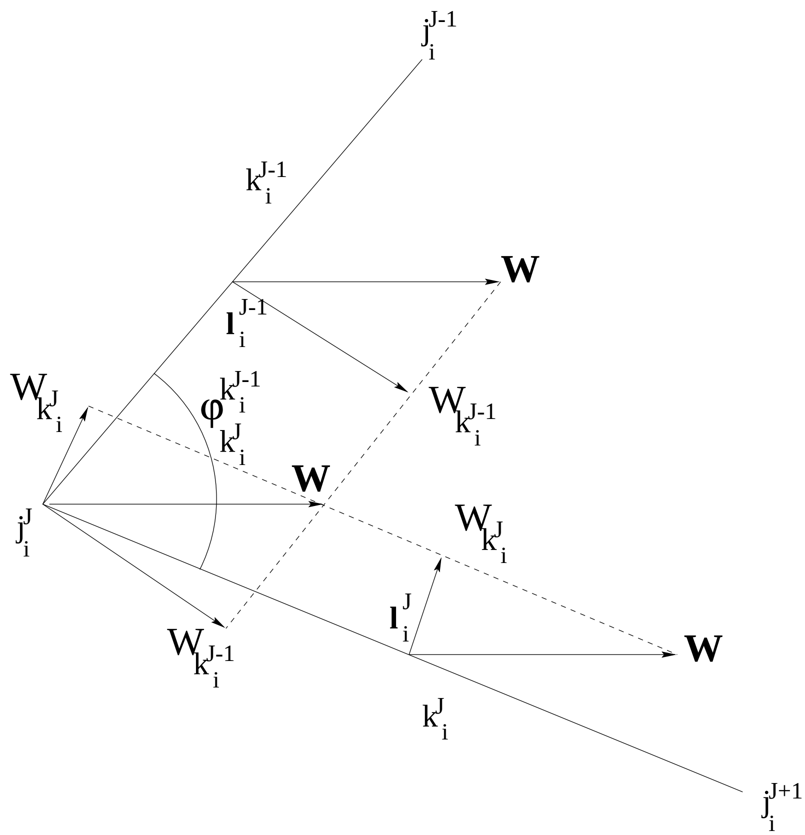


Figure 4: Projections of the vector $\mathbf{W}$ on two edge normals is moved into the join vertex of these two edges for computing the inner product $(\mathbf{A}, \mathbf{B})_{j_i^J}$ in triangle i with respect to the vertex j_i^J.

4.3 Formal inner products

Below we will need to derive discrete gradient as an adjoint operator to the operator of discrete extended divergence, so we will need to use discrete approximation to the continuum identity (12) using the discrete inner products (23) and (26). To make the transition between the discrete inner products (23) and (26) easier to describe it is usefull to intorduce the formal inner products which in fact are just sums of products of values at all the points where the discrete functions are defined. To distinguish the formal inner products from natural ones denoted by $(\cdot, \cdot)$ we denote the formal products by $[\cdot, \cdot]$.

The formal inner product on on the space HC of scalar grid functions U discretized inside the triangles by U_i and in the center of boundary edges by U_k is defined as

$$[U, V]_{HC} = \sum_{i=1}^{N_t} U_i V_i + \sum_{k=1}^{N_{eb}} U_k V_k. \tag{27}$$

It is usefull to introduce the operator M connecting natural and formal inner products on the space HC so that

$$(U, V)_{HC} = [MU, V]_{HC}. \tag{28}$$

From this definition and from (23) and (27) follows the explicit form of the operator M

$$\begin{aligned} (MU)_i &= U_i VC_i, && \text{for all triangles } i \\ (MU)_k &= U_k S_k, && \text{for all boundary edges } k \end{aligned} \tag{29}$$

The formal inner product on the space **HL** of vector grid functions **A** discretized in the centers of the triangles edges by the scalar component A_k orthogonal to the edge is given by

$$[\mathbf{A},\mathbf{B}]_{\mathbf{HL}} = \sum_{k=1}^{N_e} A_k B_k \tag{30}$$

Again it is usefull to define the operator L connecting natural and formal inner products on the space **HL** of vector grid functions

$$(\mathbf{A},\mathbf{B})_{\mathbf{HL}} = [L\mathbf{A},\mathbf{B}]_{\mathbf{HL}}. \tag{31}$$

To get the explicit for of the operator L is not so easy as in the previous case. The formal inner product on the right hand side of (31) is by definition

$$[L\mathbf{A},\mathbf{B}]_{\mathbf{HL}} = \sum_{k=1}^{N_e} (L\mathbf{A})_k B_k \tag{32}$$

The natural inner product on the left hand side (31) is by definition

$$(\mathbf{A},\mathbf{B})_{\mathbf{HL}} = \sum_{i=1}^{N_t}\sum_{J=1}^{3} \frac{A_{k_i^J}B_{k_i^J} + A_{k_i^{J-1}}B_{k_i^{J-1}} + (A_{k_i^J}B_{k_i^{J-1}} + A_{k_i^{J-1}}B_{k_i^J})\cos\phi_{k_i^J}^{k_i^{J-1}}}{K_i \sin^2\phi_{k_i^J}^{k_i^{J-1}}} V_{k_i^J}^{k_i^{J-1}}. \tag{33}$$

When we change the summation over all triangles into the summation over all edges we get

$$\begin{aligned}
(\mathbf{A},\mathbf{B})_{\mathbf{HL}} \;=\; & \sum_{k=1}^{N_e} B_k\Big[\frac{A_k + A_{k_k^{11}}\cos\phi_k^{k_k^{11}}}{\sin^2\phi_k^{k_k^{11}}\,K_{i_k^1}} V_k^{k_k^{11}} + \\
& \frac{A_k + A_{k_k^{12}}\cos\phi_k^{k_k^{12}}}{\sin^2\phi_k^{k_k^{12}}\,K_{i_k^1}} V_k^{k_k^{11}} + \frac{A_k + A_{k_k^{21}}\cos\phi_k^{k_k^{21}}}{\sin^2\phi_k^{k_k^{21}}\,K_{i_k^2}} V_k^{k_k^{21}} + \\
& \frac{A_k + A_{k_k^{22}}\cos\phi_k^{k_k^{22}}}{\sin^2\phi_k^{k_k^{22}}\,K_{i_k^2}} V_k^{k_k^{22}}\Big]
\end{aligned}$$

where for boundary edges either first two terms or second two terms are zero as the weight V_k^0 is zero outside the region V. Comparing this with (32) we get the explicit form of the operator L

$$(LA)_k = \sum_{l=1}^{2}\frac{1}{K_{i_k^l}}\sum_{m=1}^{2}\frac{A_k + A_{k_k^{lm}}\cos\phi_k^{k_k^{lm}}}{\sin^2\phi_k^{k_k^{lm}}} V_k^{k_k^{lm}}. \tag{34}$$

The operator L is a local operator at the edge k with the stencil including also four other edges $k_k^{11}, k_k^{12}, k_k^{21}, k_k^{22}$ of the two triangles joining at the edge k (see Fig. 2 for edge numbering). The stencil of the operator L is shown at Fig. 5.

4.4 Discrete approximations of continuum operators

In the support operator method we choose the prime operator, discretize it and then construct the other derived operator from approximation of integral identities (here from Gauss theorem (11)). Here we choose as the prime operator the extended divergence operator (6).

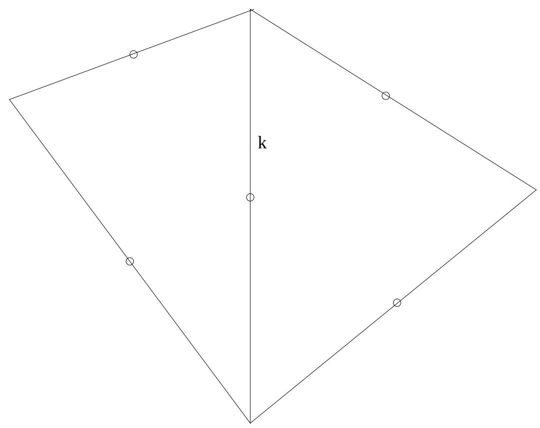


Figure 5: The stencil of the operator L on the edge k.

When we apply the divergence theorem (Green formula)

$$\int_V \operatorname{div} \mathbf{w} d\ V = \oint (\mathbf{w}, \mathbf{n}) d\ S \tag{35}$$

to the triangle i we obtain

$$(DW)_i VC_i = \sum_{J=1}^{3} W_{k_i^J} \operatorname{sign}(j_i^{J+1} - j_i^J) S_{k_i^J}, \tag{36}$$

where D is the discrete operator appoximating continuum operator $\mathbf{D}$. The sign is comming from the fixed orientation of normal on the edge (by the left hand rule). From this we can express operator D at the triangle i

$$(DW)_i = \frac{1}{VC_i} \sum_{J=1}^{3} W_{k_i^J} S_{k_i^J} \operatorname{sign}(j_i^{J+1} - j_i^J). \tag{37}$$

On the boundary edge k we approximate the continuum operator $\mathbf{D}$ by the discrete operator D

$$(DW)_k = -W_k n_k. \tag{38}$$

The discrete operator D works from the space $\mathbf{HL}$ of vector grid functions into the space HC of scalar grid functions.

For continuum operators (5) and (6) we have shown that $\mathbf{G} = \mathbf{D}^*$ (13), i.e. that the gradient is the adjoint of the extended divergence, and we want this identity to hold also in the discrete case with discrete operators and natural inner products. G will be the diskrete approximation of the continuum operator $\mathbf{G}$ and will be constructed from

$$G = D^* \tag{39}$$

where the adjoint is ment in the sence of discrete natural inner products (23) and (26). The discrete operator G works from the space HC of scalar grid functions into the space $\mathbf{HL}$ of vector grid functions.

The definition of the adjoint operator is

$$(D\mathbf{W}, U)_{HC} = (\mathbf{W}, D^* U)_{\mathbf{HL}} \tag{40}$$

and can be changed into formal inner products (27) and (30) by using previously introduced operators M (28) and L (31)

$$[D\mathbf{W}, MU]_{HC} = [\mathbf{W}, LD^*]_{\mathbf{HL}} \tag{41}$$

Using the formal adjoint operator $D^{\otimes}$ of the operator D on the left hand side we get

$$[\mathbf{W}, D^{\otimes}MU]_{\mathbf{HL}} = [\mathbf{W}, LD^*U]_{\mathbf{HL}} \tag{42}$$

and as this has to be valid for all $\mathbf{W}$ and U we see that

$$LD^* = D^{\otimes}M \tag{43}$$

Now we try to derive formal adjoint operator $D^{\otimes}$. First just from the definition of formal adjoint and definition of formal scalar product we have

$$[DW, U]_{HC} = [W, D^{\otimes}U] = \sum_{k=1}^{N_e} W_k (D^{\otimes}U)_k. \tag{44}$$

Now we express $[DW, U]_{HC}$ from the explicit form of the operator D (37), (38)

$$[DW, U]_{HC} = \sum_{i=1}^{N_t}\sum_{J=1}^{3} W_{k_i^J} S_{k_i^J} \mathrm{sign}(j_i^{J+1} - j_i^J)\frac{U_i}{VC_i} - \sum_{k=1}^{N_{eb}} W_k n_k U_k. \tag{45}$$

We rearrange the summation over all triangles and all boundary edges into the summation over all edges

$$\begin{aligned} [DW, U]_{HC} &= \sum_{k=1}^{N_{eb}} W_k \left[-U_k n_k + \mathrm{sign}(i_k^1) S_k \frac{U_{i_k^1}}{VC_{i_k^1}} - \mathrm{sign}(i_k^2) S_k \frac{U_{i_k^2}}{VC_{i_k^2}} \right] && (46)\\ &+ \sum_{k=N_{eb}+1}^{N_e} W_k S_k \left(\frac{U_{i_k^1}}{VC_{i_k^1}} - \frac{U_{i_k^2}}{VC_{i_k^2}} \right), && (47) \end{aligned}$$

where the sign just distinguishes which side of the boundary edge is inside our region V and which is outside. From this we can easily get the explicit form of the formal adjoint operator $D^{\otimes}$

$$\begin{aligned} (D^{\otimes}U)_k &= -U_k n_k + \mathrm{sign}(i_k^1) S_k \frac{U_{i_k^1}}{VC_{i_k^1}} - \mathrm{sign}(i_k^2) S_k \frac{U_{i_k^2}}{VC_{i_k^2}} && (48)\\ &\qquad \text{for boundary edges },k = 1, \cdots, N_{eb} \\ (D^{\otimes}U)_k &= S_k \left(\frac{U_{i_k^1}}{VC_{i_k^1}} - \frac{U_{i_k^2}}{VC_{i_k^2}} \right) && (49)\\ &\qquad \text{for internal edges },k = N_{eb}+1, \cdots, N_e \end{aligned}$$

where the first case is for the boundary edges $k = 1, \cdots, N_{eb}$ and the second case is for the internal edges $k = N_{eb} + 1, \cdots, N_e$.

Having the formal adjoint operator $D^{\otimes}$ we can from (43) write the natural adjoint operator as

$$D^* = L^{-1} D^{\otimes} M. \tag{50}$$

We have the explicit form of operators M (29), L (34) and $D^{\otimes}$ (48) however the explicit form of the operator inverse to L cannot be constructed, thus when we need to apply the operator of minus gradient $G = D^*$ to a scalar grid function U from HC to obtain the vector grid function $\mathbf{W} = GU$ from $\mathbf{HL}$ we have to solve the system of linear equations

$$L\mathbf{W} = D^{\otimes} MU \tag{51}$$

for unknown vector grid function $\mathbf{W}$. The system (51) the system of N_b linear algebraic equations for N_b scalar components W_k of $\mathbf{W}$ at the centers of all edges of our triangulation. The matrix of this system is given by the explicit form of the operator L (34).

4.5 Treating boundary conditions

On the boundary edges we have to distinguish if there is either Dirichlet or Neumann boundary condition on each edge. When Neumann boundary condition is on the boundary boundary edge k then we already know the value of the flux W_k through this edge and we do not include the equation for this flux W_k in the system (51) and in the final form of the discrete equation

$$(DW)_i = f_i \tag{52}$$

for the triangle i which includes this boundary edge k (there is only one such triangle) we move the known term $W_k S_k / VC_i$ from (37) with appropriate sign into the right hand side f_i.

On the other hand for the boundary edge k with Dirichlet boundary condition we have to include the equation for the flux W_k in the system (51) and this equation includes the given value $U_k = \psi_k$ through formal adjoint on the boundary (48).

For numerical solution of the system of linear equations (51) (after excluding from it equations for fluxes on boundary edges with Neumann boundary conditions) we use Gauss-Seidel iterative method. The global discrete operator $P = DG$ is symmetric and positive definite and for solving the global system $(PU)_i = f_i, i = 1, \cdots, N_t$ we use the conjugate gradient iterative method.

5 Numerical examples

In this section we provide several numerical examples which demonstrate the properties of the developed numerical method. First we evaluate the convergence rate of our support operator method and show that it is exact for linear solutions even with discontinous diffusion coefficients. Then we show that our method works reasonably well also on rather bad quality triangulation including triangles with very small angles and clearly is superior to a standard linear finite element method. In the last part of this section we present several examples of stationary heat flow through heat conducting rectangle having either areas of heat isulating material or holes. For triangular grid construction we have been using Bank's code [14].

5.1 Convergence examples with known solution

All examples here solve the generalized Laplace equation $\operatorname{div} k\, \mathbf{grad}\, u = f$ on the unit square $(x, y) \in (0, 1) \times (0, 1)$. When not noted otherwise we use diffusion coefficient $k = 1$. For a chosen exact solution u^e we analytically derive the right hand side $f = \operatorname{div} k\, \mathbf{grad}\, u^e$ and after numericaly solving the Laplace equation with this right hand side we can easily get the error of the numerical solution.

The developed support operator method discretization is exact for linear solutions and second order [13] for others and we verify here this convergence rate. The asymptotic error E_h estimate on a trianglular grid is given by

$$\|E_h\| = Ch^q + O(h^{q+1}), \tag{53}$$

where h is the maximal length of edge in the grid, q defines the order of the truncation error, the convergence-rate constant C is a positive constant independent of h, and $\|\cdot\|$ is some norm. To estimate the order q we evaluate the error on a sequence of refined grids. Using the error estimate on two grids with the parameters h and $h/2$ the order of the truncation error is approximatelly

$$q \approx \log_2 \|E_h\|/\|E_{h/2}\|.$$

In our convergence study we use the maximum norm

$$E_{\max} = \|U - p_h u^e\|_{\max} = \max_{i=1,\cdots,N_t} |U_i - (p_h u^e)_i|, \tag{54}$$

where U_i is the numerical solution of the finite-difference scheme and u^e is the exact solution of the given problem. Operator p_h is the point projection for scalar function given by $(p_h u^e)_i = u(\mathbf{m}_i)$, where $\mathbf{m}_i$ is the median of the triangle i, and we take maximum over all triangles.

In order to find the order of convergence numerically and to show that the constant C in (53) does not depend on the parameter h, we have to be sure that while refining the grid the ratio of the maximum value of the triangle area $V_{\max}$ to the minimum value $V_{\min}$ remains constant [13]. To achieve this we use the uniform refinement which in one refinement step puts new vertex into the center of each edge and divides each triangle into four similar triangles so that the sequence of grid parameters h is given by $h/2^n$. Such refinement keeps the ratio $V_{\max}/V_{\min}$ constant. Table 1 presents values of h, number of triangles N_t, areas $V_{\max}$, $V_{\min}$ and the ratio $V_{\max}/V_{\min}$ for particular sequence of refined grids which we use in the convergence analysis in this subsection.

h	N_t	$V_{\max}$	$V_{\min}$	$V_{\max}/V_{\min}$
0.4	34	0.0472	0.0157	3.0
0.2	136	0.0118	0.0039	3.0
0.1	544	0.0029	0.00097	3.0
0.05	2176	0.0007	0.00024	3.0
0.025	8704	0.000184	0.000061	3.0

Table 1: Grid parameters for used sequence of refined grids: maximum edge length h, number of triangles N_t, maximum and minimum of triangle area $V_{\max}$ and $V_{\min}$ and their ratio.

5.1.1 Smooth solutions

Our method is exact on constant and linear solutions which has been confirmed numerically on several examples like $u = 1$, $u = x$ and $u = x + y$. Next we try three nonlinear solutions $u = x^2$, $u = x(1-x)y(1-y)$ and $u = \sin(\pi x)\sin(\pi y)$. First we numericaly solve the Laplace equation with these exact solutions with Dirichlet boundary conditions everywhere. The errors and orders of convergence on our sequance of refined grids is presented in Table 2. The errors and orders of convergence of the same problems, now however with Dirichlet boundary conditions only at the right side of the solution square and with Neumann boundary conditions on the remaining three sides, is shown in Table 3. Both tables confirm the second order convergence rate of our method.

h	$u = x^2$		$u = x(1-x)y(1-y)$		$u = \sin(\pi x)\sin(\pi y)$	
	$E_{\max}$	q	$E_{\max}$	q	$E_{\max}$	q
0.4	0.0064	1.96	0.0055	1.89	0.087	1.96
0.2	0.0016	2.0	0.0015	1.93	0.022	1.96
0.1	0.00042	1.96	0.00039	1.96	0.0057	1.96
0.05	0.00011	1.96	0.0001	2.0	0.0014	2.0
0.025	0.000027		0.000025		0.00036	

Table 2: Maximum error and order of convergence for nonlinear solutions with Dirichlet boundary conditions on refined grids.

h	$u = x^2$		$u = x(1-x)y(1-y)$		$u = \sin(\pi x)\sin(\pi y)$	
	$E_{\max}$	q	$E_{\max}$	q	$E_{\max}$	q
0.4	0.007	1.96	0.0039	1.96	0.143	2.0
0.2	0.0018	1.93	0.0011	1.96	0.035	1.96
0.1	0.00047	1.96	0.00028	1.96	0.009	1.96
0.05	0.00012	1.96	0.000072	1.96	0.0023	2.0
0.025	0.000031		0.000018		0.00057	

Table 3: Maximum error and order of convergence for nonlinear solutions with Neumann and Dirichlet boundary conditions on refined grids.

5.1.2 Discontinuous coefficient problem

Here we present several examples with discontinuous piecewise constant diffusion coefficient

$$k = \begin{cases} k_1, & 0 < x < 0.5, \\ k_2, & 0.5 < x < 1. \end{cases}$$

which have exact solution:
Test DC1 – in this problem comming from [15], [6] the exact solution is a piecewise linear function

$$u = \begin{cases} \frac{k_2 x + 2k_1k_2}{0.5(k_1+k_2+4k_1k_2} & 0 < x < 0.5 \\ \frac{k_1 x + 2k_1k_2 + 0.5(k_2-k_1)}{0.5(k_1+k_2)+4k_1k_2} & 0.5 < x < 1, \end{cases} .$$

We solve this problem with Dirichlet boundary conditions for particular values of diffusion coefficient $k_1 = 1$, $k_2 = 2$. Of course the triangulation is done such a way that the whole discontinuity line $x = 0.5$ is covered by the edges and no triangle intersects it.
Test DC2 – this test is the same as the preceding one with the value $p_1 y$ being added to the solution so that the exact solution is

$$u = \begin{cases} \frac{k_2 x + 2k_1k_2}{0.5(k_1+k_2+4k_1k_2} + p_1 y & 0 < x < 0.5 \\ \frac{k_1 x + 2k_1k_2 + 0.5(k_2-k_1)}{0.5(k_1+k_2)+4k_1k_2} + p_1 y & 0.5 < x < 1 \end{cases}$$

We solve this problem with Dirichlet boundary conditions for particular values of diffusion coefficient $k_1 = 1$, $k_2 = 10$ and parameter $p_1 = 0.1$.

Test DC3 – in this problem comming from [16], [6] the exact solution is a piecewise quadratic function

$$u = \begin{cases} a_1 \frac{x^2}{2} + b_1 x & 0 \le x \le 0.5 \\ a_2 \frac{x^2}{2} + b_2 x + c_2 & 0.5 \le x \le 1 \end{cases}$$

where

$$a_i = -\frac{1}{k_i}, \quad b_1 = \frac{3a_2 + a_1}{4} \frac{k_2}{k_1 + k_2}, \quad b_2 = \frac{k_2}{k_1} b_1, \quad c_2 = -\left(b_2 + \frac{a_2}{2}\right).$$

We solve this problem with Dirichlet boundary conditions for particular values of diffusion coefficient $k_1 = 1$, $k_2 = 2$.

The convergence analysis for three examples with a discontinuous coefficient is presented in Table 4 and confirms that our method is exact for even for non-smooth piecewise linear solutions and second order for non-linear solutions even in case of discontinuous diffusion coefficients.

h	DC1	DC2	DC3	
	$E_{\max}$	$E_{\max}$	$E_{\max}$	q
0.4	$2.2 \cdot 10^{-11}$	$8.2 \cdot 10^{-12}$	0.011	1.89
0.2	$1.0 \cdot 10^{-11}$	$2.8 \cdot 10^{-12}$	0.0030	1.93
0.1	$5.7 \cdot 10^{-12}$	$1.3 \cdot 10^{-12}$	0.00082	1.96
0.05	$3.1 \cdot 10^{-12}$	$7.3 \cdot 10^{-12}$	0.00021	2.0
0.025	$1.6 \cdot 10^{-12}$	$3.6 \cdot 10^{-12}$	0.000053	

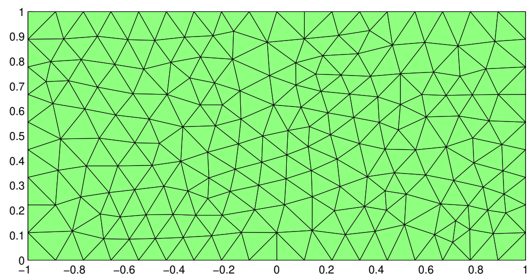
Table 4: Maximum error and order of convergence for problems with discontinuous diffusion coefficient with Dirichlet boundary conditions on refined grids.

5.2 Unisotropic triangulation

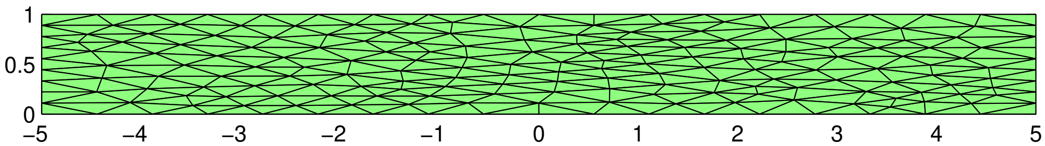
Here we compare our method with standard linear finite element method (FEM) [17] on unisotropic triangulation including triangles with very small angles. It is known that such triangulation causes troubles to FEM and we show that our method deals with this problems much better. Naturally FEM should not be used with such a triangulation, on the other hand in some problems such triangulation appears and we need a method working well also for such triangulation (of bad quality for FEM). An example of a problem where such triangulation might easily appear is Lagrangian hydrodynamics which moves grid cells and vertices with the fluid flow and where we need to treat parabolic part of the model equation.

In this example we use the problem with exact solution $u = x^2/a^2$ with parameter a on the rectangular region $(x, y) \in (-a, a) \times (0, 1)$. The grids for parameters $a = 1$ and $a = 5$ are shown on Fig. 6. For the values of the parameter a other than $a = 1$ we use the same grid, only we stretch x coordinates of all objects by multiplying them by a so that the grid covers the region $(-a, a) \times (0, 1)$. For $a >> 1$ all the tringles in the triangulation become very long in the x direction and thus have very small angles.

For different values of the parameter a we solve the Laplace equation $\operatorname{div} \mathbf{grad}\, u = 2/a^2$ with Dirichlet boundary conditions $u = 1$ on the left and right boundaries $x = \pm a$ and zero Neumann boundary conditions $(\mathbf{grad} u, \mathbf{n}) = 0$ on the lower and upper boundaries $y = 0, y = 1$. This problem has a unique solution $u = x^2/a^2$. In Table 5 we compare results of our method with results of standard FEM with linear elements as implemented in [14]. The table presents for



(a)



(b)

Figure 6: Grid used for stretching the triangulation; (a) for parameter $a = 1$, i.e. $x \in (-1, 1)$, (b) for parameter $a = 5$, i.e. $x \in (-5, 5)$

several values of the parameter a the error in maximum norm of the computed solution (in all cases solution $u \in (0, 1)$) and the minimal value of the computed solution u.

	SOM		FEM	
a	E_{max}	$min(u)$	E_{max}	$min(u)$
1	0.0037	-0.0028	0.0018	-0.0008
10	0.0019	-0.0015	0.05	0.02
25	0.0019	-0.0015	0.18	0.15
50	0.0019	-0.0015	0.45	0.45
100	0.0019	-0.0015	0.77	0.77
1000	0.0069	-0.0043	0.997	0.997
10000	0.2016	-0.2012	1.0	1.0

Table 5: Maximum error and minimum of numerical solution on unisotropic triangulation stretched by a by our SOM (support operator method) and linear FEM.

From Table 5 we can see the difference between FEM and our method in this case. As is well known FEM is not working well for very long triangles with very small angles which appear in the triangulation in the case of big parameter a. The maximum error for FEM is already 5% for $a = 10$ and 18% for $a = 25$ getting much worse, while our method is giving the same errors up to $a = 10^2$ and still for $a = 10^3$ its error is less than 1%. If we increase a further also our method starts to give larger errors as 20% for $a = 10^4$. Apparently for our method the problems are starting for much higher a than for FEM, we are getting similar error for $a = 10^4$ as FEM is getting for $a = 25$.

The origin of the FEM troubles lies in fact that for big a there are no edges in the triangulation which are parallel to the axis y. For such a grid and the exact solution with curvature in the s direction the linear interpolation on the edges (which for big a are almost parallel to the axis x) introduces zig-zagging in the y direction which is eating too much of the overall energy and the parabola in the x direction is not resolved well. Basically for high a the minimum $u = 0$ of the parabola $u = x^2/a^2$ is getting higher until for biggest $a > 10^3$ the FEM solution is very close to constant solution $u = 1$.

Table 6 presents convergence results of our method for these problems for different values of a. The grid has been refined by the same way as in the previous section by introducing new vertices in the middle of each edge, so that each triangle is divided into four smaller similar triangles. We see that our method has reasonable convergence even on very stretched grids with very small angles up to $a = 100$. For $a = 1000$ the convergence is lost.

a	$h = 0.18$		$h = 0.09$		$h = 0.045$	
	$E_{\max}$	q	$E_{\max}$	q	$E_{\max}$	q
1	0.00372	1.92	0.00097	1.96	0.00025	...
10	0.00193	1.8	0.00057	1.8	0.00016	...
25	0.00193	1.8	0.00057	1.8	0.00016	...
50	0.00193	1.8	0.00057	1.8	0.00016	...
100	0.00193	1.8	0.00055	1.7	0.00017	...
1000	0.069	0.5	0.0048	...	0.099	...

Table 6: Errors and convergence rate for triangulation with triangles with small angles by our support operators method for different values of a.

5.3 Examples of heat flow

In this section we present five examples of solving the Laplace equation (1)-(3) with zero source $f = 0$ on a rectangle with regions of different diffusion coefficients k or rectangle with holes. These examples present stationary heat flow through the rectangle and diffusion coefficient is the coefficient of heat conductivity. All these examples have Dirichlet boundary conditions on the left $u = 1$ and the right $u = 0$ and zero Neumann boundary conditions on the bottom and top. This means that we fix the temperature on the left and right and assume no heat flow through the top and bottom boundary of the rectangle.

For first three examples most of the rectangle is the region with high conductivity $k = 1$ and inside the rectangle there are some regions of heat insulator material with very low heat conductivity $k = 10^{-6}$. The problems differ in the geometry of insulator material regions:
Cicrle – problem is solved on the rectangle $(x, y) \in (-2, 2) \times (-2, 2)$ and the insulator region is the circle with center in the origin and radius one, see Fig. 7.
Fingers – problem is solved on the rectangle $(x, y) \in (0, 1) \times (0, 1)$ and there are two rectangular insulator areas $(0.2, 0.3) \times (0, 0.8)$ and $(0.6, 0.7) \times (0.3, 1)$, fingers from bottom and top, see Fig. 8.
Streak – problem is solved on the rectangle $(x, y) \in (0, 1) \times (0, 1)$ and insulator area is the curved streak between two arcs with center at $(0.1, -0.4)$ and radiuses 1.1 and 1.2, see Fig. 9.

Next two problems use only one heat conductivity $k = 1$ however the rectangle has several holes without any material, so the solution domain is non-convex. On the boundaries of the holes we use zero Neumann boundary conditions. These two examples differ in the position of three circular holes:
Three holes arranged vertically – problem is solved on the rectangle $(x, y) \in (-1, 1) \times (-1, 1)$ and insulator areas are three vertically arranged circles with the same radius $1/6$ and centres at

$(0, 0), (0, \pm 2/3)$, see Fig. 10.

Three holes arranged randomly – problem is solved on the rectangle $(x, y) \in (-1, 1) \times (-1, 1)$ and insulator areas are three randomly arranged circles with the same radius 1/6 and centres at $(0.6, -0.1), (-0.5, 0.7), (0.4, -0.5)$, see Fig. 11.

For each problem we present numerical results in four figures:

(a) triangular grid with material property, heat conductivity coefficient plotted by color; white here means no material presented in examples with holes

(b) colormap of temperature with triangular grid

(c) arrow plot of heat flux; one can notice here the directions of stationary heat flux alog internal boundaries

(d) temperature contours (isolines of constant temperature) with triangular grid

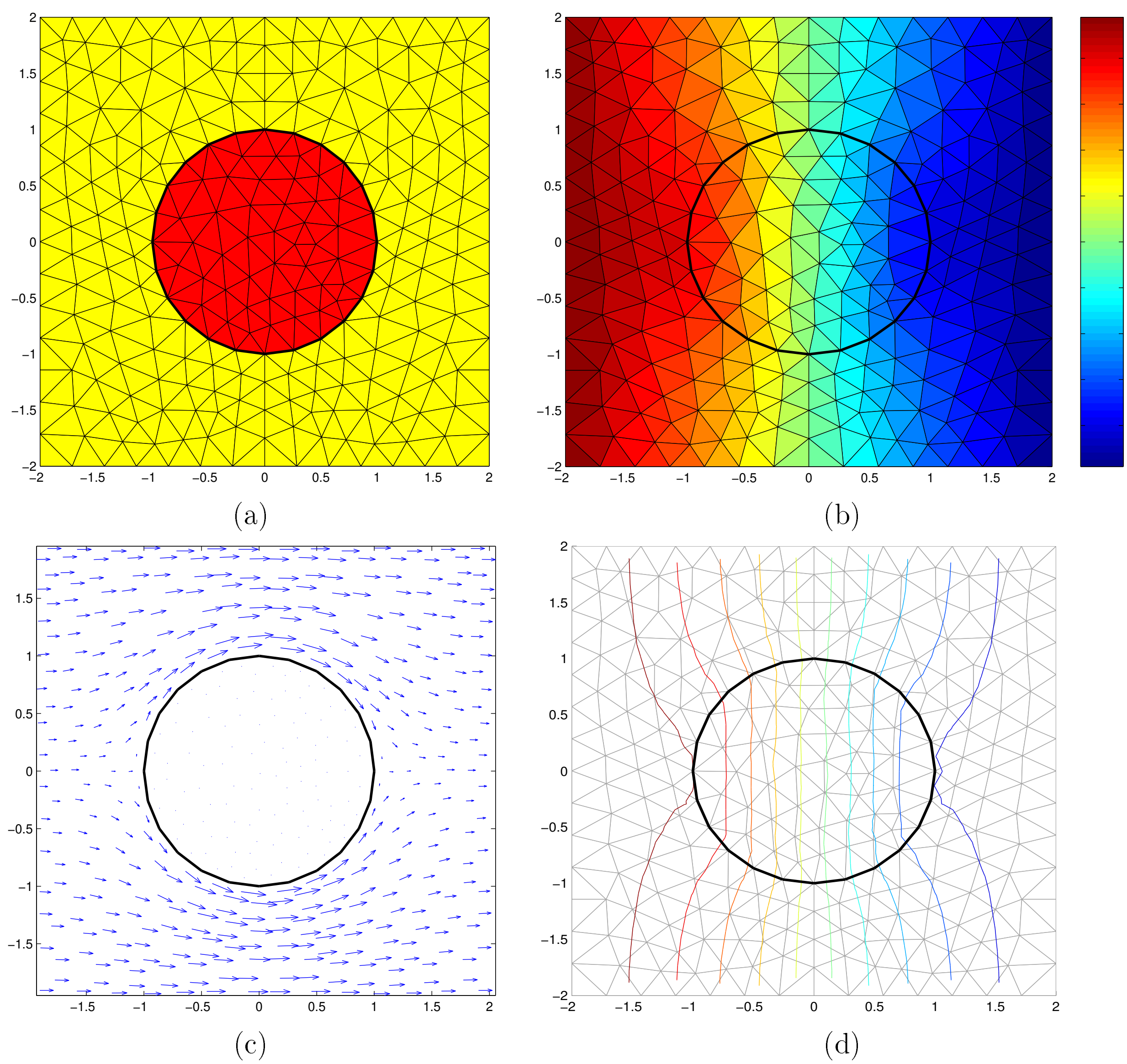


Figure 7: Circle problem: (a) grid with material properties, (b) temperature distribution, (c) heat flux distribution, (d) temperature contour isolines

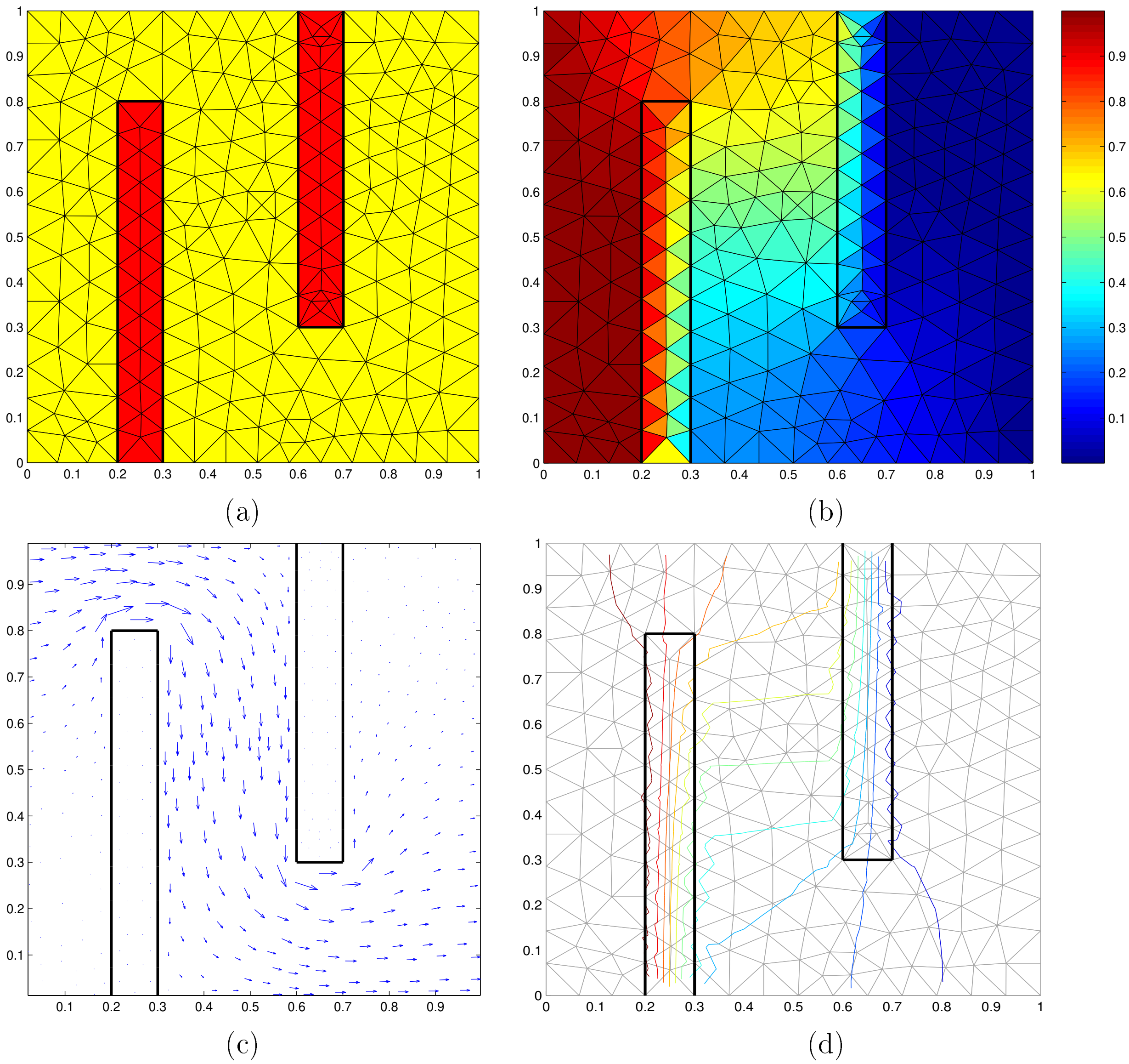


Figure 8: Fingers problem: (a) grid with material properties, (b) temperature distribution, (c) heat flux distribution, (d) temperature contour isolines

6 Conclusion

We have developed support operator discretization method for generalized Laplace equation on unstructured triangular grid with Dirichlet and Neumann boundary conditions. The method works very well for discontinuous diffusion coefficients. It is exact for linear solution and second order for nonlinear solution. It works remarkably well also for bad quality triangulations having triangles with very small angles. Presented sample numerical results confirm these properties of the developed numerical method.

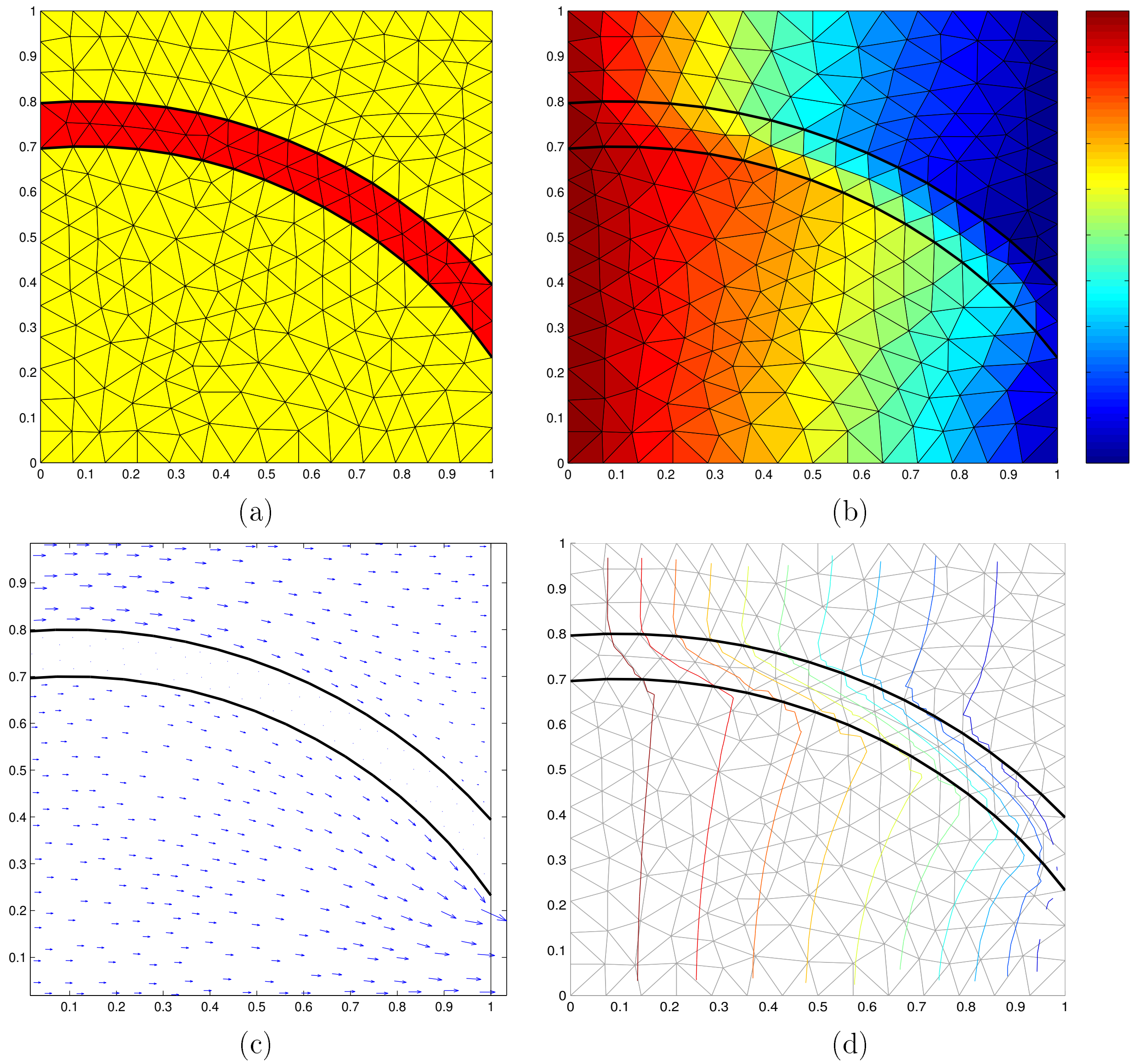


Figure 9: Streak problem: (a) grid with material properties, (b) temperature distribution, (c) heat flux distribution, (d) temperature contour isolines

Acknowledgement

This research has been partially supported by International Bureau of the BMBF and Czech Ministry of Education in the framework of German-Czech join research project nr. CZE-00-010 and by the project nr. ME 436/2001 from the program Kontakt of Czech Ministry of Education. The work of M. Shashkov was performed under auspices of the US Department of Energy under contract W-7405-ENG-36 and DOE/BES Program in the Applied Mathematical Sciences Conratct KC-07-01-01.

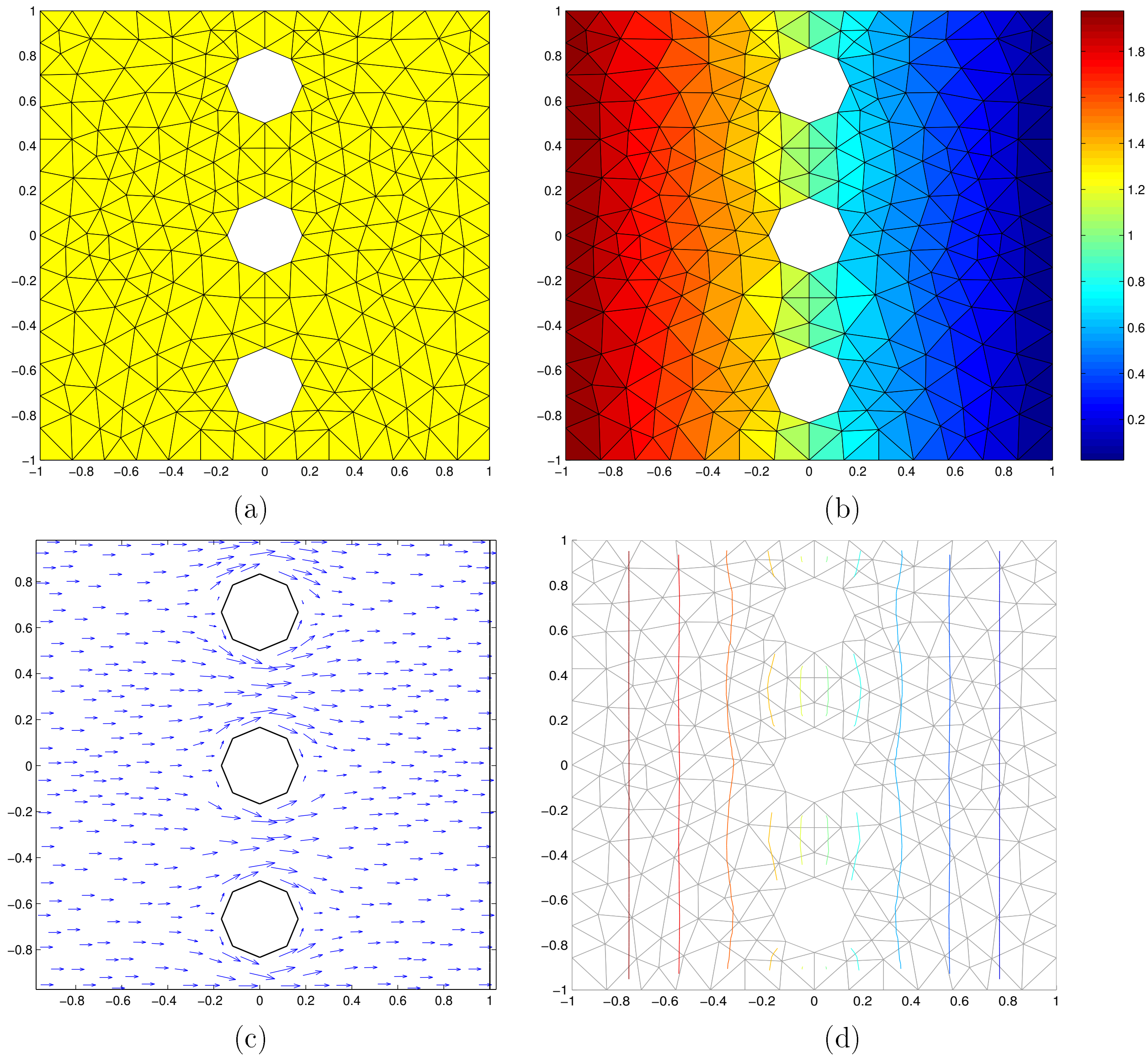


Figure 10: Three holes arranged vertcally problem: (a) grid with material properties, (b) temperature distribution, (c) heat flux distribution, (d) temperature contour isolines

References

[1] J. M. Hyman and M. Shashkov. Natural discretizations for the divergence, gradient, and curl on logically rectangular grids. *An International Journal Computers & Mathematics with Applications*, 33:81–104, 1997.

[2] J. M. Hyman and M. Shashkov. The approximation of boundary conditions for mimetic finite difference methods. *An International Journal Computers & Mathematics with Applications*, 36:79–99, 1998.

[3] J. Hyman, M. Shashkov, and S. Steinberg. The effect of inner products for discrete vector fields on the accuracy of mimetic finite difference methods. *Computers & Mathematics with Applications*. to appear.

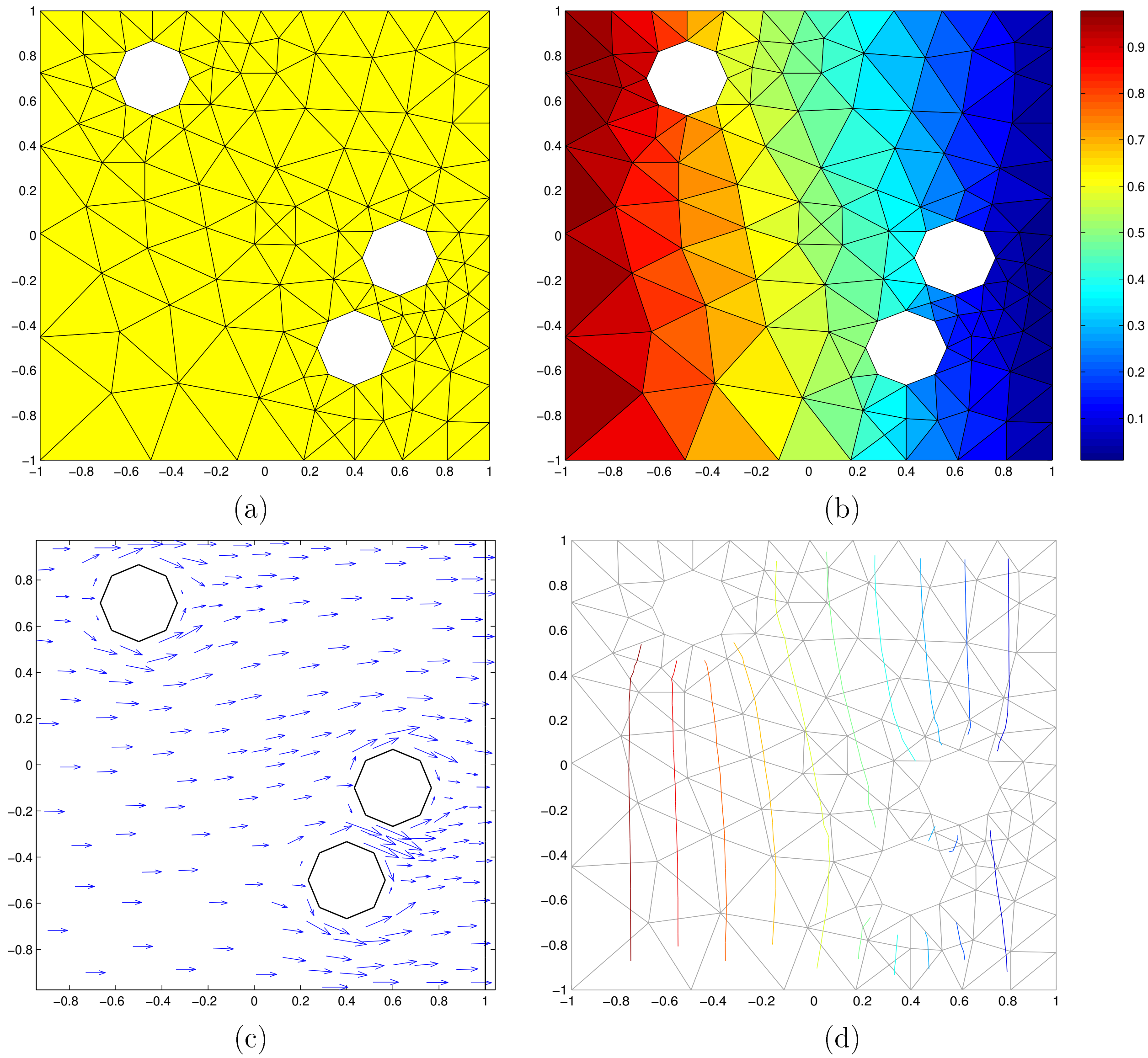


Figure 11: Three holes arranged randomly problem: (a) grid with material properties, (b) temperature distribution, (c) heat flux distribution, (d) temperature contour isolines

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

[5] M. Shashkov and S. Steinberg. Support-operator finite-difference algorithms for general elliptic problems. *J. Comp. Phys.*, 118:131–151, 1995.

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

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

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

[9] J. Morel, M. Hall, and M. Shashkov. A local support-operators diffusion discretization scheme for hexahedral meshes. *J. Comp. Phys.* Submitted.

[10] E. J. Caramana, D. E. Burto, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comp. Phys.*, 146, 1998.

[11] J. M. Hyman and M. Shashkov. Mimetic discretizations for maxwell's equations. *J. Comp. Phys.*, 151:881–909, 1999.

[12] J.M. Hyman and M. Shashkov. Approximation of boundary conditions for mimetic finite-difference methods. *Computers Math. Applic.*, 36:79–99, 1998.

[13] M. Berndt, K. Lipnukov, J. D. Moulton, and M. Shashkov. Convergence of mimetic finite difference discretizations of the diffusion equation. Technical Report LA-UR-01-6187, Los Alamos National Laboratory, Los Alomos, USA, 2001.

[14] R.E. Bank. *PLTMG: A Software Package for Solving Elliptic Partial Differential Equtions. Users' Guide 8.0.* SIAM Publisher, Philadelphia, 1998.

[15] J.M. Morel, J.E. Dendy Jr, M.L. Hall, and S.W. White. A cell-centered lagrangian-mesh diffusion differencing scheme. *J. Comp. Phys.*, 103, 1992.

[16] R. J. MacKinnon and G. F. Carey. Analysis of material interface discontinuities and superconvergent fluxes in finite difference theory. *J. Comp. Phys.*, 75, 1988.

[17] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics.* Cambridge University Press, 1997.