THE MORTAR FINITE ELEMENT METHOD IN 2D: IMPLEMENTATION IN MATLAB

J. Daněk [1], *H. Kutáková* [2]

[1] Department of Mathematics, University of West Bohemia, Pilsen
[2] MECAS ESI s.r.o., Pilsen

Abstract

The paper is focused on the mortar finite element method for solving linear elliptic problems in 2D. The mortar finite element method is a nonconforming domain decomposition technique tailored to handle problems posed on domains that are partitioned into independently triangulated subdomains. In the contribution we explained the principle and properties of the method. A significant part of the paper is dedicated to the implementation of the mortar method in the Matlab system. The numerical results are showing both the principle and the possibility of practical use of the method.

1 Introduction

The finite element method is applicable to a wide range of physical and engineering problems which can be described by means of partial differential equations. The mortar finite element discretization is a discontinuous Galerkin approximation. The functions in the approximation subspaces have jumps across subdomain interfaces and are standard finite element functions when restricted to the subdomains. The jumps across subdomains interfaces are constrained by conditions associated with one of the two neighboring meshes so that a weak continuity condition must be fulfilled. Because of the discontinuity on the interface we classify the mortar finite element method as a nonconforming finite element method (see [6]).

Mortar finite elements were first introduced in 1994 by Christine Bernardi, Yvon Maday and Anthony T. Patera in [1]. Our paper is focused on the mortar finite element method in two dimensions and its implementation in the Matlab system. Most of the literature describing the mortar finite element method deal with the geometrically conforming partition, which is easier. Therefore, this paper is focused on the nonconforming case.

2 Mortar Finite Element Method

In this section we briefly describe the mortar finite element method in two dimensions. Let $\Omega \subset \mathbb{R}^2$ be a polygonal computational domain. We decompose the domain into P nonoverlapping polygonal subdomains

$$\overline{\Omega} = \bigcup_{i=1}^{P} \overline{\Omega}_i, \qquad \Omega_j \cap \Omega_k = \emptyset \quad \text{for } j \neq k, \quad j,k = 1, \ldots, P.$$

The partition can be:

- *Geometrically conforming* – The intersection between two closure of any two subdomains $\overline{\Omega}_i \cap \overline{\Omega}_j$, $i \neq j$ is either an entire edge, a vertex or empty. Figure 1 shows an example of the partition.

- *Geometrically nonconforming* – All the other cases. See example of the nonconforming partition in Figure 2.
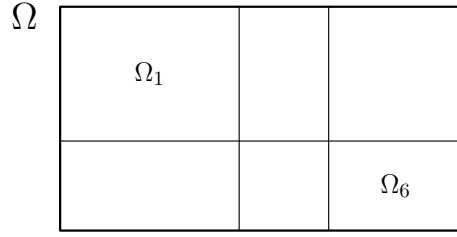
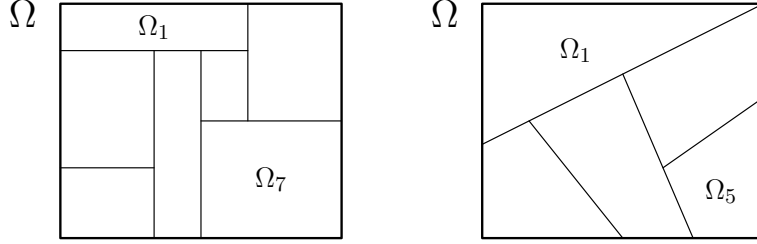Figure 1: Example of geometrically conforming partition of Ω into subdomains Ω_i.

Figure 2: Example of geometrically nonconforming partition of Ω into subdomains Ω_i.

The subdomains Ω_i form together a coarse mesh of the whole domain Ω. We discretize each subdomain by triangular elements known from the finite element method. The size of the triangles can be chosen with regard to the problem. We use finer mesh in subdomains, where big changes in behaviour of the solution are expected, etc. The resulting triangulation can be nonmatching across the interfaces of the subdomains, as you can see in Figure 4.

We define the interface Γ between subdomains Ω_i as a closure of the union of the parts of the boundaries of $\partial\Omega_i$, $i = 1, \ldots P$, that are interior to Ω

$$\Gamma = \overline{\cup_{i=1}^{P}(\partial\Omega_i \setminus \partial\Omega)}. \tag{1}$$

The interface can be considered also as a set of nodes, that belong to the boundary of at least two subdomains.

We denote

$$V^h(\Omega) = V_1^h(\Omega_1) \times V_2^h(\Omega_2) \times \cdots \times V_P^h(\Omega_P) \tag{2}$$

the space of mortar finite elements defined on Ω. We consider the low order basis functions, for example the piecewise linear basis functions. $V_i^h(\Omega_i)$ is a finite element space in each subdomain Ω_i. $V_i^h(S)$ is a restriction of functions from V^h to a set S.

For further analysis, we introduce some more notation. A main edge will represent a side of a planar n-agle, see Figure 3. A square has four main edges, n-agle has n main edges. Let Γ_{ij} be an open common edge or a part of an edge of two adjacent subdomains Ω_i and Ω_j

$$\overline{\Gamma}_{ij} = \overline{\Omega}_i \cap \overline{\Omega}_j. \tag{3}$$

The edge Γ_{ij} is a part of a main edge (or is a main edge) both of the subdomain Ω_i and Ω_j. We choose the main edge of one subdomain as a master (mortar) and the main edge of the second subdomain as a slave (nonmortar). We use a new notation:

- *Mortar edge γ* – If it belongs to a particular main edge of a boundary $\partial\Omega_i$, we denote it γ_i.

- *Nonmortar edge δ* – If it belongs to a particular main edge of a boundary $\partial\Omega_j$, we denote it δ_j.

Figures 3 and 4 show examples of situations, that can occur on the interface. If we consider geometrically conforming partition of Ω, it is obvious, that for two subdomains Ω_i and Ω_j with a commom edge holds an equality $\gamma_i = \Gamma_{ij} = \delta_j$, or $\gamma_j = \Gamma_{ij} = \delta_i$ (it's important which edge is chosen as a mortar), see Figure 4. The situation is more complicated in the nonconforming case. There is arising a question, how to choose mortar and nonmortar edges. An example of such a choice is in Figure 5. It can be proven that the partition always exists (see [3] or [6]).

In term of a new notation we can write:

$$\Gamma = \bigcup_{k=1}^{K} \overline{\gamma_k}, \quad \gamma_m \cap \gamma_n = \emptyset, \quad \text{if } m \neq n, \quad m, n = 1, \ldots, K, \tag{4}$$

where K is the number of all mortar edges. Obviously also

$$\Gamma = \bigcup_{l=1}^{L} \overline{\delta_l}, \quad \delta_m \cap \delta_n = \emptyset, \quad \text{if } m \neq n, \quad m, n = 1, \ldots, L, \tag{5}$$
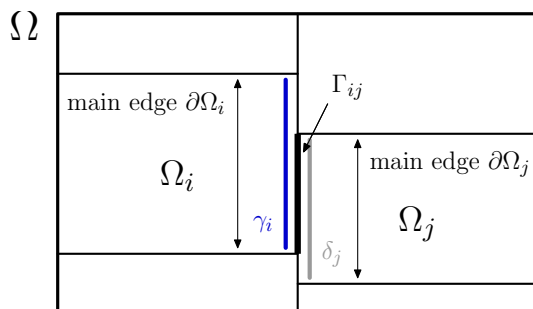
where L is the number of all nonmortar edges.

Figure 3: Interface Γ_{ij} of two subdomains Ω_i and Ω_j and signification of edges as mortar and nonmortar.
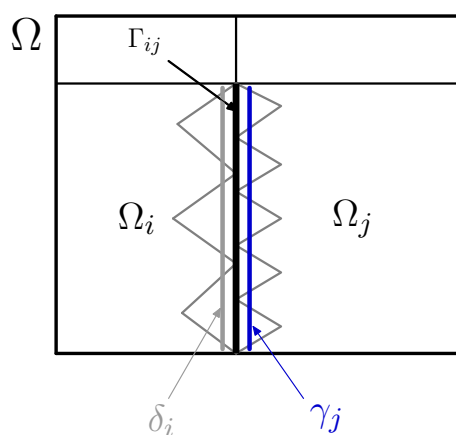
Figure 4: Mortar edge γ and nonmortar edge δ on the interface Γ_{ij} of two subdomains Ω_i and Ω_j.

It remains to solve the most important point of the method - the situation on the interface Γ. It's necessary to join somehow the values of the searched function (we call it mortar function)
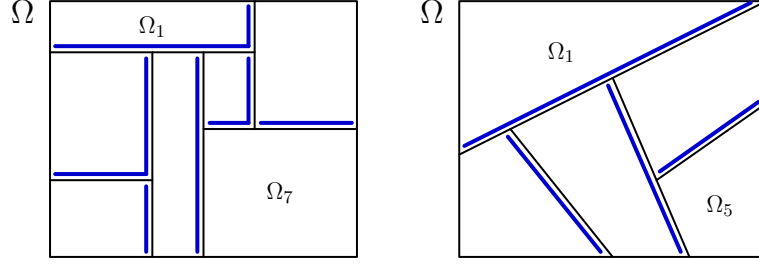
Figure 5: Example of choice of mortar edges in geometrically nonconforming case (mortar edges are blue).

on the interface. Instead of a pointwise continuity we require fulfilment of a weak continuity condition. The exact formulation will follow after some necessary definitions.

Each nonmortar edge δ_l belongs to exactly one subdomain, we denote it Ω_l. Let Γ_l be the union of q parts of the mortar edges $\gamma_{l,i}$ which correspond geometrically to nonmortar edge δ_l

$$\Gamma_l = \bigcup_{i=1}^{q} (\overline{\gamma_{l,i}} \cap \overline{\delta_l}). \tag{6}$$

For each nonmortar edge δ_l we choose a space of test functions $\Psi^h(\delta_l)$, which is a subspace of the space $V_l^h(\delta_l)$, that is a restriction of functions from $V_l^h(\Omega_l)$ to δ_l. So if we choose piecewise linear basis functions, $V_l^h(\Omega_l)$ will be a space of piecewise linear functions. $\Psi^h(\delta_l)$ will be a restriction of functions from $V_l^h(\Omega_l)$ to δ_l with requirement that these continuous, piecewise linear functions are constant in the first and last mesh intervals of δ_l (see Figure 6). Other possibility how to establish the space of test functions is described in [7].
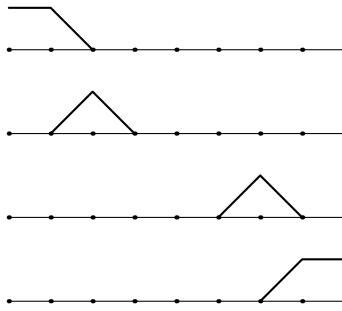

Figure 6: Test functions on δ_l.

We define the mortar projection on δ_l as $\pi_{q_1,q_2}(u_l) : L^2(\Gamma_l) \to V_l^h(\delta_l)$. For two arbitrary values q_1 and q_2 and a function $u_l \in L^2(\Gamma_l)$ it satisfies

$$\int_{\delta_l} (u_l - \pi_{q_1,q_2}(u_l))\psi \; ds = 0 \quad \forall \psi \in \Psi^h(\delta_l). \tag{7}$$

The condition means, that the jump of the mortar function across each nonmortar edge must be orthogonal (in $L^2(\Gamma_l)$) to the space of test functions defined on δ_l. The condition is called the weak continuity condition or the mortar condition.

3 Formulation of the Mortar Problem

3.1 Variational (Weak) Formulation of the Mortar Problem

Let us remind a variational (weak) formulation of a Poisson problem in two dimensions. We need to find a solution $u \in W_2^1(\Omega)$ of the Poisson equation

$$-\triangle u = f \quad \text{in } \Omega \subset \mathbb{R}^2, \tag{8}$$

$$u = g_1 \quad \text{on } \partial\Omega_D, \quad \frac{\partial u}{\partial n} = g_2 \quad \text{on } \partial\Omega_N, \tag{9}$$

where $f \in L_2(\Omega)$, $g_1 \in L_2(\partial\Omega_D)$, $g_2 \in L_2(\partial\Omega_N)$. The solution $u \in W_2^1(\Omega)$ must be from the set of admissible functions

$$V_g = \{u \in W_2^1(\Omega) : u = g_1 \text{ na } \partial\Omega_D \text{ in the sense of traces}\} \tag{10}$$

and for every test function v must satisfy

$$a(u, v) = F(v) \quad \forall v \in V, \tag{11}$$

where

$$V = \{v \in W_2^1(\Omega) : v = 0 \text{ on } \partial\Omega_D \text{ in the sense of traces}\}. \tag{12}$$

$$a(u, v) = \int_\Omega [\operatorname{grad} u \operatorname{grad} v] \, d\Omega, \tag{13}$$

$$F(v) = \int_\Omega fv \, d\Omega + \int_{\partial\Omega_N} g_2 v \, dS. \tag{14}$$

We can formulate the problem (8), (9) in terms of a decomposition of the domain Ω into subdomains Ω_i, $i = 1, 2, \ldots, P$ and by using an appropriate mortar finite element space V^h. We appear from the weak formulation (11) of the problem (8), (9) and we rewrite it to the discrete form.

We know, that $V_i^h(\Omega_i) \subset W_2^1(\Omega_i)$, $i = 1, \ldots, P$. Thus, we can write

$$a^\Gamma(u, v) = F^\Gamma(v) \quad \forall v \in W_2^{1,0}(\Omega). \tag{15}$$

$a^\Gamma(\cdot, \cdot)$ is a bilinear form, which is defined as a sum of contributions from the particular subdomains

$$a^\Gamma(u, v) = \sum_{i=1}^P \int_{\Omega_i} [\operatorname{grad} u \operatorname{grad} v] \, d\Omega_i \tag{16}$$

and

$$F^\Gamma(v) = F(v) = \int_\Omega fv \, d\Omega + \int_{\partial\Omega_N} g_2 v \, dS. \tag{17}$$

A variational (weak) formulation of the mortar problem represents such a task: Find $u_h \in V^h$ which satisfies

$$a^\Gamma(u_h, v_h) = F^\Gamma(v_h) \quad \forall v_h \in V^h. \tag{18}$$

The existence and uniqueness of the solution of (18) follows i.a. from the Lax-Milgram lemma.

3.2 Mixed Formulation of the Mortar Problem

First, we present some findings relating to dual spaces. We know, that $V_i^h(\Omega_i) \subset W_2^1(\Omega_i)$. A restriction of a mortar function u to a nonmortar edge δ_l belongs to the space $W_2^{1/2}(\delta_l)$. The space of test functions $\Psi^h(\delta_l)$ can then be a subset of the dual space of space $W_2^{1/2}(\delta_l)$ with respect to the L^2 inner product, thus $\Psi^h(\delta_l) \subset W_2^{-1/2}(\delta_l)$.

For introduction of a mixed formulation we use the mortar condition (7), whose satisfaction is demanded on the interface. We denote $[u_l]$ the jump of $u_h \in V^h$ across δ_l. The test functions from the mortar condition can be considered as Lagrange multipliers. A function u belongs to the space V^h if and only if for all the nonmortar edges δ_l and for all the Lagrange multipliers μ_l, which form a basis of $\Psi^h(\delta_l)$, holds

$$\int_{\delta_l} [u_l]\mu_l \, \mathrm{d}s = 0. \tag{19}$$

From the first paragraph of this subsection results, that $[u_l] \in W_2^{1/2}(\delta_l)$. Lagrange multipliers μ_l must then be from the dual space $W_2^{-1/2}(\delta_l)$. Let $M^h = \prod_l \Psi^h(\delta_l) \subset \prod_l W_2^{-1/2}(\delta_l)$ and $\mu_h \in M^h$, where $\mu_h = (\mu_l)_{l=1:L}$, L is the number of nonmortar edges. We define a bilinear form

$$b(u_h, \mu_h) = \sum_{l=1}^{L} \int_{\delta_l} [u_h]\mu_l \mathrm{d}s. \tag{20}$$

A function u_h is a mortar function if and only if

$$b(u_h, \mu_h) = 0 \quad \forall \mu_h \in M^h. \tag{21}$$

We can rewrite the discrete problem (18) to the mixed formulation: Find a couple $(u_h, \lambda_h) \in V^h \times M^h$ which satisfies

$$\begin{aligned} a^\Gamma(u_h, v_h) \;+\; b(v_h, \lambda_h) \;&=\; F^\Gamma(v_h) \qquad &\forall v_h \in V^h, \\ b(u_h, \mu_h) \qquad\qquad &=\; 0 &\forall \mu_h \in M^h. \end{aligned} \tag{22}$$

As well as in other mixed formulations, it is important to satisfy the Brezzi-Babuška condition (see [7]) also in formulation (22). This is important for the existence of the solution and for the error estimate.

4 Implementation of the Mortar Finite Element Method

We describe briefly the key points of the implementation of the mortar finite element method in the Matlab system. We started from the program [4], which deals with the conforming partition of the computational domain into squares and rectangles, whose sides are parallel to the coordinate axes. The generalization of the program to the nonconforming case is not a trivial task. Our program solves the Poisson or Laplace equation in two dimensions with Dirichlet boundary condition and can deal with polygonal computational domains, that can be divided into general polygonal convex subdomains. Since the conforming partition is a specific case of the nonconforming partition, it's obvious, that the program can handle with conforming case also.

The input data are the geometry of the computational domain, the right hand side and the Dirichlet boundary condition. The geometrical data contain informations about the subdomains - each subdomain and its triangulations is inscribed with a triplet of matrices $\mathbf{p}$, $\mathbf{e}$, $\mathbf{t}$ representing matrices of points, edges and triangles. It's necessary to save the information on the mutual

relationships of the edges and subdomains. We must distinguish the edges on the boundary because of the fulfilment of the boundary condition and we need to know the adjacent edges and subdomains of each edge and subdomain.

4.1 Mutual Relationship of the Edges

So let us consider a polygonal computational domain Ω. We divide the domain into polygonal subdomains Ω_i. There are, in fact, two types of nonconforming partitions, see Figure 2. In the first case, the subdomains are squares and rectangles, whose sides are parallel to the coordinate axes. The second case is more general – the subdomains are general n-agles or rectangles, whose sides are not parallel to the coordinate axes. Both cases can be combined.

In the following text we will use the terms edge and main edge, see Section 2 for the explanation. The edges and their belongings to the main edges are described in matrices **e**. The comparison of the mutual relationships in the input geometry data is realized through the edges, not the main edges. We look for the parallelism of the edges. The parallelism is indicated by the identical multiples of x- and y- component of the directional vectors. If an edge is parallel to a coordinate axis, we can profit from the fact, that one component of its directional vector is equal to zero. We must identify the edges, that are overlapping. The necessary condition of the overlapping is, that the edges lie on the same line (a special case of parallelism). Two edges can overlap with parts of the edges, one edge can be inside the second one or they can coincide. The information on the mutual relationships of the edges and subdomains is stored in a matrix by a numerical value. On the basis of described information we can do the partition into mortar and nonmortar edges.

4.2 Division of the Edges into Mortar and Nonmortar

We divide the main edges, that are not lying on the outer boundary, into mortars and non-mortars. We have already mentioned, that the partition is always possible. But there is no universal instruction, how to choose the mortars. We introduce some possibilities. First of all, we focus on the conforming case with rectangular subdomains. Each main edge is either on the outer boundary or has one adjacent main edge with which it coincides. The choice of mortar is without any problems. Let's show two possibilities:

- Neumann-Dirichlet partition – Each subdomain has either all the main edges mortar or nonmortar.

- We sign the first two main edges of the subdomain as mortar, the second two as nonmortar. We omit the main edges on the boundary.

The situation is more complicated in the nonconforming case. Neither of the above-mentioned methods can be used. With regard to a lack of information in a literature we use our own way of a choice of the mortar edges.

We will do the following. We go through the particular subdomains in the same sequence as they were entered. If a main edge is unsigned yet and if it is possible, we sign the main edge as mortar and its adjacent main edge (or edges) as nonmortar. So we go through all the main edges of all the subdomains. If a main edge is once called mortar (or nonmortar), we don't change it again. The process of signing of the main edges is shown in Figure 7.

The information on the adjacent edges is known from the previous research and is stored in a matrix. On the basis of a detection of an overlap there is assigned a type of the appropriate edge and of the adjacent edge. It's necessary to verify if one of these two edges already has a value (mortar or nonmortar). In this case, we must keep the value and the type is assigned

only to the second edge. The edges, that belong to the same main edge, are of the same type. The type (mortar, nonmortar, outer boundary) is stored as a numerical parameter added in the matrices **e** (0 = outer boundary, 1 = nonmortar edge, 2 = mortar edge), see Figure 7.
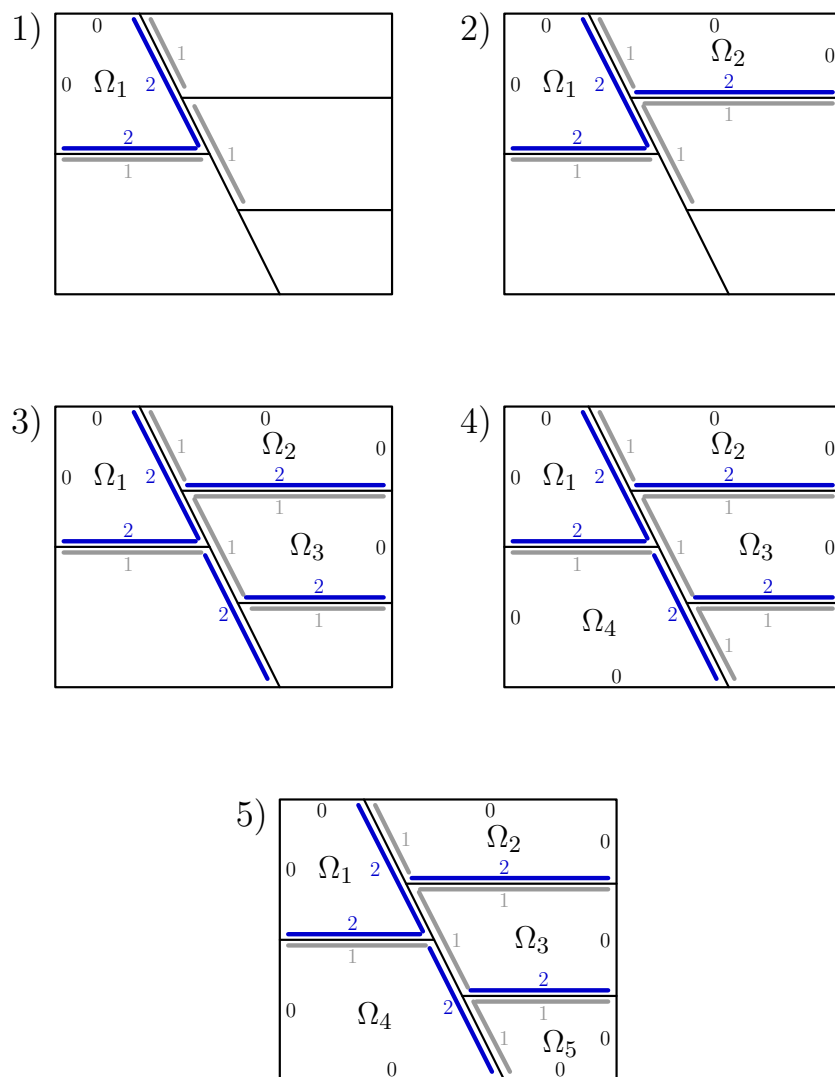

Figure 7: Process of signing of main edges (mortar edges are blue (2), nonmortar edges are grey (1) and 0 is for edges of outer boundary).

4.3 Assembling

We briefly describe the assembling of the stiffness matrix and the right hand side. We add in the assembling the requierement of the fulfilment of the mortar condition (see Section 2) also. First, we need to do some auxiliary steps. Let us consider all the nodes of all the subdomains. For further computation we divide the nodes into several groups - in the local meaning (within the subdomains) and also in global meaning (in terms of the whole computational domain). We divide the nodes into active and inactive, into interior and boundary, etc. We mention closely the global division of the nodes. We introduce a set of global active interior nodes, that contains interior nodes of all the subdomains, interior nodes of the mortar edges and all the corner nodes, that don't lie on the outer boundary. All the nodes lying on the outer boundary are called global active boundary nodes, etc. The sense of these sets will be clear later.

For each mortar (master) edge we assemble the so-called master matrix and for each

nonmortar (slave) edge the so-called slave matrix, that introduce the right and left hand side matrix in applying the mortar condition. In the computation of the master matrix we deal with the test functions on the nonmortar edge described in Section 2. Since the test functions have zero values at the end points of the nonmortar edges, it's necessary to have the nonmortar edges with at least three nodes (including the end nodes).

Each subdomain is first assembled in a usual way as in the case of the finite element method. The values on the subdomain edges are re-counted according to whether the appropriate main edge is mortar or nonmortar.

4.4 Solving the Resulting System of Equations

For solving the resulting system of equations, we use the conjugate gradient method with pre-conditioning, that is implemented in Matlab. This method is highly suitable for solving system of equations with symmetric positive definite and sparse matrix. The method is convenient for large matrices because of the iterative character of the method. The preconditioning accelerates the computation and improves stability.

The system of equations is solved only for the global active interior nodes. After the computation it's necessary to compute the solution on the outer boundary, where the fulfilment of the Dirichlet boundary condition is required, and the solution in the inactive (nonmortar) nodes, which is computed by the mortar projection.

5 Numerical Results

In this section, we present two practical examples showing both the principle and the possibility of practical use of the method. The examples are solved by our program for solving linear elliptic partial differential equations in 2D by the mortar finite element method.

As first example, we consider a Poisson problem on a general computational domain Ω (see Figure 8):

$$
\begin{aligned}
-\triangle u &= 30 \quad \text{on } \Omega, \\
u &= 0 \quad \text{on } \Gamma_0, \\
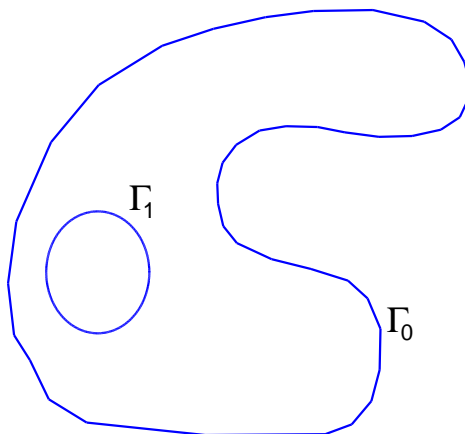u &= 1 \quad \text{on } \Gamma_1.
\end{aligned}
\tag{23}
$$

Figure 8: Computational domain Ω for problem (23).

The partition of the computational domain with the appropriate nonmatching mesh is on Figure 9 and the solution of the problem (23) is displayed in Figures 10 and 11. This example serves as an illustration of the principle of the mortar finite element method. There is shown the commonness of the usage of the method.
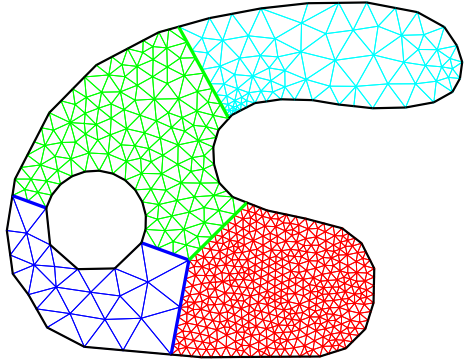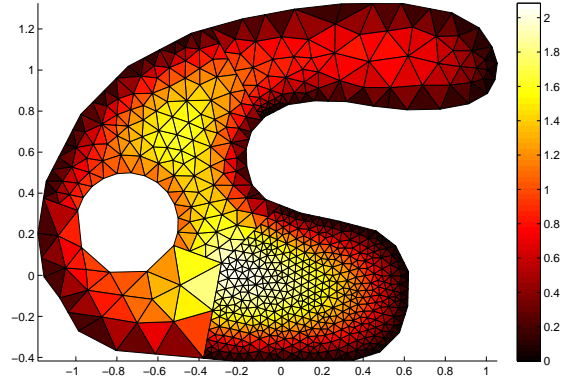

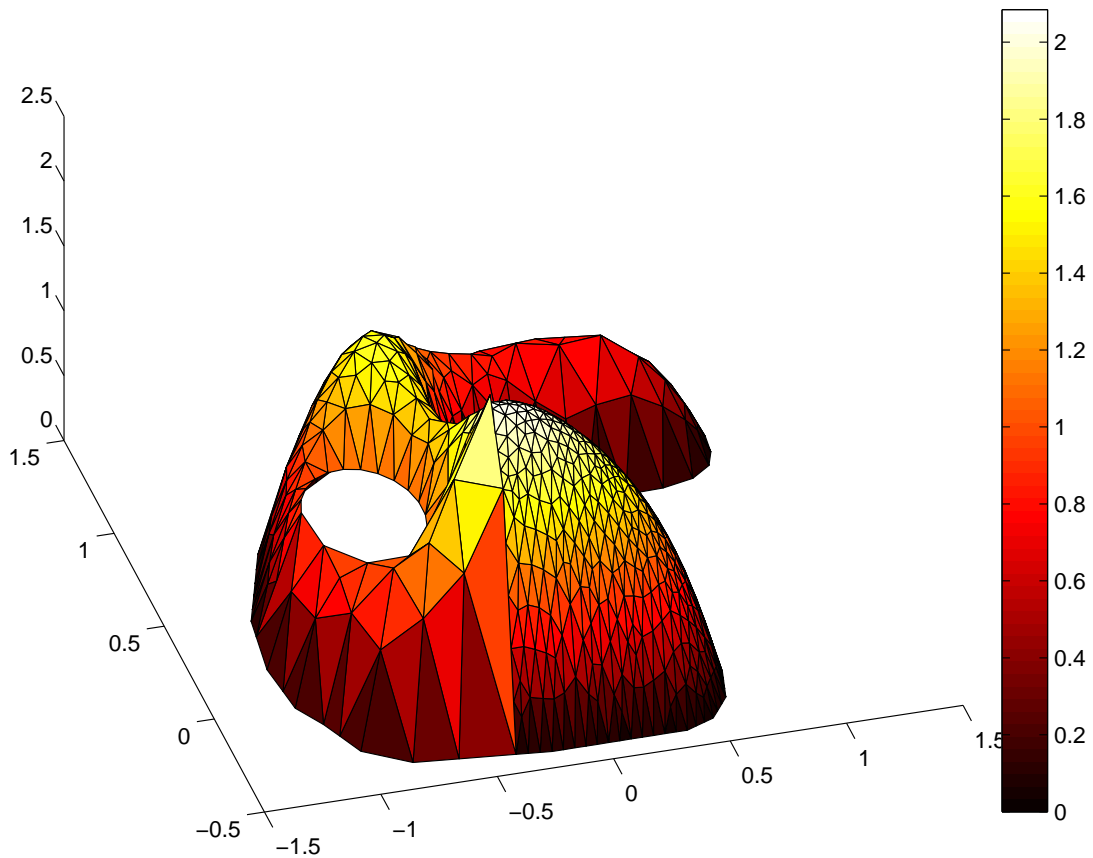Figure 9: Partition of Ω for problem (23).

Figure 10: Solution of (23).

Figure 11: Solution of (23).

As the second example, we consider a Laplace problem:

$$-\triangle u = 0 \quad \text{on } \Omega = \{(x,y) \in \mathbb{R}^2 : \frac{1}{4} < x^2 + y^2 < 1\},$$

$$u(\cos\varphi, \sin\varphi) = \frac{5\pi}{4} - \varphi, \quad \varphi \in (\frac{\pi}{4}, \frac{9\pi}{4}), \qquad (24)$$

$$u(\frac{1}{2}\cos\varphi, \frac{1}{2}\sin\varphi) = \frac{5\pi}{4} - \varphi, \quad \varphi \in (\frac{\pi}{4}, \frac{9\pi}{4}).$$

Figures 12(a),(b) show the partition of the computational domain with two versions of the appropriate mesh. The solution of the problem (24) is displayed in Figures 13(a),(b) and 14(a),(b). The usage of the mortar finite element method enables a choice of a mesh with respect to the discontinuity of the boundary condition. The subdomain with the jump can be meshed much finer than the other one.
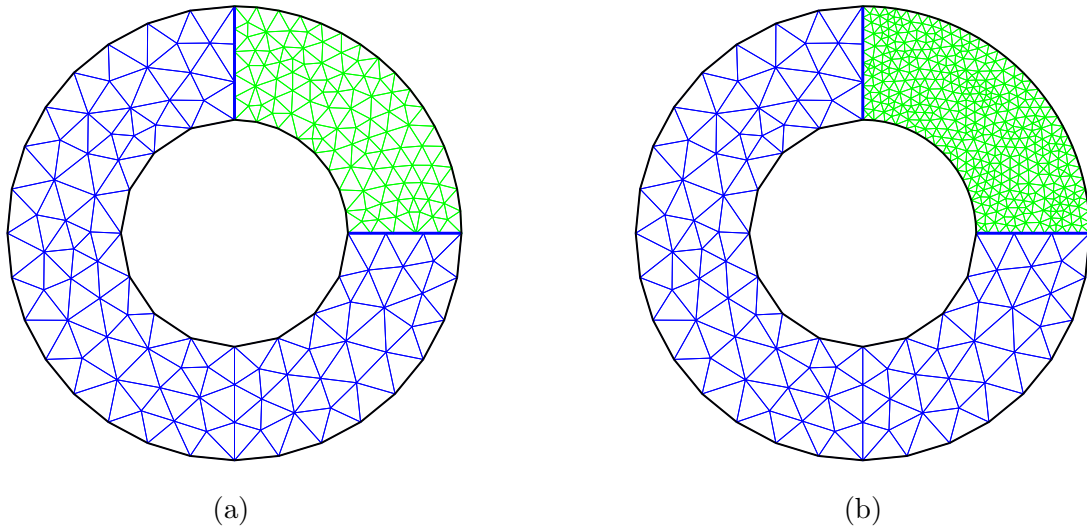

(a)   (b)

Figure 12: Partition of Ω and two versions of the appropriate mesh for problem (24).
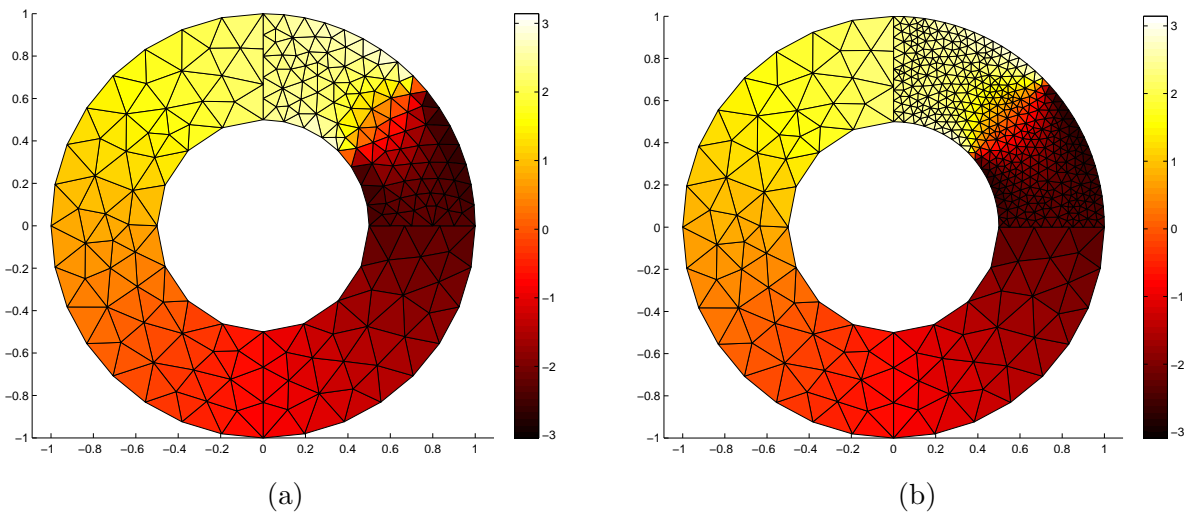

(a)   (b)
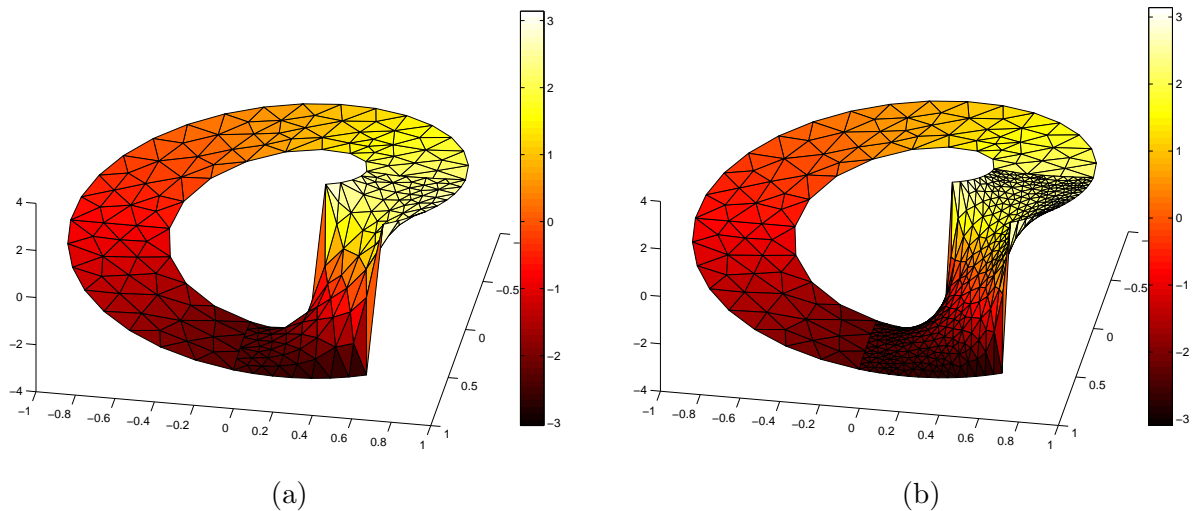
Figure 13: Solution of (24) for two versions of the appropriate mesh.

Figure 14: Solution of (24) for two versions of the appropriate mesh.

Acknowledgements

References

[1] Bernardi, C., Maday, Y., Patera, A. T.: *A new nonconforming approach to domain decompositon: The mortar element method*. Nonlinear partial diferential equations and their applications: Collège de France Seminar, 1994, Vol. 299, pages 13–51.

[2] Brezzi, F., Fortin, M.: *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, 1991. ISBN 0-387-97582-9.

[3] Kutáková, H.: *Mortar finite element method for linear elliptic problems in 2D*. Diploma Thesis (written in czech), Pilsen: University of West Bohemia, Fakulty of Applied Sciences, 2008.

[4] Rahman, T.: *Mortar FE on nonmatching grid*. Matlab File, University of Augsburg, 2002. From: ⟨ http://matlabdb.mathematik.uni-stuttgart.de/download.jsp?MC_ID=1&SC_ID=1 &MP_ID=21 ⟩.

[5] Smith, B., Bjørstad, P. E., Gropp, W.: *Domain Decomposition Parallel Multilevel Methods For Elliptic Partial Differential Equations*. Cambridge University Press (United Kingdom), 2004, ISBN 0-521-49589-X.

[6] Stefanica, D.: *Domain Decomposition Methods for Mortar Finite Elements*. Ph.D. Thesis, Courant Institute of Mathematical Sciences, New York University, 1999. From: ⟨ http://www.cs.nyu.edu/csweb/Research/TechReports/TR2000-804/TR2000-804.pdf ⟩.

[7] Wohlmuth, B.: *Discretization Methods and Iterative Solvers Based on Domain Decomposition*. Lectures Notes in Computational Science and Engineering, Vol 17. 2001.

Ing. Josef Daněk, Ph.D.
Department of Mathematics
University of West Bohemia
Univerzitní 22
306 14 Pilsen
Czech Republic
Email: danek@kma.zcu.cz

Ing. Hana Kutáková
MECAS ESI s.r.o.
Brojova 16
326 00 Pilsen
Czech Republic
Email: kutakova@volny.cz