PRZEGLĄD ELEKTROTECHNICZNY

Ukazuje się od 1919 roku 12b'12

Organ Stowarzyszenia Elektryków Polskich • Wydawnictwo SIGMA-NOT Sp. z o.o.

Dániel MARCSA, Miklós KUCZMANN

"Széchenyi István" University

Comparison of domain decomposition methods for elliptic partial differential problems with unstructured meshes

Abstract. *The paper presents the parallelisation of sequential (single-processor) finite element simulations with the use of domain decomposition methods. These domain decomposition methods are the Schur complement method and the Finite Element Tearing and Interconnecting (FETI) method. The execution time and speedup of these parallel finite element methods have been compared to each other and to the sequential one.*

Streszczenie. *W artykule zaprezentowano paralelizację sekwencyjnych (jednoprocesorowych) symulacji metodą elementów skończonych z wykorzystaniem metod dekompozycji obszaru. Metody te to metoda uzupełnienia Schura oraz metoda FETI (Finite Element Tearing and Interconnecting). Czas obliczeń i przyspieszenie paralelizowanych metod elementów skończonych zostały porównae między sobą oraz z procesem sekwencyjnym. (**Porównanie metod dekompozycji obszaru dla rozwiązywania równań różniczkowych cząstkowych eliptycznych z niestrukturalną siatką**)*

Keywords: Parallel finite element method, Domain decomposition, Schur complement method, FETI method.
Słowa kluczowe: paralelizowana metoda elementów skończonych, dekompozycja obszaru, metoda uzupełnień Schura, metoda FETI.

Introduction

Different applications of domain decomposition method [1, 2] have a long history in computational science. The reason for employing the substructuring technique was the small memory of computers. To solve large scale problems, a domain has been divided into subdomains that fit into the computer memory. However, the computer memory grows, the demand for solution of large real life problems is always ahead of computer capabilities. The large scale computations and simulations performed with finite element method (FEM) [3, 4] often require very long computation time. While limited progress can be reached with improvement of numerical algorithms, a radical time reduction can be made with multiprocessor computation. In order to perform finite element analysis a computer with parallel processors, computations should be distributed across processors.

The finite element method [3, 4] is an important technique for the solution of a wide range of problems in science and engineering. In electromagnetic computation, it is based on the weak formulation of the partial differential equations, which can be obtained by Maxwell's equations and the weighted residual method [3]. The most time consuming part of finite element computations is the solution of the large sparse system of equations. Therefore, the solution of a large system of equations must be parallelized in order to speedup the numerical computations [1].

Two non-overlapping domain decomposition methods, the Schur complement method [1, 2, 5] and the Finite Element Tearing and Interconnecting (FETI) method [2, 6, 7, 8] have been used after the partition of the finite element mesh into subregions or also called subdomains, and it reduces the large mass matrix into smaller but interconnecting parts. The independent subdomains, and the assembling of these equation systems can be handled by the independent processors of a supercomputer or by the independent computers of a computer grid, i.e. in a parallel way. Furthermore, after assembling, the system of linear equations has been solved in a parallel way, too.

The paper presents a parallel approach for the solution of two dimensional linear elliptic partial differential equation problems by parallel finite element method. These problems are case studies to show the steps of the Schur complement method [1, 2, 5], and of the Finite Element Tearing and Interconnecting method [2, 6, 7, 8] with parallel finite element technique. The comparison focused on the time, speedup and numerical behaviour of Schur method and FETI method.

Parallel Finite Element Method with Domain Decomposition

The parallel finite element based numerical analysis on supercomputers or on clusters of PCs (Personal Computers) need the efficient partitioning of the finite element mesh. This is the first and the most important step of parallelization with the use of domain decomposition methods.

The efficient mesh partitioning is necessary for the distributed computation, because each subdomain should contain approximately the same number of node points, i.e. the load balance of processors is nearly equal. When the parallel system includes p processors, usually the problem domain is partitioned into p subdomains. The number of subdomain elements assigned to each processors and the number of common elements assigned to different processors are minimized. These are important because of the load balance of the computations and the minimum communication among the processors.

Many domain decomposition or graph-partitioning algorithms can be found in the literature [1, 9]. Gmsh [10] combined with METIS algorithm [8] has been used for the discretization of the domain of problem and for the mesh partitioning. The parallel finite element program has been implemented in a MATLAB script [11].

The main idea of domain decomposition method is to divide the domain Ω into several subdomains in which the unknown potentials could be calculated simultaneously, i.e. parallel.

The general form of a linear algebraic problem arising from the discretization of a static field problem defined on the domain Ω can be written as [1, 2, 12]

(1) $$\mathbf{K}\mathbf{x} = \mathbf{b},$$

where $\mathbf{K} \in R^{n \times n}$ is the symmetric positive definite matrix, $\mathbf{b} \in R^n$ on the right hand side of the equations represents the excitation, and $\mathbf{x} \in R^n$ contains the unknown nodal potentials. Here *n* is a number of unknowns.

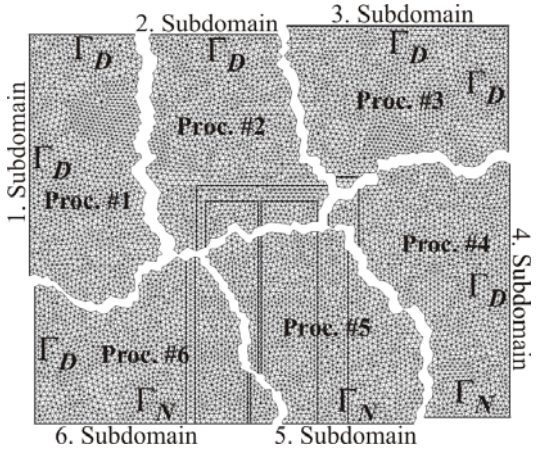

Fig.1. Partitioned two-dimensional problem

Schur Complement Method

The Schur complement method [1, 2, 5] was started to use many decades ago, and nowadays also a very popular non-overlapping domain decomposition technique among engineers.

After the mesh partitioning, equations in (1) has been split into particular blocks. In the case of Fig. 1, the j^{th} processor contains blocks ($j = 1,\ldots,6$) [2, 5]

(2) $$\begin{bmatrix} \mathbf{K}_{jj} & \mathbf{K}_{j\Gamma_j} \\ \mathbf{K}_{\Gamma_j j} & \mathbf{K}_{\Gamma_j\Gamma_j} \end{bmatrix} \begin{bmatrix} \mathbf{x}_j \\ \mathbf{x}_{\Gamma_j} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_j \\ \mathbf{b}_{\Gamma_j} \end{bmatrix},$$

where $\mathbf{K}_{jj}$ is the symmetric positive definite submatrix of the j^{th} subdomain, $\mathbf{b}_j$ is the vector of right hand side defined inside the subdomain. The Γ_j is the interface boundary index of the j^{th} subdomain. The submatrix $\mathbf{K}_{j\Gamma_j}$ contains the nodal value of j^{th} subdomain, which connect to the interface boundary nodes of that region. The $\mathbf{K}_{j\Gamma_j}$ of the upper part of the matrix is the transpose of $\mathbf{K}_{\Gamma_j j}$ of lower part of the matrix. The $\mathbf{K}_{\Gamma_j\Gamma_j}$ and $\mathbf{b}_{\Gamma_j}$ expresses the coupling of the interface unknowns.

Each subdomain will be allocated to an independent processor, because the submatrix $\mathbf{K}_{jj}$ with the $\mathbf{K}_{j\Gamma_j}, \mathbf{K}_{\Gamma_j j}$ and the right-hand side $\mathbf{b}_j$ are independent, i.e. these can be handled in a parallel way. Only the $\mathbf{K}_{\Gamma_j\Gamma_j}$ and $\mathbf{b}_{\Gamma_j}$ are not independent, these submatrices and vectores are stored on the distributed memories. The $\mathbf{K}_{\Gamma\Gamma}$ and $\mathbf{b}_\Gamma$ are the sum of $\mathbf{K}_{\Gamma_j\Gamma_j}$ and $\mathbf{b}_{\Gamma_j}$, where *j* is the index of sub-domains.

After some algebraic manipulations, the unknowns in $\mathbf{x}_\Gamma$ can be calculated by the solution of [2, 5]

(3) $$\left[\mathbf{K}_{\Gamma\Gamma} - \sum_{j=1}^{N_S} \mathbf{K}_{\Gamma_j j}\mathbf{K}_{jj}^{-1}\mathbf{K}_{j\Gamma_j}\right]\mathbf{x}_\Gamma = \mathbf{b}_\Gamma - \sum_{j=1}^{N_S} \mathbf{K}_{\Gamma_j j}\mathbf{K}_{jj}^{-1}\mathbf{b}_j$$

where the term inside the bracket is the so called Schur complement, and N_S is the number of subdomain. The original system of equations contains *n* unknowns, while the reduced system (3) contains only the unknowns of $\mathbf{x}_\Gamma$. This reduction of unknowns is an important feature of the Schur complement method. Equation (3) is also called the coarse grid problem, because only the unknowns of interior boundaries are used.

The inverse of the matrix $\mathbf{K}_{jj}$ is needed to compute the corresponding subdomain of the solution vector. However, matrix $\mathbf{K}_{jj}$ never inverted explicitly in practical computing, because it is very time consuming. Instead of an inverse matrix, the incomplete LU-factorization [12] has been used here.

The unknowns of all the other subdomains $\mathbf{x}_j$ can be calculated simultaneously [2, 5] i.e.

(4) $$\mathbf{K}_{jj}\mathbf{x}_j = \mathbf{b}_j - \mathbf{K}_{j\Gamma_j}\mathbf{x}_\Gamma,$$

where $j = 1,\ldots,6$ in the case of Fig. 1.

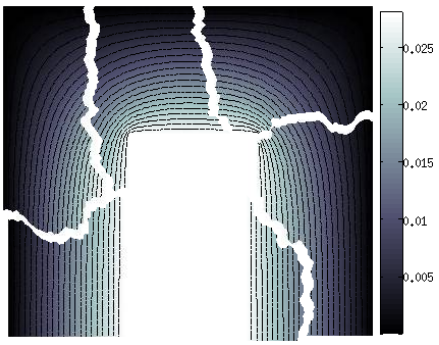

Fig.2. The three sub-solution of the partitioned problem

The possibility of parallel computation can be decreased the computation time. The assembly of the submatrices can be performed parallel by independent processors. However, for the solution of equation (3) use the submatrices from the independent processors. After obtaining $\mathbf{x}_\Gamma$, it must be sent back to the independent processors to calculate the subsolutions by equation (4). If the problem is large enough, the data exchange is a small amount while solving the problem.

In this paper, the problems are quiet small examples, this is why a direct solver, the parallel forward-backward algorithm [1] has been used to solve the equation in (4).

The six subsolutions can be calculated as it illustrated in Fig. 2. This figure shows the potential distribution and the equipotential lines of the single-phase transformer.

Finite Element Tearing and Interconnecting

The Finite Element Tearing and Interconnecting (FETI) method was introduced by Farhat and Roux in reference [6]. In the last decade, the FETI method [2, 6, 7, 8] has seemed as one of the most powerful and the most popular solvers for numerical computation.

If domain Ω is partitioned into a set of N_S disconnected subdomains (Fig. 1), the FETI method consists in replacing equation in (1) with the equivalent system of substructure equations (where $j = 1,\ldots,N_S$) [2, 6, 7, 8]

$$\mathbf{K}_j \mathbf{x}_j = \mathbf{b}_j - \mathbf{B}_j^{\mathrm{T}} \boldsymbol{\lambda}, \tag{5}$$

$$\sum_{j=1}^{N_S} \mathbf{B}_j \mathbf{x}_j = \mathbf{0}, \tag{6}$$

where $\mathbf{K}_j$ is the j^{th} subdomain mass matrix, $\mathbf{b}_j$ is the j^{th} vectors of right-hand side, $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers [2, 6] introduced for enforcing the constraint (6) on the subdomain interface boundary Γ_j, and $\mathbf{B}_j$ is a signed ($\pm$) Boolean mapping matrix, which used to express the compatibility condition at subdomain interface Γ_j. The superscript T denotes the transpose.

Usually, the partitioned problem may contain $N_f \le N_S$ floating subdomain, where matrices $\mathbf{K}_j$ from being singular. The floating subdomain is a subdomain without enough Dirichlet boundary conditions. In Fig. 1, the Subdomain 5 is a floating subdomain, because the outer boundary is not Dirichlet boundary condition (Γ_D), but Neumann boundary condition (Γ_N). In this case N_f of local Neumann problems (Eq. (6)) are ill-posed. To guarantee the solvability of these problems, we require that [2, 6]

$$\left(\mathbf{b}_j - \mathbf{B}_j^{\mathrm{T}} \boldsymbol{\lambda}\right) \perp \mathrm{Ker}\left(K_j\right), \tag{7}$$

and compute the solution of equation in (5) as [2, 6, 7, 8]

$$\mathbf{x}_j = \mathbf{K}_j^{+}\left(\mathbf{b}_j - \mathbf{B}_j^{\mathrm{T}} \boldsymbol{\lambda}\right) + \mathbf{R}_j \boldsymbol{\alpha}_j, \tag{8}$$

where $\mathbf{K}_j^{+}$ is a pseudo-inverse of $\mathbf{K}_j$, $\mathbf{R}_j = \mathrm{Ker}\left(\mathbf{K}_j\right)$ is the null space [12] of $\mathbf{K}_j$, and $\boldsymbol{\alpha}_j$ is the set of amplitudes that specifies the contribution of the null space $\mathbf{R}_j$ to the solution $\mathbf{x}_j$. Instead of a pseudo-inverse of matrix, the Moore-Penrose matrix inverse [12] has been used here. The introduction of the $\boldsymbol{\alpha}_j$ is compensated by the additional equations resulting from (4) [2, 6, 7, 8]

$$\mathbf{R}_j^{\mathrm{T}}\left(\mathbf{b}_j - \mathbf{B}_j^{\mathrm{T}} \boldsymbol{\lambda}\right) = \mathbf{0}. \tag{9}$$

Substituting equation (8) into the equation (6) and exploiting the solvability condition (9) leads after some algebraic manipulations to the following FETI interface problem [2, 6, 7, 8]

$$\begin{bmatrix} \mathbf{F}_I & -\mathbf{G}_I \\ -\mathbf{G}_I^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ -\mathbf{e} \end{bmatrix} \tag{10}$$

where [2, 6, 7, 8]

$$\begin{aligned} \mathbf{F}_I &= \sum_{j=1}^{N_S} \mathbf{B}_j \mathbf{K}_j^{+} \mathbf{B}_j^{\mathrm{T}}, \\ \mathbf{G}_I &= \left[\mathbf{B}_1 \mathbf{R}_1 \ \ldots \ \mathbf{B}_{N_S} \mathbf{R}_{N_S}\right], \\ \mathbf{d} &= \sum_{j=1}^{N_S} \mathbf{B}_j \mathbf{K}_j^{+} \mathbf{b}_j, \\ \mathbf{e} &= \left[\mathbf{b}_1^T \mathbf{R}_1 \ \ldots \ \mathbf{b}_{N_S}^T \mathbf{R}_{N_S}\right]^{\mathrm{T}}. \end{aligned} \tag{11}$$

In order to solve equation in (10) for the Lagrange multiplier vector $\boldsymbol{\lambda}$, the following splitting of $\boldsymbol{\lambda}$ is performed [2, 6, 7, 8]

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_0 + \mathbf{P}(\mathbf{Q})\bar{\boldsymbol{\lambda}}, \tag{12}$$

where $\boldsymbol{\lambda}_0 = \mathbf{Q}\mathbf{G}_I\left(\mathbf{G}_I^T \mathbf{Q}\mathbf{G}_I\right)^{-1}\mathbf{e}$, which is a particular solution of $\mathbf{G}_I^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{e}$, and $\mathbf{P}(Q)$ is a projection operator [2], which is for any matrix $\mathbf{Q}$, $\mathbf{G}_I^{\mathrm{T}}\mathbf{P}(\mathbf{Q}) = \mathbf{0}$ by $\mathbf{P}(Q) = \mathbf{I} - \mathbf{Q}\mathbf{G}_I\left(\mathbf{G}_I^{\mathrm{T}}\mathbf{Q}\mathbf{G}_I\right)^{-1}\mathbf{G}_I^{\mathrm{T}}$. In this paper, $\bar{\boldsymbol{\lambda}} = \mathbf{F}_I^{+}\mathbf{d}$ and $\mathbf{Q} = \mathbf{F}_I^{+}$ choosen, based on reference [8]. After some algebraic manipulations equation in (12) leads to the following equation

$$\boldsymbol{\lambda} = \mathbf{F}_I^{+}\left(\mathbf{d} + \mathbf{G}_I \boldsymbol{\alpha}\right), \tag{13}$$

where $\boldsymbol{\alpha} = -\left(\mathbf{G}_I^T \mathbf{F}_I^{+} \mathbf{G}_I\right)^{-1}\left(\mathbf{G}_I^T \mathbf{F}_I^{+} \mathbf{d} - \mathbf{e}\right)$.

The interface problem (10) is the best solved by an iterative algorithm [2, 6, 7, 8]. However, in this paper a direct solver has been used by the above mentioned parameters.

Test Problems

Two test problems have been used for the comparison, which can be seen in Fig. 3. The first benchmark is a single-phase transformer, and the second one is a parallel-plate capacitor [5].

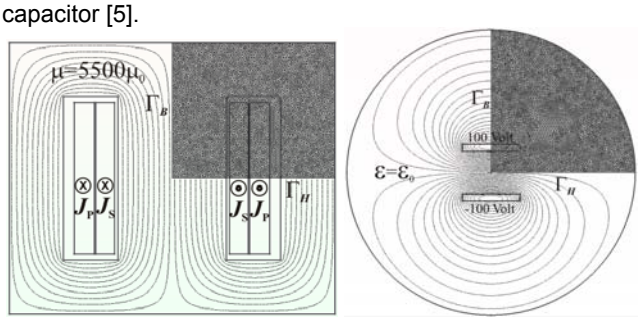

Fig.3. The test problems.

The choosen test problems are static problems, where the partial differential equations are elliptic type [3, 5]. The 2D problem is dicretized by triangle elements and linear nodal shape functions have been used for the test problems. These problems are enough to simulate the quarter of the problem because of symmetry.

Results and Discussion

The computations have been carried out on a massively parallel computer (SUN Fire X2250). This computer works with a shared memory topology. The parallel programs have been implemented under the operating system Linux.

Table 1, Table 2 and Table 3 presents the comparison of domain decomposition methods at three different mesh sizes. The 23110 and 45967 number of unknowns (DOF) problems are the static magnetic field problem with. the single-phase transformer. The 37661 DOF problem is the electrostatic problem with the parallel-plate capacitor. In the tables, N_P is the number of processors and N_{DOF} is the number of unknowns on each sub-domain.

The times of sequential computation of 23110, 37661 and 45967 problems are 31.272 sec, 87.9131 sec and 118.5621 sec, respectively.

The times in the brackets at FETI method is the computation time of the full problem with a same mesh size as the quarter one. If the full problem is calculated, the problem does not contain floating sub-domain, because the outer boundaries are Dirichlet boundary condition.

Fig. 4 shows the speedup tests of the function of the number of the applied processors. The speedup of problems clearly show, if the size of full problem increased the speedup of FETI method (FETI-full) is also increased. However, this is not true at FETI-quarter, i.e. when the problem contains floating subdomain. The speedup of Schur method is increased when the number of processors are increased.

Table 1. Performance comparison of Schur and FETI methods at 23100 DOF.

N_P	N_{DOF}	Schur	FETI
		Time [sec]	Time [sec]
2	11633	31.8679	19.1888
3	7804	14.4887	11.2188
4	5853	9.1514	7.7721
5	4703	6.5774	6.7625
6	3930	5.4711	5.8121
7	3389	4.8756	16.4467 (4.8933)
8	2965	4.5103	14.6080 (5.3837)

Table 2. Performance comparison of Schur and FETI methods at 37661 DOF.

N_P	N_{DOF}	Schur	FETI
		Time [sec]	Time [sec]
2	19014	83.3859	48.7743
3	12713	43.0057	26.3409
4	9613	29.8087	18.0200
5	7671	23.8673	14.3775
6	6402	16.2492	57.7801 (12.2929)
7	5490	13.8095	47.0895 (9.6963)
8	4865	10.8847	43.2187 (9.4909)

Table 3. Performance comparison of Schur and FETI methods at 45967 DOF.

N_P	N_{DOF}	Schur	FETI
		Time [sec]	Time [sec]
2	23091	106.1977	68.2186
3	15429	68.8367	37.9652
4	11577	50.3323	23.6598
5	9290	32.5471	18.2715
6	7752	24.6692	16.3725
7	6671	20.9650	72.7768 (13.8424)
8	5847	17.6793	56.9673 (11.9318)

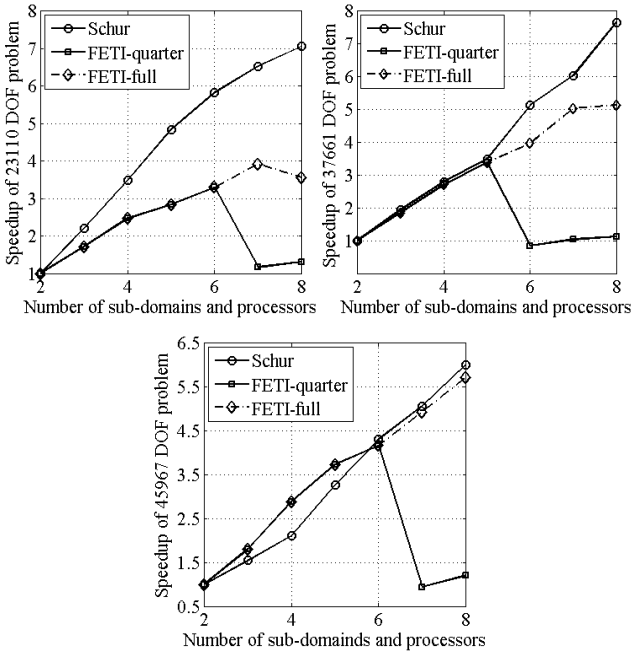

Fig.4. Speedup tests of static field problems.

Conclusion

One electrostatic and one static magnetic field problems have been solved by parallel finite element method. The parallel finite element program with Schur complement method and finite element tearing and interconnecting method works properly, because the time is decreased when the number of processors is increased. The speedup achieved over 7-fold and 5-fold speedup by 8 processors at the Schur and the FETI methods, respectively.

The aim of future research is to solve more complex, large two dimensional and three dimensional problems, and to realize preconditioned iterative solvers for FETI method, which handle the sub domains with Neumann boundary condition.

Acknowledgments

This paper was supported by the Interdisciplinary Doctoral School of Engineering of "Széchenyi István" University, and by the "Széchenyi István" University (15-3210-02) and by the TáMOP 4.2.2.A.

REFERENCES

[1] Magoulés F., *Mesh Partitioning Techniques and Domain Decomposition Method*, Kippen, Stirling, Scotland, Saxe-Coburg Publication, 2007.

[2] Kruis J., *Domain Decomposition Methods for Distributed Computing*, Kippen, Stirling, Scotland, Saxe-Coburg Publication, 2006.

[3] Kuczmann M., Iványi A., *The Finite Element Method in Magnetics*, Budapest: Academic Press, 2008.

[4] Šturmberger, B., Šturmberger, G., Hadžiselimović, M., Marčič, T., Virtič, P., Trlep, M., Goričan, V., Design and Finite-element Analysis of Interior Permanent Magnet Synchronous Motor with Flux Barriers, *IEEE Transactions on Magnetics*, 44 (2008), nr. 11, 4389-4392.

[5] Marcsa D., Kuczmann M., Parallel Solution of Electrostatic and Static Magnetic Field Problems by Domain Decomposition Method, *Przeglad Elektrotechniczny*, printing (2001).

[6] Farhat C., Roux F. X., A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm, *International Journal for Numerical Methods in Engineering*, 32 (1991), nr. 6, 1205–1227.

[7] Fragakis Y., Papadrakakis M., The Mosaic of High Performance Domain Decomposition Methods for Structural Mechanics: Formulation, Interrelation and Numerical Efficiency of Primal and Dual Methods, *Computer Methods in Applied Mechanics and Engineering*, 192 (2003), nr. 35-36, 3799–3830.

[8] Gosselet P., Rey C., Rixen D. J., On the Initial Estimate of Interface Forces in FETI Methods, *Computer Methods in Applied Mechanics and Engineering*, 192 (2003), nr. 25, 2749–2764.

[9] `http://glaros.dtc.umn.edu/gkhome/views/metis`. (Last visited 27 August 2012)

[10] Geuzaine C., Remacle J. F., Gmsh: A Three-Dimensional Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities, *International Journal for Numerical Methods in Engineering*, 79 (2009), nr. 11, 1309-1331.

[11] `http://www.mathworks.com` (Last visited 20 June 2012)

[12] Saad Y., *Iterative Methods for Sparse Linear Systems*, 3rd edition, SIAM, 2003.

***Authors**: Dániel Marcsa, MSc "Széchenyi István" University, Department of Automation, Laboratory of Electromagnetic Field, Egyetem tér 1, H-9026 Győr, Hungary, E-mail: marcsad@sze.hu.; Prof. Dr. Habil. Miklós Kuczmann, "Széchenyi István" University, Department of Automation, Laboratory of Electromagnetic Field, Egyetem tér 1, H-9026 Győr, Hungary, E-mail: kuczmann@sze.hu.*