Dániel MARCSA, Miklós KUCZMANN

"Széchenyi István" University

Parallel solution of electrostatic and static magnetic field problems by domain decomposition method

Abstract*. The paper presents a parallel approach for the efficient solution of a one-dimensional and a two-dimensional problem by parallel finite element method. These problems are case studies. The non-overlapping domain decomposition method has been used to cut the problem into sub-regions or also called sub-domains, and it reduces the large mass matrix into smaller parts. The independent sub-domains, and the assembling of these equation systems can be handled by the independent processors of a supercomputer, i.e. in a parallel way. The execution time and speedup of parallel finite element method have been compared to the serial one.*

Streszczenie. *W artykule opisano metodę efektywnego rozwiązywania problemów jedno- i dwuwymiarowych, poprzez równoległe analizy metodą elementów skończonych. Analizowany obiekt jest dzielony na podregiony co zmniejsza rozmiary jego macierzy i dzieli ją na mniejsze (poddziedziny). Te z kolei mogą być obliczane przez niezależne procesory superkomputera. (**Metoda równoległego rozwiązywania pól elektrostatycznych i magnetycznych – dekompozycja dziedziny**).*

Keywords: Finite element method, Parallel computing, Domain decomposition, Schur complement method.
Słowa kluczowe: metoda elementów skończonych, obliczanie równoległe, dekompozycja dziedziny, metoda dopełnień Schura.

Introduction

The finite element method (FEM) [1]-[3] is an important technique for the solution of a wide range of problems in science and engineering. It is based on the weak formulation of the partial differential equations, which can be obtained by the Maxwell's equations and the discretization of the analyzed problems geometry. The most time consuming part in finite element computation 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 FEM computation.

Over the past decade, high-performance computing has been achieved via multiprocessing. In a massively parallel environment, traditional sequential algorithms will not necessarily scale and can leads to a very poor utilization of the multiprocessor's architecture. As a result, specialized algorithms that directly exploit the parallel architecture must be developed. For the solution of sparse matrices, parallel algorithms based on domain decomposition method [4] - [8].

The non-overlapping domain decomposition method [4] - [8] has been used to cut the finite element mesh into sub-regions or also called sub-domains (see in Fig. 1) and it reduces the large mass matrix into smaller parts. The independent sub-domains, 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 the assembling, the systems of linear equation have also been solved in parallel way.

The paper presents a parallel approach for the efficient solution of a one-dimensional and a two-dimensional problems by parallel finite element method. These problems are case studies to show the steps of the Schur complement method [4] - [8] with parallel finite element technique.

Test Problems

The paper presents the steps of parallel finite element method through two simple problems, which can be seen in Fig. 2. The first benchmark is a parallel-plate capacitor, which is a electrostatic field problem. The second one is a quarter of the single-phase transformer, which is a static magnetic field problem. The detailed description of the problems you can find in [8] and [9].

The chosen test problems are static problems, where the partial differential equations are the Laplace-Poisson equation [1], [7], [8]. The 2D problem is dicretized by triangle elements and linear nodal shape functions are used for the test problems.

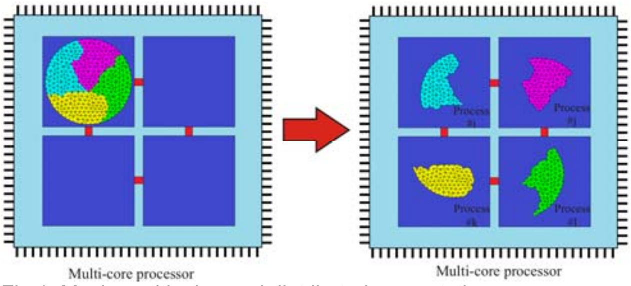

Fig.1. Mesh partitioning and distributed computation

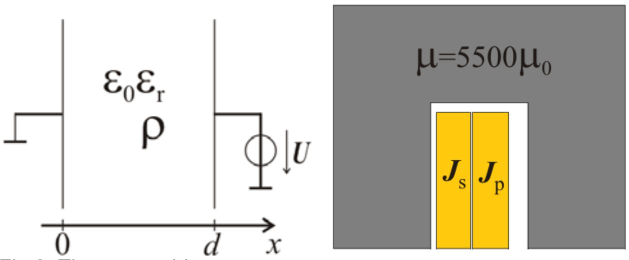

Fig.2. The test problems.

Parallel Finite Element Method with Domain Decompositon

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 parallel finite element method.

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

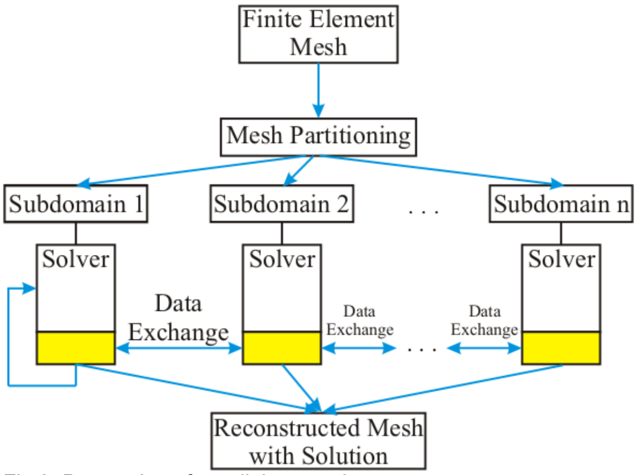

Fig.3. Processing of parallel computing

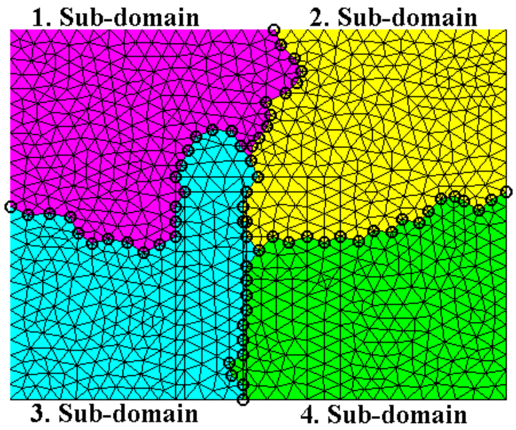

Fig.4. Partitioned two-dimensional problem

Many domain decomposition or graph-partitioning algorithms can be found in the literature [4], [10], [11]. The METIS algorithm [10] has been used in 2D case, and an own code has been used in 1D, because it is very easy to realize. The parallel finite element program has been implemented in a MATLAB script [12]. The main processes of parallel FEM can be seen in Fig. 3.

Domain Decomposition Method

The main idea of domain decomposition method is to divide the domain Ω into several sub-domains in which the unknown potentials could be calculated simultaneously, i.e. parallel. In this paper the Schur complement method (sub-structuring method) [4] - [8] has been used.

The system of linear algebraic equation can be written as

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

where $\mathbf{K} \in R^{n\times n}$ is the symmetric mass 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.

Equations in (1) has a special block structure when applying the mesh partitioning technique, for example equation (1) has the following form in the case of Fig. 4

(2) $$\begin{bmatrix} \mathbf{K}_{11} & 0 & 0 & 0 & \mathbf{K}_{15} \\ 0 & \mathbf{K}_{22} & 0 & 0 & \mathbf{K}_{25} \\ 0 & 0 & \mathbf{K}_{33} & 0 & \mathbf{K}_{35} \\ 0 & 0 & 0 & \mathbf{K}_{44} & \mathbf{K}_{45} \\ \mathbf{K}_{51} & \mathbf{K}_{52} & \mathbf{K}_{53} & \mathbf{K}_{54} & \mathbf{K}_{55} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{bmatrix}.$$

In equation (2), $\mathbf{K}_{11}$, $\mathbf{K}_{22}$, $\mathbf{K}_{33}$ and $\mathbf{K}_{44}$ are the symmetric positive definite sub-matrices of the four sub-domain (Fig. 4), $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ and $\mathbf{b}_4$ are the vectors of right hand sides defined inside the four sub-domain. The sub-matrix $\mathbf{K}_{i5}$ contains the nodal value of i^{th} sub-domain, which is connected to the adjacent boundary nodes of that region. The adjacent boundary nodes (interior boundaries) have been denoted by black circle in Fig. 4. The $\mathbf{K}_{i5}$ of the upper part of the matrix is the transpose of $\mathbf{K}_{5i}$ of lower part of the matrix. The sub-matrix $\mathbf{K}_{55}$ is the interaction coefficient between degree of freedom (DOF) attached to nodes located on the interior boundaries. The same is true for the right hand side subpart $\mathbf{b}_5$.

Each sub-domain will be allocated to an independent processor, because the sub-matrices $\mathbf{K}_{11}$, $\mathbf{K}_{22}$, $\mathbf{K}_{33}$, $\mathbf{K}_{44}$ with the $\mathbf{K}_{5i}$, $\mathbf{K}_{i5}$ and the right-hand side $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ and $\mathbf{b}_4$ are independent, i.e. these can be handled in a parallel way. Only the $\mathbf{K}_{55}$ and $\mathbf{b}_5$ are not independent, these sub-matrices are stored on the distributed memories. The i^{th} processor handled only the i^{th} sub-domain data, which leads to the following mass matrix and right-hand side vector:

(3) $$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{i5} \\ \mathbf{K}_{5i} & \mathbf{K}_{55}^{(i)} \end{bmatrix} \text{ and } \begin{bmatrix} \mathbf{b}_{\mathrm{i}} \\ \mathbf{b}_5^{(i)} \end{bmatrix}.$$

The block $\mathbf{K}_{55}$ and $\mathbf{b}_5$ are the sum of $\mathbf{K}_{55}^{(i)}$ and $\mathbf{b}_5^{(i)}$, where i is the index of sub-domains.

After some algebraic manipulation, the unknowns in $\mathbf{x}_5$ can be calculated by the solution of [6] - [8]

(4) $$\left[\mathbf{K}_{55} - \sum_{i=1}^{4} \mathbf{K}_{5i}\mathbf{K}_{ii}^{-1}\mathbf{K}_{i5}\right]\mathbf{x}_5 = \mathbf{b}_5 - \sum_{i=1}^{4} \mathbf{K}_{5i}\mathbf{K}_{ii}^{-1}\mathbf{b}_i$$

where the term inside the bracket is the so called Schur complement. The original system of equations contains n unknowns, while the reduced system (4) contains only the unknowns of $\mathbf{x}_5$. This reduction of unknowns is an important feature of the Schur complement method. Equation (4) is also called the coarse grid problem, because only the unknowns of interior boundaries are used.

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

The unknowns of all the other sub-domains $\mathbf{x}_i$ can be calculated simultaneously [6] - [8] i.e.

(5) $$\mathbf{K}_{ii}\mathbf{x}_i = \mathbf{b}_i - \mathbf{K}_{i5}\mathbf{x}_5\,,$$

where i = 1,2,3,4 in the case of Fig. 4.

The possibility of parallel computation can be decreased the computation time. The assembly of the sub-matrices can be performed parallel by independent processors. However, for the solution of equation (4) needs the sub-matrices from the independent processors. After obtaining $\mathbf{x}_5$, it must be sent back to the independent processors to calculate the sub-solutions by equation (5). If the problem is large enough, the data exchange is a small amount while solving the problem.

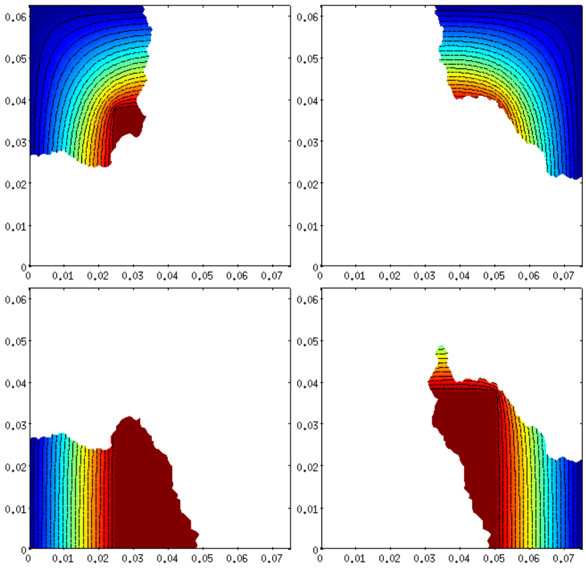

Fig.5. The four sub-solution of the partitioned problem

In this case study, the problems are quiet small examples. This is why a direct method has been used to solve the system of equations.

The four sub-solutions can be calculated as it illustrated in Fig. 5. This figure shows the potential distribution and the equipotential lines of the single-phase transformer.

Results and Discussion

The computations have been carried out on a SUN Fire X2250 computer with a following data. CPU: 2x Quad-Core Intel Xeon L5420 @ 2.5GHz; RAM: 8x 4GB DDR2 ECC 800MHz; HDD: 2x SAMSUNG HD502IJ 500GB SATA II RAID1. This computer works with a shared memory topology. The parallel program has been implemented under the operating system Linux.

Table 1 and Table 2 presents the computation time for different mesh size and number of processors. The regions of problems have been discretized using two fine meshes to increase the number of unknowns (DOF), i.e. the size of the problem. The finite element mesh of 1D problem consists of 50000 and 90000 linear line elements. The finite element mesh of 2D problem consists of 23110 and 45967 linear triangle elements. In these tests, the mesh has been divided into number of processors part. The Serial row shows the reference solution, but the second time of 1D problem is missing, because in this case out of memory has appeared.

In the case of the 1D and 2D problems the computation time is decreased when the number of processors is increased. When increase the number of processors, the communication time between processors has also increasing. Furthermore, the computation costs are increased because the number of internal boundary nodes is increased, but if the problem is large enough, the computation time in Eq. (5) is decreased.

Table 1. Time of the solution of the one-dimensional problem

	Number of Processors	DOF	
		50000	90000
Serial	1	55.57 sec	-
Parallel	2	16.25 sec	49.69 sec
	3	8.9 sec	25.56 sec
	4	5.45 sec	14.65 sec
	5	4.37 sec	11.02 sec
	6	3.48 sec	8.21 sec
	7	3.25 sec	6.84 sec
	8	3.27 sec	6.48 sec

Table 2. Time of the solution of the two-dimensional problem

	Number of Processors	DOF	
		23110	45967
Serial	1	344.17 sec	1652.24 sec
Parallel	2	215.48 sec	1099.22 sec
	3	129.85 sec	434.829 sec
	4	53.6 sec	268.634 sec
	5	31.723 sec	182.385 sec
	6	24.503 sec	123.882 sec
	7	19.861 sec	89.267 sec
	8	17.14 sec	78.403 sec

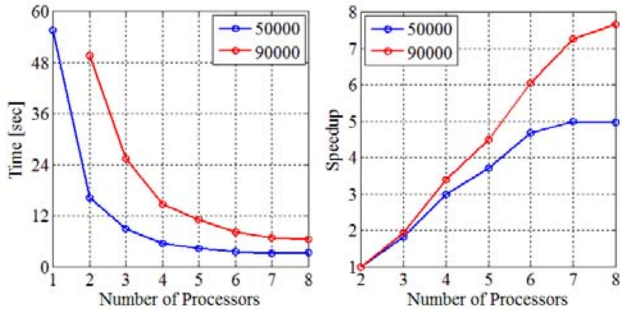

Fig.6. Comparison of different solutions the function of the number of the applied processors in 1D case

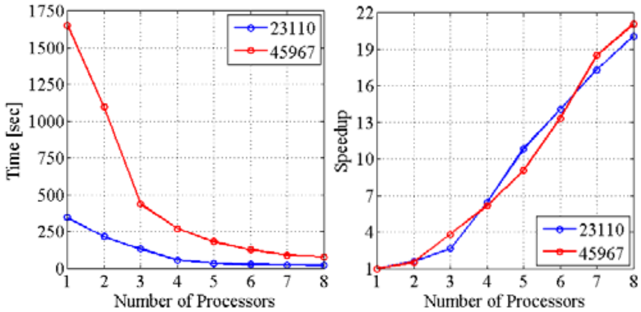

Fig.7. Comparison of different solutions the function of the number of the applied processors in 2D case

The computation time and the speedup ratio the function of the number of the applied processors can be seen in Fig. 6 and Fig. 7. The optimal number of processors for the one-dimensional 50000 DOF problem is 7. Therefore, an additional increase in the number of processors does not lead to a faster solution or a correspondingly shorter elapsed time. In the case of 1D 90000 DOF problem and the two-dimensional problems, the time is decreased and the speedup is increased with 8 processors.

Conclusion

Two very simple electrostatic and static magnetic field problems have been solved by parallel finite element method. The parallel finite element program with Schur complement method works properly, because the time is decreased when the number of processors is increased. The presented method achieved over 20-fold speedup by 8 processors at the one-dimensional and the two-dimensional problem, respectively.

The aim of future research is to solve more complex, large two-dimensional and three-dimensional problems with other domain decomposition methods and parallelization techniques, and to realize a parallel finite element program on General-purpose computing on graphics processing units (GPGPU).

Acknowledgments
This paper was supported by the Hungarian Scientific Research Fund (OTKA PD 73242), and by the Széchenyi István University (15-3210-02).

REFERENCES

[1] Kuczmann M., Iványi A., *The Finite Element Method in Magnetics*, Akadémiai Kiadó, Budapest, 2008.
[2] 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), No. 11, 4389-4392.
[3] Šturmberger, B., Šturmberger, G., Hadžiselimović, A. Hamler, M. Trlep, V. Goričan, M. Jesenik, High-performance permanent magnet brushless motor with balanced concentrated windings and similar slot and pole numbers, *Journal of Magnetism and Magnetic Materials*, 304 (2006), iss. 2, 829-831.
[4] Magoulés F., *Mesh partitioning techniques and domain decomposition method*, Saxe-Coburg Publication, Kippen, Stirling, Scotland, 2007.
[5] Magoulés F., *Substructuring techniques and domain decomposition method*, Saxe-Coburg Publication, Kippen, Stirling, Scotland, 2010.
[6] Kruis J., *Domain decomposition methods for distributed computing*, Saxe-Coburg Publication, Kippen, Stirling, Scotland, 2006.
[7] Kuczmann M., Parallel Finite Element Method, *Przegląd Elektrotechniczny*, 87 (2011), nr 12b, 100–103.
[8] Marcsa D., Kuczmann M., Parallel Solution of an Electrostatic Field Problem - Case Study, *Pollack Periodica*, 7 (2012), nr 2, 25-34.
[9] Bianchi N., *Electrical Machine Analysis Using Finite Elements*, Taylor & Francis, Boca Rotan, FL, USA, 2005.
[10] http://glaros.dtc.umn.edu/gkhome/views/metis. (Last visited 20 June 2012)
[11] http://morpheus.pte.hu/~peteri (Last visited 20 June 2012)
[12] http://www.mathworks.com (Last visited 20 June 2012)

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

The Correspondence address is:
e-mail: marcsa@maxwell.sze.hu