

## SCHUR-COMPLEMENT BASED PARALLEL FINITE ELEMENT ANALYSIS COUPLED WITH CIRCUIT AND MECHANICAL EQUATIONS

Dániel Marcsa, Miklós Kuczmann

Széchenyi István University, Győr, Hungary

*marcsad@sze.hu, kuczmann@sze.hu*

© Marcsa D., Kuczmann M., 2014

**Abstract:** The analysis and design of electro-mechanic devices involve the solution of large sparse linear systems, and require, therefore, high performance algorithms. In this paper, the Schur complement method with a parallel preconditioned conjugate gradient (PCG) solver is introduced into two-dimensional parallel time-stepping finite element formulation to analyze a rotating machine considering an electromagnetic field, external circuit and rotor movement. The proposed parallel solver with two preconditioners is analyzed concerning its computational efficiency and number of iterations. The results of the simulation of a rotating machine are also presented.

**Key words:** Parallel finite element method, coupled problem, Schur complement method, movement modeling.

### 1. Introduction

The numerical design of electromechanic devices is a very complex task because many different physical aspects should be considered. The performance of electric equipment cannot be described only by electromagnetic field equations because electromagnetic field distribution depends on mechanical quantities, characteristics of external circuits, etc. Induction machines are the most obvious examples.

The finite element method (FEM) [1–2] is a well-known technique for the solution of a wide range of problems in science and engineering. However, a few years ago, the simulation of complex structures considering multiple aspects in the same set of equations was restrictive due to the unavailability of sufficient computer capabilities for data processing. However, nowadays, thanks to the improvements in computer architecture, the analysis of complex electromagnetic systems is more affordable.

Nevertheless, the analysis of complex systems, e.g. electric machine analysis considering the movement and the voltage source requires a computing effort to solve large sparse linear systems. These large linear systems arise from the discretization by FEM. The solution to these large equation systems are very resource-intensive and time consuming, wherein the resources and time of

the simulation plays an important role for designers and researchers. Therefore, the solution of a complex system should be parallelised in order to speed up the numerical computations with less computer requirements.

In this paper, we propose to solve a two-dimensional parallel time-stepping finite element problem using the domain decomposition method (DDM) [3, 4]. The used DDM is the Schur complement method with a parallel Krylov method, the parallel preconditioned conjugate gradient method [3], which is currently one of the most popular methods for the systems with real symmetric positive definite matrices. Two preconditioners, and namely Jacobi and Neumann-Neumann preconditioners are used in the solver algorithm to improve the convergence behaviour. We present the numerical behaviour of a parallel preconditioned conjugate gradient solver with preconditioners for modeling an electric machine with direct-coupled field formulations.

The paper is organized as follows. The next section briefly describes the equations and methods used to introduce the formulation of the parallel time-stepping finite element method coupled with circuit and mechanical equations. The third section describes the Schur complement method and the way in which this method and its iterative solver algorithm can be used to formulate and solve a coupled problem. Section 4 collects numerical results to illustrate the potential of the method, and present an induction machine with different mesh size. Finally, some extensions of the method are discussed.

### 2. Formulation

The electric machine is modelled in 2D using the FEM to discretize the domain, which is based on the weak formulation of partial differential equations, which can be obtained by Maxwell's equations and the weighted residual method [1]. The magnetic vector potential formulation has been applied, and the temporal derivatives are discretized by the backward Euler scheme. The field and external circuit equations are combined together using the direct coupling method [2, 5, 6]. Equation (1) shows the matrix system of the field equations [2, 5]:

$$\begin{aligned} \mathbf{S}\mathbf{A}(t) + \mathbf{N}\frac{d}{dt}\mathbf{A}(t) - \mathbf{P}\mathbf{I}(t) &= \mathbf{0}, \\ \mathbf{Q}\frac{d}{dt}\mathbf{A}(t) + \mathbf{R}\mathbf{I}(t) + \mathbf{L}\frac{d}{dt}\mathbf{I}(t) &= \mathbf{U}(t), \end{aligned} \quad (1)$$

$$\mathbf{a} = \begin{bmatrix} \mathbf{A}(t) \\ \mathbf{I}(t) \end{bmatrix},$$

where  $\mathbf{A}$  is the vector of magnetic vector potential,  $\mathbf{I}$  is the vector of currents in the windings,  $\mathbf{U}$  is the vector of voltages at the terminal of the winding,  $\mathbf{S}$  is the matrix related to permeability,  $\mathbf{N}$  is the matrix related to electric conductivity,  $\mathbf{Q}$  is the matrix associated with flux linkage,  $\mathbf{R}$  is the matrix of d.c. resistance of windings,  $\mathbf{L}$  is the matrix of the end-windings inductances.

In order to simulate the rotation of the rotor in a two-dimensional case, we used one of the most common methods the so-called sliding surface technique with the first order nodal interpolation method [7]. The interpolation method is necessary when the fixed (stator) and mobile (rotor) part of the mesh are non-conforming because of variation of angular speed. The new angular speed and rotor displacement are evaluated by the mechanical oscillation equation [2, 7]:

$$J_r \frac{d}{dt}\omega_r - D_r \omega_r = T_e - T_L, \quad \frac{d}{dt}\alpha_r = \omega_r, \quad (2)$$

where  $J_r$  is the rotor inertia moment,  $D_r$  is the friction damping coefficient,  $T_e$  is the electromagnetic torque,  $T_L$  is the load torque acting on the mechanical axis,  $\omega_r$  is the rotor speed, and  $\alpha_r$  is the rotor angular position. At each time step, the electromagnetic torque is calculated via the Maxwell's stress tensor [2] by the following relationship:

$$T_e = L \int_{\Gamma} \left\{ \mathbf{r} \times \left[ \frac{1}{\mu_0} (Bn)B - \frac{1}{2\mu_0} B^2 n \right] \right\} d\Gamma, \quad (3)$$

where  $L$  is the length of the domain in  $z$ -direction, and  $r$  is the position vector linking the rotation axis to the element  $d\Gamma$ , and  $\Gamma$  is a surface, which is placed in the middle of the air gap,  $B$  is the magnetic flux density,  $\mu_0$  is the permeability of vacuum, and  $n$  is the normal unit vector to the surface.

### 3. Schur Complement Method

When the domain decomposition method is used, the problem domain  $\Omega$  is to be divided into several sub-domains in which the unknown potentials can be calculated simultaneously, i.e. in parallel. The general form of a linear algebraic problem arising from the discretization of parabolic type problems defined on the domain  $\Omega$  can be written as  $\mathbf{K}\mathbf{a}=\mathbf{b}$  [2, 4], in more detail:

$$\mathbf{K} = \begin{bmatrix} \mathbf{S} + \frac{\mathbf{N}}{\Delta t} & -\mathbf{P} \\ \frac{\mathbf{Q}}{\Delta t} & \mathbf{R} + \frac{\mathbf{L}}{\Delta t} \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} \frac{\mathbf{N}}{\Delta t} & \mathbf{0} \\ \frac{\mathbf{Q}}{\Delta t} & \frac{\mathbf{L}}{\Delta t} \end{bmatrix} \begin{bmatrix} \mathbf{A}(t-\Delta t) \\ \mathbf{I}(t-\Delta t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{U}(t) \end{bmatrix}, \quad (4)$$

where  $\mathbf{K} \in R^{(n \times n)}$  is the mass matrix,  $\mathbf{b} \in R^{(n \times 1)}$  on the right hand side of the equations, and  $\mathbf{a} \in R^{(n \times 1)}$  contains the unknowns. Here  $n$  is a number of unknowns.

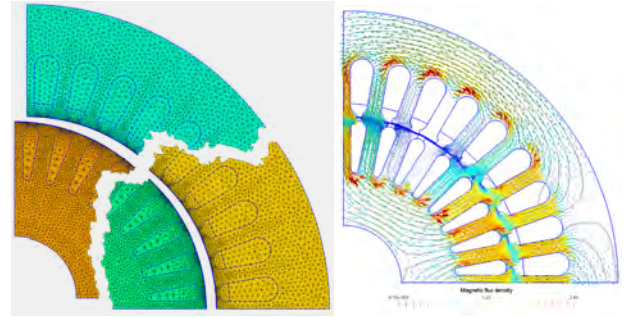


Fig. 1. The partitioned finite element mesh and the assembled results, the equipotential lines of magnetic vector potential and the magnetic flux density vectors.

After the problem is partitioned into a set of  $N_S$  disconnected sub-domains, as we can see in Fig. 1, and the linear sparse system,  $\mathbf{K}\mathbf{a}=\mathbf{b}$  is split into  $N_S$  particular blocks [3, 4]

$$\begin{bmatrix} \mathbf{K}_{jj} & \mathbf{K}_{j\Gamma_j} \\ \mathbf{K}_{\Gamma_j j} & \mathbf{K}_{\Gamma_j \Gamma_j} \end{bmatrix} \begin{bmatrix} \mathbf{a}_j \\ \mathbf{a}_{\Gamma_j} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_j \\ \mathbf{b}_{\Gamma_j} \end{bmatrix}, \quad (5)$$

where  $j=1 \dots N_S$ ,  $\mathbf{K}_{jj}$  is the positive definite sub-mass matrix of the  $j^{\text{th}}$  sub-domain,  $\mathbf{b}_j$  is the vector of the right hand side defined inside the sub-domain. The sub-matrix  $\mathbf{K}_{j\Gamma_j} = \mathbf{K}_{\Gamma_j j}^T$  contains the coefficients of  $j^{\text{th}}$  sub-domain, which are connected to the interface boundary unknowns of that region. The superscript T denotes the transpose.  $\mathbf{K}_{\Gamma_j \Gamma_j}$ , and  $\mathbf{b}_{\Gamma_j}$  express the coupling of the interface unknowns. It should be noted, that it is much easier in parallel computation if the sliding surface is used as an interface boundary in the air gap, as it can be seen in Fig. 1.

Each sub-domain will be allocated to an independent processor core, because the sub-matrix  $\mathbf{K}_{jj}$  with the  $\mathbf{K}_{j\Gamma_j}$ ,  $\mathbf{K}_{\Gamma_j j}$  and the right-hand side  $\mathbf{b}_j$  are independent, i.e. they can be assembled in parallel on distributed memory. Only the matrix  $\mathbf{K}_{\Gamma_j \Gamma_j}$ , and the vector  $\mathbf{b}_{\Gamma_j}$  are not independent.

The matrix  $\mathbf{K}_{\Gamma\Gamma}$  and the vector  $\mathbf{b}_{\Gamma}$  are assembled after interprocess data transfer because they are the assembly of  $\mathbf{K}_{\Gamma_j\Gamma_j}$  and  $\mathbf{b}_{\Gamma_j}$ , where  $j$  is the index of sub-domains.

The assembly and solution of the sub-matrices can be performed parallelly by independent processors. However, the solution requires exchange of the interface values,  $\mathbf{a}_{\Gamma_j}$  between the processes in charge of the various sub-domains. In many practical applications, the preconditioned conjugate gradient (PCG) method is used because of its simplicity and efficiency. The parallel implementation of the preconditioned conjugate gradient method can be presented by algorithm 1 [3].

In the parallel PCG algorithm [3],  $\mathbf{K}_S$  and  $\mathbf{b}_S$  are the mass matrix and right-hand side of the sub-domain,  $\mathbf{a}$  is the vector of unknown potentials,  $\mathbf{r}$  is the residual vector, subscript  $\Gamma_{int}$  denotes the external interface entries from neighbouring sub-domains,  $\mathbf{M}$  is the preconditioning matrix [8],  $\mathbf{w}$  and  $\mathbf{p}$  are working vectors, and  $\varepsilon$  is the specified error tolerance.

*Algorithm 1. Parallel PCG Algorithm.*

- 1 Initialization:  $a_0 = 0$ ,
- 2  $r_0 = b_s$
- 3 Assembly local  $r_0$  with  $r_{\Gamma_{int}}$  entries  $\rightarrow \bar{r}_0$ ,
- 4 **for**  $i = 0, 1, \dots$  **do**
- 5  $\mathbf{w}_i = \mathbf{M}\bar{\mathbf{r}}_i$ ,
- 6  $\mathbf{g}_i = r_i^T \mathbf{w}_i$ ,
- 7 Assembling  $\mathbf{g}_i$ ,
- 8 Assembling local  $\mathbf{w}_i$  with  $w_{\Gamma_{int}}$  entries  $\rightarrow \bar{\mathbf{w}}_i$ ,
- 9 **if**  $i = 0$  **then**
- 10  $\bar{\mathbf{p}}_i = \bar{\mathbf{w}}_i$ ,
- 11 **else**
- 12  $\bar{\mathbf{p}}_i = \bar{\mathbf{w}}_i + (\mathbf{g}_i / \mathbf{g}_{i-1})\bar{\mathbf{p}}_{i-1}$ ,
- 13  $\mathbf{w}_i = \mathbf{K}_s \bar{\mathbf{p}}_i$ ,
- 14  $\mathbf{b}_i = \mathbf{p}_i^T \bar{\mathbf{w}}_i$ ,
- 15 Assembling  $\mathbf{b}_i$
- 16 Assembling local  $\mathbf{w}_i$  with  $w_{\Gamma_{int}}$  entries  $\rightarrow \bar{\mathbf{w}}_i$
- 17  $\bar{\mathbf{a}}_i = \bar{\mathbf{p}}_{i-1} + (\mathbf{g}_i / \mathbf{b}_i)\bar{\mathbf{p}}_i$ ,
- 18  $\bar{r}_i = \bar{r}_{i-1} + (\mathbf{g}_i / \mathbf{b}_i)\bar{\mathbf{w}}_i$ ,
- 19 **if**  $\mathbf{g}_i / \mathbf{g}_0 < \varepsilon$  **then**
- 20 **return**

In this case, two preconditioners have been used, the Jacobi preconditioner [3], and the Neumann-Neumann preconditioner [8]. The Jacobi preconditioner is one of the simplest forms of preconditioning, in which the preconditioner is chosen to be the diagonal of the matrix  $\mathbf{M} = [\text{diag}(\mathbf{K}_S)]^{-1}$ . The Neumann-Neumann preconditioner is defined by approximating the inverse of the sum

of the local Schur complement matrices by the weighted sum of the inverses [7],

$$\mathbf{M} = \sum_{j=1}^{N_s} \mathbf{D}_j \left( \mathbf{K}_{\Gamma_j\Gamma_j} - \mathbf{K}_{j\Gamma_j}^T \mathbf{K}_{jj}^{-1} \mathbf{K}_{j\Gamma_j} \right) \mathbf{D}_j, \quad (6)$$

where  $\mathbf{D}_j$  is the diagonal weight matrix, which must verify  $\sum_{j=1}^{N_s} \mathbf{D}_j = \mathbf{I}$ .

To illustrate how the above mentioned domain decomposition method with parallel PCG is implemented into the field-circuit coupled finite element method.

#### 4. Application

In this section, to demonstrate the operation of the methods presented, a 4-pole 3-phase 3kW induction motor with unskewed rotor slot fed by sinusoidal voltage is tested. The test problem and its parameters were described in [7], and the GMSH model was taken from the free GetDP models in [9].

The domain being studied consists of one pole of the machine, i.e. a  $45^\circ$  domain, as you can see in Fig. 1. The anti-periodic boundary conditions are used to represent the whole problem. In this simulation, 20 periods have been calculated, and a period of the 50Hz voltage excitation has been divided into 300 time steps.

The numerical experiments have been performed on the platform composed of four CPUs Intel Xeon L5420. Each CPU is a Dual-core processor running at 2.5 GHz. It supplies 8x4GB RAM memory. The benchmark presented in this paper consists in performing 10 times the same operation in order to overcome the problem caused by the finite precision of the clock. The implemented program has been developed under the Matlab computing environment in C language and in own scripting language of Matlab.

We compare the implemented method for different mesh size. Table 1 contains data on the partitioned finite element mesh in various global element size factors.

In order to use the same stop criterion for the methods,  $e = 10^{-8}$ , the speedup was calculated by the following formula,

$$\text{Speedup}_{sf} = \text{Time}_1 / \text{Time}_{sf}, \quad (7)$$

where  $\text{Time}_1$  is the running time in the case of the smallest element size factor, and  $\text{Time}_n$  is the running time of the different size factor [10]. The efficiency was calculated by the following formula

$$\text{Efficiency} = \text{Speedup}_{sf} / n, \quad (8)$$

where  $\text{Speedup}_n$  is the speedup in the case of the different element size factors, and  $n$  is the applied processor cores [10].

In Table 1, the number of degrees of freedom of the unpartitioned problem,  $GDoF$ , the number of degrees of freedom of one sub-domain,  $DoF$ , the number of interface unknowns,  $CDoF$  are summarized.

The results on the performance of the parallel program for all element size factors are reported in Fig. 3–5. The speedups (Fig. 3–4) are computed using the wall clock time of smallest problem (2900  $GDoF$ ) as the reference point. The results show the speedup as high as 9.8 with the Jacobi preconditioner, and 10.8 with the Neumann-Neumann preconditioner. In both case, the speedup is continuously increase until the 0.004 element size factor, because the time of the interprocess communication is relatively smaller, than the time of the parallel PCG. This is not true for the largest test case, when the subproblems are too big, and the operations of the parallel algorithm are very time consuming. Further, the memory requirement of the program is also very high in this case. These conclusions are also supported by the figures of efficiency, as you can see in Fig. 4 in the case of Jacobi preconditioner, and in Fig. 5 in the case of Neumann-Neumann preconditioner.

Table 1

## Data on different finite element meshes used

Global element size factor (GDoF)	Number of processor cores		
	CDoF (DoF)		
	2	4	8
0.05 (2900)	33 (1450)	71 (725)	130 (363)
0.025 (5832)	52 (2916)	118 (1458)	206 (729)
0.01 (23208)	102 (11604)	265 (5802)	445 (2901)
0.008 (35421)	128 (17711)	321 (8855)	508 (4427)
0.006 (58724)	155 (29362)	420 (14681)	679 (7340)
0.004 (129194)	249 (64597)	644 (32299)	1140 (16150)
0.003 (224308)	339 (112154)	802 (56077)	1343 (28039)

Fig. 6 and Fig. 7 show the running performance of a preconditioned conjugate gradient solver, the number of iterations versus the global element size factors at three different number of processor cores. The interprocess communication hardly depends on the number of interface unknowns (CDoF in Table 1) and the number of applied processor cores. However, the number of iterations shows the robustness of the presented algorithm because the curves continuously increase, so the solver is more or less independent from the number of degrees of freedom and the number of interface unknowns.

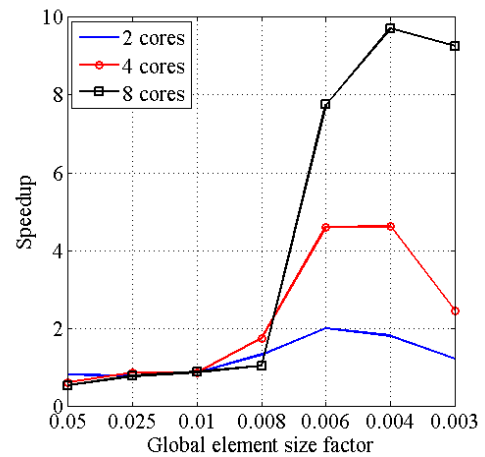


Fig. 2. Speedup of parallel FEM procedure with Jacobi preconditioner.

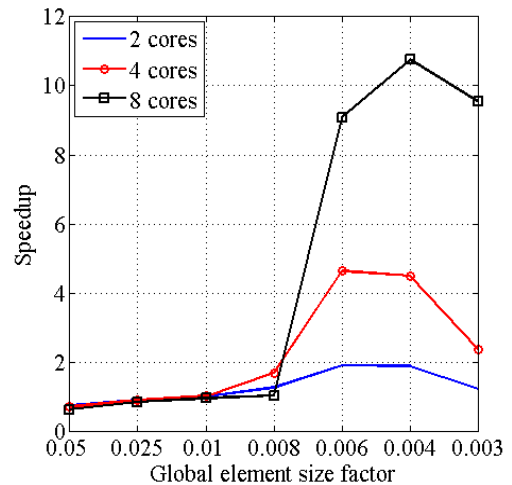


Fig. 3. Speedup of parallel FEM procedure with Neumann-Neumann preconditioner.

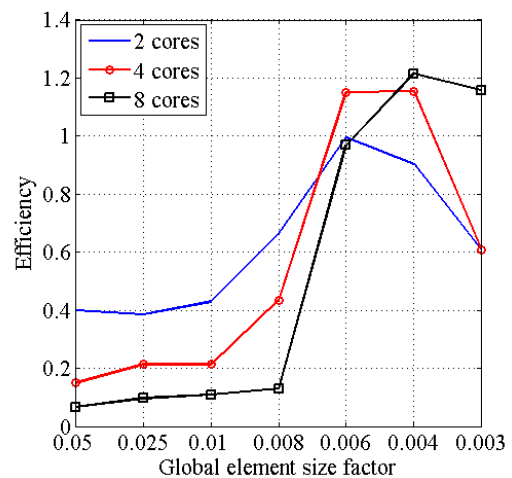


Fig. 4. Efficiency of parallel FEM procedure with Jacobi preconditioner.

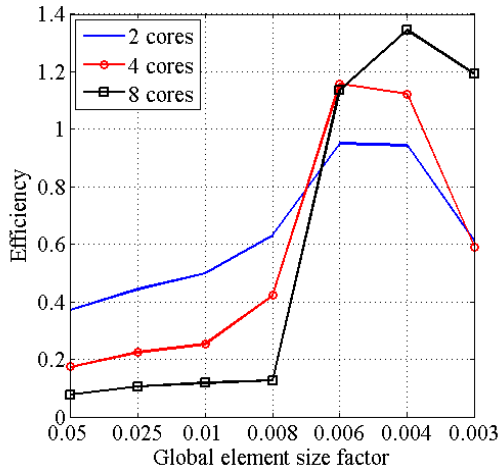


Fig. 5. Efficiency of parallel FEM procedure with Neumann-Neumann preconditioner.

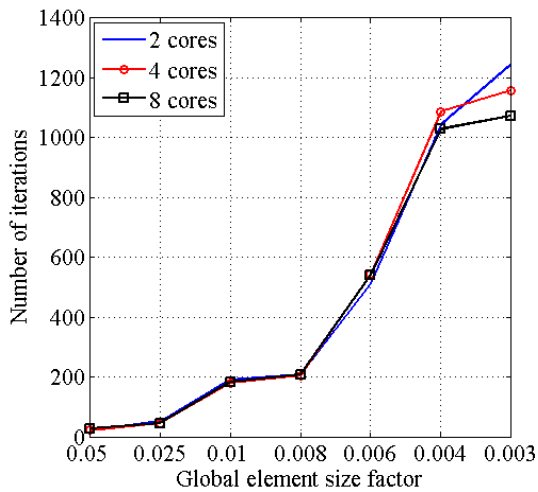


Fig. 6. Number of iterations of PCG with Jacobi preconditioner versus global element size factor.

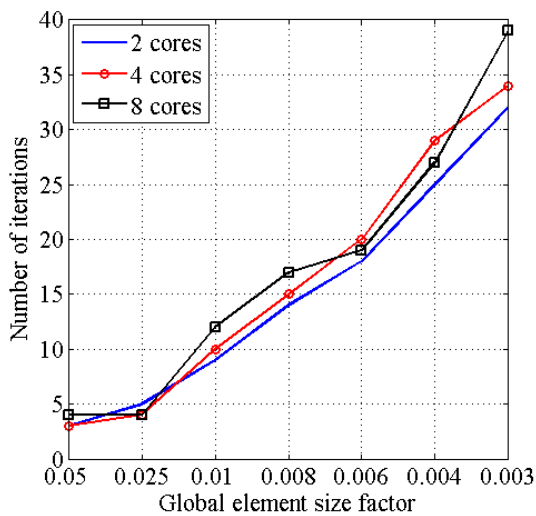


Fig. 7. Number of iterations of PCG with Neumann-Neumann preconditioner versus global element size factor.

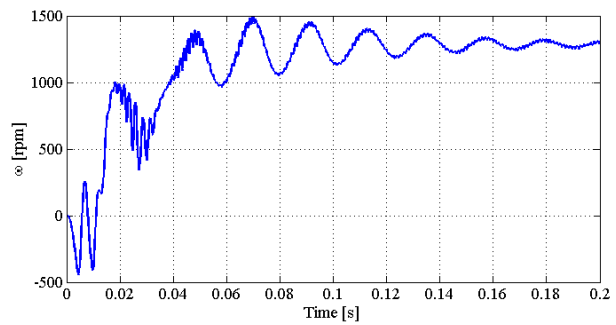


Fig. 8. Speed variation in time.

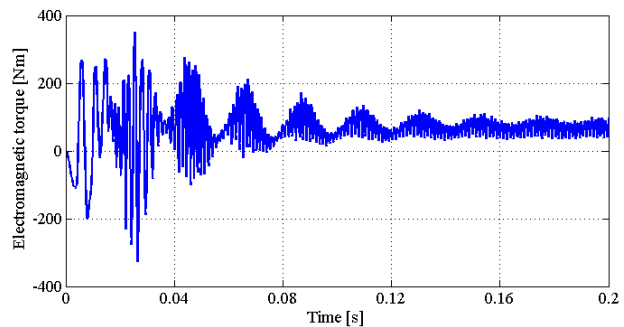


Fig. 9. Electromagnetic torque variation in time.

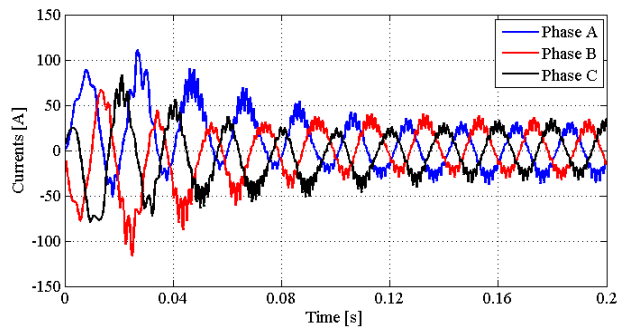


Fig. 10. Time variation of three phase currents.

Fig. 8–10 show the simulation results of an induction machine. These Figures depict the first ten periods of the simulation. Fig. 8 represents the transient speed waveform. Fig. 9 shows the transient torque waveform of the machine. Fig. 10 demonstrates the transient current waveforms of the stator windings.

### 5. Conclusion

In this paper, a two-dimensional parallel finite- element modeling of an induction machine has been presented taking the rotor movement and field-circuit equation of the windings into account. To study the performance of the implemented method, different global finite element size factors have been considered. Results of numerical experiments on all mesh sizes compared. Furthermore, the results obtained for the simulation of the induction machine have also been presented.

The numerical experiments show that the work on the implemented program depends, to a large extent, on

the size of the problem. If the problem size is too large, the efficiency of the computation decreases, so the running performance of the implemented program depends on the size of the problem. However, the parallel PCG solver with preconditioners works properly, as the presented results show.

It should be noted, that only a two-dimensional benchmark has been used for the numerical tests. The tests with more complex three-dimensional problems will be the subject of a forthcoming work.

#### Acknowledgement

This paper was supported by Széchenyi István University, Faculty of Engineering Science, Hungary.

#### References

- [1] M. Kuczmann and A. Iványi, *The Finite Element Method in Magnetics*. Budapest, Hungary: Akadémiai Kiadó, 2008.
- [2] J. P. A. Bastos and N. Sadowski, *Electromagnetic Modeling by Finite Element Methods*. New York, USA: Marcel Dekker Inc, 2003.
- [3] G. P. Nikishkov, “Basics of the Domain Decomposition Method for Finite Element Analysis” In F. Magoulés, eds. *Mesh Partitioning Techniques and Domain Decomposition Methods*, pp. 119–142. Stirlingshire, UK: Saxe-Coburg Publications, 2007.
- [4] D. Marcsa and M. Kuczmann, “Parallel solution of electrostatic and static magnetic field problems by domain decomposition method”, *Przegląd Elektrotechniczny*, vol. 89, no. 2b, pp. 49–52, 2013.
- [5] A. M. De Oliveira, R. Antunes, P. Kuo-Peng, N. Sadowksi, and P. Dular, “Electrical machine analysis considering field-circuit-movement and skewing effects”, *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 23, no. 4, pp. 1080–1091, 2004.
- [6] X. Shi, Y. Le Menach, J. P. Ducreux, and F. Piriou, “Comparison of slip surface and moving band techniques from modelling movement in 3D with FEM”, *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 25, no. 1, pp. 17–30, 2006.
- [7] J. Gyselinck, L. Vandeveld, P. Dular, C. Geuzaine, and W. Legros, “A general method for the frequency domain FE modelling of rotating electromagnetic devices”, *IEEE Transactions on Magnetics*, vol. 39, no. 3, pp. 1147–1150, 2003.
- [8] J. M. Cros, “A preconditioner for the Schur complement domain decomposition method”, in *Proc. XIV-th International Conf. on Domain Decomposition Methods*, pp. 373–380, Cocoyoc, Mexico, 2002.

- [9] “ONELAB – Open Numerical Engineering LABoratory”, [http://onelab.info/wiki/Electric\\_Machines](http://onelab.info/wiki/Electric_Machines)
- [10] J. Jajczyk, “Optimization with the Use of Genetic Algorithm Method and Parallel Computing”, *Computational Problems of Electrical Engineering*, vol. 3, no. 2, pp. 24–27, 2013.

### РОЗРАХУНОК ПОЄДНАНИХ КОЛОВОЇ ТА МЕХАНІЧНОЇ МОДЕЛІ МЕТОДОМ СКІНЧЕННИХ ЕЛЕМЕНТІВ З РОЗПАРАЛЕЛЕННЯМ РОЗРАХУНКІВ ІЗ ЗАСТОСУВАННЯМ ДОПОВНЕННЯ ШУРА

Даніел Маркса, Міклош Кучманн

Розрахунок та проектування електромеханічних пристроїв передбачають розв’язування великих лінійних розріджених систем рівнянь, що потребують застосування високоефективних алгоритмів. У цій роботі застосовано техніку доповнення Шура із попередньо зумовленим методом спряжених градієнтів до аналізу режиму електричної машини – розрахунку електромагнітного поля, зовнішнього кола та механіки ротора. Задачу сформульовано у двовимірній постановці методом скінченних елементів з проведенням розпаралелення розрахунків. Запропоновану методіку розпаралелення з двома операторами попереднього обумовлення проаналізовано стосовно її обчислювальної ефективності та кількості ітерацій. Продемонстровано результати моделювання електричної машини.



finite element method and non-overlapping domain decomposition techniques in electromagnetic computation.

**Dániel Marcsa** – B. Sc. in Electrical Engineering (2008), M. Sc. in Mechatronics Engineering (2010) awarded by the Széchenyi István University, Hungary. Assistant Professor at the Széchenyi István University, Hungary. He carries out research into static magnetic and eddy current potential formulations, numerical simulation of the electric machines by



the “Best PhD Dissertation” award (2006) from the Hungarian Academy of Sciences.

**Miklós Kuczmann** – M.S c. in Electrical Engineering (2000), Ph. D. in Electrical Engineering (2005) awarded by Budapest University of Technology and Economics, Department of Electromagnetic Theory, Hungary. Head of the Department of Automation, Széchenyi István University, Hungary. Dr. Kuczmann won the Bolyai János Scholarship (2006) and