Primal Domain Decomposition Method with Direct and Iterative Solver for Circuit-Field-Torque Coupled Parallel Finite Element Method to Electric Machine Modelling

Daniel MARCSA, Miklos KUCZMANN

Department of Automation, Faculty of Computer Science, Mechanical and Electrical Engineering,
Szechenyi Istvan University, Egyetem ter 1, 9026 Gyor, Hungary

marcsad@sze.hu, kuczmann@sze.hu

DOI: 10.15598/aeee.v13i5.1410

Abstract. *The analysis and design of electromechanical devices involve the solution of large sparse linear systems, and require therefore high performance algorithms. In this paper, the primal Domain Decomposition Method (DDM) with parallel forward-backward and with parallel Preconditioned Conjugate Gradient (PCG) solvers are introduced in two-dimensional parallel time-stepping finite element formulation to analyze rotating machine considering the electromagnetic field, external circuit and rotor movement. The proposed parallel direct and the iterative solver with two preconditioners are analyzed concerning its computational efficiency and number of iterations of the solver with different preconditioners. Simulation results of a rotating machine is also presented.*

Keywords

Field-circuit-torque coupling, parallel finite element method, primal domain decomposition method.

1. Introduction

The numerical field calculation of electromechanical devices is a very complex task, because a lot of different physical aspects should be considered for the appropriate modelling. The performances of electrical equipments are not defined only by their electromagnetic field, because the electromagnetic field has strong interaction between the following quantities: electromagnetic field distribution, mechanical oscillation equation, external circuits, etc. The electric 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 back, the simulation of complex structures considering multiple aspects in a same set of equations was restrictive, due to the unavailability of enough computer capabilities for data processing. However, nowadays, thanks to the improvements in the computer architecture, the analysis of complex electromagnetic systems is more affordable.

But, the analysis of complex systems, e.g. rotating electrical machine analysis considering movement and voltage drive source require a computing effort to solve large sparse linear systems. These large linear systems arise from the discretization with the help of finite element method. The solution of these large equation systems are very resource-intensive and time consuming, wherein the resources and time of the calculation 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 requirement.

In this paper, we propose to solve a two-dimensional parallel time-stepping finite element problem using primal Domain Decomposition Method [3], [4], [5], [6], [7]. The used primal DDM is also called the static condensation method, the method of sub-structuring or the Schur complement method [3]. The direct solver is the parallel forward-backward method with parallel factorisation [4], [5]. The iterative solver is a parallel Krylov method, the parallel preconditioned conjugate gradient method [3], [6], which is currently one of the most popular method for systems with real symmetric positive definite matrices. Two preconditioners, Jacobi and Neumann-Neumann preconditioner [7] used in the solver algorithm to improve the convergence behaviour. We present the numerical behaviour of the parallel direct solver and the parallel PCG solver with preconditioners for the modelling of electrical machine with direct coupled field-circuit formulations [2].

The paper is organized as follows. The next section briefly describes the used equations and methods to give an introduction to the formulation of the parallel time-stepping finite element method coupled with circuit and mechanical oscillation equations. The third section describes the Schur complement method and how this method, and its direct and iterative solver algorithms can be used to formulate and solve a coupled problem. Section 4. collect numerical results to illustrate the potential of the method, an induction machine with different mesh size are then presented. Finally, some extensions of the method are discussed.

2. Field-Circuit Coupling Finite Element Formulation

The electrical machine is modelled in two-dimensional space, using the FEM to discretize the domain, which is based on the weak formulation of the 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's scheme [2]. The field and external circuit equations are combined together using the direct coupling method [2], [6]. Equation (1) shows the matrix system of the field equations [2], [6]:

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

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, $\mathbf{P}$ is the matrix associated with constant coil current, $\mathbf{Q}$ 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 the two-dimensional case, we used one of the most common method, the so called sliding surface technique with first order nodal interpolation method [8]. 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 [6]:

$$\begin{aligned}&J_r\frac{\mathrm{d}}{\mathrm{d}t}\omega_r-D_r\omega_r=T_e-T_L,\\&\frac{\mathrm{d}}{\mathrm{d}t}\alpha_r=\omega_r,\end{aligned}\tag{2}$$

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, ω_r is the rotor speed, and α_r is the rotor angular position. At each time step, the electromagnetic torque is calculated via the Maxwell's stress tensor [2], [6] by the following relationship:

$$T_e=L\int_\Gamma\left\{\vec{r}\cdot\left[\frac{1}{\mu_0}(B\vec{n}B)-\frac{1}{2\mu_0}B^2\vec{n}\right]\right\}\mathrm{d}\Gamma,\tag{3}$$

where L is the axial length of the domain, and $\vec{r}$ is the position vector linking the rotation axis to the element $\mathrm{d}\Gamma$, and Γ is a surface, which is placed in the middle of the air gap, B is the magnetic flux density, μ_0 is the permeability of vacuum, and $\vec{n}$ is the normal unit vector to the surface.

3. Primal Domain Decomposition Method

When domain decomposition method is used, the problem domain Ω is to divide into several sub-domains in which the unknown magnetic vector potentials and currents can be calculated simultaneously, i.e. in a parallel way. The general form of a linear algebraic problem arising from the discretization of a parabolic type problems defined on the domain Ω can be written as $\mathbf{Ka}=\mathbf{b}$ [3], [5], in more detail:

$$\begin{aligned}&\mathbf{K}=\begin{bmatrix}\mathbf{S}+\frac{\mathbf{N}}{\Delta t} & -\mathbf{P}\\ \frac{\mathbf{Q}}{\Delta t} & \mathbf{R}+\frac{\mathbf{L}}{\Delta t}\end{bmatrix},\\&\mathbf{a}=\begin{bmatrix}\mathbf{A}(t)\\ \mathbf{I}(t)\end{bmatrix},\\&\mathbf{b}=\begin{bmatrix}\frac{\mathbf{N}}{\Delta t} & 0\\ \frac{\mathbf{Q}}{\Delta t} & \frac{\mathbf{L}}{\Delta t}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{A}\left(t-\Delta t\right)\\ \mathbf{I}\left(t-\Delta t\right)\end{bmatrix}+\begin{bmatrix}0\\ \mathbf{U}\left(t\right)\end{bmatrix},\end{aligned}\tag{4}$$

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

After the problem is partitioned into a set of N_S disconnected sub-domains, as it can be seen in Fig. 1 and the linear sparse system, $\mathbf{Ka}=\mathbf{b}$ has been split into N_S particular blocks [3], [4], [5], [6]. For each sub-domains, the nodes are partitioned into interior nodes designated by the subscript i, and interface boundary nodes designated by the subscript Γ. If the interior nodes are numbered first and the interface boundary nodes are numbered last, then the sumdomain equation system can be written in the following matrix form [3], [6].

$$\begin{bmatrix} \mathbf{K}_{jj} & \mathbf{K}_{j\Gamma} \\ \mathbf{K}_{\Gamma j} & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a}_j \\ \mathbf{a}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{b}_j \\ \mathbf{b}_\Gamma \end{bmatrix}, \tag{5}$$

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

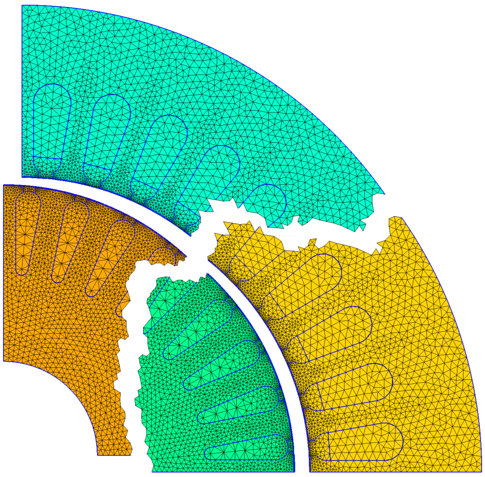

Fig. 1: Domain decomposition of the finite element mesh of the induction motor.

Each sub-domain will be allocated to an independent processor core, because the sub-matrix $\mathbf{K}_{jj}$ with the $\mathbf{K}_{j\Gamma}$, $\mathbf{K}_{\Gamma j}$ and the right-hand side $\mathbf{b}_j$ are independent, i.e. they can be assembled in parallel on distributed memory. Only the matrix $\mathbf{K}_{\Gamma\Gamma}$, and the vector $\mathbf{b}_\Gamma$ 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 $\mathbf{K}_{\Gamma\Gamma}$ and $\mathbf{b}_\Gamma$. $\mathbf{K}_{\Gamma\Gamma}$ is called the local Schur complement, and $\mathbf{K}_{\Gamma\Gamma}$ is the mass matrix of the reduced system, the Schur complement of the problem.

The assembly and the solution of the sub-matrices can be performed parallel by independent processors. However, the solution requires exchange of interface values, $\mathbf{a}_\Gamma$ between the processes in charge of the various sub-domains. In many practical applications, the forward-backward method with LU factorisation, or the preconditioned conjugate gradient method is used because of its simplicity and efficiency.

The practical implementation of the parallel forward-backward substitution with parallel factorisation can be found in [4] and [5].

The parallel implementation of the preconditioned conjugate gradient method can be found in [3] and [6].

In this case, two preconditioners have been used, the Jacobi preconditioner [3], [6] and the Neumann-Neumann preconditioner [7]. The Jacobi preconditioner is one of the simplest forms of preconditioning, in which the preconditioner is chosen to be the diagonal of the matrix. The Neumann-Neumann preconditioner is defined by approximating the inverse of the sum of local Schur complement matrices by the weighted sum of the inverses [7].

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

4. Application

In this section, to demonstrate the operation of the presented methods, a 4-pole 3-phase 3 kW induction motor with un-skewed rotor slot fed by sinusoidal voltage is tested. The test problem and parameters, and the GMSH model is from the free GetDP models it can find in [9], [10].

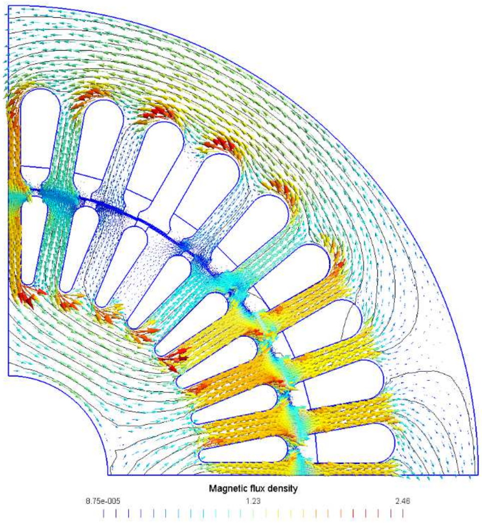

Fig. 2: The assembled results, the equipotential lines of magnetic vector potential and the magnetic flux density vectors.

The studied domain consists of one pole of the machine, i.e. a 45 ° domain, as you can see in Fig. 2.

Anti-periodic boundary conditions are used to represent the whole problem. In this simulation, 20 periods have been calculated, and a period of the 50 Hz voltage excitation has been divided into 300 time steps.

Numerical experiments have been performed on platform composed of four CPUs Intel Xeon L5420. Each CPU is a Dual-core processor running at 2.5 GHz. It supplies 8 × 4 GB RAM memory. The benchmark presented in this paper consists in performing 15 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 [11] computing environment in C language and in own scripting language of the MATLAB [11].

We compare the implemented method for different mesh size. Table 1 contains the data about the partitioned finite element mesh in various global element size factors. In the Tab. 1, the number of degree of freedom of the unpartitioned problem, GDoF, the number of degree of freedom of one sub-domain, DoF, the number of interface unknown, CDoF are summarized.

Tab. 1: Data of the different used finite element meshes.

Globalelement 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)

In order to use the same stop criterion for the methods, $\varepsilon = 10^{-8}$. The speedup has been calculated by the following formula,

$$Speedup_{sf} = \frac{Time_1}{Time_{sf}}, \tag{6}$$

where $Time_1$ is the running time in the case of the smallest element size factor, and $Time_n$ is the running time of the different size factor [6]. The efficiency has been calculated by the following formula:

$$Efficiency = \frac{Speedup_{sf}}{n}, \tag{7}$$

where $Speedup_{sf}$ is the speedup in the case of the different element size factor, and n is the applied processor cores [6].

The performance results of the parallel program are reported in Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7 and Fig. 8 for all element size factor. The speedups (Fig. 3, Fig. 4 and Fig. 5) are computed using the wall clock time of smallest problem (2900 GDoF) as the reference point. The results show the speedup as high as 6.6 with direct solver, 9.8 and 10.8 with iterative solver for the Jacobi preconditioner, and the Neumann-Neumann preconditioner, respectively. In both iterative 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 also true for the direct solver until 0.006 element size factor. However, this is not true for the largest test cases. When the sub-problems are too big, and the operations of both parallel solvers are very time consuming. Further the memory requirement of the program is also very high in this case. This conclusions are also supported by the figures of efficiency, as it can be show in Fig. 6 in the case of direct solver, Fig. 7 in the case of Jacobi preconditioner, and in Fig. 8 in the case of Neumann-Neumann preconditioner.

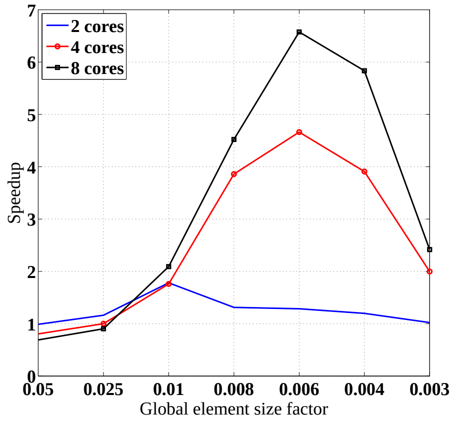

Fig. 3: The speedup of the parallel FEM procedure with direct solver.

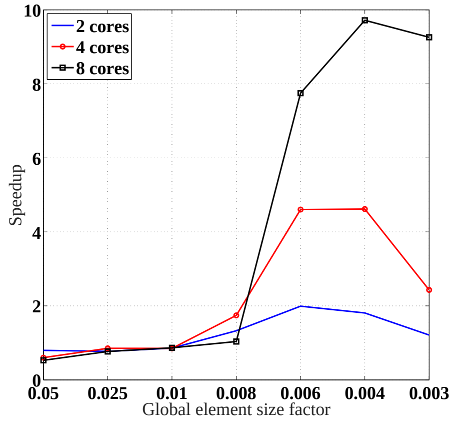

Fig. 4: The speedup of the parallel FEM procedure with Jacobi preconditioner.

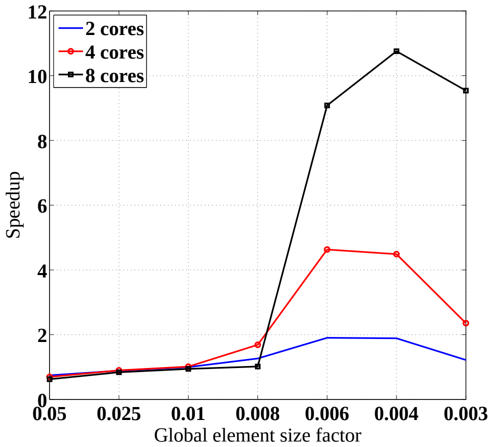

Fig. 5: The speedup of the parallel FEM procedure with Neumann-Neumann preconditioner.

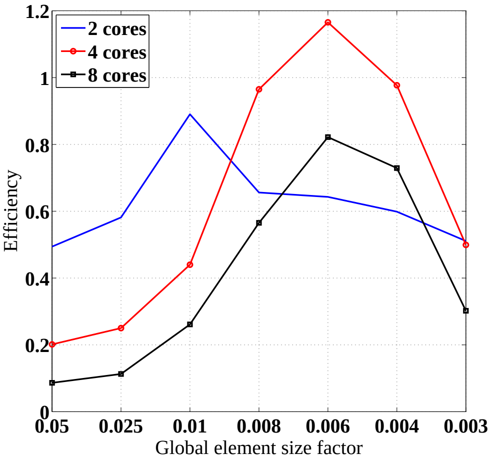

Fig. 6: The efficiency of the parallel FEM procedure with direct solver.

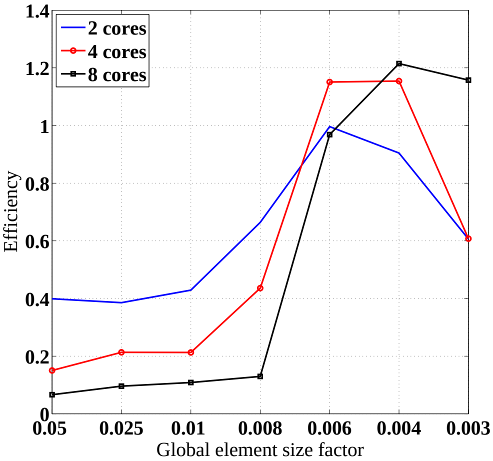

Fig. 7: The efficiency of the parallel FEM procedure with Jacobi preconditioner.

Figure 9 and Fig. 10 show the running performance of the preconditioned conjugate gradient solver, the number of iteration versus global element size factors at three different number of processor cores. The interprocess communication hardly depend on the number of interface unknowns (CDoF in Tab. 1) and the number of applied processor cores. However, the number of iteration shows the robustness of the presented algorithm, because the curves are continuously increase, so the solver more or less independent from the number of degree of freedom and the number of interface unknown.

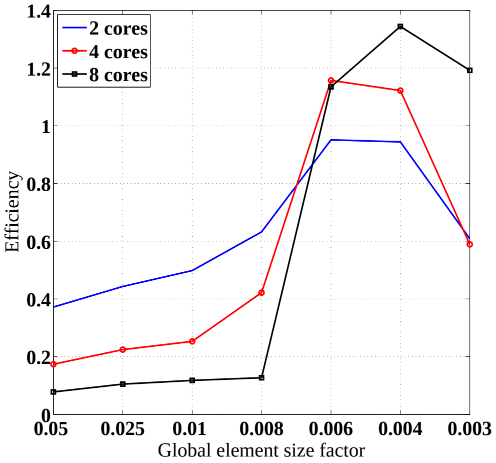

Fig. 8: The efficiency of the parallel FEM procedure with Neumann-Neumann preconditioner.

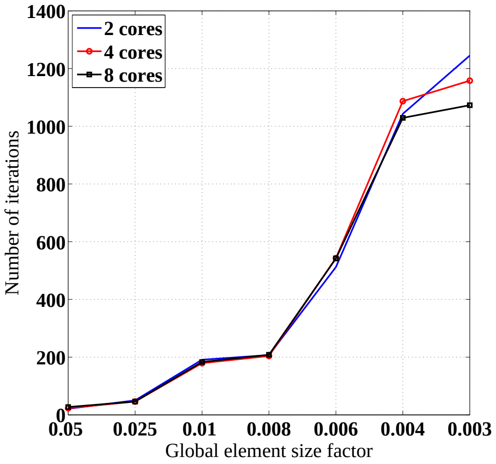

Fig. 9: Number of iteration of PCG with Jacobi preconditioner.

Figure 11, Fig. 12 and Fig. 13 show the simulation results of the induction machine. These figures show the first ten periods of the simulations. Figure 11 shows the transient speed waveform. Figure 12 shows the transient torque waveform of the machine. Figure 13 shows the transient current waveforms of the stator windings.

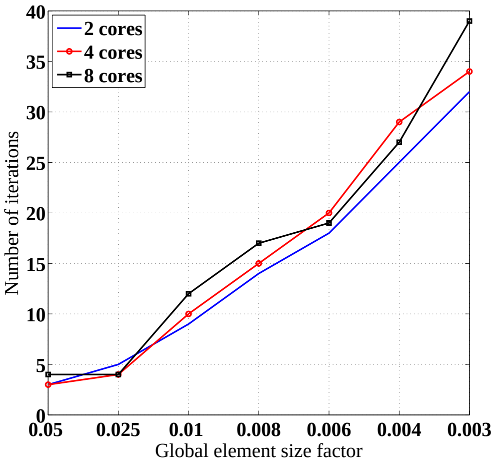

Fig. 10: Number of iteration of PCG with Neumann-Neumann preconditioner.

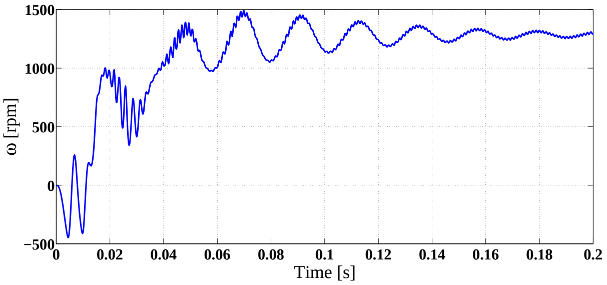

Fig. 11: The speed variation in time.

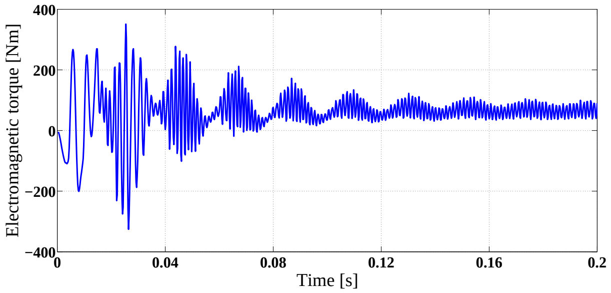

Fig. 12: Electromagnetic torque variation in time.

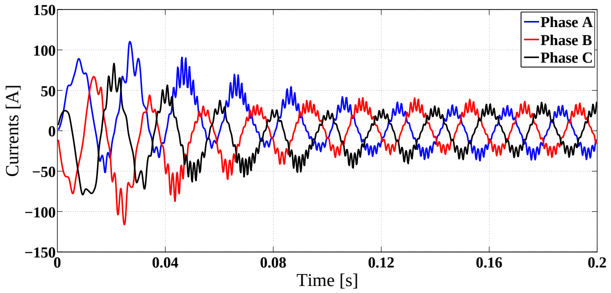

Fig. 13: The time variation of the three phase currents.

5. Conclusion

In this paper, a two-dimensional parallel finite element modelling of an induction machine have been presented taking rotor movement and field-circuit equation of the windings into account. To study the operation of the implemented method, different global finite element size factor have been considered. Results of numerical experiments on all mesh sizes compared. The parallel direct and the parallel PCG solver with the preconditioners are work properly, as the presented results show. Furthermore, the results obtained for the simulation of the induction machine have also been presented.

The numerical experiments show the work of the implemented program is hardly depend on the size of the problem. If the problem size is too large, the efficiency of the computation is decreased, so the running performance of the implemented program is depend on the size of the problem. However, it can be concluded based on the presented results, the parallel direct solver is more efficient for the smaller problems, whereas the iterative solver is more useful for larger problem.

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.

Acknowledgment

This paper was supported by Szechenyi Istvan University, Faculty of Engineering Science.

References

[1] KUCZMANN, M. and A. IVANYI. *The Finite Element Method in Magnetics*. 1st ed. Budapest: Akademiai Kiado, 2008. ISBN 978-963-05-8649-8.

[2] BASTOS, J. P. A. and N. SADOWSKI. *Electromagnetic modeling by finite element methods*. 1st ed. New York: Marcel Dekker, 2003. ISBN 0-8247-4269-9.

[3] KRUIS, J. *Domain decomposition methods for distributed computing*. 1st ed. Stirling: Saxe-Coburg Publications, 2006. ISBN 1-874672-23-7.

[4] NIKISHKOV, G. P., A. MAKINOUCHI, G. YAGAWA and S. YOSHIMURAR. Performance Study of the Domain Decomposition Method with Direct Equation Solver for Parallel Finite Element Analysis. *Computational Mechanics*. 1996,

vol. 19, iss. 1, pp. 84–93. ISSN 1432-0924. DOI: 10.1007/BF02824847.

[5] MAGOULES, F. and F. X. ROUX. *Mesh Partitioning Techniques and Domain Decomposition Methods: Algorithms and Theory for Substructuring and Domain Decomposition Methods.* 1st ed. Striling: Saxe-Coburg Publications, 2007. ISBN 978-1-874672-29-6.

[6] MARCSA, D. and M. KUCZMANN. Schur complement method with iterative solver for 2D fieldcircuit coupling finite element problem with movement. *Przeglad Elektrotechniczny.* 2014, vol. 90, no. 12, pp. 145–148. ISSN 0033-2097. DOI: 10.12915/pe.2014.12.35.

[7] CROS, J. M. A preconditioner for the Schur complement domain decomposition method. In: *Fourteenth International Conference on Domain Decomposition Methods.* Cocoyoc: National Autonomous University of Mexico, 2003, pp. 373–380. ISBN 82-994951-1-3.

[8] SHI, X., Y. L. MENACH, J. P. DUCREUX and F. PIRIOU. Comparison of slip surface and moving band techniques for modelling movement in 3D with FEM. *The international journal for computation and mathematics in electrical and electronic engineering.* 2006, vol. 25, iss. 1, pp. 17–30. ISSN 0332-1649. DOI: 10.1108/03321640610634290.

[9] SABARIEGO, V., J. GYSELINCK and C. GEUYAINE. ONELAB (Open Numerical Engineering LABoratory). *Wikipedia* [online]. 2012. Available at: http://onelab.info/wiki/ONELAB.

[10] GYSELINCK, J., L. VANDEVELDE and J. MELKEBEEK. Multi-slice FE modeling of electrical machines with skewed slots-the skew discretization error. *Transactions on Magnetics.* 2001, vol. 37, iss. 5, pp. 3233–3237. ISSN 0018-9464. DOI: 10.1109/20.952584.

[11] MATLAB and Statistics Toolbox Release 2012b. *MathWorks* [online]. 2012. Available at: http://www.mathworks.com/products/matlab.

About Authors

Daniel MARCSA was born in Keszthely, 8th of August, 1984, in Hungary. He has become B.Sc. in Electrical Engineering in 2008, and M.Sc. in Mechatronics Enginnering in 2010 at the Szechenyi Istvan University. He is recently professor's assitant at the Szechenyi Istvan University, Gyor, Hungary. His research is on static magnetic and eddy current potential formulations, numerical simulation of the electric machines by finite element method, and non-overlapping domain decomposition techniques in electromagnetic computation. He is an active member of the Laboratory of Electromagnetic Field and Hungarian Electrotechnical Association, and member of IEEE and International COMPUMAG Society. He received a Best Presenter Award at the 6th International Ph.D. & DLA Symposium in 2010.

Miklos KUCZMANN was born in Kapuvar, 31st of May, 1977, in Hungary. He has become M.Sc. in Electrical Engineering in 2000, and Ph.D. in Electrical Engineering in 2005 at the Budapest University of Technology and Economics, Department of Electromagnetic Theory. He is presently the dean of the Faculty of Computer Science, Mechanical and Electrical Engineering, and the head of the Department of Automation, Szechenyi Istvan University, Gyor, Hungary, where he is full professor since 2012. Dr. Kuczmann has won the Bolyai Janos Scholarship from the Hungarian Academy of Sciences in 2006 and the "Best Ph.D. Dissertation" award in 2006 from the same Institute.