Dániel MARCSA[1], **Miklós KUCZMANN**[1]

Széchenyi István University (1)

Schur complement method with iterative solver for 2D field-circuit coupling finite element problem with movement

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 Schur complement method with parallel preconditioned conjugate gradient (PCG) solver is introduced in two-dimensional parallel time-stepping finite element formulation to analyse rotating machine considering the electromagnetic field, external circuit and rotor movement. The proposed parallel solver is analysed concerning its computational efficiency and number of iterations. Simulation results of a rotating machine are also presented.*

Streszczenie. *Analiza i projektowanie urządzeń elektromechanicznych zawierają rozwiązanie dużych układów równań z macierzami rzadkimi, i dlatego wymagają algorytmów o wysokiej wydajności. W artykule metodę uzupełnień Schura z algorytmem opartym na metodzie gradientów sprzężonych wprowadzono do dwuwymiarowego, równoległego, czasowokrokowego sformułowania elementów skończonych do analizy maszyny elektrycznej od strony pola elektromagnetycznego, obwodu elektrycznego zewnętrznego oraz ruchu wirnika. Proponowany równoległy algorytm został poddany analizie pod kątem jego wydajności obliczeniowej, oraz liczby iteracji. Rezultaty symulacji maszyny wirującej są także przytoczone. (****Metoda uzupełnienia Schura z iteracyjnym algorytmem dla dwuwymiarowego obwodowo-polowego sprzężonego elementowo-skończeniowego problemu z uwzględnieniem ruchu****).*

Keywords: Parallel finite element method, coupled problem, Schur complement method, movement modelling.
Słowa kluczowe: równoległa metoda elementów skończonych, zagadnienie sprzężone, metoda uzupełnień Schura, modelowanie ruchu

doi:10.12915/pe.2014.12.35

Introduction

The numerical design of electromechanical devices is a very complex task, because a lot of different physical aspects should be considered. 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 equation, external circuits, etc. The 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 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. electrical machine analysis considering movement and voltage source require a computing effort to solve large sparse linear systems. These large linear systems arise from the discretization by FEM. The solution of 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 speedup the numerical computations with less computer requirement.

In this paper, we propose to solve a two-dimensional parallel time-stepping finite element problem using domain decomposition method (DDM) [3-5]. 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 systems with real symmetric positive definite matrices. We present the numerical behaviour of the parallel preconditioned conjugate gradient solver for the modelling of electrical machine with direct coupled field formulations.

The paper is organised 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 equations. The subsequent section describes the Schur complement method and how this method, and its iterative solver algorithm can be used to formulate and solve a coupled problem. Section 3 collect numerical results to illustrate the potential of the method, an induction machine with two different mesh size are then presented. Finally, some extensions of the method are discussed.

Formulation

The electrical machine is modelled in 2D, 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. The field and external circuit equations are combined together using the direct coupling method [2, 6-8]. Eq. (1) shows the matrix system of the field equations [2,6]:

$$\begin{aligned}&\mathbf{SA}(t)+\mathbf{N}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{A}(t)-\mathbf{PI}(t)=\mathbf{0},\\&\mathbf{Q}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{A}(t)+\mathbf{RI}(t)+\mathbf{L}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{I}(t)=\mathbf{U}(t),\end{aligned}\tag{1}$$

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, $\mathbf{Q}$ 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 methods, the so called sliding surface technique with first order nodal interpolation method [9]. 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 [2,6]:

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

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, 10] by the following relationship:

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

where L is the length of the domain in z-direction, and r is the position vector linking the rotation axis to the element $\mathrm{d}\Gamma$, 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 n is the normal unit vector to the surface.

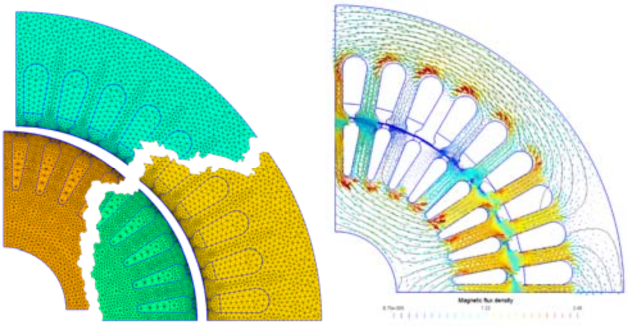

Fig.1. The partitioned finite element mesh and the assembled results, the equipotential lines of magnetic vector potential and the magnetic flux density vectors.

Schur Complement Method

When domain decomposition method is used, the problem domain Ω is to divide into several sub-domains in which the unknown potentials 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 $\mathbf{Ka}=\mathbf{b}$ [2,5], in more detail:

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

$$\mathbf{a}=\begin{bmatrix}\mathbf{A}(t)\\ \mathbf{I}(t)\end{bmatrix}, \tag{4}$$

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

where $\mathbf{K}\in\mathrm{R}^{(n\times n)}$ is the mass matrix, $\mathbf{b}\in\mathrm{R}^{(n\times 1)}$ on the right hand side of the equations, and $\mathbf{a}\in\mathrm{R}^{(n\times 1)}$ 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 [3-5]

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

where j=1…N_S, $\mathbf{K}_{jj}$ is the positive definite sub-mass 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 $\mathbf{K}_{j\Gamma_j}$ $=\mathbf{K}_{\Gamma_j j}{}^{\mathrm{T}}$ contains the coefficients of j^{th} sub-domain, which connect to the interface boundary unknowns of that region. The superscript T denotes the transpose. $\mathbf{K}_{\Gamma_j\Gamma_j}$, and $\mathbf{b}_{\Gamma_j}$ 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.

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 the solution of the sub-matrices can be performed parallel by independent processors. However, the solution requires exchange of 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 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 sub-domain, $\mathbf{a}$ is the vector of unknown potentials, $\mathbf{r}$ is the residual vector, subscript Γ_{int} denotes the external interface entries from neighbouring sub-domains, $\mathbf{M}$=diag($\mathbf{K}_S$) is the diagonal preconditioning matrix [5], $\mathbf{w}$ and $\mathbf{p}$ are working vectors, and ε is the specified error tolerance.

Algorithm 1. Parallel PCG Algorithm.

```
Initialization:  $\mathbf{a}_0 = \mathbf{0}$,
                 $\mathbf{r}_0 = \mathbf{b}_S$
Assembly local $\mathbf{r}_0$ with $\mathbf{r}_{\Gamma_{int}}$ entries → $\bar{\mathbf{r}}_0$,
for i = 0, 1, … do
    $\mathbf{w}_i = \mathbf{M}^{-1}\mathbf{r}_i$,
    $\gamma_i = \mathbf{r}_i^{\mathrm{T}}\mathbf{w}_i$,
    Assembling $\gamma_i$,
    Assembling local $\mathbf{w}_i$ with $\mathbf{w}_{\Gamma_{int}}$ entries → $\bar{\mathbf{w}}_i$,
    if i = 0 then
        $\mathbf{p}_i = \bar{\mathbf{w}}_i$,
    else
        $\mathbf{p}_i = \bar{\mathbf{w}}_i + (\gamma_i/\gamma_{i-1})\mathbf{p}_{i-1}$,
    $\mathbf{w}_i = \mathbf{K}_S\mathbf{p}_i$,
    $\beta_i = \mathbf{p}_i^{\mathrm{T}}\bar{\mathbf{w}}_i$,
    Assembling $\beta_i$,
    Assembling local $\mathbf{w}_i$ with $\mathbf{w}_{\Gamma_{int}}$ entries → $\bar{\mathbf{w}}_i$,
    $\mathbf{a}_i = \mathbf{p}_{i-1} + (\gamma_i/\beta_i)\mathbf{p}_i$,
    $\mathbf{r}_i = \mathbf{r}_{i-1} + (\gamma_i/\beta_i)\bar{\mathbf{w}}_i$,
    if $\gamma_i/\gamma_0 < \varepsilon$ then
        return
```

To illustrate how the above mentioned domain decomposition method with parallel PCG is implemented into the field-circuit coupled finite element method, Fig. 2 shows the flowchart with its main parts. The gray background blocks are the parallel part of the program.

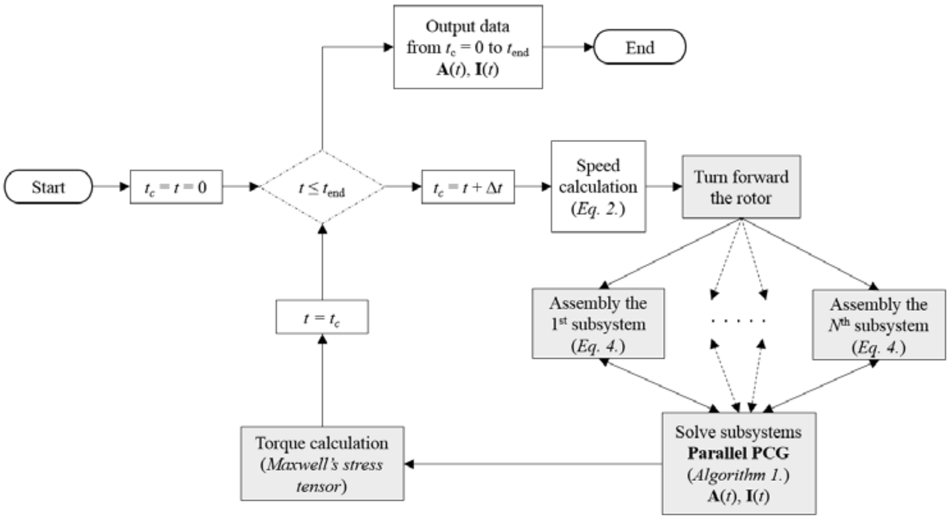
Output data
from $t_c = 0$ to t_{end}
$\mathbf{A}(t)$, $\mathbf{I}(t)$
End
Start
$t_c = t = 0$
$t \le t_{end}$
$t_c = t + \Delta t$
Speed calculation (*Eq. 2.*)
Turn forward the rotor
Assembly the 1st subsystem (*Eq. 4.*)
Assembly the N^{th} subsystem (*Eq. 4.*)
$t = t_c$
Torque calculation (*Maxwell's stress tensor*)
Solve subsystems **Parallel PCG** (*Algorithm 1.*) $\mathbf{A}(t)$, $\mathbf{I}(t)$

Fig.2. The flowchart of the implemented program.

Table 1. Running performance - First test case (35451 DoF).

N_p		2	3	4	5	6	7	8
	DoF	17726	11817	8861	7090	5906	5062	4429
	C.DoF	145	217	295	366	439	502	683
First test case	*Time*	2194.9	1904.3	1671.3	1739.8	1962.6	2231.2	2810.4
	NoIt	207	195	204	201	193	207	209

Table 2. Running perfrormance - Second test case (192747 DoF).

N_p		2	3	4	5	6	7	8
	DoF	96373	64249	48185	38548	32124	27533	24092
	C.DoF	513	648	834	1023	1340	1538	1793
Second test case	*Time*	53450	42888	29774	17698	9383.1	7245.4	6860.4
	NoIt	1146	1141	1127	1119	1086	1105	1069

Application

In this section, to demonstrate the operation of the presented methods, a 4-pole 3-phase 3kW induction motor with unskewed rotor slot fed by sinusoidal voltage is tested. The test problem and parameters it can find in [6], and the GMSH model is from the free GetDP models in [11].

The studied domain consists of one pole of the machine, i.e. a 45° domain, as you can see in Fig. 1. 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.

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 8×4GB 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 the Matlab [12].

We compare the implemented method for two different mesh size. The first mesh contains 73399 3-node triangular elements, which means 35451 nodes. The second test case contains 390887 3-node triangular elements, which means 192747 nodes.

In order to use the same stop criterion for the methods, $\varepsilon = 10^{-8}$. The speedup has been calculated by the following formula, $Speedup = Time_1 / Time_n$, where $Time_1$ is the running time of the sequential algorithm or the running time with least processor number, and $Time_n$ is the running time of the parallel algorithm executed on n processors [4].

In the tables the number of applied processor cores, N_p, the number of degree of freedom of one sub-domain, *DoF*, the number of interface unknown, *C.DoF*, the computation time (wall clock time) in seconds, *Time* and the number of iteration of the iterative solvers, *NoIt* are summarized.

For $1 \le N_p \le 8$, the performance results of the Schur complement method are reported in Fig. 3 for both test cases. The speedups are computed using the wall clock time of sequential calculations (first test case – 2914 sec, second test case – 62319 sec) as the reference point. The results show a speedup as high as 1.8 for the first test case, and 9 for the second test case. In the second test case, the speedup is higher and continuously increases, because the time of the interprocess communication is relatively smaller, than the time of the parallel PCG. This is not true for the first or smaller test case, when the subproblems are too smalls, and the overhead of the computation is too high, because of the communication between the processor cores.

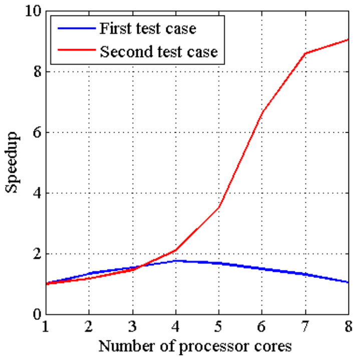

Fig.3. The speedup of the two test cases.

This conclusion is also supported by the *Time* and *C.DoF*, gives in Table 1 and 2. These tables show the data of running performance at different number of cores. The interprocess communication hardly depends on the number of interface unknowns and the number of applied processor cores. Furthermore, the number of iteration shows the robustness of the presented algorithm, because the *NoIt* is more or less independent from the number of degree of freedom and the number of interface unknown.

Figure 4-6 show the simulation results of the induction machine. These figures show the first ten periods of the simulations. Fig. 4 shows the transient speed waveform. Fig. 5 shows the transient torque waveform of the machine. Fig. 6 shows the transient current waveforms of the stator windings.

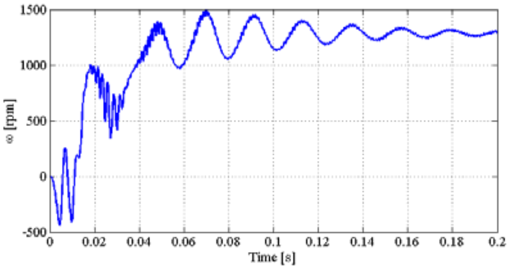

Fig.4. The speed variation in time.

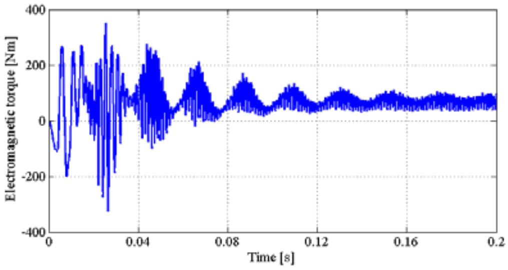

Fig.5. Electromagnetic torque variation in time.

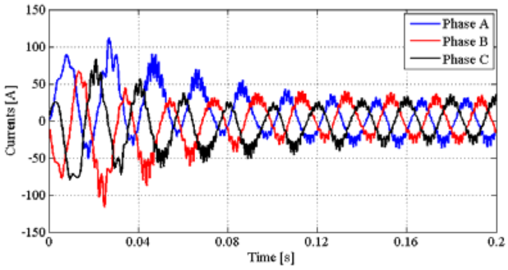

Fig.6. The time variation of the three phase currents.

Conclusions

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, two different mesh have been considered. Results of numerical experiments on both mesh compared. 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 small, the sequential computation is more efficient, than the parallel computation. However, if the problem is large enough, as in the second test case, the parallel computation is much faster than the sequential.

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.

This paper was supported by the TÁMOP-4.2.2.A-11/1/KONV-2012-0012: Basic research for the development of hybrid and electric vehicles - The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

REFERENCES

[1] Kuczmann M., Iványi A., *The Finite Element Method in Magnetics*, Akadémiai Kiadó, Budapest, 2008.
[2] Bastos J. P. A., Sadowski N., *Electromagnetic Modeling by Finite Element Methods*, Marcel Dekker, Inc., New York, 2003.
[3] Nikishkov G. P., Basics of the Domain Decomposition Method for Finite Element Analysis, in F. Magoulès, (Editor), *Mesh Partitioning Techniques and Domain Decomposition Methods*, Saxe-Coburg Publications, Stirlingshire, UK, Chapter 5, pp. 119-142, 2007.
[4] Kruis J., *Domain decomposition methods for distributed computing*, Saxe-Coburg Publication, Kippen, Stirling, Scotland, 2006.
[5] Marcsa D., Kuczmann M., Parallel solution of electrostatic and static magnetic field problems by domain decomposition method, *Przegląd Elektrotechniczny*, 89 (2013), nr 2b, 49-52.
[6] Gyselinck J., Vandevelde L., Dular P., Geuzaine C., Legros W., A general method for the frequency domain FE modelling of rotating electromagnetic devices, *IEEE Transactions on Magnetics*, 39 (2003), nr 3, 1147-1150.
[7] Vassent E., Meunier G., Foggia A., Reyne G., Simulation of Induction Machine Operation Using a Step by Step Finite Element Method Coupled with Circuit and Mechanical Equations, *IEEE Transactions on Magnetics*, 27 (1991), nr 6, 5232-5234.
[8] De Oliveira A. M., Antunes R., Kuo-Peng P., Sadowski N., Dular P., Electrical machine analysis considering field - circuit - movement and skewing effects, *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 23 (2004), nr 4, 1080-1091.
[9] Shi X., Le Menach Y., Ducreux J.-P., Piriou F., 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*, 25 (2006), nr 1, 17-30.
[10] Marcsa D., Finite Element Analysis of Solid-Rotor Induction Machine, *Acta Technica Jaurinensis*, 3 (2010), nr 2, 61-70.
[11] `http://onelab.info/wiki/Electric_Machines` (Visited - 2014.05.15.)
[12] Kuczmann M., Budai T., Kovács G., Marcsa D., Friedl G., Prukner P., Unger T., Tomozi Gy. Application of PETSc and other useful packages in finite element simulation, *Pollack Periodica*, 8 (2013), nr. 2, 141-148.

***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