ELSEVIER

Journal of Computational and Applied Mathematics 115 (2000) 51–61

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Parallel electromagnetic modelling for the nonlinear electrotechnic systems

M. Azizi[a], E.M. Daoudi[b, 1], R. El Hani[a], A. Lakhouaja[b,1, *]

[a] *UFR Eléctronique et Systèmes, Ecole Supérieure de Technologie, B.P. 473, Oujda, Morocco*

[b] *Département de Mathématiques et d'Informatique, Faculté des Sciences, Université Mohamed 1er, Oujda, Morocco*

Received 28 September 1998; received in revised form 2 July 1999

Abstract

We present a parallel implementation for the nonlinear electromagnetic modelling using PVM environment. Theoretical study shows that the computation can be highly parallelizable but the communication can be a handicap if we do not overlap communication with computation especially when the machine communication parameters values are very high.

Keywords: Electrotechnic systems; Parallel numerical algorithms and architecture; Electromagnetic modelling; Finite element method; Successive approximations method

1. Introduction

The validation of the electrotechnic systems by experimentation requires an important investment associated with some sophisticated measurement devices. The modelling offers extended possibilities for constructors and operators, it permits design optimization which leads to high performances [2,9,12]. In this context, the numerical computation of spatial and temporal distribution of the magnetic field in the electrotechnic systems, is based on the resolution of the equation of magnetic vector potential. The finite elements method (FEM) [5] is one of the more used methods for approximating this type of equation, it takes into account the saturation effects, the boundary conditions and also allows the study of domains with complex geometrical shapes [1]. In order to solve real problems, this method requires a large memory space and a prohibitive computation time. These two constraints can be avoided by using parallelism concepts [7,10].

* Corresponding author.

E-mail addresses: azizi@est.univ-oujda.ac.ma (M. Azizi), mdaoudi@sciences.univ-oujda.ac.ma (E.M. Daoudi), elhani@est.univ-oujda.ac.ma (R. El Hani), lakhouaja@sciences.univ-oujda.ac.ma (A. Lakhouaja)

[1] Supported by the European projects ITDC'94 "DOONS" and INCO-DC "DAPPI".

PII: S0377-0427(99)00307-6

In this work we study and present a parallel implementation of this method, taking into account the saturation effects, on a distributed memory architecture composed of p processors. For the linearization we have used successive approximations method. In Sections 2–4 we give a general presentation of the model. In Section 5 we present the sequential algorithm and give the complexity of the computation time. Sections 6 and 7 are devoted to the present parallel version of the algorithm with the theoretical complexities of the computation and the communication times; after this we present the experimental results obtained on a real parallel machine.

2. Magnetic vector potential formulation

2.1. Maxwell's equations

Maxwell's equations provide the mathematical basis for rigorous analysis of electromagnetic phenomenon [5]:

$$\operatorname{curl}(\boldsymbol{H}) = \boldsymbol{J} + \frac{\partial \boldsymbol{D}}{\partial t}, \tag{1}$$

$$\operatorname{curl}(\boldsymbol{E}) = -\frac{\partial \boldsymbol{B}}{\partial t}, \tag{2}$$

$$\operatorname{div}(\boldsymbol{B}) = 0, \tag{3}$$

$$\operatorname{div}(\boldsymbol{D}) = \rho \tag{4}$$

with $\boldsymbol{H}$ is the magnetic field intensity (A/m), $\boldsymbol{B}$ the magnetic induction (T), $\boldsymbol{E}$ the electric field intensity (V/m), $\boldsymbol{D}$ the electric induction (C/m^2), $\boldsymbol{J}$ the current density (A/m^2), and ρ the volumes charge density (C/m^3).

- Eqs. (1) and (2) express the coupling between the electric and magnetic sizes. Eq. (1) implies that a variation in the time of the electric induction creates a variation of magnetic field. In the same way, Eq. (2), called Faraday law, implies that a variation in the time of the magnetic induction provokes a variation of electric field.
- Eqs. (3) and (4) represent the fluxes conservation.

From Eqs. (1) and (4), we obtain the conservation of electric charge

$$\operatorname{div}(\boldsymbol{J}) + \frac{\partial \rho}{\partial t} = 0. \tag{5}$$

2.2. General electrotechnic case

In the electrotechnic domains, the frequency of the studied phenomenon is such that the displacement current ($\partial \boldsymbol{D}/\partial t$) and ($\partial \rho/\partial t$) could be neglected. In this case Eqs. (1) and (5) become

$$\operatorname{curl}(\boldsymbol{H}) = \boldsymbol{J}, \tag{6}$$

$$\operatorname{div}(\boldsymbol{J}) = 0. \tag{7}$$

The fields $\boldsymbol{H}$, $\boldsymbol{B}$, $\boldsymbol{E}$ and $\boldsymbol{J}$ satisfy additional relations which take into account material properties:

$$\boldsymbol{B} = \mu \boldsymbol{H}, \tag{8}$$

$$\boldsymbol{J} = \sigma \boldsymbol{E}, \tag{10}$$

where μ is the magnetic permeability (H/m), and σ is the electric conductivity (Ω^{-1}/m).

In this work, we assume that the conductivity is constant. Since we take into account the saturation effects, the permeability will be nonlinear, $\mu^{-1} = \nu(|\boldsymbol{B}|)$, where ν is the magnetic reluctivity.

Eq. (3) shows the existence of a magnetic vector potential $\boldsymbol{P}$ such that

$$\boldsymbol{B} = \operatorname{curl}(\boldsymbol{P}). \tag{11}$$

From this relation and Eq. (2), the electric field can be written as follows:

$$\boldsymbol{E} = -\frac{\partial \boldsymbol{P}}{\partial t} - \operatorname{grad}(V), \tag{12}$$

where V is the electric scalar potential. By using relations (8) and (10) and expressions (11) and (12), Eq. (6) becomes

$$\operatorname{curl}(\nu \operatorname{curl}(\boldsymbol{P})) + \sigma \frac{\partial \boldsymbol{P}}{\partial t} = -\sigma \operatorname{grad}(V). \tag{13}$$

2.3. 2D electrotechnic case

In the remaining part of this work, we only consider the two-dimensional problems. The 2D problems represent an invariance of the solution in one direction. They could be expressed in terms of the vector potential where its direction is fixed and only the modulus is unknown.

With a current density of excitation following Oz direction: $\boldsymbol{J}_0 = (0, 0, J_0) = -\sigma \operatorname{grad}(V)$, the direction of the vector potential also follows the same axis, $\boldsymbol{P} = (0, 0, P)$, and Eq. (13) becomes

$$-\operatorname{div}(\nu \operatorname{grad}(P)) + \sigma \frac{\partial P}{\partial t} = J_0 \quad \text{in } D, \tag{14}$$

where D is the study global domain delimited by a border Γ. The boundary conditions are those of Dirichlet (potential is fixed) or of Neumann (normal derivative of potential vanishes).

3. Discretization

3.1. Spatial discretization by FEM

In order to make a spatial discretization by FEM of Eq. (14), we use the weighted residual method which consists in replacing (14) by an integral shape [2]:

$$\int_D \phi \left[-\operatorname{div}(\nu \operatorname{grad}(P)) + \sigma \frac{\partial P}{\partial t} - J_0 \right] \mathrm{d}D = 0, \tag{15}$$

where ϕ is an arbitrary function, called test function with $\phi = 0$ on the Dirichlet boundary.

By application of the Green formula to Eq. (15) and taking into account the boundary conditions, we obtain

$$\int_D \nu \operatorname{grad}(\phi) \operatorname{grad}(P) \, \mathrm{d}D + \int_D \sigma \phi \frac{\partial P}{\partial t} \, \mathrm{d}D = \int_D \phi J_0 \, \mathrm{d}D. \tag{16}$$

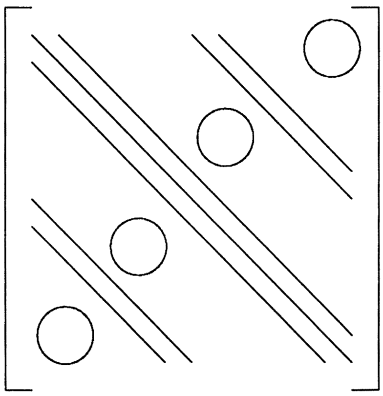

Fig. 1. Structures of matrices $[A]$ and $[B]$.

For the FEM discretization we proceed as follows:

- We partition D into subdomains D_e.
- In (16), we replace the test functions ϕ by a set of N independent functions ϕ_i, $1 \leqslant i \leqslant N$, where N is the number of nodes in the domain D and N_e is the number of nodes in each subdomain D_e.
- We replace P in every subdomain D_e, by a polynomial approximation:

$$P(x, y, t) = \sum_{j=1}^{N_e} \psi_j(x, y) P_j(t),$$

where ψ_i are the nodal approximation functions. The Galerkin method [5,11] leads to the following matrix equation (after insertion of the boundary conditions):

$$[A]\{P(t)\} + [B]\frac{\mathrm{d}\{P(t)\}}{\mathrm{d}t} = \{F(t)\}, \tag{17}$$

where $\{P(t)\}$ is the unknown potential vector of size N. Matrices $[A], [B]$ and vector $\{F(t)\}$ are obtained after assembling the following elementary matrices and vectors:

$$[A]^e_{i,j} = \int_{D_e} v_e \,\mathrm{grad}(\psi_i)\mathrm{grad}(\psi_j)\,\mathrm{d}D_e, \tag{18}$$

$$[B]^e_{i,j} = \int_{D_e} \sigma_e \psi_i \psi_j \,\mathrm{d}D_e, \tag{19}$$

$$\{F(t)\}^e_i = \int_{D_e} \psi_i J^e_0(t)\,\mathrm{d}D_e, \tag{20}$$

where $[A]$ and $[B]$ are sparse, symmetric and positive-definite matrices. These characteristics are exploited for the storage scheme.

With triangular finite elements, and using natural ordering of domain nodes, the structure of the matrices $[A]$ and $[B]$ is given in Fig. 1.

3.2. Temporal discretization

We note that Eq. (14) depends on the space and time. FEM is used to approximate the spatial derivation and the temporal derivation is approximated by the finite difference method. For this, we retained the implicit Euler method where the convergence and the stability are independent from the discretization step Δt. Then Eq. (17) becomes

$$\left([A]+\frac{1}{\Delta t}[B]\right)\{P(t+\Delta t)\}=\frac{1}{\Delta t}[B]\{P(t)\}+\{F(t+\Delta t)\}. \tag{21}$$

4. Saturation effects

In the nonlinear material, the magnetic reluctivity v is a nonlinear function of $|\boldsymbol{B}|$. In order to introduce this nonlinearity, the first magnetization curve is necessary and generally given by the constructor for increasing magnetic induction and fields. For the numerical modelling one must eliminate a continuous expression. From the different methods describing the curve $v(|\boldsymbol{B}|)$, we use the approach of Hecht and Marrocco [8]:

$$v(|\boldsymbol{B}|)=v_0\left(\varepsilon+(c-\varepsilon)\frac{|\boldsymbol{B}|^{2\alpha}}{|\boldsymbol{B}|^{2\alpha}+\tau}\right) \tag{22}$$

where $v_0=1/4\pi\times10^{-17}$ is the magnetic reluctivity of air, ε, c, α and τ are positive constants depending on the materials.

In this case, $[A]$ depends on the magnetic induction, therefore on the potential vector. Then it is necessary to use an iterative method. The two mainly used methods are the successive approximations and the Newton–Raphson.

4.1. Successive approximations method

The successive approximations algorithm of resolution of nonlinear systems (21) consists, at every instant $t+\Delta t$, in:

1. *Computation of vector* $\{F(t+\Delta t)\}$ *from* (20).
2. *Initialization:* $k=0$ *and* $\{P(t+\Delta t)\}^0=\{P(t)\}$.
3. *while err* $>$ *tol do* (*tol* is a given precision).
   - 3.1. *Computational of* $v(|\boldsymbol{B}|)$ *from* (22), *in each subdomain.*
   - 3.2. *Computation of the matrix* $[A]$ (18).
   - 3.3. *Computation of* $\{P(t+\Delta t)\}^{k+1}$ *from* (21) *by the Conjugate Gradient* (*CG*) *method.*
   - 3.4. *Approximation error:*

$$\text{err}=\frac{\|\{P(t+\Delta t)\}^{k+1}-\{P(t+\Delta t)\}^{k}\|}{\|\{P(t+\Delta t)\}^{k}\|}.$$

4. *End do.*

This method converges only for "small" values of the densities J_0. In the case where J_0 is "large" enough, we prefer to use the Newton–Raphson method.

We note that from Eq. (11), we compute the magnetic induction $\boldsymbol{B}$ in each triangle, then from Eq. (22) we obtain the reluctivity v, therefore v can be computed in terms of potential P in Section 3.1.

4.2. Newton–Raphson method

The ideal of this method is: if x^k is an approached solution of the equation $g(x)=0$ then a new approached solution x^{k+1} is given by the relation $g'(x^k)\Delta x^{k+1} = -g(x^k)$ with $\Delta x^{k+1} = x^{k+1} - x^k$.

In our case, we apply Newton–Raphson method to Eq. (20), and obtain

$$\left([A]^k + [NR]^k + \frac{1}{\Delta t}[B]\right)\{\Delta P\}_{t+\Delta t}^{k+1} = -\left([A]^k + \frac{1}{\Delta t}[B]\right)\{\Delta P\}_{t+\Delta t}^{k} + \frac{1}{\Delta t}[B]\{P\}_t + \{F\}_{t+\Delta t}. \tag{23}$$

The terms of the matrix $[NR]$ are given in every subdomain D_e by the following expression:

$$[NR]_{i,j}^e = 2\int_{D_e} \frac{\partial v(|\boldsymbol{B}|)}{\partial |\boldsymbol{B}|^2} \sum_{n=1}^{N_e} \operatorname{grad}(\psi_i)\operatorname{grad}(\psi_n)P_n \sum_{m=1}^{N_e} \operatorname{grad}(\psi_j)\operatorname{grad}(\psi_m)P_m \, \mathrm{d}D_e. \tag{24}$$

One notes that the matrices of system (23) remains band ones and symmetric positive definite. The algorithm of the resolution of this system remains the same as the one presented in Section 4.1, with the following modifications: at stage 3.2, we also compute the matrix $[NR]$ (24) and at stage 3.3 we solve the system (23) in order to calculate $\{P\}_{t+\Delta t}^{k+1}$. The convergence of this method is fast. When it exists, the error decreases quadratically. The choice of the initial guess determines the convergence.

5. Sequential algorithm

In this work we have only implemented the successive approximations method which can be easily adapted to the Newton–Raphson method. According to the algorithm of Section 4.1, at each step of time we need to compute the vector $\{P\}$ from relation (21) using the Conjugate Gradient method since the matrix is sparse, symmetric and positive definite. Note that matrix $[B]$ is computed one time while matrix $[A]$ is computed at each iteration of linearization.

Let I_{sa} denote the number of iterations of the successive approximations and I_{cg} the number of iterations of the conjugate gradient method.

a. *Computation of vector* $\{F(t)\}$ *from relation* (20) *requires* $7N$ *flops.*
b. *Computation of* $v(|\boldsymbol{B}|)$ *from relation* (22) *requires* $25N$ *flops.*
c. *Computation of the second member of Eq.* (21) *requires* $15N$ *flops.*
d. *Computational of the matrix* $[A]$ *from relation* (18) *requires* $22N$ *flops.*
e. *Computational of* $\{P(t)\}$ *from* (21) *by the conjugate gradient* (CG) *algorithm requires* $23N$ *flops in the initialization phase and* $28N$ *flops in each iteration, which means that it requires* $(23 + 28I_{\mathrm{cg}})N$ *flops.*
f. *Approximation error requires* $5N$ *flops.*

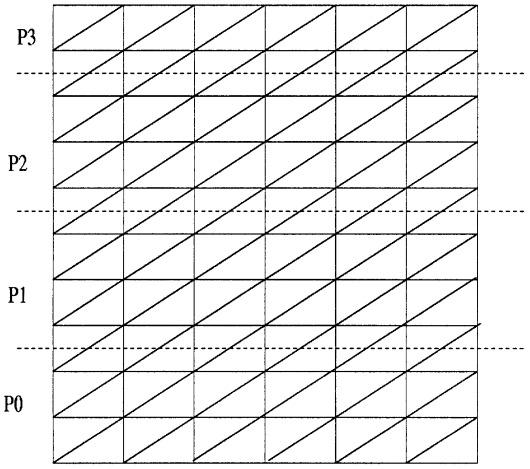

Fig. 2. Partitioning domain into strips.

We deduce that the execution time of the sequential successive approximations algorithm, at each step of time, is estimated to

$$T_{\text{seq}} = (7N + (90N + 28N * I_{\text{cg}}) * I_{\text{sa}}) * \omega,$$

where ω is the execution time for one *flop*.

6. Parallel implementation

In this section we present a parallel implementation of the successive approximations on a parallel distributed machine composed of p processors numbered by P_i, $0 \leqslant i \leqslant p-1$. The communication of m data between two neighbor processors is modeled by $\beta + m\tau$, where β is the start-up time, τ is the time to transmit one data. The parallelization consists in partitioning the global domain D into p strips each of size N_x by N_y/p, where N_x is the number of nodes in the x-axis, N_y is the number of nodes in the y-axis.

At each processor P_i, for $0 \leqslant i \leqslant p-1$, we affect one strip as shown in Fig. 2.

The computation of $\{F\}$ and $[A]$ is highly parallelizable since it does not need any communication between processors. In parallel, each processor computes one part of the matrix $[A]$ and the vector $\{F\}$. We deduce that the computation of $[A]$ and $\{F\}$ requires, respectively, $22N/p$ flops and $7N/p$.

In the following, two processors are neighbors if the strips assigned to these processors are neighbors, which means that each processor has two neighbors except the processors P_0 and P_{p-1}.

For the computation of the reluctivity $\nu(|\boldsymbol{B}|)$ in each subdomain one needs communication between neighbor strips since the calculation of the induction requires the values of the potential vector at each node of the subdomain. We deduce that the computation time is estimated to be $25N/p$ *flops* and the communication time is $2(\beta + N_x\tau)$.

The conjugate gradient method needs two inner products, one matrix-vector product and one vector update in the initialization phase and at each step, three major operations: two inner products, one

matrix-vector product and three vector updates of the form $y = y + \alpha x$ ($x, y \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$). For the parallelization of each operation is done as follows:

- *Inner product*: The computation of the inner product is done in two steps. Each processor:
  (1) Computes its local inner product.
  (2) Exchanges the local inner product with the other processors in order to compute the global inner product.

  Note that the performance of (2) depends on the architecture of the parallel machine. We deduce that the computation time is estimated to be $2N/p$ *flops* and the communication time is $\log_2(p)(\beta+\tau)$ for an hypercube.

- *Matrix-vector product*: Unlike the inner product where global communications are done, the matrix-vector product needs only communication between neighbor processors. For $0 \leqslant i \leqslant p-2$, P_i sends N_x data to P_{i+1} and for $1 \leqslant i \leqslant p-1$, P_i sends N_x data to P_{i-1}. We estimate the computation time to $18N/p$ *flops* and the communication time to $2(\beta + N_x\tau)$.
- *Vector updates*: The vector updates needs only local computation. The cost is $2N/p$ *flops*.

We deduce that, at each step of time: the computation time is estimated to be

$$T_{\text{comp}} = \frac{T_{\text{seq}}}{p},$$

and the communication time is estimated to be:

$$T_{\text{com}} = ((3 + I_{\text{cg}})(\beta + N_x\tau) + 2\log_2(p)(\beta + \tau)(I_{\text{cg}} + 1))I_{\text{sa}}.$$

When communication and computation do not overlap, the execution of the algorithm, at each step of time, is $T_{\text{par}} = T_{\text{comp}} + T_{\text{com}}$.

7. Experimental results

We have implemented this method on the distribution memory architecture TN310 of Telmat composed of 32 Transputers under PVM environment with real double precision. Using PVM, the estimated values for communication parameters are $\beta = 0.002$ s, $\tau = 22$ μs and for computation $\omega = 0.25$ μs. This shows that the communication for our system is very expensive. As a test example, we have considered the electromagnetic relay (Fig. 3) [3], taking into account the saturation effects and assuming that the current in the spool (electrical material) is constant. The system is placed in a box of air with Dirichlet conditions. The geometrical structure is symmetric following the y-axis; this leads to the treatment of only one half of the domain with Neumann condition on the symmetric axis. Fig. 4 shows the equipotential lines obtained for this example.

In Table 1 we compare the execution times obtained with different number p of processors and different number N of unknowns (Fig. 5).

As shown in Table 1, the use of p processors improves the execution time compared to the sequential one but the ratio sequential time over the parallel one is very small compared to the number of processors (we hope that this ratio will be equal to p).

In Table 2, we give the efficiency (efficiency $= T_{\text{seq}}/(T_{\text{par}} * p)$) which measures the rate of the use of p processors (see Fig. 6).

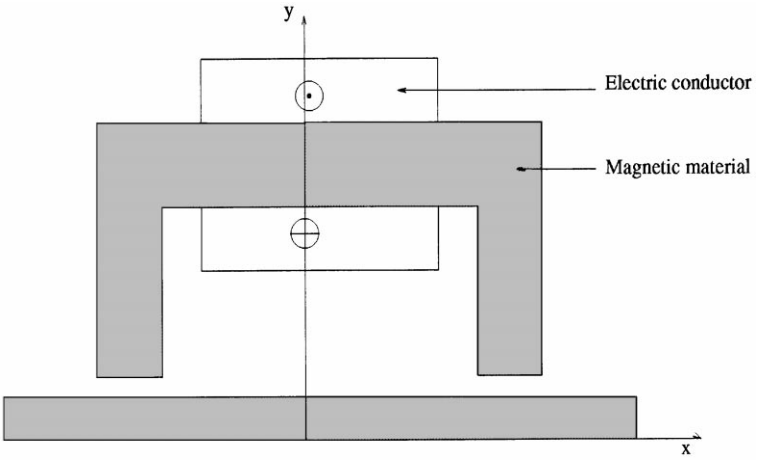

Fig. 3. Example of electromagnetic relay.

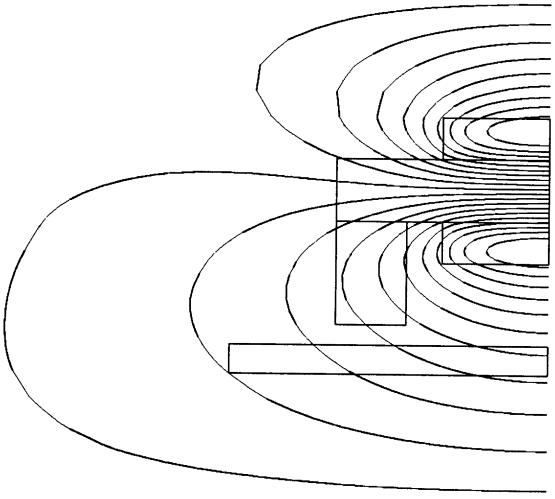

Fig. 4. Equipotential lines.

Table 1
Execution time for different number of processors and unknowns

Number of unknowns	Number of processors				
	1	2	4	8	16
3290	10.16	06.15	04.78	04.55	05.54
4410	13.82	07.96	05.78	05.35	06.42
5530	19.77	12.04	08.27	07.89	10.12
6650	30.94	24.58	16.48	16.40	22.29

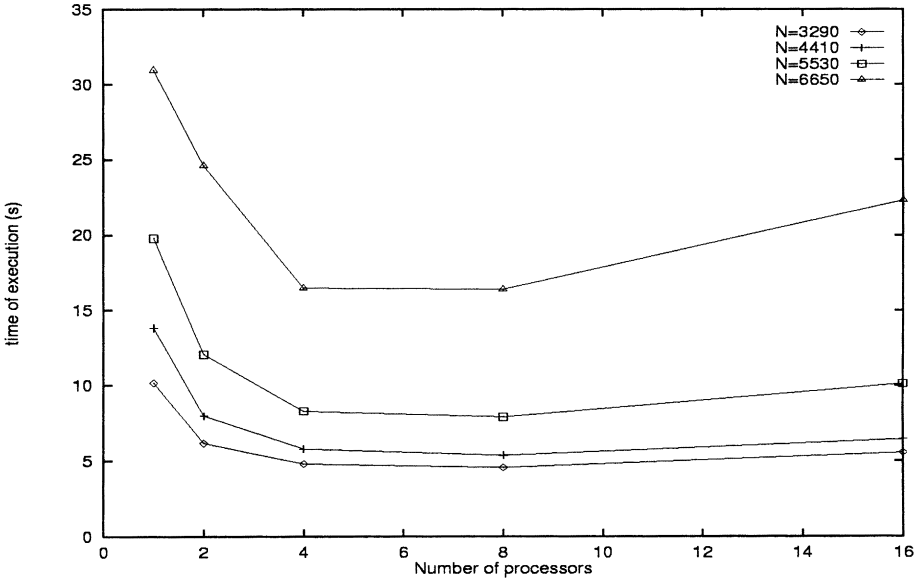

Fig. 5. Time of execution.

Table 2
Efficiency for different number of processors and unknowns

Number of unknowns	Number of processors			
	2	4	8	16
3290	82.59	53.08	27.91	11.45
4410	86.76	59.77	32.26	13.44
5530	82.07	59.73	31.31	12.20
6650	62.93	46.91	23.57	08.67

These results show clearly the penalty due to the communication. Notes that the communication time increases with the number of processors (factor $\log_2(p)$ in T_{com}) and also with the number of unknowns (I_{sa} and I_{cg} increase with N).

8. Conclusion and remarks

In this work we have presented a parallel version for the nonlinear electromagnetic systems. The effective implementation are done on a parallel distributed system based on transputers. The experimental results are not those desired since the communication parameter values of our system are very big. In order to improve the global execution time we will use the techniques to overlap the communication by the computation in the phases of the computation of the matrix-vector product and the magnetic induction. On the other hand, we will use the preconditioning techniques to improve the convergence of the CGM [4,6].

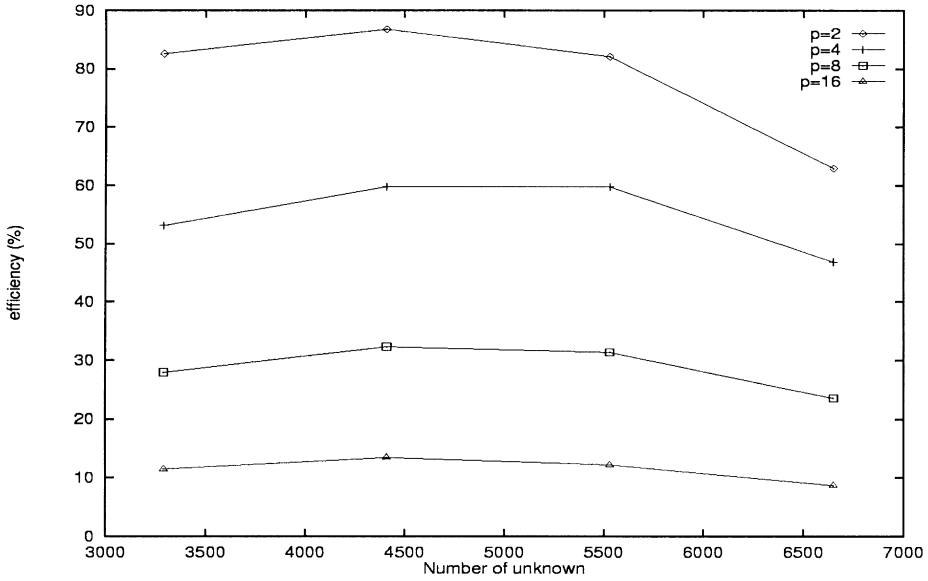

Fig. 6. Efficiency.

References

[1] M. Azizi, R. El Hani, Elaboration d'un code de calcul bidimensionnel pour la modélisation des systèmes électrotechniques complexes, Proceedings of the JNAEETSA, Marrakech, Morocco, October 1996, pp. 246–251.
[2] M. Azizi, R. El Hani, Les efforts électromagnétiques des machines polydiscoïdes à aimants alternés, Proceedings of the ISVANEM, Béthune, France, May 1998, pp. 129–133.
[3] F. Boualem, Contribution à la modélisation des systèmes électrotechniques à l'aide des formulations en potentiels: Application à la machine asynchrone, Ph.D. Thesis à l'UST de Lille, 1997.
[4] E.M. Daoudi, P. Manneback, Implementation of ICCG algorithm on distributed memory architecture, Iterative Methods in Linear Algebra, North-Holland, Amsterdam, 1992, pp. 339–347.
[5] R. Dautray, J.L. Lions, Mathematical Analysis and Numerical Methods for Science and Technology, Vol. 1, Springer, New York, 1990.
[6] H.C. Elman, Ordering technics for the preconditioned conjugate gradient method on parallel computers, Computers Physics Communications, Vol. 53, North-Holland, Amsterdam, 1989, pp. 253–269.
[7] I.T. Foster, Designing and Building Parallel Programs, Addison-Wesley, Reading, MA, 1995.
[8] F. Hecht, A. Marrocco, A finite element simulation of an alternator connected to an nonlinear external circuit, IEEE Trans. Magn. 26 (2) (1990) 964–967.
[9] T. Kawai, Role of computer simulation in scientific research and engineering development. Electromagnetic Phenomena and Computational Techniques, Elsevier, Amsterdam, 1992, pp. 3–18.
[10] A. Kumar et al., Introduction to Parallel Computing: Design and Analysis of Algorithms, Benjamin/Cummings, Menlo Park, CA, 1994.
[11] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.
[12] S. Williamson, Induction motor modelling using finite elements, RGE 8 (1994) 2–8.