A Survey of Parallel Solvers for the Finite Element Method in Computational Electromagnetics

B. Butrylo [(1)], F. Musy [(2)], L. Nicolas [(3)], R. Perrussel [(2,3)], R. Scorretti [(3)], C. Vollaire [(3)]

[(1)] *Bialystok Technical University*
ul. Wiejska 45A,
PL 15-893 Bialystok, Poland
bogb@cksr.ac.bialystok.pl

[(2)] *MAPLY, UMR CNRS 5005,*
Ecole Centrale de Lyon,
69134 Ecully cedex, France
Francois.Musy@ec-lyon.fr

[(3)] *CEGELY, UMR CNRS 5585,*
Ecole Centrale de Lyon,
69134 Ecully cedex, France
Laurent.Nicolas@ec-lyon.fr

Abstract

New trends in parallel methods used to solve Finite Element matrix systems are presented: standard iterative and direct solving methods first, and then domain decomposition methods. For example, the current status and properties of two prevailing programming environments (PVM and MPI) are finally given and compared when implemented together with a Finite Element Time Domain formulation.

Keywords

Finite element methods, Parallel algorithms, linear systems, electromagnetic analysis.

1. Introduction

Numerical computation is more and more used in engineering sciences to develop new devices or to optimize existing one. Only parallel computers provide the increase in computing performances necessary to solve today's problems. Two reasons may be highlighted: large memory is required because of a large amount of data, or speed is required to obtain the solution [1].

We have previously presented a survey of the parallelization of numerical techniques used in computational electromagnetics [2]. In this paper, it is proposed to focus now our attention on numerical solving methods for the parallelized Finite Element Method (FEM) in electromagnetics [3]. The objective is actually to describe lately appeared methods which can be useful to solve efficiently linear systems issued from FE discretization. The main characteristic of such systems is that they are generally sparse. The FEM is attractive for modeling 2D and 3D devices because of its adaptability for designing complex systems with highly heterogeneous materials. Modeling realistic devices requires the solution of a large

problem which can be afforded only by parallel computation [4].

The first section deals with standard numerical solving methods, such as preconditioned conjugate gradient. In the second section, domain decomposition methods (DDM) are described. Since these methods are less known in computational electromagnetics, their basic principle is presented. Special attention is also paid to multilevel methods, which can be seen as a DDM method from a strict mathematical point of view. As example, the last section shows the implementation of a Finite Element Time Domain (FETD) algorithm used for a wave propagation problem. A comparison between the two message passing libraries PVM and MPI is also performed.

2. Standard Parallel Solving Methods for the FEM

2.1. The Conjugate Gradient Method

The Conjugate Gradient (CG) method is one of the most popular linear solvers for Finite Element (FE) analysis, since it is well adapted to solve sparse matrix systems. In order to improve the solving, preconditioning techniques have to be used together with the solving method. The basic idea consists in replacing the initial system A x = b by a new system

$$Ax = b \Leftrightarrow \tilde{A}x = \tilde{b}$$
$$\text{with } \tilde{A} = M^{-1}A \text{ and } \tilde{b} = M^{-1}b$$

(1)

where M is the preconditioning matrix. Two conditions are required for efficiency:

- the conditioning number of the new matrix $\tilde{A}$ has to be lower than the initial conditioning number,

- and the matrix system M has to be to not expensive to build or to solve.

Several preconditioning techniques exist: Incomplete Cholesky (ICP), SSOR, Diagonal (DP) preconditioning. The following example shows the efficiency of the preconditioning method. It is used for the modeling of radiofrequency hyperthermia [5]. The problem is computed with more than 130 000 degrees of freedom. From Table I, by comparing both DP and ICP, it is shown that better results are obtained with the ICP in terms of number of iterations. On the other hand, since the cost of one iteration is greater than when using the DP, the total CPU time of computation is larger. In this case, the best results are obtained with a SSOR preconditioning: both number of iterations and CPU time are divided by two.

TABLE I: COMPARISON OF THE EFFICIENCY OF SEVERAL PRECONDITIONING TECHNIQUES FOR THE CG METHOD

Preconditioning method	Number of iterations	CPU time
Diagonal	13561	22356 s

Incomplete Cholesky	7551	25761 s
SSOR	3481	12225 s

Several papers published recently propose new parallel-processing techniques of the preconditioned CG solver.

In [6], a parallelized transient eddy current FE analysis is presented. Due to the presence of moving conductors, the right-hand side of the system of equations is changed at each time step. First-order brick-type edge elements are used for the FE discretization. Parallel solving is performed using the ICCG, leading to difficulties due to the forward-backward substitutions. Two parallel implementations are compared. First, on N_p processors, the block ICCG (BICCG) method divides the global matrix into N_p submatrices and performs the IC factorization for the local submatrix in each processor, with the entries between different processors being ignored in the factorization. Second, a renumbering process allows to reorder the matrix into a form similar to the dissection ordering case (PICCG-RP) (fig. 1). Since no entries are neglected during the preconditioning step, a better convergence rate is observed with the PICCG-RP method, leading to a better parallel efficiency (fig. 2).
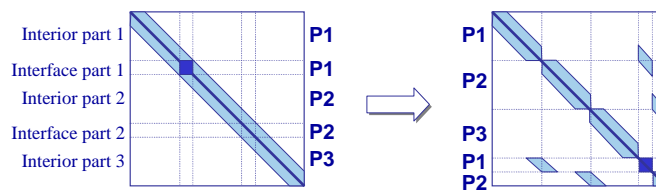

Fig. 1. Basic principle of the renumbering process [6].

This reordering technique is extended in [7] for the same 3D eddy-current analyses. The key of the method is coloring of the unknowns. Several colors are assigned to the unknowns, such that the unknowns belonging to the same color must have no data dependency on each other. Due to the presence of edge elements, the assignment is algebraic: it is carried out by using only information obtained from a coefficient matrix. This method is actually a complete black-box solver, and its implementation appears to be easy. Parallel performances obtained on a distributed memory Fujitsu VPP-800 are also better than when using a classical BICCG (fig. 3). It is also shown that the use of many colors improves the convergence rate, but it causes an increase in communication costs and a decline in parallelism. For the example of transient eddy currents analysis, made of more than 10^6 unknowns, best results are obtained with 60 colors.
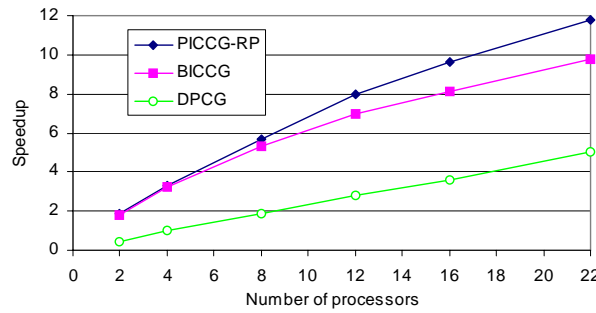
Fig. 2. Comparison of several preconditioning techniques for the CG solver [5]. Computation is performed on a Hitachi SR2201.
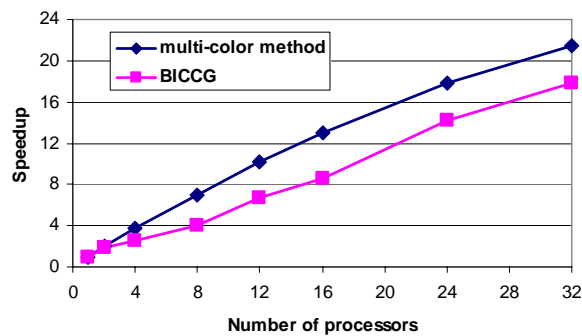

Fig. 3. Comparison of multi-color method for the CG solver and BICCG [6] for a transient eddy currents problem (1011920 unknowns).

The Conjugate Gradient Squared (CGS) is also used for the modeling of 2D open-region frequency-domain scattering problems by the FEM in [8]. By modifying the computation sequence in the algorithm, the synchronization overhead is reduced by a factor of two. The basic idea of this modification is to merge the inner products present in the CGS algorithm so that they can be evaluated using a single global summing operation per iteration of the "while" loop. From both theoretical and experimental analyses it is found that this Modified CGS (MCGS) algorithm becomes slightly faster as the number of processors increases. A set of algorithms is then proposed, where either CGS or MCGS is selected depending on the number of unknowns and number of processors.

2.2. Other parallel standard iterative methods

Other parallel iterative solvers may be used for the FE method and can be more efficient than the preconditioned CG. As example, the Quasi-Minimal Residual (QMR) method and the BI-Conjugate Gradient STAbilized (BICGSTAB) method are compared in [9] for the computation of 2D axisymmetric magnetic and electric fields on printed circuit board (PCB) containing magnetic films. Both solvers and

preconditioners are parallelized by matrix and vector partitioning, the mesh partition being obtained by using METIS [10]. The results obtained on different parallel machines show the efficiency of this approach (Table II).

TABLE II: SPEEDUP ON 48 PROCESSORS FOR A 16593 DOF MATRIX [9]. COMPUTATION ON A NEC CENJU-4.

Solver	Speedup
ILUT perconditioned BICGSTAB	14.4
Diagonaly scaled BICGSTAB	22.9
Diagonaly scaled QMR	31.9

2.3. Direct methods

Iterative solvers are not suitable for any type of problem. For example, the matrix equations obtained from the FEM with perfect matched layers (PML) can be rather ill-conditioned and can lead to very slow or non-convergent results. Direct solvers have then to be employed, since they are much more robust, reliable and efficient. As example, a sparse LU decomposition solver is presented in [11]. It is used to compute 3D microstrip discontinuities. Several implementation techniques are used to speedup the computation: dynamic partial pivoting, reordering scheme, supernodal approach. The parallel version of the package is implemented on a PC-based parallel platform, leading to decent speedups (fig. 4).
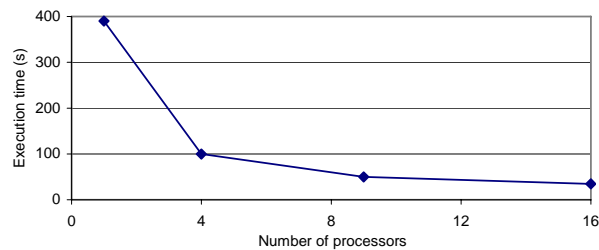

Fig.4. LU decomposition for the computation of 3D microstrip discontinuities (21408 dof) [11].

3. Domain Decomposition Methods

A natural way to create solvers for partial differential equations on parallel computers is the decomposition technique based on a decomposition of the spatial domain of the problem into subdomains. Most domain decomposition methods can be classified as either overlapping or nonoverlapping subdomain approach. For storage and time efficiency, iterative algorithms are preferred to direct solvers. To make the methods numerically scalable, a global coupling between subdomains must be introduced, using a coarse global mesh for example.
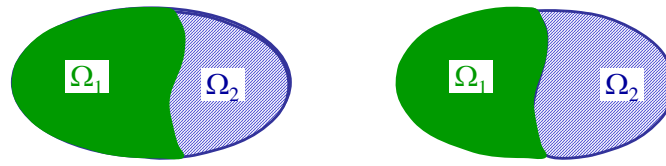
Fig.5. Overlapping (left) and nonoverlapping (right) domains.

3.1. Overlapping domains methods

Overlapping domains methods are easy to implement as additive Schwarz iteration. In this method, the computation is first performed on each subdomain with known boundary condition and with unknown boundary condition on the interface with other subdomains. From this computation, new boundary conditions are obtained on the interfaces and transferred to the other subdomains. This process is made iteratively until convergence (fig. 6). On each subdomain, an exact solver (LU) or an iterative solver can be used. Of course, this method is easy to parallelize, since each subdomain can be computed in parallel. So it is well adapted to coarse grain parallelism.
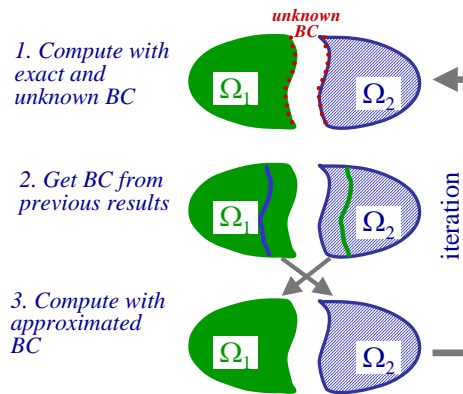

Fig.6. Basic idea of the additive Schwarz iteration.

On the other hand, the performance of multiplicative Schwarz iteration is often better, since the unknown boundary conditions are faster approximated (fig.7). However this method seems to be more difficult to parallelize, since it appears to be sequential by nature. The technique of multicoloring must then be used to improve its parallel efficiency .
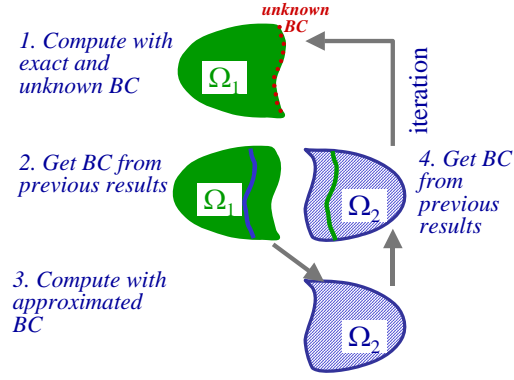
Fig.7. Basic idea of the multiplicative Schwarz iteration

3.2. Non overlapping domain methods

Algorithms with non overlapping domains involve Schur complement systems. If the system matrix is made of four blocks, the Schur complement matrix S is defined by:

$$\begin{bmatrix} A & B \\ B' & C \end{bmatrix} \Rightarrow S = C - B'A^{-1}B \quad (2)$$

It is then shown that solving the global system leads first to solve the Schur system:

$$\begin{bmatrix} A & B \\ B' & C \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \Rightarrow \begin{cases} x = A^{-1}b_1 - A^{-1}By \\ (C - B'A^{-1}B)y = b_2 - B'A^{-1}b_1 \end{cases} \quad (3)$$

The unknowns located on the interface separating two neighbouring subdomains give the Schur complement system. Since the subdomains Ω_i become then disconnected, the corresponding block matrices A_i become also disconnected, allowing an easy parallelization (fig. 8). The Schur matrix system is solved by the CG method for symmetric definite matrix or more generally by Krylov subspace method such as GMRES. Parallel interface preconditioners are often introduced to improve the convergence. Many ways for preconditioning exist in the literature for 2D problems. Preconditioners which not destroy the original scalability and which are not expensive are complicated to obtain for complex geometries and irregular discretizations.
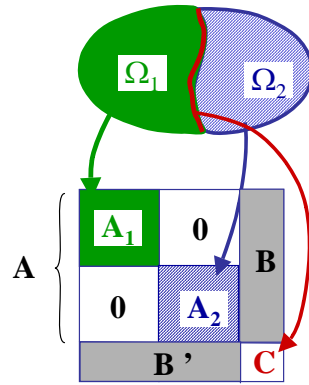
Fig. 8. Basic idea of the Schur complement system method.

A smart manner to obtain non overlapping domain decomposition algorithm showing a good parallel efficiency is to enforce the continuity of boundary conditions by introducing Lagrange multipliers [12]. The FE Tearing and Interconnecting method is an important class of such hybrid methods [13]. It can be viewed as a dual Schur Complement method . The Lagrange multipliers are solution of a small-size linear system compared to the global problem size. When solved with an iterative method (CG or BiCG method), the corresponding matrix does not need to be explicitly assembled. Each local field problem can be treated independently on each processor. In [14] the resulting algorithm for the solution of the vector wave equation discretized with tetrahedral elements is presented . It is used to modelize 3D PEC wedge in a rectangular cavity. The continuity of tangential electric fields on the interface between subdomains is used to build the Schur system. This system is calculated using parallel conjugate gradient, while the local matrices A_i factorized only once. From Fig. 9, this algorithm is shown to be highly scalable.

An other interesting advantage of the FETI method follows from [15]. For the solution of Maxwell's equation in the frequency domain, its performance is less affected by the presence of perfectly matched layers solution than a direct iterative method (a divide-and-conquer-type preconditioned BiCGM). On the other hand, the convergence of the method is sensitive to irregular subdomains. The question of constructing efficient parallel preconditionners is also a remaining problem for such a method.
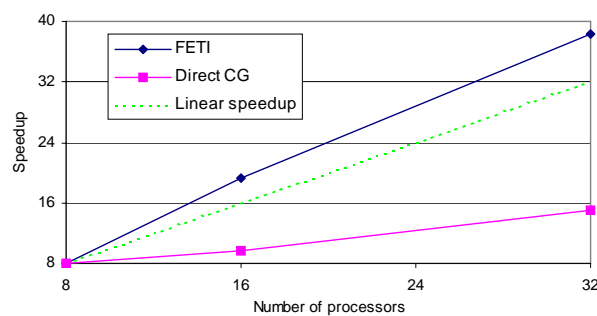

Fig.9. Speedup of the FETI algorithm for the computation of a PEC wedge into a cavity [13]. Computation is performed on a 32-node Intel iSPC/860 hypercube.

The domain decomposition approach offers many opportunities for the coupling of boundary element method (BEM) and FEM. An unified variational formulation has been proposed in [16]. In [17, 18], the z-component of the vector potential is determined from the solution of linear, symmetric but indefinite system of coupled equations which are reformulated as symmetric and positive definite systems. A different way to exploit the BEM-FEM coupling is presented in [19] for 3D transient problem. In this case, the coupling with the BEM is made necessary due to the presence of moving part into the electromechanical device. The developed method is based on a Dirichlet-Neumann algorithm: sequential solutions of the FEM part are linked within a preconditioned GMRES method together with parallel solution of the BEM part (fig. 10). On a NEC-SX4 with 16 processors, the obtained speedup is about 6.
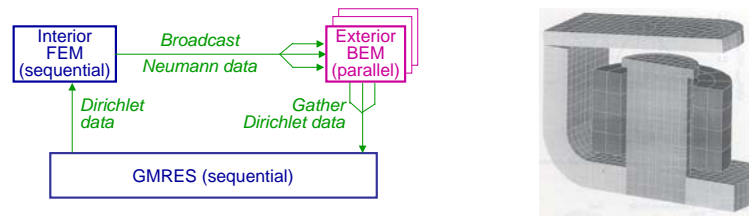

Fig. 10. BEM-FEM coupling (left) to modelize electromechanical device (right) with moving part [19].

3.3. Multilevel methods

Multilevel methods can be included in the class of decomposition method, since these algorithms have similarities with Schwarz type domain decomposition method with inexact solvers. Different grid levels or subspaces play the role of subregions. Schematically, each iteration may be seen as performed into 3 steps (fig. 11). A few iterations are first performed on a fine grid, leading to an approximative solution u_h. This leads to a residual vector r_h which is transferred to the coarse grid and then exactly computed (second step). The solution is finally re-transferred onto the fine grid, allowing to modify the previous values of u_h. It is followed by a post-smoothing step.

Multilevel methods can be generalized to several grid sizes. Two types of multigrid methods exist [20]:

- the Geometric Multigrid Method (GMM), based on the effective use of mesh grids,

- the Algebraic Multigrid Method (AMG), where there is no mesh, and the building of the operators is based on the values of the coefficients of the matrix.
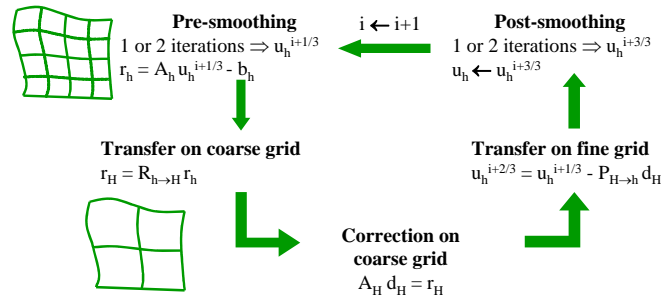
Fig. 11. Basic idea of the multilevel method (example with 2 grids).

Parallel implementation of the AMG is presented in [21] for eddy-current analyses and magnetostatics analyses. The parallel performance of the algorithm is improved in the setup phase by the use of long-range interpolation instead of the conventional direct interpolation. The numerical results obtained with a rectangular domain show that the AMG is a fast solver, since the CPU time is divided by 10 compared to a classical ICCG (Table III). As shown in Fig. 12, this intrisic property is not destroyed by the parallelization of the method, since both speedups remain comparable.

TABLE III: COMPARISON OF CPU TIMES (IN S) OBTAINED WITH THE AMG AND ICCG SOLVERS ON A 80601 DEGREES OF FREEDOM PROBLEM.

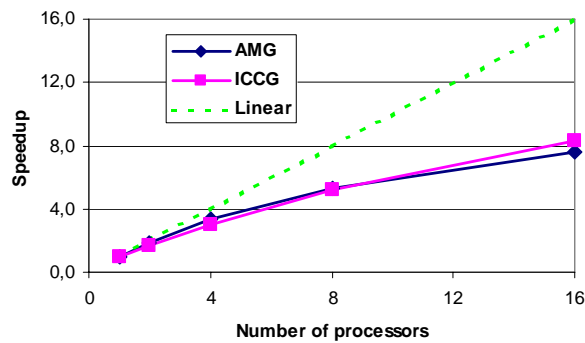| Number of processors	AMG	ICCG
1	513	5219
2	273	3163
4	153	1748
8	97.2	998
16	67.5	630

Fig. 12. Speedup of AMG and ICCG methods [19].

It would be of great interest to test the AMG method with more complicated domains, since the GMG requires a hierarchy of coarse grids which is actually difficult to manage in 3D. An experience of implementation of the GMG has been reported in [22] for 3D magnetostatics. Note that all these methods (domain decomposition or multigrid methods) can be used as preconditioners for the Conjugate Gradient algorithm [23].

4. An example: the Finite Element Time Domain method for wave propagation

Flexibility of the finite element technique and right physical sense of the edge elements make this formulation useful in modeling of electromagnetic phenomena. High performance simulation of the time domain problem enables to reduce either memory cost or time of computation. As example of implementation, the properties of the vector finite element time domain algorithm are evaluated in the known distributed multi-computer environments MPI and PVM [24-27].

4.1. Finite Element formulation

The distribution of electric field is given by time dependent vector wave equation

$$\nabla \times \frac{1}{\mu} \nabla \times \mathbf{E} + \sigma \frac{\partial \mathbf{E}}{\partial t} + \varepsilon \frac{\partial^2 \mathbf{E}}{\partial t^2} + \frac{\partial \mathbf{J}_{imp}}{\partial t} = 0, \qquad (4)$$

where μ, σ, ε are the parameters of the media and J_{imp} is the source current. The domain of analysis is truncated with first order Engquist-Majda absorbing boundary condition (ABC) [27]. In the case of the excitation by a plane wave, according to general Galerkin scheme, the weak form of eq. (4) is given by expression

$$\int_V \frac{1}{\mu} (\nabla \times \mathbf{E})(\nabla \times \mathbf{W}) dV + \int_V \mathbf{W} \sigma \frac{\partial \mathbf{E}}{\partial t} dV + \int_V \mathbf{W} \varepsilon \frac{\partial^2 \mathbf{E}}{\partial t^2} +$$
$$\int_{S_{ABC}} \mathbf{W} \frac{1}{\mu c} \left(\frac{\partial \mathbf{E}}{\partial t} \times \vec{\mathbf{n}} \right) dS = \int_{S_{ABC}} \mathbf{W} \left[\frac{1}{\mu c} \left(\frac{\partial \mathbf{E}^i}{\partial t} \times \vec{\mathbf{n}} \right) - \vec{\mathbf{n}} \times \nabla \times \mathbf{E}^i \right], \qquad (5)$$

where $\mathbf{W}$ is the test vector function, S_{ABC} is the external surface of the model and E^i is the incident field. This equation is discretized in time domain and in space domain to yield a system of linear equations which must be solved. Space discretization is achieved using incomplete first order $H(curl, \Omega)$ tetrahedral edge elements. Considering the central Euler difference approximations of the first and the second order derivatives, the final form of the equation is

$$\left(\mathbf{T} + \frac{\Delta t}{2} \mathbf{R} \right) \cdot \mathbf{E}_{n+1} =$$
$$\left(2\mathbf{T} - \Delta t^2 \mathbf{S} \right) \cdot \mathbf{E}_n + \left(\frac{\Delta t}{2} \mathbf{R} - \mathbf{T} \right) \cdot \mathbf{E}_{n-1} - \Delta t^2 \mathbf{f}_n, \qquad (6)$$

where $\mathbf{R}$, $\mathbf{T}$, $\mathbf{S}$ are mass, damping and stiffness matrices. The $\mathbf{f}_n$ vector represents the dynamic load in

the analyzed model.

4.2. Distributed implementation of the formulation

The Single Program Multiple Data (SPMD) paradigm is used in the elaborated parallel code. Because the unknowns are connected with the edges, the set of edges is decomposed in the distributed algorithm. Since the matrices are assembled with respect to the degrees of freedom, there are no geometrical restrictions of the data decomposition.

Each computing unit assembles the matrices only for its local subset of degrees of freedom. During the assembling stage, the processing units have to exchange information only about the spatial distribution of boundary conditions (fig. 13). The numerical integration of (5) requires the solution of the linear system $\mathbf{A}\ \mathbf{x}_n = \mathbf{b}_n$ (fig. 14). Conjugate gradient method with diagonal preconditioning is used. The basic matrix and vector operations in the solver procedure are parallelized. The interdependence of the processes is different in the assembling stage and in the time integration loop. The assembling thread in a processing unit is loosely connected with the others threads, while the subtasks in the distributed implementation of the CG algorithm are tightly coupled.
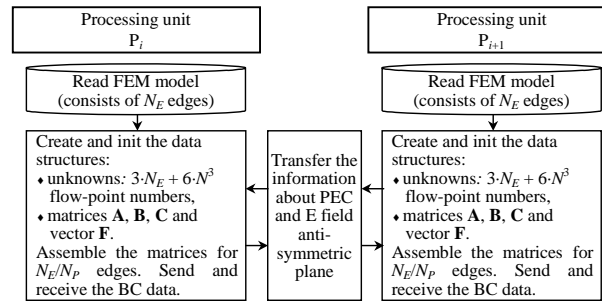
Processing unit P_i		Processing unit P_{i+1}
Read FEM model (consists of N_E edges)		Read FEM model (consists of N_E edges)
Create and init the data structures: ♦ unknowns: $3 \cdot N_E + 6 \cdot N^3$ flow-point numbers, ♦ matrices **A**, **B**, **C** and vector **F**. Assemble the matrices for N_E/N_P edges. Send and receive the BC data.	Transfer the information about PEC and E field anti-symmetric plane	Create and init the data structures: ♦ unknowns: $3 \cdot N_E + 6 \cdot N^3$ flow-point numbers, ♦ matrices **A**, **B**, **C** and vector **F**. Assemble the matrices for N_E/N_P edges. Send and receive the BC data.

Fig. 13. Relation between two processes in the assembling stage.

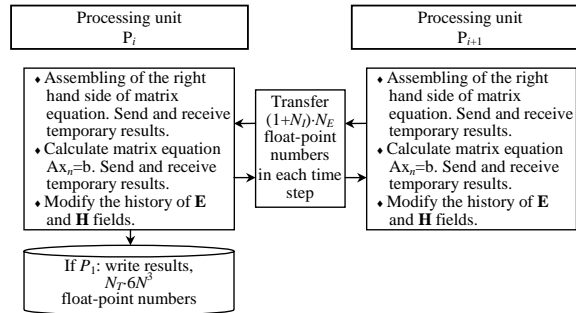| Processing unit P_i		Processing unit P_{i+1}
♦ Assembling of the right hand side of matrix equation. Send and receive temporary results. ♦ Calculate matrix equation $Ax_n=b$. Send and receive temporary results. ♦ Modify the history of **E** and **H** fields.	Transfer $(1+N_I) \cdot N_E$ float-point numbers in each time step	♦ Assembling of the right hand side of matrix equation. Send and receive temporary results. ♦ Calculate matrix equation $Ax_n=b$. Send and receive temporary results. ♦ Modify the history of **E** and **H** fields.
If P_1: write results, $N_T \cdot 6N^3$ float-point numbers		

Fig. 14. Relation between two processes in the time integration loop.

4.3. Comparison of PVM and MPI performances

Both implementations of the FEM algorithm with PVM and MPI have been tested on an heterogeneous cluster of 4 PC. The modelized problem is the propagation of a plane wave in free space.

Several mesh sizes have used for the comparison. The total time of execution of the PVM version is about 20% longer than the MPI implementation of the algorithm. However the relative parameters of PVM implementation are better (fig. 13, 14). The speedup of MPI and PVM implementations depends on the numbers of unknowns in the analysed model. If the relation between time of computation and time of data transfer is small (i.e. for relatively small model) the PVM and MPI implementations do not cause the speedup of the calculation process. The performance of the MPI version is shown to be a non-monotone function. The model of medium size gives the best speedup of the computation (about 2.7 for 4 processing units). The speedups of PVM and MPI are approximately equal only for this type of the FEM model. The efficiency and speedup of MPI version decrease when the limit of RAM memory in the cluster is achieved. The efficiency of MPI version converges to 0.5 for the largest FEM model. In this case the speedup and efficiency of PVM implementation achieve the ideal values.
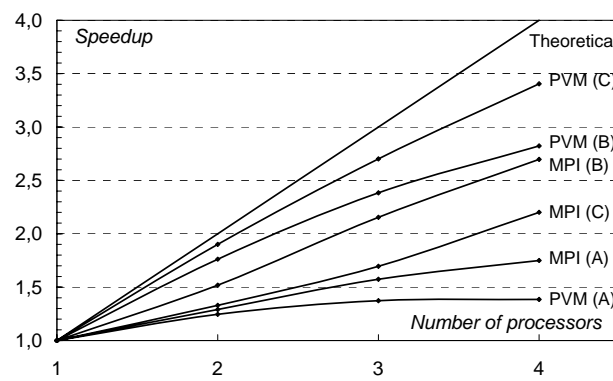

Fig. 13. Speedup of the FETD algorithm**-**A: 6930 DOF, B: 77832 DOF, C: 187488 DOF.
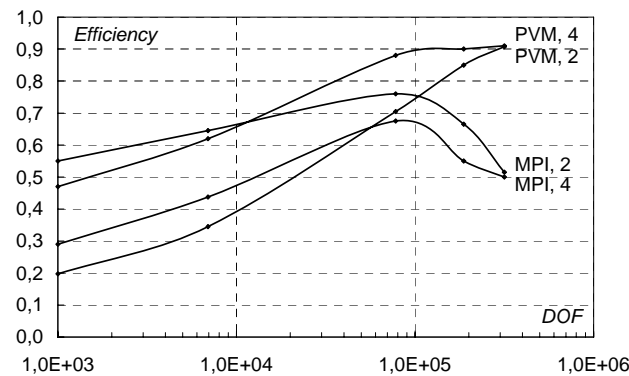

Fig. 14. Efficiency of MPI and PVM implementations for cluster with 2 and 4 processing units.

These results indicate better numerical performance of PVM version of the presented FETD algorithm. However, the configuration of the hardware seems to have significant influence on the performance of the parallel algorithm.

7. Conclusion

It appears from this review that the parallelization of standard solving methods, such as Conjugate Gradient method preconditioned with classical technique (SSOR, Incomplete Cholesly, …), has reached a sufficient maturity level. On the other hand, progress have to be made regarding the use of new methods, such as domain decomposition techniques or multigrid solvers. These methods may be used themselves as preconditioning techniques for Krylov subspace methods. Domain decomposition methods may be classified into overlapping and non-overlapping methods. Regarding the overlapping methods, additive Schwarz method is easy to parallelize, and it is well adapted to coarse grain parallelism. On the other hand, multiplicative Schwarz iteration is often better, but it appears more difficult to parallelize. Non overlapping domains involve Schur complement systems, and a good parallel efficiency may be obtained by introducing Lagrange multipliers to enforce the continuity of boundary conditions. Multilevel methods can be included in the class of decomposition method. These methods are generally efficient when implemented on sequential computer, and only some parallel implementations have been reported yet. However more complex geometries and domains have to solved in order to obtain more significant results.

Like in many other disciplines, computational electromagnetics requires a computing power considerably higher than that offered by today's conventional supercomputers, and parallel computing is destined to have a large influence in the next scientific developments.

8. References

[1] R.K. Agarwal, "Parallel Computers and Large Problems in Industry," Computational Methods in Applied Sciences, Elsevier Science Pub., pp. 1-11, 1992.

[2] T. Jacques, L. Nicolas, C. Vollaire, "Le calcul de champs électromagnétiques sur architectures MIMD," Revue Internationale de Génie Electrique, vol. 2, no2/1999, pp. 185-214.

[3] B. Butrylo, F. Musy, L. Nicolas, R. Scorretti, C. Vollaire, "New trends in parallel electromagnetic fields computation", International Conference on Parallel Computing in Electrical Engineering, Varsaw, Poland, 22-25 september 2002.

[4] C. Vollaire, L. Nicolas, A. Nicolas: "Parallel Computing for the Finite Element Method", The European Physical Journal Applied Physics, vol.1, 1998, pp. 305-314.

[5] N. Siauve, L. Nicolas, C. Vollaire, C. Marchal: "3D Modeling of the SAR Distribution for RF Hyperthermia with External Waveguide Applicator", COMPUMAG 2001, 13th Conference on the Computation of Electromagnetic Fields, Evian, 2/7/01-5/7/01

[6] T. Iwashita, M. Shimasaki, "Parallel Processing of 3-D Eddy Current Analysis with Moving Conductor Using Parallelized nICCG Solver with Renumbering Process," IEEE Trans. on Mag., vol. 36, no 4, july 2000, pp. 1504-1509.

[7] T. Iwashita, M. Shimasaki, "Algebraic Multicolor Ordering for Paralleized ICCG Solver in Finite-Element Analyses," IEEE Trans. on Mag., vol. 38, no 2, march 2002, pp. 429-432.

[8] M. Maheswaran, K.J. Webb, H.J. Siegel, "MCGS: a Modified Conjugate Gradient Squared Algorithm for Nonsymmetric Linear Systems," *The Journal of Supercomputings*, 14, 1999, pp. 257-280.

[9] T. Iwashita, M. Shimasaki, "Parallel Solvers for Electromagnetic Field Computations on Printed Circuit Boards," IEEE Trans. on Mag., vol. 36, no 4, july 2000, pp. 1504-1509.

[10] G. Karypis, V. Kumar, "Analysis of Multilevel Graph Partitioning," Univ. of Minnesota, Dept of Computer Science, Tech. Rep. 95-037, 1995.

[11] X.Q. Sheng, E.K.N. Yung, C.H. Chan, "Parallel Electromagnetic Modeling of 3D Microstrip Discontinuities using FEMand PML," John Wiley and Sons, 2001, pp. 38-47.

[12] F.I. Zyserman, J. E. Santos, "Parallel finite element algorithm with domain decomposition for three-dimensional magnetotelluric modelling," Journal of Applied Geophysics 44 (2000) 337-351

[13] C. Farhat and F. X. Roux, "A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm, " Int. J. Numer. Methods Eng. , vol. 32, 1991, pp. 1205-1227.

[14] U. D. Navsariwala and S.D. Gedney, "An Efficient Implementation of the Finite-Element Time Domain algorithm on parallel Computers using a Finite-Element Tearing and Interconnecting Algorithm," Microwave and Optical Technology Letters/Vol. 16, no 4, Nov. 1997.

[15] C. T. Wolfe, U. Navsariwala and S. D. Gedney, "A parallel Finite-Element Tearing and Interconnecting Algorithm for Solution of the Vector Wave Equation with PML Absorbing Medium, " IEEE Trans. on Antennas and Propagation, Vol. 48, no 2, Feb. 2000.

[16] M.Costabel, "Symmetric methods for the coupling of finite elements and boundary elements," in: C.A. Brebbia, W.L. Wendland and G. Kuhn, eds., *Boundary Elements IX*, Springer, Berlin,1987, pp. 411-420.

[17] B. Heise, "Nonlinear simulation of electromagnetic fields with domain decomposition methods on MIMD parallel computers," Journal of Computational and Applied Mathematics 63, 1995, pp. 373-381.

[18] M. Kuhn, "The application of coupled BE/FE formulations in technical magnetic field computations," Comput. Methods Appl. Mech. Engrg. 157 (1998) 193-204.

[19] V. Rischmüller, M. Haas, S. Kurz, and Wolfgang M. Rucker, "3D Transient Analysis of Electromagnetic Devices Using Parallel BEM Coupled to FEM," IEEE Trans. on Magnetics, Vol. 36, No. 4, July 2000.

[20] J. Ruge and K. Stüben, "Algebraic multigrid," in *Multigrid Method*, S. McCormick, Ed. Philadelphia, PA: SIAM, 1986.

[21] T. Mifune, T. Iwashita, and M. Shimasaki, "A Fast Solver for FEM Analyses Using the Parallelized Algebraic Multigrid Method," IEEE Trans. on Mag., vol. 38, no 2, March 2002.

[22] G. Haase, M. Kuhn, U. Langer, "Parallel multigrid 3D Maxwell solvers", Parallel computing, 27, 2001, pp. 761-775.

[23] J. Xu, "Iterative methods by space decomposition and subspace correction," SIAM Review, no 34, 4, 1992, pp. 581-613.

[24] R. Buyya, *High Performance Cluster Computing*, vol. 2, Prentice Hall PTR, New Jersey, USA, 1999.

[25]   G.A. Geist, J.A. Kohl, P.M. Papadopoulos, PVM and MPI: a Comparison of Features, *Calculateurs Paralleles*, 1996, vol. 8, no 2, pp. 137-150.

[26]   P.S. Pacheco, *Parallel Programming with MPI*, Morgan Kaufman Publishers, San Francisco, California, 1997.

[27]   V.S. Sunderam, G.A. Geist, Heterogeneous Parallel and Distributed Computing, *Parallel Computing*, Elsevier, 1999, vol. 25, pp. 1699-1721.

[28]   B. Engquist, A. Majda, "Absorbing Boundary Conditions for Numerical Simulation of Waves," Math. of Computation, vol. 31, No. 139, pp. 629-651, 1977.