3D Harmonic Maxwell Solutions on Vector and Parallel Computers using Controllability and Finite Element Methods

Marie-Odile Bristeau, Roland Glowinski, Jacques Périaux, Tuomo Rossi

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

3D Harmonic Maxwell Solutions on Vector and Parallel Computers using Controllability and Finite Element Methods.

Marie-Odile Bristeau , Roland Glowinski , Jacques Périaux ,

Tuomo Rossi

No 3607

Janvier 1999

THÈME 4

ISSN 0249-6399

3D Harmonic Maxwell Solutions on Vector and Parallel Computers using Controllability and Finite Element Methods.

Marie-Odile Bristeau *, Roland Glowinski †, Jacques Périaux ‡,
Tuomo Rossi §

Thème 4 — Simulation et optimisation
de systèmes complexes
Projet M3N

Rapport de recherche n ° 3607 — Janvier 1999 — 31 pages

Abstract: We consider the scattering problem for 3-D electromagnetic harmonic waves. The time-domain Maxwell's equations are solved and Exact Controllability methods improve the convergence of the solutions to the time-periodic ones for non-convex obstacles. A least-squares formulation solved by a preconditioned conjugate gradient is introduced. The discretization is achieved in time by a centered finite difference scheme and in space by Lagrange finite elements. Numerical results for 3-D nonconvex scatterers illustrate the efficiency of the method on vector and parallel computers.

Key-words: Maxwell equations, Scattering, Controllability, Least-Squares /Shooting method, Conjugate Gradient, Finite Elements, Parallelization.

(Résumé : tsvp)

* M3N, Email : Marie-Odile.Bristeau@inria.fr
† Dept. of Mathematics, University of Houston, Houston, Texas 77204-3476, U.S.A.
‡ Dassault Aviation, 78, quai M. Dassault, 92214 Saint-Cloud, France
§ Dept. of Mathematics, University of Jyvaskyla, Jyvaskyla,Finland, M3N, Inria and Dassault Aviation

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Méthodes de Contrôlabilité et d'Eléments finis pour les équations de Maxwell en régime harmonique.

Résumé : On s'intéresse au problème tridimensionnel de la diffraction d'ondes électromagnétiques en régime harmonique. On considère une formulation temporelle des équations de Maxwell et, dans le cas d'obstacles non convexes, une méthode de Contrôlabilité Exacte permet d'améliorer la convergence vers la solution périodique. On introduit une formulation moindres-carrés du problème qui peut ensuite être résolu par un algorithme de gradient conjugué. Le problème est discrétisé en temps par un schéma aux différences finies centré et en espace par éléments finis de Lagrange. Des résultats numériques pour des obstacles tridimensionnels non convexes montrent l'efficacité de la méthode sur machines vectorielle ou parallèle.

Mots-clé : Equations de Maxwell, Diffraction, Contrôlabilité, Moindres-Carrés/ Méthode de tir, Gradient Conjugué, Eléments finis, Parallélisation.

1 Introduction

Exact Controllability Methods have been applied in Bristeau, Glowinski, Périaux [6, 7, 8] to the solution of scalar harmonic waves scattering on complex geometry obstacles. In this paper, we apply this method to the solution of 3-D Maxwell equations for time harmonic fields. We consider the second-order time derivative formulation of Maxwell equations; for nonconvex reflectors, if we integrate in time these equations with periodic data, the convergence to the periodic solution may be very slow and even the correct solution may be not reached. The method described in the present article improves the speed of convergence to the periodic solution.

Once the problem is set as an Exact Controllability one, its solution via a least-squares formulation and a preconditioned conjugate gradient algorithm is discussed. Some details on the discretizations and the numerical implementation are given and numerical examples are presented.

2 Maxwell Equations

We consider the scattering problem of an electromagnetic wave by a perfectly conducting obstacle B surrounded by vacuum. Let $\Omega \subset \mathbb{R}^3$ be a bounded domain with $\gamma = \partial B$ the boundary of the obstacle and Γ the artificial boundary (see Figure 2.1). Let $T > 0$ be given.

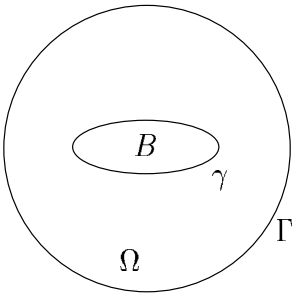

Figure 2.1:

The electric field E of the scattered wave satisfies the equations (Maxwell equations in which the magnetic field H is eliminated) :

$$\frac{\partial^2 E}{\partial t^2} + \nabla \times (\nabla \times E) = 0 \quad in \quad Q = \Omega \times (0, T) \tag{2.1}$$

$$\nabla . E = 0 \quad in \quad Q \tag{2.2}$$

$$E \times n = -E^{inc} \times n \quad on \quad \sigma = \gamma \times (0,T) \tag{2.3}$$

$$(\nabla \times E) \times n = (\frac{\partial E}{\partial t} \times n) \times n \quad on \quad \Sigma = \Gamma \times (0,T) \tag{2.4}$$

with E^{inc} the incident wave and n the unit outward normal; on the artificial boundary we prescribe the first-order (Silver-Muller) radiation condition (2.4). We use the non-dimensionalized equations with $t = c\tau$, c light speed.

We assume that some initial conditions are given (we use the notation $E(t) \equiv E(.,t)$) :

$$E(0) = E_0, \tag{2.5}$$

$$\frac{\partial E}{\partial t}(0) = E_1, \tag{2.6}$$

satisfying the conditions:

$$\nabla \cdot E_0 = 0, \tag{2.7}$$

$$\nabla \cdot E_1 = 0. \tag{2.8}$$

3 Variational Formulation

We introduce the following spaces :

$$H(curl;\Omega) = \{z \in (L^2(\Omega))^3 | \nabla \times z \in (L^2(\Omega))^3\}$$

$$H(div;\Omega) = \{z \in (L^2(\Omega))^3 | \nabla \cdot z \in L^2(\Omega)\}$$

$$V_g = \{z \in H(curl;\Omega) \cap H(div;\Omega) | z \times n = g \times n \quad on \quad \gamma\}$$

Multiplying (2.1) by a function $z \in H(curl;\Omega)$ and integrating by parts over Ω, we obtain :

$$\int_\Omega \frac{\partial^2 E}{\partial t^2} \cdot z dx + \int_\Omega (\nabla \times E) \cdot (\nabla \times z) dx - \int_{\Gamma \cup \gamma} (\nabla \times E) \times n . z d\Gamma = 0. \tag{3.1}$$

The bilinear form $\int_\Omega (\nabla \times \varphi) \cdot (\nabla \times z) dx$ is not coercive on $H(curl;\Omega)$, an usual way to remedy this problem is to add the term $\int_\Omega (\nabla . \varphi)(\nabla . z) dx$. (see [19],[1],[13]).

So we introduce the following variational formulation to be satisfied by $E(t)$:

$$\begin{cases} \int_\Omega \frac{\partial^2 E}{\partial t^2} \cdot z dx + \int_\Omega [(\nabla \times E) \cdot (\nabla \times z) + (\nabla \cdot E)(\nabla \cdot z)] dx \\ - \int_\Gamma (\frac{\partial E}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\ E(t) \in V_g, \end{cases} \tag{3.2}$$

with $g = -E^{inc}$.

We use the relation $\int_\gamma (\nabla \times E) \times n \cdot z d\gamma = - \int_\gamma (\nabla \times E) \cdot (z \times n) d\gamma$, so this integral is zero for $z \in V_0$.

We do not examine here the problem of the equivalence between (2.1)-(2.4) and (3.2).

4 Time Harmonic Problem. Exact Controllability

We consider now that the incident wave is a time harmonic one, then, the scattered field is also periodic and satisfies particularly:

$$E(0) = E(T), \tag{4.1}$$

$$\frac{\partial E}{\partial t}(0) = \frac{\partial E}{\partial t}(T), \tag{4.2}$$

where T denotes now the period ($T = c/f$, f being the frequency of the incident wave).

Then, the problem (3.2) associated with the conditions (4.1)-(4.2) can be formulated as an Exact Controllability problem (see [18],[11],[12]):

Find $e = \{e_0, e_1\}$ *such that* :

$$\begin{cases} \int_\Omega \frac{\partial^2 y}{\partial t^2} \cdot z dx + \int_\Omega [(\nabla \times y) \cdot (\nabla \times z) + (\nabla \cdot y)(\nabla \cdot z)] dx \\ - \int_\Gamma (\frac{\partial y}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\ y \in V_g, \end{cases} \tag{4.3}$$

$$y(0) = e_0, \quad \frac{\partial y}{\partial t}(0) = e_1, \tag{4.4}$$

$$y(T) = e_0, \quad \frac{\partial y}{\partial t}(T) = e_1. \tag{4.5}$$

We shall not address here the existence and uniqueness of solutions to problem (4.3)-(4.5); instead we shall focus on the calculations of such solutions, assuming they do exist.

In order to solve practically this problem, we introduce a least-squares approach, leading to the following minimization problem :

$$\min_{v \in W_g} J(v) \tag{4.6}$$

with

$$\begin{cases} J(v) = \frac{1}{2}[\int_\Omega |y(T) - v_0|^2 dx + \int_\Omega (|\nabla \times (y(T) - v_0)|^2 \\ +|\nabla \cdot (y(T) - v_0)|^2) dx + \int_\Omega |\frac{\partial y}{\partial t}(T) - v_1|^2 dx] \end{cases} \tag{4.7}$$

where y is a function of v through the equations (4.3) and

$$y(0) = v_0, \quad \frac{\partial y}{\partial t}(0) = v_1. \tag{4.8}$$

We denote $W_g = V_{g(0)} \times (L^2(\Omega))^3$. The space W_0 is equipped with the scalar product $(.,.)$ defined by

$$\begin{cases} (u,v)_{W_0} = \int_\Omega u_0 \cdot v_0 dx + \int_\Omega [(\nabla \times u_0) \cdot (\nabla \times v_0) + (\nabla \cdot u_0)(\nabla \cdot v_0)] dx \\ + \int_\Omega u_1 \cdot v_1 dx, \quad \forall u, v \in W_0, \end{cases} \tag{4.9}$$

we denote $|.|_{W_0}$ the associate norm.

The choice of the space W_0 and of the associated scalar product is very important, it determines the expression of J and then the scalar products and norms used in the conjugate gradient algorithm introduced to solve (4.6).

Remark 4.1 *The well-posedness of problem (4.6) is not discussed here; we refer to Bardos and Rauch [2] for a study of the functional analogous to (4.7) but associated to the 2D scalar problem, the same authors have proposed also an alternative functional with better coercivity properties.*

5 Gradient calculation

Assuming that e is the solution of the least-squares problem (4.6), it will satisfy:

$$< J'(e), z >= 0, \quad \forall z \in W_0, \tag{5.1}$$

where $< .,. >$ denotes the duality pairing between W_0' and W_0 (W_0' dual space of W_0).

To compute the derivative J', as in [6],[8], we use the classical perturbation analysis. Let δv be a perturbation of $v \in W_0$, from (4.7), we have :

$$
(5.2)\quad \left\{
\begin{array}{l}
< J'(v), \delta v >= \displaystyle\int_\Omega (y(T) - v_0) \cdot \delta y(T) dx \\
+ \displaystyle\int_\Omega [(\nabla \times (y(T) - v_0)) \cdot (\nabla \times \delta y(T)) \\
+ (\nabla \cdot (y(T) - v_0))(\nabla \cdot \delta y(T))] dx \\
+ \displaystyle\int_\Omega (\frac{\partial y}{\partial t}(T) - v_1) \cdot \frac{\partial \delta y}{\partial t}(T) dx + \int_\Omega (v_0 - y(T)) \cdot \delta v_0 dx \\
+ \displaystyle\int_\Omega [(\nabla \times (v_0 - y(T))) \cdot (\nabla \times \delta v_0) + (\nabla \cdot (v_0 - y(T)))(\nabla \cdot \delta v_0)] dx \\
+ \displaystyle\int_\Omega (v_1 - \frac{\partial y}{\partial t}(T)) \cdot \delta v_1 dx.
\end{array}
\right.
$$

In (5.2), the perturbation δy of y is solution of the following system

$$
(5.3)\quad \left\{
\begin{array}{l}
\displaystyle\int_\Omega \frac{\partial^2 \delta y}{\partial t^2} \cdot z dx + \int_\Omega [(\nabla \times \delta y) \cdot (\nabla \times z) + (\nabla \cdot \delta y)(\nabla \cdot z)] dx \\
\displaystyle - \int_\Gamma (\frac{\partial \delta y}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\
\delta y \in V_0,
\end{array}
\right.
$$

$$
(5.4)\qquad \delta y(0) = \delta v_0, \quad \frac{\partial \delta y}{\partial t}(0) = \delta v_1.
$$

In (5.3), we choose $z = p$, we integrate on $(0, T)$ and integrate by parts, we obtain :

$$
(5.5)\quad \left\{
\begin{array}{l}
\displaystyle\int_Q \frac{\partial^2 p}{\partial t^2} \cdot \delta y dx dt - \int_\Omega \frac{\partial p}{\partial t} . \delta y dx |_0^T + \int_\Omega p . \frac{\partial \delta y}{\partial t} dx |_0^T \\
\displaystyle + \int_Q [(\nabla \times p) \cdot (\nabla \times \delta y) + (\nabla \cdot p)(\nabla \cdot \delta y)] dx dt \\
\displaystyle + \int_\Sigma (\frac{\partial p}{\partial t} \times n) \times n \cdot \delta y d\Gamma dt + \int_\Gamma (p \times n) \times n \cdot \delta y d\Gamma |_0^T = 0.
\end{array}
\right.
$$

Suppose that function p satisfies

$$
(5.6)\quad \left\{
\begin{array}{l}
\displaystyle\int_\Omega \frac{\partial^2 p}{\partial t^2} \cdot z dx + \int_\Omega [(\nabla \times p) \cdot (\nabla \times z) + (\nabla \cdot p)(\nabla \cdot z)] dx \\
\displaystyle + \int_\Gamma (\frac{\partial p}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0,
\end{array}
\right.
$$

$$p \times n = 0 \quad on \quad \sigma, \tag{5.7}$$

then, using (5.4), equation (5.5) reduces to

$$\begin{cases} \int_\Omega p(T) \cdot \dfrac{\partial \delta y}{\partial t}(T) dx - \int_\Omega \dfrac{\partial p}{\partial t}(T) \cdot \delta y(T) dx \\ + \int_\Gamma (p(T) \times n) \times n \cdot \delta y(T) d\Gamma = \int_\Omega p(0) \cdot \delta v_1 dx \\ - \int_\Omega \dfrac{\partial p}{\partial t}(0) \cdot \delta v_0 dx + \int_\Gamma (p(0) \times n) \times n \cdot \delta v_0 d\Gamma. \end{cases} \tag{5.8}$$

Let us define $p(T)$ and $\dfrac{\partial p}{\partial t}(T)$ by

$$p(T) = \frac{\partial y}{\partial t}(T) - v_1, \tag{5.9}$$

$$\begin{cases} \int_\Omega \dfrac{\partial p}{\partial t}(T) \cdot z dx = \int_\Gamma ((\dfrac{\partial y}{\partial t}(T) - v_1) \times n) \times n \cdot z d\Gamma \\ - \int_\Omega (y(T) - v_0) \cdot z dx - \int_\Omega [(\nabla \times (y(T) - v_0)) \cdot (\nabla \times z) \\ + (\nabla \cdot (y(T) - v_0))(\nabla \cdot z)] dx, \quad \forall z \in V_0. \end{cases} \tag{5.10}$$

We then have from (5.2) and (5.6)-(5.10) (in this last equation, we choose $z = \delta y(T)$),

$$\begin{cases} < J'(v), w >= \int_\Omega (v_0 - y(T))) \cdot w_0 dx \\ + \int_\Omega [(\nabla \times (v_0 - y(T))) \cdot (\nabla \times w_0) + (\nabla \cdot (v_0 - y(T)))(\nabla \cdot w_0)] dx \\ + \int_\Omega (v_1 - \dfrac{\partial y}{\partial t}(T)) \cdot w_1 dx + \int_\Omega p(0) \cdot w_1 dx - \int_\Omega \dfrac{\partial p}{\partial t}(0) \cdot w_0 \\ + \int_\Gamma (p(0) \times n) \times n \cdot w_0 d\Gamma, \quad \forall w = \{w_0, w_1\} \in W_0. \end{cases} \tag{5.11}$$

Remark 5.1 *Relations (5.10), (5.11) are largely formal ; however, it is worth mentioning that the discrete variants of the above two relations make sense and lead to algorithms with fast convergence properties (in order to make (5.10) rigorous we should replace the integral in the left-hand side of the above equation by a duality pairing; similarly, we should replace the boundary integral in the right-hand side of (5.10) by a well-chosen duality pairing).*

6 Conjugate gradient algorithm

We propose to solve the problem (5.1) by a conjugate gradient algorithm written as follows:

Step 0 : Initialization

(6.1) $e^0 = \{e_0^0, e_1^0\} \in W_0$ *is given* (see Sec. 8 for the practical choice of e^0).

Solve the generalized forward wave equation

$$\text{(6.2)} \qquad \begin{cases} \displaystyle\int_\Omega \frac{\partial^2 y^0}{\partial t^2} + \int_\Omega [(\nabla \times y^0) \cdot (\nabla \times z) + (\nabla \cdot y^0)(\nabla \cdot z)]dx \\ \displaystyle - \int_\Gamma (\frac{\partial y^0}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\ y^0 \in V_g, \end{cases}$$

$$\text{(6.3)} \qquad y^0(0) = e_0^0, \quad \frac{\partial y^0}{\partial t}(0) = e_1^0.$$

Solve the generalized backward wave equation

$$\text{(6.4)} \qquad \begin{cases} \displaystyle\int_\Omega \frac{\partial^2 p^0}{\partial t^2} \cdot z dx + \int_\Omega [(\nabla \times p^0) \cdot (\nabla \times z) + (\nabla \cdot p^0)(\nabla \cdot z)]dx \\ \displaystyle + \int_\Gamma (\frac{\partial p^0}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\ p^0 \in V_0, \end{cases}$$

with the final conditions

$$\text{(6.5)} \qquad p^0(T) = \frac{\partial y^0}{\partial t}(T) - e_1^0,$$

$$\text{(6.6)} \qquad \begin{cases} \displaystyle\int_\Omega \frac{\partial p^0}{\partial t}(T) \cdot z dx = \int_\Gamma ((\frac{\partial y^0}{\partial t}(T) - e_1^0) \times n) \times n \cdot z d\Gamma \\ \displaystyle - \int_\Omega (y^0(T) - e_0^0) \cdot z dx - \int_\Omega [(\nabla \times (y^0(T) - e_0^0)) \cdot (\nabla \times z) \\ +(\nabla \cdot (y^0(T) - e_0^0))(\nabla \cdot z)]dx, \quad \forall z \in V_0. \end{cases}$$

Define next $g^0 = \{g_0^0, g_1^0\} \in W_0$ *by*

$$(6.7)\quad \begin{cases} \displaystyle\int_\Omega g_0^0 \cdot z dx + \int_\Omega [(\nabla \times g_0^0) \cdot (\nabla \times z) + (\nabla \cdot g_0^0)(\nabla \cdot z)] dx = \\ \displaystyle\int_\Omega (e_0^0 - y^0(T)) \cdot z dx \\ \displaystyle + \int_\Omega [(\nabla \times (e_0^0 - y^0(T))) \cdot (\nabla \times z) + (\nabla \cdot (e_0^0 - y^0(T))(\nabla \cdot z)] dx \\ \displaystyle - \int_\Omega \frac{\partial p^0}{\partial t}(0) \cdot z dx + \int_\Gamma (p^0(0) \times n) \times n \cdot z d\Gamma, \quad \forall z \in V_0, \end{cases}$$

$$(6.8)\qquad g_1^0 = p^0(0) + e_1^0 - \frac{\partial y^0}{\partial t}(T),$$

and then

$$(6.9)\qquad w^0 = g^0.$$

For $k \geq 0$, *suppose that* e^k, g^k, w^k *are known; we then compute their updates* e^{k+1}, g^{k+1}, w^{k+1}, *as follows:*

Step 1 : Descent

Solve

$$(6.10)\quad \begin{cases} \displaystyle\int_\Omega \frac{\partial^2 \bar{y}^k}{\partial t^2} \cdot z dx + \int_\Omega [(\nabla \times \bar{y}^k) \cdot (\nabla \times z) + (\nabla \cdot \bar{y}^k)(\nabla \cdot z)] dx \\ \displaystyle - \int_\Gamma (\frac{\partial \bar{y}^k}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\ \bar{y}_k \in V_0, \end{cases}$$

$$(6.11)\qquad \bar{y}^k(0) = w_0^k, \quad \frac{\partial \bar{y}^k}{\partial t}(0) = w_1^k.$$

Solve

$$(6.12)\quad \begin{cases} \displaystyle\int_\Omega \frac{\partial^2 \bar{p}^k}{\partial t^2} \cdot z dx + \int_\Omega [(\nabla \times \bar{p}^k) \cdot (\nabla \times z) + (\nabla \cdot \bar{p}^k)(\nabla \cdot z)] dx \\ \displaystyle + \int_\Gamma (\frac{\partial \bar{p}^k}{\partial t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\ \bar{p}^k \in V_0, \end{cases}$$

$$(6.13)\qquad \bar{p}^k(T) = \frac{\partial \bar{y}^k}{\partial t}(T) - w_1^k,$$

$$(6.14)\qquad \begin{cases} \displaystyle\int_\Omega \frac{\partial \bar{p}^k}{\partial t}(T)\cdot z dx = \int_\Gamma ((\frac{\partial \bar{y}^k}{\partial t}(T) - w_1^k)\times n)\times n\cdot z d\Gamma \\ \displaystyle -\int_\Omega (\bar{y}^k(T) - w_0^k)\cdot z dx - \int [(\nabla\times(\bar{y}^k(T)-w_0^k))\cdot(\nabla\times z) \\ \displaystyle +(\nabla\cdot(\bar{y}^k(T)-w_0^k))(\nabla\cdot z)]dx, \quad \forall z\in V_0. \end{cases}$$

Define $\bar{g}^k = \{\bar{g}_0^k, \bar{g}_1^k\} \in W_0$ *by*

$$(6.15)\begin{cases} \displaystyle\int_\Omega \bar{g}_0^k\cdot z dx + \int_\Omega [(\nabla\times\bar{g}_0^k)\cdot(\nabla\times z) + (\nabla\cdot\bar{g}_0^k)(\nabla\cdot z)]dx = \\ \displaystyle\int_\Omega (w_0^k - \bar{y}^k(T))\cdot z dx \\ \displaystyle +\int_\Omega [(\nabla\times(w_0^k-\bar{y}^k(T)))\cdot(\nabla\times z) + (\nabla\cdot(w_0^k-\bar{y}^k(T)))(\nabla\cdot z)]dx \\ \displaystyle -\int_\Omega \frac{\partial\bar{p}^k}{\partial t}(0)\cdot z dx + \int_\Gamma (\bar{p}^k(0)\times n)\times n\cdot z d\Gamma, \quad \forall z\in V_0, \end{cases}$$

$$(6.16)\qquad \bar{g}_1^k = \bar{p}^k(0) + w_1^k - \frac{\partial\bar{y}^k}{\partial t}(T).$$

Compute ρ_k *by*

$$(6.17)\qquad \rho_k = \frac{|g^k|^2_{W_0}}{(\bar{g}^k, w^k)_{W_0}},$$

$(.,.)_{W_0}$ *and* $|.|_{W_0}$ *being defined by (4.9).*

We then update e^k *and* g^k *by*

$$(6.18)\qquad e^{k+1} = e^k - \rho_k w^k,$$

$$(6.19)\qquad g^{k+1} = g^k - \rho_k \bar{g}^k.$$

Step 2: Test for convergence and construction of the new descent direction. *If*

$$(6.20)\qquad \frac{|g^k|_{W_0}}{|g^0|_{W_0}} \le \varepsilon$$

take $e = e^{k+1}$ *; else, compute*

$$(6.21)\qquad \gamma_k = \frac{|g^{k+1}|^2_{W_0}}{|g^k|^2_{W_0}}$$

and update w^k *by*

$$w^{k+1} = g^{k+1} + \gamma_k w^k. \tag{6.22}$$

Set $k = k + 1$ *and go to (6.10).*

Remark 6.1 *Each iteration of the above conjugate gradient algorithm requires basically the solution of two generalized wave equations such as (6.10),(6.11) and (6.12)-(6.14) and of an elliptic problem such as (6.15) (see Sec. 8 for details concerning the numerical implementation).*

7 Time and Space Discretization

For the practical implementation of the above algorithm, we give the space and time discretization of the equations (4.3), (4.8) which can be considered as model for the generalized wave equations to be solved in algorithm (6.1)-(6.22).

Concerning the time discretization, we use a second-order finite difference centered scheme. Let $\Delta t = T/N$ (N : a positive integer) be the time step, we obtain after time discretization of (4.3), (4.8)

$$\left\{\begin{array}{l} \displaystyle\int_\Omega \frac{y^{n+1} - 2y^n + y^{n-1}}{\Delta t^2} \cdot z dx \\ \displaystyle + \int_\Omega [(\nabla \times y^n) \cdot (\nabla \times z) + (\nabla \cdot y^n)(\nabla \cdot z)] dx \\ \displaystyle - \int_\Gamma (\frac{y^{n+1} - y^{n-1}}{2\Delta t} \times n) \times n \cdot z d\Gamma = 0, \quad \forall z \in V_0, \\ y^{n+1} \in V_{g^{n+1}}, \quad n = 0, \ldots, N, \\ y^0 = v_0, \\ \displaystyle\frac{y^1 - y^{-1}}{2\Delta t} = v_1 \end{array}\right. \tag{7.1}$$

with $g^{n+1} = g(t^{n+1})$.

The backward adjoint equation is discretized by the same scheme. It is important to have an explicit scheme to avoid the solution of a linear system at each time-step. As we simulate a time harmonic phenomenon, we have to use in any case enough time steps per period and the time step prescribed by the CFL condition related to the explicit scheme is not very smaller than the one prescribed by the accuracy of the solution; so it would not be interesting to introduce an implicit scheme.

For the space discretization, we introduce a tetrahedrisation $\mathcal{T}_h$ of Ω, and we consider the following discrete set:

$$V_g^h = \{z_h \in C^0(\bar{\Omega})^3, z_h|_T \in P_1^3, \forall T \in \mathcal{T}_h, z_h \times n_h = g_h \times n_h \ on\ \gamma\} \tag{7.2}$$

with P_k the space of polynomials in three variables of degrees $\leq k$; n_h will be defined in the following.

For simplicity, we have chosen an approximation by continuous linear finite elements; we will not discuss here the advantages and the drawbacks of this approximation compared to edge elements, we refer for instance to P. Monk ([19]). To apply this approximation, we assume that the boundary γ is smooth enough not to generate singularities.

So we approximate (4.3),(4.8) by :

$$\left\{\begin{array}{l} \frac{1}{\Delta t^2}\int_\Omega y_h^{n+1}\cdot z_h dx - \frac{1}{2\Delta t}\int_\Gamma (y_h^{n+1}\times n_h)\times n_h \cdot z_h d\Gamma = \\ \frac{1}{\Delta t^2}\int_\Omega (2y_h^n - y_h^{n-1})\cdot z_h dx \\ -\int_\Omega [(\nabla\times y_h^n)\cdot(\nabla\times z_h) + (\nabla\cdot y_h^n)(\nabla\cdot z_h)]dx \\ -\frac{1}{2\Delta t}\int_\Gamma (y_h^{n-1}\times n_h)\times n_h\cdot z_h d\Gamma, \quad \forall z_h \in V_0^h, \\ y_h^{n+1} \in V_{g^{n+1}}^h, \quad n = 0,\ldots,N, \\ y_h^0 = v_{0h}, \\ \frac{y_h^1 - y_h^{-1}}{2\Delta t} = v_{1h}, \end{array}\right. \tag{7.3}$$

where v_{0h} and v_{1h} are approximations of v_0 and v_1 belonging to V_0^h.

The first terms of the two sides of the equation are calculated with mass lumping to obtain a diagonal matrix. Using the same idea for the boundary terms, leads to a 3×3 block diagonal matrix (at each node the 3 components of y_h are coupled) each block is inverted analytically. The derivatives of y_h and z_h being constant by tetrahedron, the integral $\int_\Omega[(\nabla\times y_h^n)\cdot(\nabla\times z_h) + (\nabla\cdot y_h^n)(\nabla\cdot z_h)]dx$ is computed exactly (see the following section for the implementation).

Concerning the approximate normale n_h, it is defined at each node M_i of the boundary by the following usual formula; we compute

$$n_i = \sum_{T\in V_i} V(T)\nabla\varphi_i|_T,$$

or which is equivalent

$$n_i = \frac{1}{3} \sum_{S \in B_i} A(S) n_s,$$

with

φ_i a scalar basis function,
V_i the set of the tetrahedra T such as $M_i \in T$,
B_i the set of the boundary faces S such as $M_i \in S$,
$V(T)$ the volume of the tetrahedron T,
$A(S)$ the area of the face S,
n_s the unit normal of the face S,

then n_h is obtained at each node by normalization of n_i.

As the scheme is explicit, the tangential condition $y_h \times n_h = g_h \times n_h$ is satisfied by projection ; we can use the relation

$$y_h = (y_h \cdot n_h) n_h - (y_h \times n_h) \times n_h$$

8 Numerical Implementation

In the algorithm (6.1)-(6.22), the more time consuming steps are the integration of the generalized wave equations (6.10),(6.11) and (6.12)-(6.14) which after discretization are of the form (7.3) , and the solution of problem (6.15). In (7.3), the more expensive part is the computation of the matrix-vector product corresponding to the evaluation of the term $\int_\Omega [(\nabla \times y_h) \cdot (\nabla \times z_h) + (\nabla \cdot y_h)(\nabla \cdot z_h)] dx$; the matrix is the same as the one of the discretized problem corresponding to (6.15). As an iterative process is used for the solution of this last problem, the main cost is also related to the matrix-vector product. So, for the efficiency of the global algorithm, the matrix-vector product has to be as optimized as possible.

First the following identity is used :

$$\left\{ \begin{array}{l} \displaystyle\int_\Omega [(\nabla \times y) \cdot (\nabla \times z) + (\nabla \cdot y)(\nabla \cdot z)] dx \equiv \\ \displaystyle\int_\Omega \nabla y : \nabla z dx + \sum_{i,j=1}^{3} \int_{\Gamma \cup \gamma} (\frac{\partial y_i}{\partial x_i} n_j - \frac{\partial y_i}{\partial x_j}\ n_i) z_j d\Gamma \end{array} \right. \tag{8.1}$$

or with an other notation

$$(8.2)\qquad \begin{cases} \displaystyle\int_\Omega [(\nabla \times y)\cdot(\nabla \times z) + (\nabla \cdot y)(\nabla \cdot z)]dx \equiv \\ \displaystyle\int_\Omega \nabla y : \nabla z dx + \sum_{i=1}^{3} \int_{\Gamma\cup\gamma} (\nabla y_i \times n)\cdot(e_i \times z)d\Gamma \end{cases}$$

with $\{e_i\}_{i=1,\ldots,3}$ the canonical basis of $\mathbb{R}^3$.

This identity is verified as soon as each term is defined, which is true in the discrete problem (7.3).

So, using the right-hand side formulation the same laplacian matrix is applied to the three components of y which are coupled only by the boundary integral.

In view of efficiency on the vector computer C90, the nonzero terms of the laplacian sparse matrix and of the boundary matrix are stored diagonal by diagonal (see [9],[10]). For the parallel computations, the row storage is used because in this case it is better to handle short vectors.

Concerning the e_0^0 and e_0^1 initial control values in algorithm (6.1)-(6.22), we need smooth initial solutions satisfying the boundary conditions on the obstacle, this can be obtained by prescribing the time harmonic sources progressively during a transient time interval $[0, t_{tr}]$ as suggested by G. Mur [20]. On this time interval, we just integrate in time the Maxwell equation, the right-hand side $g(x,t)$ in (4.3) is multiplied by a smooth transient function $\theta_{tr}(t)$ increasing from 0 to 1. Different functions can be considered, we have used the one proposed by G. Mur [20]:

$$(8.3)\qquad \begin{cases} \theta_{tr}(t) = (2 - \sin((t/t_{tr})\dfrac{\pi}{2}))\sin((t/t_{tr})\dfrac{\pi}{2}), \quad \textit{if} \quad 0 \le t \le t_{tr}, \\ \theta_{tr}(t) = 1, \quad \textit{if} \quad t \ge t_{tr}. \end{cases}$$

The discrete elliptic problems associated with the preconditioning steps (6.15), have been solved by a diagonal preconditioned conjugate gradient. In the case of parallel computations, the linear problem (6.15) could also be solved by an algorithm based on the domain decomposition (see[17]).

The meshes have been generated by a new 3-D mesh generator developed at INRIA by E. Seveno ([22],[23]) and based on an advancing front method. This method gives good quality meshes implying that we can choose a reasonable time step satisfying nevertheless the CFL condition.

9 Parallelization

Basing on the introduced algorithm, we have constructed a parallel solver for the considered Maxwell equations. In the parallel implementation we use message passing conforming to the MPI standard [24]. This makes our code portable to various parallel computers. The numerical experiments were carried out in a Cray T3E computer using up to 64 processors. Similar parallelization techniques were considered for two–dimensional standard wave equation in [14].

The parallelization is based on nonoverlapping domain decomposition and it reserves the explicit nature of the algorithm. The computational domain Ω_h is decomposed into a given number of subdomains using the mesh partitioning software `decomp` of the Modulef finite element library [3]. The software `decomp` is based on K means techniques which were earlier developed in automatic data classification. This method, also called dynamic clusters, is initialized by a first partition which then is iteratively improved in order to get connected regular mesh clusters and well balanced number of vertices. The obtained decomposition is then used as an input for the K means algorithm. This approach has been compared with other mesh partitioning methods in [16] .

In our approach, each processor handles one subdomain, and the required vectors and matrices are distributed to the different processors conforming the decomposition of the domain. The nodes at the subdomain interface boundaries are shared. That is, all the processors whose subdomains meet at a given interface node have a private copy of the vector components related to this node. This strategy increases the computational overhead a bit, but it simplifies the implementation. Using this strategy, we are able to solve larger problems, or to solve a given problem faster by increasing the number of subdomains (and processors).

Our parallel implementation is algebraically equivalent to the sequential algorithm, unlike the usual domain decomposition methods, where, typically, a parallel preconditioner is constructed for the iterative solution of the linear system arising from discretization. Due to the explicit time-integration scheme and mass lumping for the generalized wave equations (6.10),(6.11) and (6.12)-(6.14), and the iterative solution of the linear problems (6.7), (6.15), almost all the communication occurs in the multiplication of a distributed vector by a distributed matrix A corresponding to the integral (8.2). To describe this in more detail, we consider the case when we have only two subdomains. The global matrix, denoted by A, and the vectors x and

y then have compatible block representations

$$A = \begin{pmatrix} A_{11} & A_{1\gamma} & \\ A_{\gamma 1} & A_{\gamma\gamma} & A_{\gamma 2} \\ & A_{2\gamma} & A_{22} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_\gamma \\ x_2 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_\gamma \\ y_2 \end{pmatrix}, \tag{9.1}$$

where the subscript $i = 1, 2$ denotes the components of the matrix and the vectors related to the nodes of subdomain i which are not at the interface boundary, and the subscript γ denotes the shared interface components. The local subdomain matrices A^i and distributed vectors x^i and y^i then are as follows

$$\begin{cases} A^1 = \begin{pmatrix} A_{11} & A_{1\gamma} \\ A_{\gamma 1} & A^1_{\gamma\gamma} \end{pmatrix}, \quad x^1 = \begin{pmatrix} x_1 \\ x_\gamma \end{pmatrix}, \quad y^1 = \begin{pmatrix} y_1 \\ y_\gamma \end{pmatrix}, \\ A^2 = \begin{pmatrix} A^2_{\gamma\gamma} & A_{\gamma 2} \\ A_{2\gamma} & A_{22} \end{pmatrix}, \quad x^2 = \begin{pmatrix} x_\gamma \\ x_2 \end{pmatrix}, \quad y^2 = \begin{pmatrix} y_\gamma \\ y_2 \end{pmatrix}, \end{cases} \tag{9.2}$$

where $A_{\gamma\gamma} = A^1_{\gamma\gamma} + A^2_{\gamma\gamma}$. Let us now consider the parallel computation of $y = Ax$, assuming that the vector x is already constructed and both processors have their own copy of the block x_γ. This can be done by the following obvious algorithm:

1. The processor i posts the nonblocking receive request for getting the block y^j_γ from the neighbor processor j.
2. Both processors compute in parallel:
$$\tilde{y}^i = A^i x^i, \ i = 1, 2, \quad \tilde{y}^1 = (y_1, y^1_\gamma), \quad \tilde{y}^2 = (y^2_\gamma, y_2). \tag{9.3}$$
3. The processor i uses the nonblocking send to submit the block y^i_γ to the neighbor processor j.
4. After processor i has received the block y^j_γ from the neighbor processor j, it can compute its final block y_γ by $y_\gamma = y^1_\gamma + y^2_\gamma$.

In the above algorithm we used nonblocking point–to–point communication. It might not be very beneficial in the case of only two subdomains, due to the extra buffer space which is needed. However, it gives more flexibility and is easier to implement in the case when we have several subdomains, and each subdomain can have quite an arbitrary number of neighbor subdomains. This is the case when using a "black box" mesh partitioning tool, instead of constructing the partitioning by hand.

In case of several subdomains the above algorithm is then modified appropriately. In step 1, a receive is posted for each neighbor of subdomain i, and in step 3, the

processor i sends a required part of the interface block to each of its neighbors. Obviously, in step 2, all the processors can compute their local vectors in parallel. In step 4, each processor waits until all the receives posted in step 1 have arrived in an auxiliary buffer. After that it computes the update of its interface components. This has to be done in fixed order in all the processors for a given shared interface component in order to avoid the incoherence of the values of the shared components due to roundoff errors.

It would be tempting to perform the update as soon as a message arrives from a neighbor, because it would probably improve the performance. Since the matrix–vector multiply is repeated quite a many times in the course of the algorithm, the roundoff errors might accumulate and disturb the accuracy of the solution. This is a topic which should be further studied numerically. In our implementation, we have used the described safe strategy.

Some communication is also required in the parallel evaluation of the global inner products in (6.17), (6.20) and (6.21). This can be done efficiently by using the linearity. Hence, processors compute in parallel a given inner product with their local vectors. After that, the global value is obtained by summing up the local inner products, provided that the contributions of the shared interface components are properly taken care of. We have used an approach, where each processor neglects the interface components which are shared with subdomains whose number is less than the number of the subdomain related to the processor itself. Hence, each interface component is evaluated only once in the inner products. In terms of communication, the cost of computing the inner products is quite small, even though collective communication (in practice, the all reduce operation in MPI) is used since only scalar values need to be summed.

10 Numerical Results

The developed code based on this algorithm has been first tested and validated on spheres where comparisons with analytical solutions can be done (see [21]). We show here the results for a sphere with radius λ, the artificial boundary is located at a distance λ from ∂B. With the angles defined as in Figure 10.1, the incident wave is defined by $\varphi = 270°$ and $E^{inc} = (E_x^{inc}, 0, 0)$.

The mesh consists of about 88,000 nodes and 500,000 tetrahedra. The mean length of the tetrahedron edges is about $\lambda/12$. In Fig.10.2, we show the mesh split up into 16 subdomains. The time step is $T/40$.

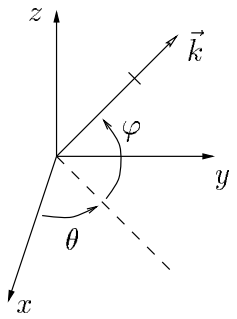

Figure 10.1: Definition of the angles.

We show in Fig.10.3 (resp., Fig.10.4) the contours of the E_x component of the exact (resp., computed) solution in a cross-section by the plane ($x = 0$). For the plotting, the sphere is considered as transparent so we see also the E_x component on the surface of the sphere.

We compare the computation times obtained with one processor of a Cray C90 computer (vectorized code) or with a given number of processors of a Cray T3E. For the sphere, the computation times correspond to 5 transition iterations and 10 control iterations, they include the initialization time (sub-domains meshes reading, matrices computation) but not the solution writing time since until now we use a global visualization code, so the global solution is written by only one processor. With one processor of Cray C90, the run time is 230 s and the run times of the parallel solver are displayed in Table 10.1. We give also in Table 10.1 the relative speed up (sp) (compared to the time required when using 4 processsors) and the optimal speed up (osp) possible for the given number of processors.

Table 10.1: Sphere: parallel solver results.

n	4	8	16	32	64
time(s)	228	118	66	37	22
sp	1	1.93	3.45	6.16	10.36
osp	1	2	4	8	16

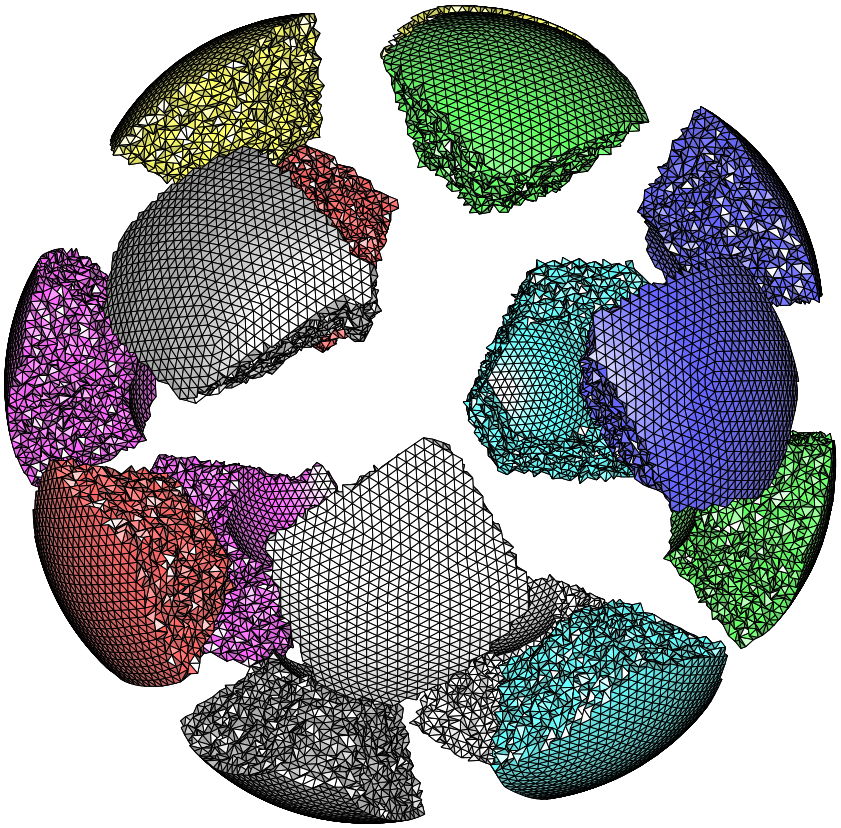

Figure 10.2: Sphere: Split up mesh.

The Exact Controllability formulation being mainly interesting for nonconvex obstacles, we show results for a 3-D semi-open cylindrical cavity and an idealized air-intake.

For the 3-D semi-open cylindrical cavity (test case of the Oxford Workshop ([25])), the internal radius is 0.5λ, the cavity length is 2λ and the wall thickness is 0.1λ. The mesh has about 357,000 nodes and 2,020,000 tetrahedra; if we denote by h_m the mean length of the triangle edges of the surface meshes, we have

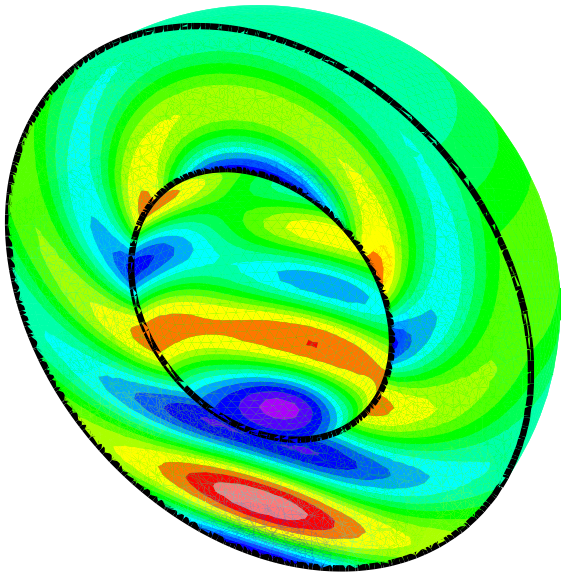

Figure 10.3: Sphere: contours of the E_x component of the scattered field. Exact solution.

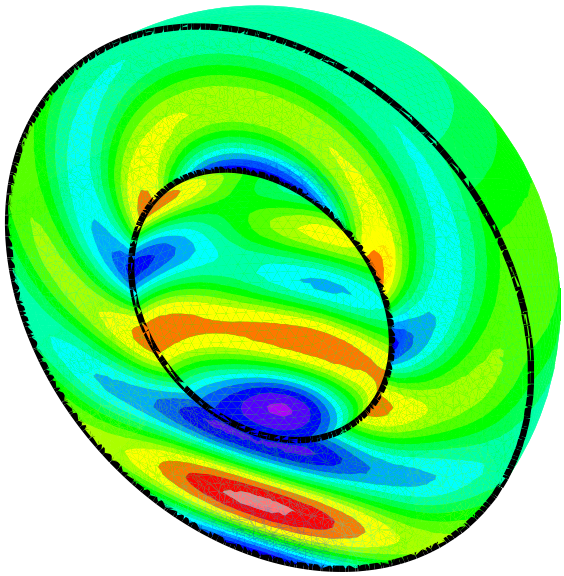

Figure 10.4: Sphere: contours of the E_x component of the scattered field. Computed solution.

$h_m \simeq \lambda/40$ inside the cavity and $h_m \simeq \lambda/15$ outside. Figure 10.5 shows the trace of the mesh on the cavity and Fig.10.6 shows the mesh split up into 32 subdomains. We use time step $\Delta t = T/160$. With the angles defined as in Fig.10.1, the incident wave is defined by $\theta = 150°, \varphi = 0°$ and $E^{inc} = (0, 0, E_z^{inc})$.

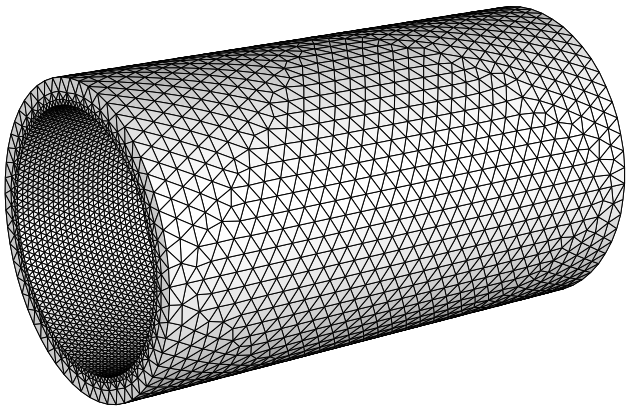

Figure 10.5: Trace of the mesh on the boundary of the cylindrical semi-open cavity.

We present on Figure 10.7 (resp. Fig. 10.8) the contours of the real part (resp. imaginary part) of the E_z component of the total field in the plane ($z = 0$). We use the terms of real and imaginary part by analogy with the frequency domain approach. These results are in good agreement with other Workshop participants' results (see e.g.[4]).

We compare the convergence history obtained with the control algorithm (Fig. 10.9) and the one obtained by just integrating in time the Maxwell equations in view to getting asymptotically the periodic solution (Fig.10.10). For this comparison, we define the L_2 residual :

$$Res = \frac{[\int_\Omega |e_0^{n+1} - e_0^n|^2 dx]^{\frac{1}{2}}}{[\int_\Omega |e_0^0|^2 dx]^{\frac{1}{2}}}$$

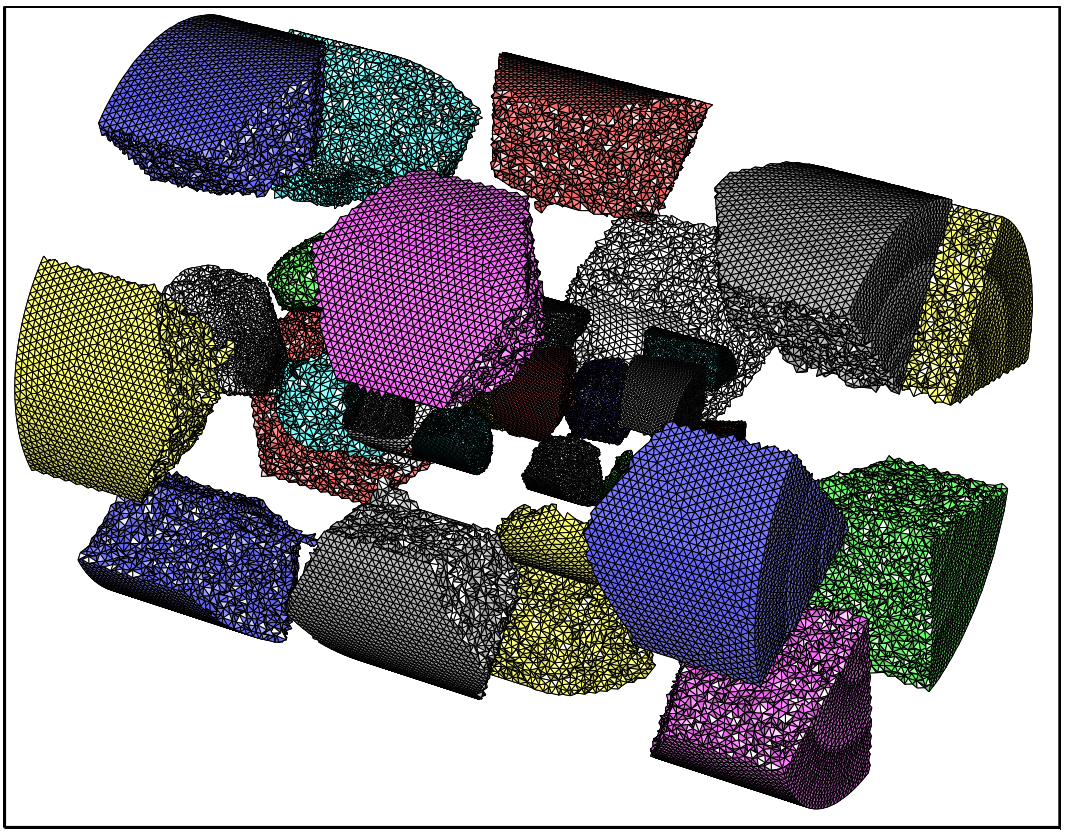

Figure 10.6: Cavity: Split up mesh.

with e_0^n denoting either the solution after n periods (asymptotic method) or the solution after n iterations of the conjugate gradient algorithm (control method); we denote e_0^0 the initial solution obtained after the transient process. We notice that, for this nonconvex obstacle, the asymptotic method does not converge (Fig.10.10) and the solution inside the cavity is not correct.

With the control approach and $t_{tr} = 2T$, the convergence is reached after 20 iterations and a run time of 44 min on one processor of a Cray C90. The run times of the parallel solver are displayed in Table 10.2. In Figure 10.11 we show the relative speed-up. In some cases the speed-up is lightly better than the optimal one, this

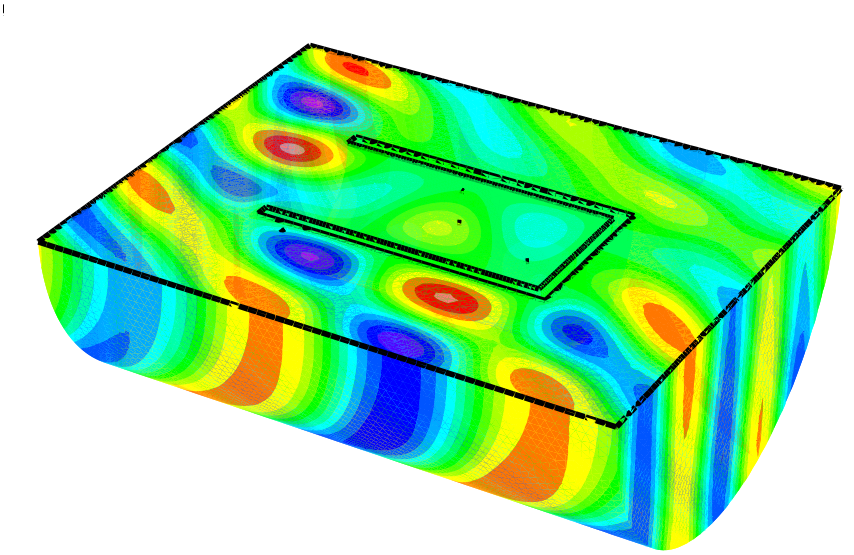

Figure 10.7: Cylindrical cavity: contours of the real part of the E_z component of the total field in a cross-section

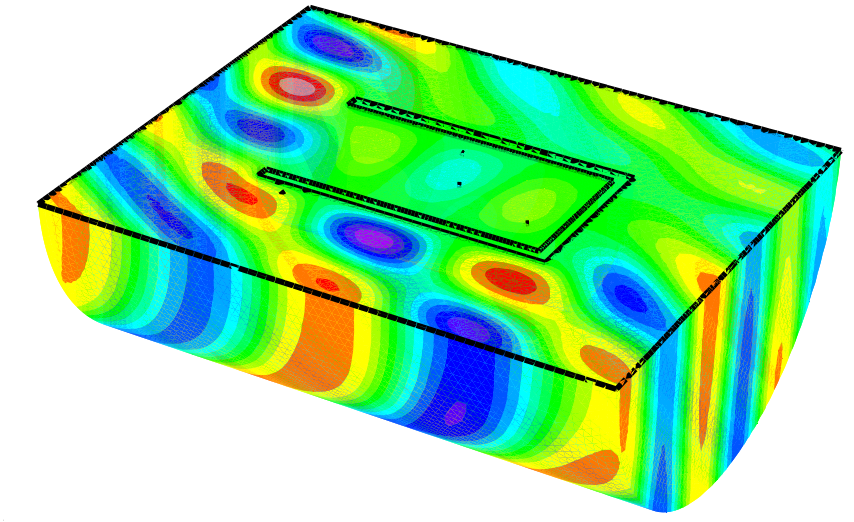

Figure 10.8: Cylindrical cavity: contours of the imaginary part of the E_z component of the total field in a cross-section

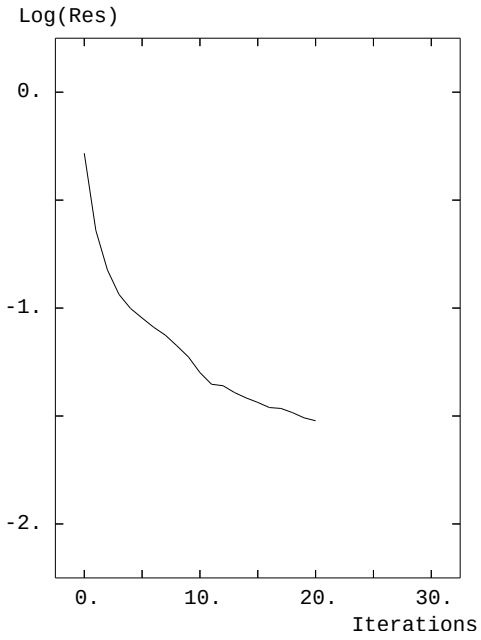

Figure 10.9: Residual convergence history with control algorithm.

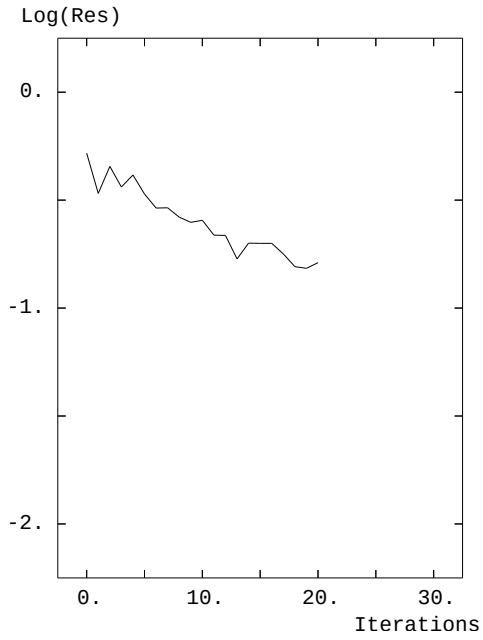

Figure 10.10: Residual convergence history without control.

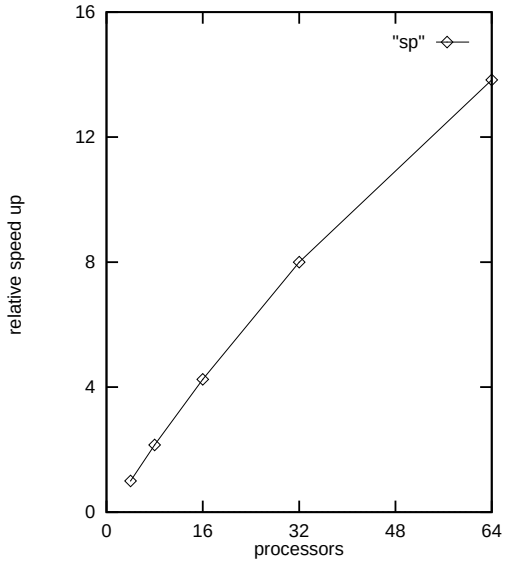

Figure 10.11: Cavity: Relative speed-up versus the number of processors.

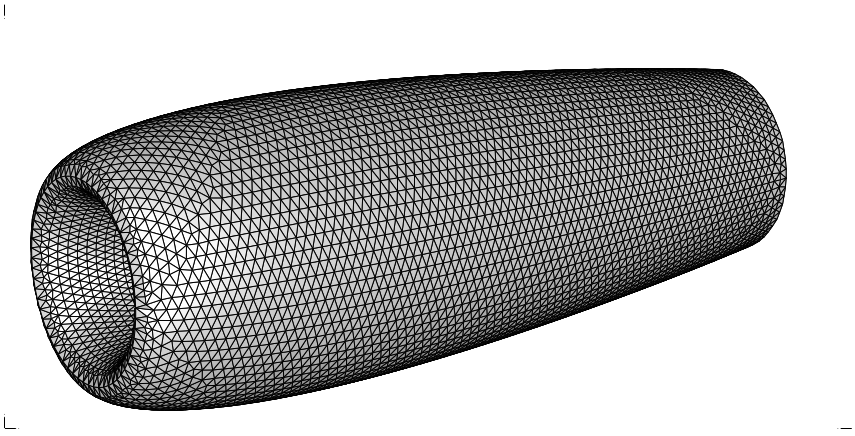

Figure 10.12: Trace of the mesh on the boundary of the air-intake.

is due to some balancing default (10% or 15%) which is worse when there are less subdomains.

Table 10.2: Cavity: parallel solver results.

n	4	8	16	32	64
time(s)	4081	1897	960	509	295
sp	1	2.15	4.25	8	13.8
osp	1	2	4	8	16

The last test case concerns an idealized air-intake as shown in Figure 10.12. The artificial boundary is a circular cylinder located, at least, at one wavelength of the obstacle. The characteristic length of the air-intake is 4 wavelengths. The Figure 10.12 presents the trace of the mesh on the obstacle. The mesh has about 327,000 nodes and 1,830,000 tetrahedra with $h_m \simeq \lambda/25$ inside the air-intake and $h_m \simeq \lambda/15$ outside. With the angles defined as in Fig. 10.1, the incident wave is defined by $\theta = 90°, \varphi = 45°$ and $E^{inc} = (E_x^{inc}, 0, 0)$. We use as time step $\Delta t = T/80$.

With $t_{tr} = 5T$, the convergence is reached after 20 control iterations and a CPU time of 37 min on a Cray C90, the parallel results are shown in Table 10.3. As in the previous case, we find some unexpected results due to balancing default, but globally on these different examples we see a good efficiency of the parallelisation.

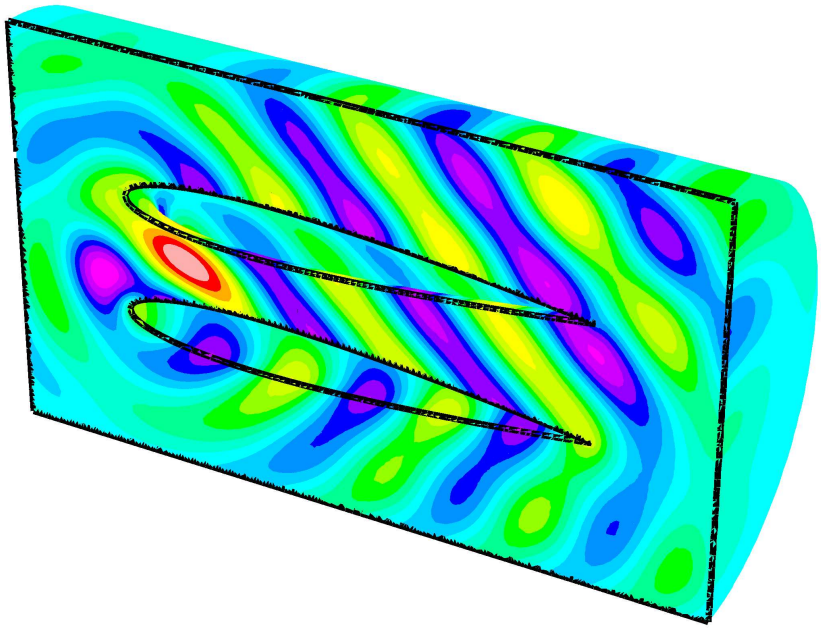

Figure 10.13: Air-intake: contours of the E_x component of the scattered field.

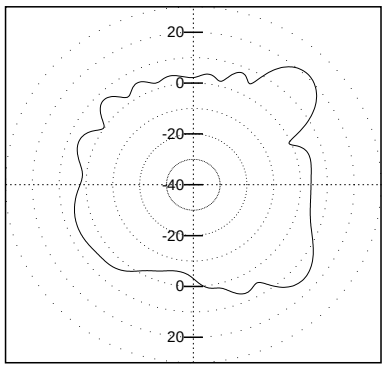

Figure 10.14: Air-intake : R.C.S. in the incidence plane.

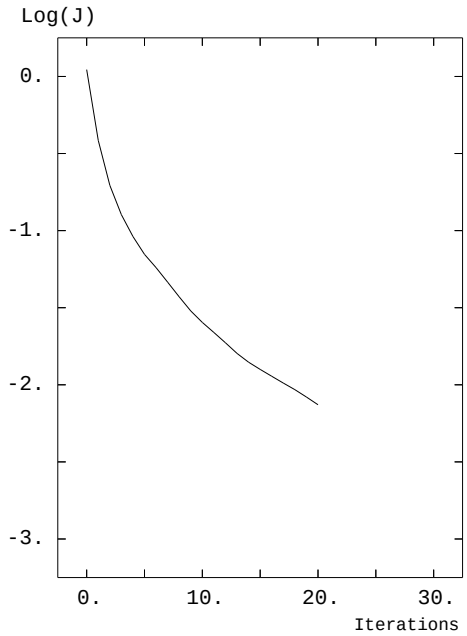

Figure 10.15: Cost function J convergence history.

In Figure 10.13 we have visualized the E_x component of the scattered field in the cross section by the plane $(x = 0)$. The RCS in the same plane is shown in Figure 10.14.

Figure 10.15 shows the convergence history of the cost function J defined by (4.7).

Table 10.3: Air intake: parallel solver results.

n	4	8	16	32	64
time(s)	3572	1695	992	397	211
sp	1	2.1	3.6	8.9	16.9
osp	1	2	4	8	16

11 Concluding Remarks

For the scattering problems considered, we obtain satisfying results with the proposed treatment of the divergence constraint (of penalization type with penalization parameter equal to one), associated wih the least-squares formulation.

Due to the field approach, the extension of the method to coated obstacles is possible, as it has been done for 2-D cases in [5, 15].

The algorithm being based on a completely explicit scheme, the parallelization of the code is easy and, due to the good convergence properties of the method, higher frequencies computations could be considered.

We have presented here an application of the Exact Controllability to the solution of a scattering problem on nonconvex reflectors, but, of course, the state equation solver can also be used for transient computations.

Acknowledgements The authors would like to thank P. Ayoub, C. Bardos, J. Erhel, J.L. Lions for helpful comments and suggestions.

Part of the computations have been done on the Cray C90 and Cray T3E of IDRIS, France.

References

[1] F. Assous, P. Degond, E. Heintze, P.A. Raviart, J. Segré, On a Finite Element Method for Solving the Three-Dimensional Maxwell Equations, *J. of Comp. Phys.*, Vol. 109, 1993, pp. 222-237.

[2] C. Bardos, J. Rauch, Variational algorithms for the Helmholtz equation using time evolution and artificial boundaries, *Asymptotic Analysis*, 9, 1994, pp. 101-117.

[3] M. Bernadou, P.-L. George, P. Joly, P. Laug, A. Perronet, E. Saltel, D. Steer, G. Vanderborck, M. Vidrascu, and A. Hassim, *MODULEF: A Modular Library of Finite Elements*, INRIA, 1986.

[4] A. Bespalov, Numerical Simulation of 3D Electromagnetic Scattering by Algebraic Fictitious Domain Method, *INRIA research report* N° 2729, 1995.

[5] M.O. Bristeau, E.J. Dean, R. Glowinski, V.K. Kwok, J. Périaux, Application of Exact Controllability to the computation of scattering waves, *Control Problems in Industry*, I. Lasiecka, B. Morton eds., Birkhäuser, 1995, pp. 17-41

[6] M.O. Bristeau, R. Glowinski, J. Périaux, Using exact controllability to solve the Helmholtz equation at high wave numbers, *Mathematical and Numerical Aspects of Wave Propagation*, R. Kleinman, Th. Angell, D. Colton, F. Santosa, I. Strakgold eds., SIAM, Philadelphia, PA., 1993, pp. 113-127.

[7] M.O. Bristeau, R. Glowinski, J. Périaux, Waves scattering using Exact Controllability, *Numerical Methods in Engineering 96*, J.A. Desideri, P. Le Tallec, E. Onate, J. Périaux, E. Stein eds, J. Wiley & Sons, 1996, pp. 96-103.

[8] M.O. Bristeau, R. Glowinski, J. Périaux, Controllability Methods for the Calculation of Time-Periodic Solutions. Application to Scattering. *INRIA research report* N° 3386, 1998.

[9] J. Erhel, Sparse matrix multiplication on vector computers, *Int. J. of High Speed Computing*, Vol. 2, N° 2, 1990, pp. 101-116.

[10] J. Erhel, M. Vidrascu, Adapting algorithms and data structures to supercomputers in finite element calculations. *Calcul des Structures et Intelligence Artificielle*, Vol. 3, J.M. Fouet, P. Ladevèze, R. Ohayon eds, Pluralis, 1989, pp. 403-417.

[11] R. Glowinski, J.L. Lions, Exact and Approximate Controllability for Distributed Parameter Systems, Part I, *Acta Numerica*, 1994, pp. 269-378.

[12] R. Glowinski, J.L. Lions, Exact and Approximate Controllability for Distributed Parameter Systems, Part II, *Acta Numerica*, 1996, pp. 159-333.

[13] B. Jiang, J. Wu, L.A. Povinelli, The Origin of Spurious Solutions in Computational Electromagnetics, *J. of Computational Physics*, Vol. 124; issue 2, 1996.

[14] M. Kern, Parallel solution of the wave equation using higher order finite elements, *Computational Science for the 21st Century*, M.O. Bristeau, G. Etgen, W. Fitzgibbon, J. L. Lions, J. Périaux, and M. F. Wheeler, eds., J. Wiley & Sons, Chichester, 1997, ch. 3, pp. 160-169.

[15] V. Kwok, Méthodes de Contrôlabilité Exacte et de Décomposition de Domaines pour la Résolution Numérique des équations de l'Electromagnétisme en régime harmonique dans un milieu hétérogène, Thèse, Univ. P. et M. Curie, 1995.

[16] P. Le Tallec, E. Saltel, M. Vidrascu, Solving large scale structural problems on parallel computers using domain decomposition techniques, *Advances in Parallel and Vector Processing for Structural Mechanics*, B. Topping and M. Papadrakakis, eds., Civil-Comp. Press, Athens, 1994, pp. 127-134.

[17] P. Le Tallec, M. Vidrascu, Solving Large Scale Structural Problems on Parallel Computers using Domain Decomposition, *Solving Large Scale Problems in Mechanics*, M. Papadrakakis, ed, J. Wiley & Sons, chap. 2, 1997, pp.49-82

[18] J.L. Lions, Exact Controllability, stabilization and perturbations for distributed systems, *SIAM Review* 30, 1988, pp. 1-68.

[19] P. Monk, A Comparison of Finite Element Methods for the Time-Dependent Maxwell Equations, *Math. and Num. Aspects of Wave Propagation Phenomena*, G. Cohen, L. Halpern, P. Joly eds, SIAM, 1991, pp. 80-88.

[20] G. Mur, The Finite Element Modeling of Three-Dimensional Electromagnetic Fields using Edge and Nodal Elements, *IEEE Trans. in Antennas and Prop.*, Vol. 41, N° 7, 1993.

[21] G.T. Ruck, D.E. Barrick, W.D. Stuart, C.K. Krichbaum, *Radar Cross Section Handbook*, G.T. Ruck ed., Plenum Press, New York, 1970.

[22] E. Seveno, Towards an adaptive advancing front method, *Proc. of 6th Int. Meshing Roundtable*, Park City, Utah, 1997.

[23] E. Seveno, Génération automatique de maillages tridimensionnels isotropes par une méthode frontale, Thèse, Univ. P. & M. Curie, 1998.

[24] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, *MPI: The Complete Reference*, The MIT Press, Cambridge, Massachusetts, 1996.

[25] Workshop on Approximations and Numerical Methods for the solution of Maxwell equations. F. El Dabaghi, K. Parrott, H. Steve eds., J. Wiley & Sons (to appear).

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399