

TWO-SCALE SIMULATION OF MAXWELL'S EQUATIONS

HYAM ABOUD¹, SÉBASTIEN JUND², STÉPHANIE SALMON², ERIC
SONNENDRÜCKER² AND HAMDI ZORGATI³

Abstract. We develop a numerical method for solving Maxwell's equations on a grid involving zones with cells of very different sizes, in order for example to compute sources coming from particles which need to be resolved on a very fine grid. The method is based on domain decomposition techniques which lead us to introduce two auxiliary problems and show theoretically how they allow us to calculate the solution of the initial problem. These two auxiliary problems are discretized using Edge Finite Elements introduced by Nédélec on two different scales which introduce some errors that we correct by setting to zero an operator we know has to be zero in the theoretical study.

1. INTRODUCTION.

In some cases, for example when a satellite or a space shuttle re-enters the atmosphere, particles are ionized around its surface and a small layer of plasma is created. This plasma generates an electromagnetic field which propagates away from the device. In order to simulate the propagation of this electromagnetic wave, it is necessary to resolve the sources quite accurately and thus use a very fine grid in a small layer around the device and a coarser grid in order to be able to compute the field which does not include very high frequencies far away. For this application a grid consisting of a coarse grid with a locally fine patch is very useful. There are quite a few other applications, including e.g. laser-plasma interaction, involving charged particles where an efficient numerical solution of Maxwell's equations depends on the use of such a type of grid based on a coarse grid with finer patches.

In order to devise a two-scale algorithm that can be used on a coarse grid with fine patches, we chose to extend the method developed by Glowinski and al. [2] for the steady-state Poisson equation using scalar Lagrange Finite Elements to the time dependent Maxwell equations solved using Raviart-Thomas-Nédélec Finite Elements. The idea is based on domain decomposition and multigrid techniques (see [1] or [4]). We first develop our method in the steady-state case and then extend it naturally to the time-dependent case using an implicit time discretization.

The paper is organized as follows: First we introduce a continuous problem based on two auxiliary problems with an artificial interface separating the two regions of interest. The continuous problem enables us to find an operator on the interface on which to iterate to obtain the continuity of the tangential component. Then we introduce the Raviart-Thomas-Nédélec finite element that we shall use and the discretized problem. In the following section we present the algorithm that is used to compute an approximation of the solution before presenting numerical results.

¹ Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie (Paris 6), Boîte Courrier 187, 4, place Jussieu, 75252 Paris cedex 05, France. Faculté des Sciences, Université Saint-Joseph, B.P 11-514 Riad El Solh, Beyrouth 1107 2050, Liban.

² Institut de Recherche Mathématiques Avancées, Université Louis Pasteur, Strasbourg, France.

³ CEREMADE, CNRS UMR 7534, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775, Paris, France.

2. CONTINUOUS PROBLEM.

Let $\Omega \subset \mathbb{R}^2$ be an open, bounded domain of \mathbb{R}^2 with sufficiently smooth boundary $\Gamma = \partial\Omega$ and let $\omega \subset \Omega$ be another open regular domain with boundary γ . We denote by \vec{n} the unit normal vector outward to the domain ω on γ .

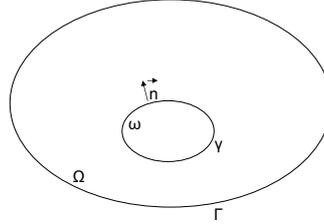


FIGURE 1. Domain.

We consider the following problem: find $U \in H(\overrightarrow{\text{curl}}, \Omega)$ such that

$$\begin{cases} U + \nabla \times \nabla \times U = f \text{ in } \Omega, \\ U \times \vec{n} = 0 \text{ on } \Gamma. \end{cases} \quad (2.1)$$

where $f \in (L^2(\Omega))^2$ is the sum of two functions $f_1, f_2 \in (L^2(\Omega))^2$, $f = f_1 + f_2$ with $\text{supp}(f_2) \subset \omega$.

Keeping in mind that Ω and ω differ by several scale factors, we introduce two auxiliary problems: find $V \in H(\overrightarrow{\text{curl}}, \Omega)$ such that

$$\begin{cases} V + \nabla \times \nabla \times V = f_1 \text{ in } \Omega \setminus \bar{\omega} \cup \omega, \\ V \times \vec{n} = 0 \text{ on } \Gamma, \\ [V] = 0 \text{ on } \gamma, \\ [(\nabla \times V) \times \vec{n}] = -\lambda \text{ on } \gamma \end{cases} \quad (2.2)$$

and find $W \in H(\overrightarrow{\text{curl}}, \Omega)$ such that

$$\begin{cases} W + \nabla \times \nabla \times W = f_1 + f_2 \text{ in } \omega, \\ W \times \vec{n} = V \times \vec{n} \text{ on } \gamma \end{cases} \quad (2.3)$$

where, $[\psi] = \psi^+ - \psi^-$ denotes the jump of ψ on γ , ψ^+ and ψ^- are respectively the restriction of ψ to $\Omega \setminus \bar{\omega}$ and ω . One can see that the solution W of problem (2.3) actually depends on the solution of problem (2.2), itself depending on the unknown λ . We now want to determine λ in order that the solution of (2.1) is given by

$$U = V^+ \cdot \chi(\Omega \setminus \bar{\omega}) + W \cdot \chi(\omega),$$

where V and W respectively solve (2.2) and (2.3). This means that we will be able to define the solution of our initial problem, as the solution of (2.2) in $\Omega \setminus \bar{\omega}$, which will be calculated on a large scale (knowing the solution will be smooth in this region), and as the solution of (2.3) in ω , which will be calculated on a small scale (knowing that f_2 , the peaking part of the external force, has its support in ω).

Since $\text{supp}(f_2) \subset \omega$, and with regard to (2.2) and (2.3), it is sufficient to remark that

Proposition 2.1. *If λ is chosen such that*

$$\lambda = (\nabla \times V^-) \times \vec{n} - (\nabla \times W) \times \vec{n}, \quad (2.4)$$

then the solution U of the initial problem (2.1) is given by

$$U = V^+ \cdot \chi(\Omega \setminus \overline{\omega}) + W \cdot \chi(\omega) \quad (2.5)$$

Proof. The proof is based on the following : Take $U \in H_0(\overrightarrow{\text{curl}}, \Omega)$ verifying (2.5). Then $\forall \phi \in H_0(\overrightarrow{\text{curl}}, \Omega)$,

$$\int_{\Omega} U \cdot \phi \, dX + \int_{\Omega} (\nabla \times U)(\nabla \times \phi) \, dX = \int_{\Omega} U \cdot \phi \, dX + \int_{\Omega \setminus \overline{\omega}} (\nabla \times V^+)(\nabla \times \phi) \, dX + \int_{\omega} (\nabla \times W)(\nabla \times \phi) \, dX \quad (2.6)$$

where

$$\int_{\Omega \setminus \overline{\omega}} (\nabla \times V^+)(\nabla \times \phi) \, dX = \int_{\Omega \setminus \overline{\omega}} (\nabla \times \nabla \times V^+) \cdot \phi \, dX + \int_{\Gamma \cup \gamma} (\nabla \times V^+ \times \vec{m}) \cdot \phi \, d\sigma \quad (2.7)$$

and

$$\int_{\omega} (\nabla \times W)(\nabla \times \phi) \, dX = \int_{\omega} (\nabla \times \nabla \times W) \cdot \phi \, dX + \int_{\gamma} (\nabla \times W \times \vec{n}) \cdot \phi \, d\sigma \quad (2.8)$$

where \vec{m} is the outbound normal on $\partial(\Omega \setminus \overline{\omega})$, within particular $\vec{m} = -\vec{n}$ on γ .

Hence combining (2.7) and (2.8), equation (2.6) becomes:

$$\begin{aligned} \int_{\Omega} U \cdot \phi \, dX + \int_{\Omega} (\nabla \times U)(\nabla \times \phi) \, dX &= \int_{\Omega \setminus \overline{\omega}} f_1 \cdot \phi \, dX + \int_{\omega} (f_1 + f_2) \cdot \phi \, dX \\ &+ \int_{\gamma} \left\{ ((\nabla \times W) \times \vec{n}) - ((\nabla \times V^+) \times \vec{n}) \right\} \cdot \phi \, d\sigma. \end{aligned} \quad (2.9)$$

Using (2.2) and (2.4) we have

$$\lambda := (\nabla \times V^-) \times \vec{n} - (\nabla \times V^+) \times \vec{n} = (\nabla \times V^-) \times \vec{n} - (\nabla \times W) \times \vec{n},$$

hence

$$\int_{\gamma} \left\{ ((\nabla \times W) \times \vec{n}) - ((\nabla \times V^+) \times \vec{n}) \right\} \cdot \phi \, d\sigma = 0.$$

Finally, we obtain

$$\int_{\Omega} U \cdot \phi \, dX + \int_{\Omega} (\nabla \times U)(\nabla \times \phi) \, dX = \int_{\Omega} (f_1 + f_2) \cdot \phi \, dX.$$

□

Therefore, defining the operator

$$T\lambda = ((\nabla \times W) \times \vec{n}) - ((\nabla \times V^+) \times \vec{n}),$$

it is sufficient to find λ such that $T\lambda = 0$. We can also prove that

Lemma 2.2. *The operator T is given by*

$$T\lambda = \lambda + ((\nabla \times \overline{W}) \times \vec{n}) - ((\nabla \times \overline{V}^+) \times \vec{n}) \quad (2.10)$$

where \overline{V} and \overline{W} satisfy respectively

$$\begin{cases} \overline{V} + \nabla \times \nabla \times \overline{V} = f_1 \text{ in } \Omega, \\ \overline{V} \times \vec{n} = 0 \text{ on } \Gamma, \end{cases} \quad (2.11)$$

and

$$\begin{cases} \overline{W} + \nabla \times \nabla \times \overline{W} = f_1 + f_2 \text{ in } \omega, \\ \overline{W} \times \vec{n} = \overline{V} \times \vec{n} \text{ on } \gamma. \end{cases} \quad (2.12)$$

Proof. We give a sketch of the proof. We introduce $\widetilde{W} = W - V^-$ and $\widetilde{\overline{W}} = \overline{W} - \overline{V}^-$ and notice that they both verify

$$\begin{cases} E + \nabla \times \nabla \times E = f_2 \text{ in } \omega, \\ E \times \vec{n} = 0 \text{ on } \gamma. \end{cases} \quad (2.13)$$

The uniqueness of the solution gives $(\nabla \times \widetilde{W}) \times \vec{n} = (\nabla \times \widetilde{\overline{W}}) \times \vec{n}$ which means that

$$(\nabla \times W) \times \vec{n} - (\nabla \times V^-) \times \vec{n} = (\nabla \times \overline{W}) \times \vec{n} - (\nabla \times \overline{V}^-) \times \vec{n}.$$

Then the result follows by adding and subtracting in each side of the above equation $(\nabla \times V^+) \times \vec{n}$ and by noting that

$$\begin{aligned} (\nabla \times \overline{V}^+) \times \vec{n} - (\nabla \times \overline{V}^-) \times \vec{n} &= 0, \\ (\nabla \times V^+) \times \vec{n} - (\nabla \times V^-) \times \vec{n} &= -\lambda, \end{aligned}$$

and

$$((\nabla \times W) \times \vec{n}) - ((\nabla \times V^+) \times \vec{n}) = T\lambda.$$

□

Thus, it is trivial that $T\lambda = 0$ is equivalent to $\lambda = ((\nabla \times \overline{V}^+) \times \vec{n}) - ((\nabla \times \overline{W}) \times \vec{n})$.

But if this is true when we consider the continuous problems (2.11) and (2.12), it is sure that solving the corresponding discretized problems and defining

$$\lambda_H = ((\nabla \times \overline{V}_H^+) \times \vec{n}) - ((\nabla \times \overline{W}_h) \times \vec{n}) \quad (2.14)$$

introduces errors that we will have to correct. But still, the approach is quite clear at this moment : by solving the discretized problems of (2.11) and (2.12) we define $\lambda_H = ((\nabla \times \overline{V}_H^+) \times \vec{n}) - ((\nabla \times \overline{W}_h) \times \vec{n})$ and then we can solve the discretized problems of (2.2) and (2.3) and finally define the solution of the initial problem (2.1) as in (2.5).

3. NUMERICAL APPROXIMATION.

We propose to use Raviart-Thomas-Nédélec finite-elements (see [3]). We recall that these finite-elements are given by the following triplet (K, P, Σ) where K denotes a quadrilateral, P denotes the polynomial space defined by $(P_0)^2 \oplus \begin{pmatrix} \alpha y \\ \beta x \end{pmatrix}$ and Σ is the set of four linear forms defined by $\sigma_i(f) = \int_{\Gamma_i} f \cdot \tau_i d\sigma$ where $\{\Gamma_i\}_{i=1..4}$ is the set of the four edges of the quadrilateral and $\{\tau_i\}_{i=1..4}$ is a set of four associated unit tangential vectors.

We give for example the four basis functions associated to a reference quadrilateral $[0, h] \times [0, h]$ (i.e. the four polynomial functions P_i in P which satisfies $\sigma_i(P_j) = \delta_{ij} \forall (i, j) \in \{1..4\}^2$):

$$P_1 = \begin{pmatrix} \frac{1}{h}(1 - \frac{y}{h}) \\ 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 \\ \frac{x}{h^2} \end{pmatrix}, \quad P_3 = \begin{pmatrix} \frac{y}{h^2} \\ 0 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 0 \\ \frac{1}{h}(1 - \frac{x}{h}) \end{pmatrix}.$$

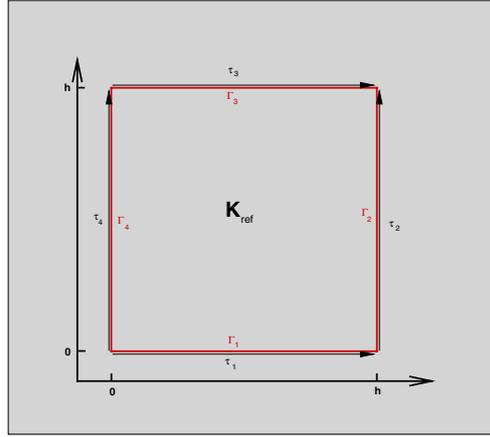


FIGURE 2. reference quadrilateral.

Once the two meshes are fixed, we define two spaces of finite-element approximation P_H and P_h respectively associated to the coarse mesh over the whole domain Ω and to the fine mesh over ω .

Assuming that the boundary γ can be written as the union of some edges of coarse mesh elements, we also define a finite-element approximation space Δ_H which corresponds to the tangential trace on γ of any function in P_H . More precisely, Δ_H is a 1D \mathbb{P}_0 finite-element space on γ discretized by the edges of the coarse mesh and for which the degrees of freedom are the integrals over each edge. It is in this last space that λ has to be searched.

The idea is to define an operator $T_H : \Delta_H \rightarrow \Delta_H$ approximating appropriately $T\lambda$ and then to search for λ_H such that $T_H\lambda_H = 0$ in order to correct the λ_H defined in (2.14). Thus, an approximation of the solution U of the problem (2.1) is given by

$$U_{Hh} = V_H^+ \cdot \chi(\Omega \setminus \bar{\omega}) + W_h \cdot \chi(\omega) \tag{3.1}$$

where V_H^+ is the restriction on $\Omega \setminus \bar{\omega}$ of the solution of the following equation

$$\int_{\Omega} V_H \cdot \phi \, dX + \int_{\Omega} (\nabla \times V_H)(\nabla \times \phi) \, dX = \int_{\Omega} f_1 \cdot \phi \, dX + \int_{\gamma} \lambda_H \cdot \phi \, d\sigma \quad \forall \phi \in P_H, \tag{3.2}$$

and W_h is the solution of

$$\int_{\omega} W_h \cdot \phi \, dX + \int_{\omega} (\nabla \times W_h)(\nabla \times \phi) \, dX = \int_{\omega} (f_1 + f_2) \cdot \phi \, dX \quad \forall \phi \in P_h \tag{3.3}$$

with boundary conditions $W_h \times \vec{n} = V_H \times \vec{n}$ on γ .

One can see that the tangential restriction $((\nabla \times V_H^+) \times \vec{n})$ on γ is by definition in Δ_H , but it is not the case of $((\nabla \times W_h) \times \vec{n})$. This means that we can not directly define $T_H = ((\nabla \times W_h) \times \vec{n}) - ((\nabla \times V_H^+) \times \vec{n})$.

Then, the most natural way to construct a consistent approximation of $T\lambda$ is to find $\delta_H \in \Delta_H$ such that

$$\int_{\gamma} \delta_H \cdot \Phi \, d\sigma = \int_{\gamma} ((\nabla \times W_h) \times \vec{n}) - ((\nabla \times V_H^+) \times \vec{n}) \cdot \Phi \, d\sigma \quad \forall \Phi \in \Delta_H, \tag{3.4}$$

and then define $T_H\lambda_H = \delta_H$. This can be computed very easily, as by using Raviart-Thomas-Nédélec finite-elements, any function in Δ_H is constant by edge.

A more sophisticated way to define T_H is the following : rather than integrating directly

$$\int_{\gamma} ((\nabla \times W_h) \times \vec{n}) \cdot \Phi \, d\sigma$$

and

$$\int_{\gamma} ((\nabla \times V_H^+) \times \vec{n}) \cdot \Phi \, d\sigma$$

in equation (3.4) which will be a very coarse approximation because we only use information on the boundary γ , we use the fact that if

$$V + \nabla \times \nabla \times V = f_1 \text{ in } \Omega, \quad (3.5)$$

then

$$\int_{\Omega \setminus \overline{\omega}} V_H^+ \cdot \phi \, dX + \int_{\Omega \setminus \overline{\omega}} (\nabla \times V_H^+) (\nabla \times \phi) \, dX = \int_{\Omega \setminus \overline{\omega}} f_1 \cdot \phi \, dX - \int_{\gamma} ((\nabla \times V_H^+) \times \vec{n}) \cdot \phi \, d\sigma \quad \forall \phi \in P_H, \quad (3.6)$$

and combining this with equation (3.2) we obtain

$$\int_{\gamma} ((\nabla \times V_H^+) \times \vec{n}) \cdot \Phi \, d\sigma = \int_{\omega} V_H^- \cdot \tilde{\Phi} \, dX + \int_{\omega} (\nabla \times V_H^-) (\nabla \times \tilde{\Phi}) \, dX - \int_{\omega} f_1 \cdot \tilde{\Phi} \, dX - \int_{\gamma} \lambda_H \cdot \Phi \, d\sigma, \quad (3.7)$$

for any Φ in Δ_H , where $\tilde{\Phi}$ is an extension of Φ in P_H .

In the same way we obtain another expression of

$$\int_{\gamma} ((\nabla \times W_h) \times \vec{n}) \cdot \Phi \, d\sigma = \int_{\omega} W_h \cdot \tilde{\Phi} \, dX + \int_{\omega} (\nabla \times W_h) (\nabla \times \tilde{\Phi}) \, dX - \int_{\omega} (f_1 + f_2) \cdot \tilde{\Phi} \, dX. \quad (3.8)$$

We finally propose to calculate $T_H \lambda_H = \delta_H$ where δ_H verifies : $\forall \Phi \in \Delta_H$,

$$\begin{aligned} \int_{\gamma} \delta_H \cdot \Phi \, d\sigma = \int_{\gamma} \lambda_H \cdot \Phi \, d\sigma &+ \left(\int_{\omega} W_h \cdot \tilde{\Phi} \, dX + \int_{\omega} (\nabla \times W_h) (\nabla \times \tilde{\Phi}) \, dX - \int_{\omega} (f_1 + f_2) \cdot \tilde{\Phi} \, dX \right) \\ &- \left(\int_{\omega} V_H^- \cdot \tilde{\Phi} \, dX + \int_{\omega} (\nabla \times V_H^-) (\nabla \times \tilde{\Phi}) \, dX - \int_{\omega} f_1 \cdot \tilde{\Phi} \, dX \right). \end{aligned} \quad (3.9)$$

4. RESOLUTION ALGORITHM.

As said at the end of section 2 the use of numerical resolution introduces some errors that have to be corrected. This means that defining λ_H by (2.14) does not ensure that $T_H \lambda_H = 0$. This is the reason why we decide to apply a fixed point method on $(I - T_H)$ (which should be accelerated by a GRMES algorithm if necessary) as follows :

- (1) Define $\lambda_H = 0$ identically as an initialisation.
- (2) Solve

$$\begin{cases} V_H + \nabla \times \nabla \times V_H = f_1 \text{ in } \Omega, \\ V_H \times \vec{n} = 0 \text{ on } \Gamma, \\ [V_H \times \vec{n}] = 0 \text{ on } \gamma, \\ [(\nabla \times V_H) \times \vec{n}] = -\lambda_H \text{ on } \gamma \end{cases} \quad (4.1)$$

This resolution gives $V_H \times \vec{n}$ on γ .

(3) Solve

$$\begin{cases} W_h + \nabla \times \nabla \times W_h = f_1 + f_2 \text{ in } \omega, \\ W_h \times \vec{n} = V_H \times \vec{n} \text{ on } \gamma. \end{cases} \quad (4.2)$$

(4) Define $T_H \lambda_H = \delta_H$ using equation (3.9).

(5) Finally define $\lambda_H = \lambda_H - T_H \lambda_H$.

(6) Return to step 2.

One can see that at the first iteration, with $\lambda_H = 0$, solving (4.1) and (4.2), defining $T_H \lambda_H = \delta_H$ and redefining $\lambda_H = \lambda_H - T_H \lambda_H$ corresponds exactly to what we said at the end of section 2, i.e solving the discretized problems of (2.11) and (2.12) and defining λ_H as in (2.14), so that defining U_{Hh} as in (3.1) with V_H and W_h calculated at the second iteration should be a good approximation of the solution of the initial problem. The further iterations correspond only to the correction of λ_H which lead to $T_H \lambda_H = 0$.

5. NUMERICAL RESULTS.

In this section we give numerical results which highlight the efficiency of the two-scale method. In the following, the solution of the initial problem using our two-scale method is called numerical solution, and the solution of the initial problem calculated directly on the extension of the fine grid over the whole domain is referred to be the reference solution. As we are only interested in the loss of precision when we use our method rather than a direct resolution on a uniformly refined mesh, we will only consider the relative error between these two solutions; which will be given in norm $L^2(\Omega)$ and $H(\overrightarrow{\text{curl}}, \Omega)$. We will split these errors into interior errors and exterior errors which respectively correspond to the errors in ω and $\Omega \setminus \overline{\omega}$. We choose our first and second test case in order that the solution of the initial problem in $\Omega \setminus \overline{\omega}$ is given by the restriction to $\Omega \setminus \overline{\omega}$ of the solution of (2.2). In test case three the solution of the initial problem in $\Omega \setminus \overline{\omega}$ really depends on λ , and for the fourth test case we take f_1 and f_2 as for the third test case, but we define ω slightly greater than $\text{supp}(f_2)$.

5.1. Test case 1.

For our first test case we consider $\Omega = [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\omega = [-\frac{\pi}{6}, \frac{\pi}{6}] \times [-\frac{\pi}{6}, \frac{\pi}{6}]$, $f_1 = \begin{pmatrix} 2 \cos(y) \\ 2 \cos(x) \end{pmatrix}$ as background source and $f_2 = 0$ identically. The exact solution is given by $U = \begin{pmatrix} \cos(y) \\ \cos(x) \end{pmatrix}$. This test case is well adapted to be solved by our low order finite-elements : the solution is constant in the tangential direction, i.e. in the direction where the finite-element is \mathbb{P}_0 .

If $f_2 = 0$, the only interest in applying our method is to verify that we already have a good approximation of the solution at the first iteration, and that this solution remains almost constant during the further iterations (the solution should be slightly modified because the resolution in ω , which makes appear $f_1 + f_2$ as source term, is better due to the finer grid, even if $f_2 = 0$).

We give in Figure 3 the first component of the reference solution and the numerical solution after one iteration. We give in Tables 1 to 3 the errors after 1, 2, 5, 10 and 50 iterations for 9×9 , 27×27 and 45×45 coarse grids and fine grids three times finer in each case.

5.2. Test case 2.

The aim of this second test case is to construct a solution of the initial problem (2.1) in $\Omega \setminus \overline{\omega}$ that does not depend on the solution of problem (2.3) in ω , this can be done easily by considering a Gaussian source term f_2 localized in ω which vanishes (numerically) on γ .

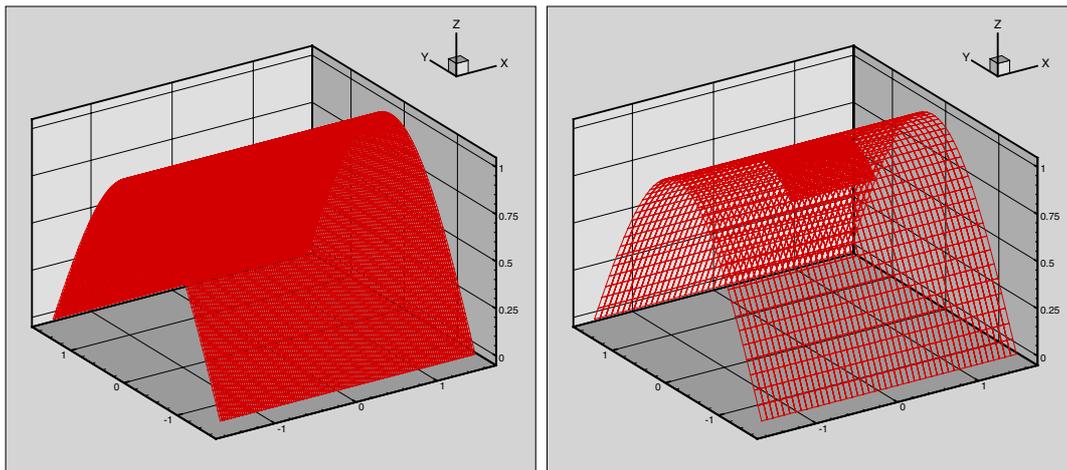


FIGURE 3. First field of the reference solution and the numerical solution after one iteration for the first test case.

Iteration	1	2	5	10
L^2 error	0.127792E-01	0.129802E-01	0.129983E-01	0.129983E-01
L^2 exterior error	0.141880E-01	0.144605E-01	0.144782E-01	0.144782E-01
L^2 interior error	0.376103E-02	0.300233E-02	0.305109E-02	0.305110E-02
$H(\overrightarrow{\text{curl}})$ error	0.599090E-01	0.598985E-01	0.598982E-01	0.598982E-01
$H(\overrightarrow{\text{curl}})$ exterior error	0.635364E-01	0.635261E-01	0.635258E-01	0.635258E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.106011E-02	0.451137E-03	0.433064E-03	0.433062E-03

TABLE 1. Numerical errors on a 9×9 grid for the first test case.

Iteration	1	2	5	10
L^2 error	0.142712E-02	0.144246E-02	0.144295E-02	0.144295E-02
L^2 exterior error	0.158463E-02	0.161307E-02	0.161352E-02	0.161352E-02
L^2 interior error	0.417327E-03	0.184581E-03	0.187740E-03	0.187741E-03
$H(\overrightarrow{\text{curl}})$ error	0.199886E-01	0.199878E-01	0.199878E-01	0.199878E-01
$H(\overrightarrow{\text{curl}})$ exterior error	0.212009E-01	0.212000E-01	0.212000E-01	0.212000E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.118266E-03	0.178047E-04	0.183512E-04	0.183514E-04

TABLE 2. Numerical errors on a 27×27 grid for the first test case.

Iteration	1	2	5	10
L^2 error	0.513971E-03	0.519377E-03	0.519473E-03	0.519473E-03
L^2 exterior error	0.570703E-03	0.580997E-03	0.581085E-03	0.581085E-03
L^2 interior error	0.150221E-03	0.595088E-04	0.602927E-04	0.602931E-04
$H(\overrightarrow{\text{curl}})$ error	0.119941E-01	0.119939E-01	0.119939E-01	0.119939E-01
$H(\overrightarrow{\text{curl}})$ exterior error	0.127216E-01	0.127214E-01	0.127214E-01	0.127214E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.425890E-04	0.753758E-05	0.769877E-05	0.769884E-05

TABLE 3. Numerical errors on a 45×45 grid for the first test case.

More precisely we consider $\Omega = [-12, 12] \times [-12, 12]$, $\omega = [-4, 4] \times [-4, 4]$, $f_1 = \begin{pmatrix} \left(\frac{\pi^2}{576} + 1\right) \cos\left(\frac{\pi y}{24}\right) \\ \left(\frac{\pi^2}{576} + 1\right) \cos\left(\frac{\pi x}{24}\right) \end{pmatrix}$

as background source and $f_2 = \begin{pmatrix} \eta e^{-\frac{(x^2+y^2)}{\varepsilon}} \left(1 + \frac{2}{\varepsilon} - 4\frac{(y^2 - xy)}{\varepsilon^2}\right) \\ \eta e^{-\frac{(x^2+y^2)}{\varepsilon}} \left(1 + \frac{2}{\varepsilon} - 4\frac{(x^2 - xy)}{\varepsilon^2}\right) \end{pmatrix}$ as local source in ω where $\varepsilon = 0.5$ and $\eta = 10$.

We give in Figure 4 the first field of the reference solution and the numerical solution after one iteration computed on a 45×45 coarse grid and a three times finer fine grid (this means the reference solution is calculated on a 135×135 grid). In Tables 4 to 6 we give the errors calculated in the same conditions as for the first test case.

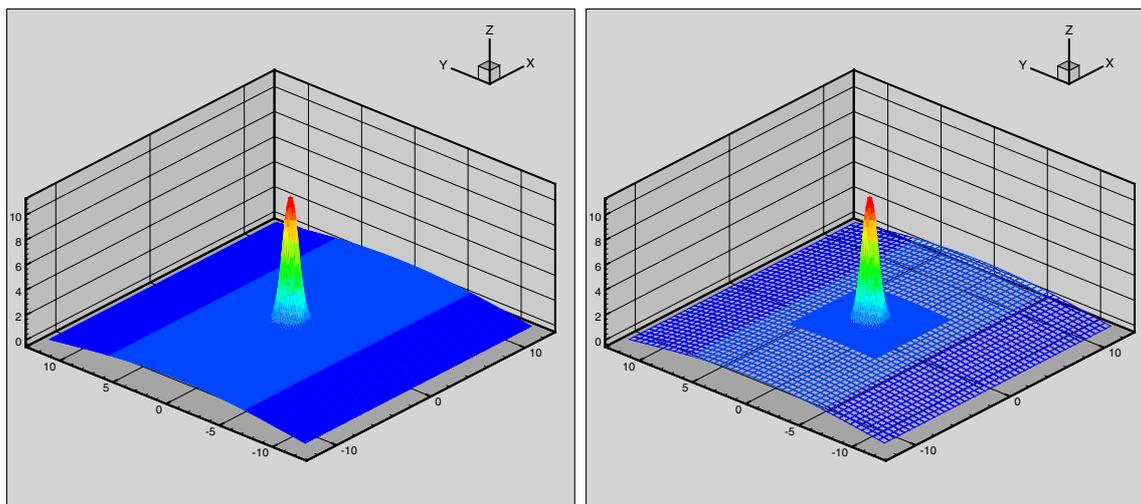


FIGURE 4. First field of the reference solution and the numerical solution after one iteration for the second test case.

Iteration	1	2	5	10
L^2 error	0.823368E-02	0.854288E-02	0.859216E-02	0.859221E-02
L^2 exterior error	0.101939E-01	0.105637E-01	0.106163E-01	0.106164E-01
L^2 interior error	0.132116E-02	0.154363E-02	0.165281E-02	0.165288E-02
$H(\overrightarrow{\text{curl}})$ error	0.946069E-02	0.947588E-02	0.948614E-02	0.948615E-02
$H(\overrightarrow{\text{curl}})$ exterior error	0.123393E-01	0.123511E-01	0.123566E-01	0.123566E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.760773E-03	0.927762E-03	0.106761E-02	0.106779E-02

TABLE 4. Numerical errors on a 9×9 grid for the second test case.

Iteration	1	2	5	10
L^2 error	0.869482E-03	0.874768E-03	0.875010E-03	0.875010E-03
L^2 exterior error	0.113507E-02	0.114716E-02	0.114750E-02	0.114750E-02
L^2 interior error	0.137202E-03	0.508224E-04	0.503533E-04	0.503526E-04
$H(\overrightarrow{\text{curl}})$ error	0.270281E-02	0.270232E-02	0.270232E-02	0.270232E-02
$H(\overrightarrow{\text{curl}})$ exterior error	0.410376E-02	0.410342E-02	0.410342E-02	0.410342E-02
$H(\overrightarrow{\text{curl}})$ interior error	0.589900E-04	0.308610E-04	0.313108E-04	0.313108E-04

TABLE 5. Numerical errors on a 27×27 grid for the second test case.

Iteration	1	2	5	10
L^2 error	0.312617E-03	0.312908E-03	0.312931E-03	0.312931E-03
L^2 exterior error	0.408765E-03	0.411261E-03	0.411291E-03	0.411291E-03
L^2 interior error	0.499898E-04	0.110458E-04	0.110195E-04	0.110195E-04
$H(\overrightarrow{\text{curl}})$ error	0.160328E-02	0.160314E-02	0.160314E-02	0.160314E-02
$H(\overrightarrow{\text{curl}})$ exterior error	0.246170E-02	0.246159E-02	0.246159E-02	0.246159E-02
$H(\overrightarrow{\text{curl}})$ interior error	0.210766E-04	0.672441E-05	0.677168E-05	0.677168E-05

TABLE 6. Numerical errors on a 45×45 grid for the second test case.

5.3. Test case 3.

In this test case, we consider $\Omega = [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\omega = [-\frac{\pi}{6}, \frac{\pi}{6}] \times [-\frac{\pi}{6}, \frac{\pi}{6}]$, $f_1 = \begin{pmatrix} 2 \cos(y) \\ 2 \cos(x) \end{pmatrix}$ as background source and $f_2 = 10 \cos(9x) \cos(9y) \chi(\omega) \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

We give in Figure 5 the first field of the reference solution, the numerical solution after one iteration and the numerical solution after two iterations. One can see that the algorithm produces exactly what we expected : what we see after the first iteration is not supposed to be an approximation of the solution but something that allows us to calculate λ_H which is an approximation of λ . Then, at the second iteration, the algorithm calculates an approximation of the solution of the initial problem. We give in Tables 7 to 9 the errors calculated in the same conditions as for the first test case.

Iteration	1	2	5	10
L^2 error	0.398497E+00	0.281039E+00	0.278372E+00	0.278371E+00
L^2 exterior error	0.493352E+00	0.379490E+00	0.387446E+00	0.387477E+00
L^2 interior error	0.347533E+00	0.223060E+00	0.211780E+00	0.211754E+00
$H(\overrightarrow{\text{curl}})$ error	0.562216E-01	0.454014E-01	0.450850E-01	0.450844E-01
$H(\overrightarrow{\text{curl}})$ exterior error	0.751863E-01	0.636354E-01	0.635357E-01	0.635356E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.302904E-01	0.164788E-01	0.151184E-01	0.151155E-01

TABLE 7. Numerical errors on a 9×9 grid for the third test case.

Iteration	1	2	5	10
L^2 error	0.387811E+00	0.740680E-01	0.731386E-01	0.731386E-01
L^2 exterior error	0.508526E+00	0.136386E+00	0.136992E+00	0.136993E+00
L^2 interior error	0.329459E+00	0.203193E-01	0.126115E-01	0.126112E-01
$H(\overrightarrow{\text{curl}})$ error	0.409944E-01	0.147756E-01	0.147125E-01	0.147125E-01
$H(\overrightarrow{\text{curl}})$ exterior error	0.515924E-01	0.222988E-01	0.222920E-01	0.222920E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.304299E-01	0.195685E-02	0.877946E-03	0.877902E-03

TABLE 8. Numerical errors on a 27×27 grid for the third test case.

5.4. Test case 4.

For our fourth test case we consider $\Omega = [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\omega = [-\frac{7\pi}{30}, \frac{7\pi}{30}] \times [-\frac{7\pi}{30}, \frac{7\pi}{30}]$, $f_1 = \begin{pmatrix} 2 \cos(y) \\ 2 \cos(x) \end{pmatrix}$ as background source and $f_2 = 10 \cos(9x) \cos(9y) \chi([-\frac{\pi}{6}, \frac{\pi}{6}] \times [-\frac{\pi}{6}, \frac{\pi}{6}]) \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

One can see that the only difference between this test case and the third test case is that we take ω

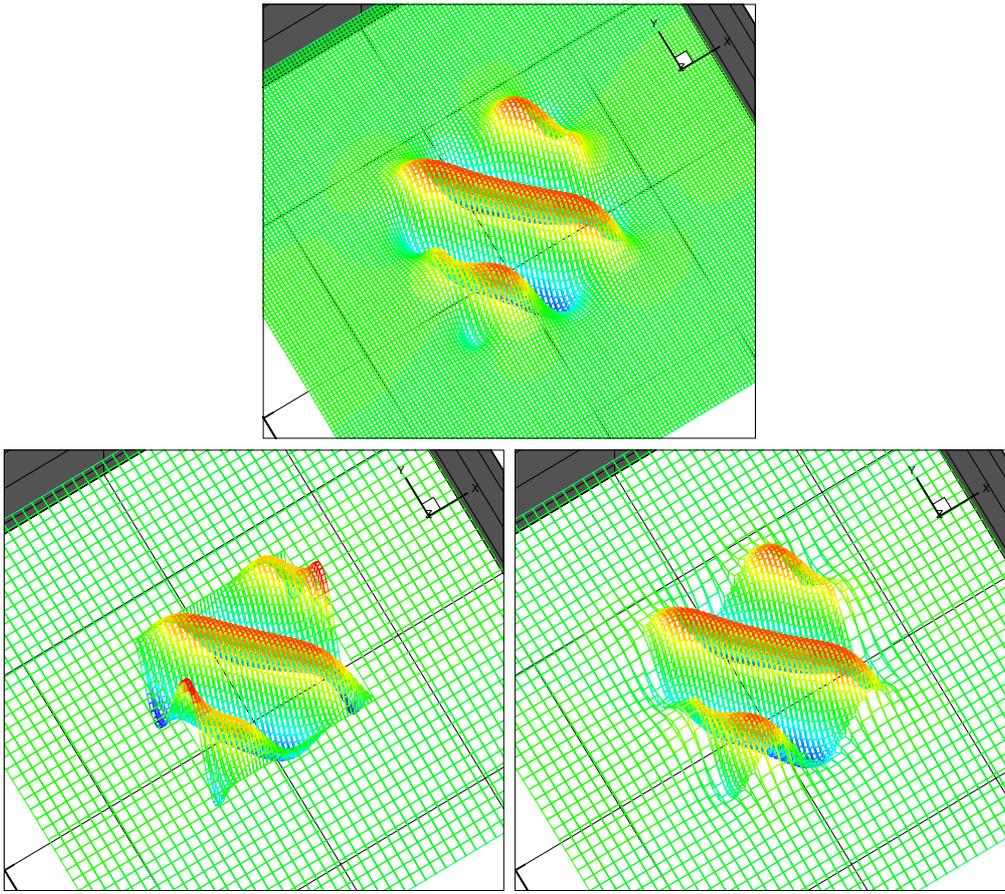


FIGURE 5. First field of the reference solution, the numerical solution after one iteration and the numerical solution after two iterations for the third test case.

Iteration	1	2	5	10
L^2 error	0.386945E+00	0.435838E-01	0.433808E-01	0.433808E-01
L^2 exterior error	0.509749E+00	0.821687E-01	0.822958E-01	0.822958E-01
L^2 interior error	0.328072E+00	0.668444E-02	0.350526E-02	0.350523E-02
$H(\overrightarrow{\text{curl}})$ error	0.396788E-01	0.886878E-02	0.885411E-02	0.885411E-02
$H(\overrightarrow{\text{curl}})$ exterior error	0.492684E-01	0.134717E-01	0.134723E-01	0.134723E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.304401E-01	0.746564E-03	0.296818E-03	0.296817E-03

TABLE 9. Numerical errors on a 45×45 grid for the third test case.

slightly greater than $\text{supp}(f_2)$. We give in Figure 6 the first field of the reference solution and the numerical solution after two iterations. We give in Table 10 the errors calculated on a 45×45 coarse grid and a three time finer fine grid.

5.5. Interpretation

One can see that the error over the whole domain does not seem to decrease any more after a few iterations. After a closer look at the results we can precise what happens : the main part of the error is due to what we called the exterior error, i.e. the error in $\Omega \setminus \overline{\omega}$. This error does not decrease any more after a few iterations; the interior error, which is considerably smaller, really decreases iteration after iteration so that we can conclude that the only error we add by using our two-scale decomposition comes

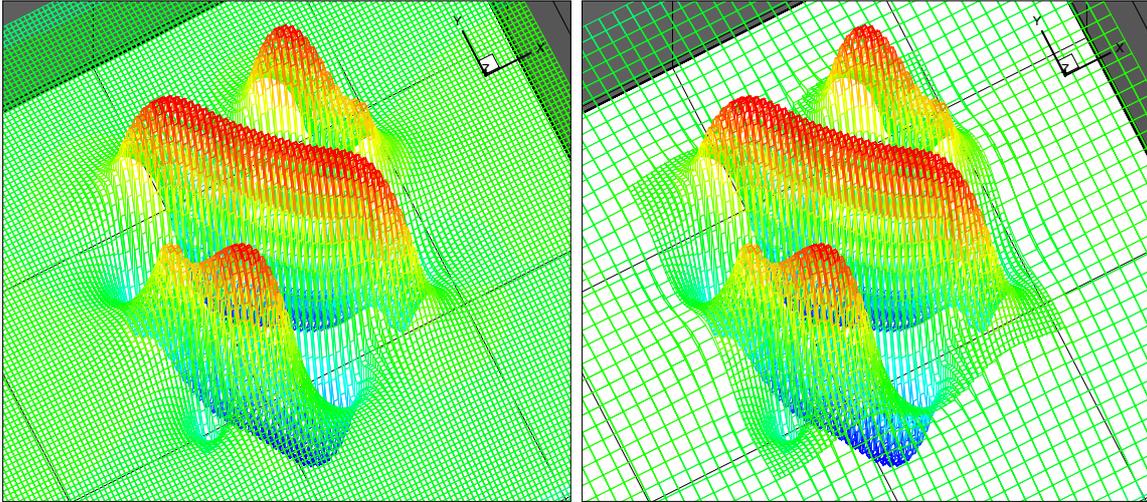


FIGURE 6. First field of the reference solution and the numerical solution after two iterations for the fourth test case.

Iteration	1	2	5	10
L^2 error	0.881475E-01	0.743515E-02	0.741946E-02	0.741946E-02
L^2 exterior error	0.145889E+00	0.182483E-01	0.182695E-01	0.182695E-01
L^2 interior error	0.714757E-01	0.768414E-03	0.398887E-03	0.398886E-03
$H(\overrightarrow{\text{curl}})$ error	0.250256E-01	0.695290E-02	0.695237E-02	0.695237E-02
$H(\overrightarrow{\text{curl}})$ exterior error	0.371278E-01	0.120720E-01	0.120716E-01	0.120716E-01
$H(\overrightarrow{\text{curl}})$ interior error	0.159048E-01	0.815904E-04	0.357121E-04	0.357123E-04

TABLE 10. Numerical errors on a 45×45 grid for the fourth test case.

from the fact that we use a coarser finite-element space in $\Omega \setminus \overline{\omega}$ which is unable to fit the solution as well as the finer space defined on the extension of the fine grid over the whole domain Ω . Indeed, it is not only necessary to assure that the finite-element space is fine enough to capture the source term (what we did by defining P_h in ω) but also that the finite-element space is fine enough to represent the solution over the whole domain Ω , which is not trivial as we use low order finite-elements, knowing that even if a peaking source term is located in ω , it also perturbs, in general, the solution in $\Omega \setminus \overline{\omega}$. We highlight this consideration by simply taking ω slightly larger than $\text{supp}(f_2)$, as in the fourth test case, and see how much better the results are, compared to the third test case.

PERSPECTIVES

One can see that this work opens many perspectives of investigation : for example some of the next studies that will be done are the use of higher order finite-elements to ensure a better resolution in $\Omega \setminus \overline{\omega}$ even on a coarse grid, and applying the multi-domain method for solving, at each time step, a multi-scale stationary Maxwell problem. Another point of interest is the following : the interest of our method remains in the fact that we use a finer finite-element space P_h where the source term is non-smooth, but rather than refining the mesh, this could be done by using local higher order finite-elements in ω ...

ACKNOWLEDGEMENTS

The authors thank the CEMRACS 2005 organizers.

REFERENCES

- [1] Y. ACHDOU, Y. MADAY The mortar element method with overlapping subdomains., *SIAM J. Numer. Anal.* 40(2), 601-628 (2002)
- [2] R. GLOWINSKI, J. HE, J. RAPPAZ & J. WAGNER A multi-domain method for solving numerically multi-scale elliptic problems , *R. Glowinski and al., C. R. Acad. Sci. Paris, Ser. I 338 (2004)*.
- [3] J.C. NÉDÉLEC A new family of mixed finite elements in \mathbb{R}^3 , *Numer. Math.*, 50:57-81, 1986.
- [4] A. QUATERONI, A. VALLI Domain Decomposition Methods for Partial Differential Equations, *Numerical Mathematics and Scientific Computation. Oxford Science publications, Oxford, 1999*.
- [5] SCILAB <http://scilab.inria.fr>