Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 2044–2072
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A domain decomposition method for solving the three-dimensional time-harmonic Maxwell equations discretized by discontinuous Galerkin methods

Victorita Dolean [a,b], Stéphane Lanteri [a,*], Ronan Perrussel [c]

[a] *INRIA, 2004 Route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France*
[b] *Université de Nice-Sophia Antipolis, Laboratoire J.A. Dieudonné, CNRS UMR 6621, 06108 Nice Cedex, France*
[c] *Laboratoire Ampère, CNRS UMR 5005, Universìté de Lyon, Ecole Centrale de Lyon, 69134 Écully Cedex, France*

Received 11 June 2007; received in revised form 4 October 2007; accepted 5 October 2007
Available online 16 October 2007

Abstract

We present here a domain decomposition method for solving the three-dimensional time-harmonic Maxwell equations discretized by a discontinuous Galerkin method. In order to allow the treatment of irregularly shaped geometries, the discontinuous Galerkin method is formulated on unstructured tetrahedral meshes. The domain decomposition strategy takes the form of a Schwarz-type algorithm where a continuity condition on the incoming characteristic variables is imposed at the interfaces between neighboring subdomains. A multifrontal sparse direct solver is used at the subdomain level. The resulting domain decomposition strategy can be viewed as a hybrid iterative/direct solution method for the large, sparse and complex coefficients algebraic system resulting from the discretization of the time-harmonic Maxwell equations by a discontinuous Galerkin method.

Keywords: Computational electromagnetism; Time-harmonic Maxwell's equations; Discontinuous Galerkin method; Unstructured meshes; Domain decomposition method; Schwarz algorithm

1. Introduction

This work aims at developing a high-performance numerical methodology for the computer simulation of time-harmonic electromagnetic wave propagation problems in irregularly shaped domains and heterogeneous media. In this context, we are naturally led to consider volume discretization methods (*i.e.* finite difference, finite volume or finite element methods) as opposed to surface discretization methods (*i.e.* boundary element method). Most of the related existing works deal with the second-order form of the time-harmonic Maxwell equations discretized by a conforming finite element method [36]. More recently, discontinuous Galerkin

* Corresponding author. Tel.: +33 4 92 38 7734.
E-mail address: Stephane.Lanteri@inria.fr (S. Lanteri).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.10.004

methods [29] have also been considered for this purpose. Here, we concentrate on the first-order form of the time-harmonic Maxwell equations discretized by discontinuous Galerkin methods on unstructured tetrahedral meshes. While it keeps almost all the advantages of the finite element method (large spectrum of applications, complex geometries, etc.), the discontinuous Galerkin method has other nice properties which explain the renewed interest it gains in various domains in scientific computing (as witnessed by books or special issues of journals dedicated to this method [9,10,12]): easy extension to higher order interpolation (one may increase the degree of the polynomials in the whole mesh as easily as for spectral methods and moreover this can be done very locally), no global mass matrix to invert (when solving time-domain systems of partial differential equations using an explicit time scheme), easy handling of complex meshes (the grid may be a classical conforming finite element mesh, a non-conforming one or even a hybrid mesh made of various types of element), natural treatment of discontinuous solutions and coefficient heterogeneities, nice parallelization properties (the compact nature of a discontinuous Galerkin scheme is in favor of high computation to communication ratio especially for high-order interpolation methods). Not all of these features are exploited in the present study: we limit ourselves to the lowest order schemes (constant and linear interpolation) and we assume conforming meshes. Higher order nodal interpolation is considered in [15] in the context of the two-dimensional time-harmonic Maxwell equations while non-conforming triangular meshes in conjunction with high-order discontinuous Galerkin time-domain methods are studied in [20].

Theoretical results concerning discontinuous Galerkin methods applied to the time-harmonic Maxwell equations have been obtained by several authors. Most of these works use a mixed formulation [37,30] but discontinuous Galerkin methods on the non-mixed formulation have also been proved to converge (interior penalty techniques [29,7] and local discontinuous Galerkin methods [7]). However, to our knowledge, a direct convergence analysis of discontinuous Galerkin methods applied to the first-order time-harmonic Maxwell system has not been conducted so far. Our contribution in [15] is a numerical evaluation of the convergence of discontinuous Galerkin methods based on centered and upwind fluxes and nodal polynomial interpolation applied to the first-order time-harmonic Maxwell system in the two-dimensional case. These methods have previously been shown to converge in the time-domain case [28,21].

In this paper, we are concerned with the application of such discontinuous Galerkin methods to the discretization of the three-dimensional time-harmonic Maxwell equations taken in the form of a first-order system of partial differential equations. Our efforts are towards the design of a parallel solution strategy for the resulting large, sparse and complex coefficients algebraic systems. Indeed, as far as non-trivial propagation problems are considered, the associated matrix operators are in most cases solved with difficulty by classical iterative methods. The preconditioning issues for highly indefinite and non-symmetric matrices is for instance discussed by Benzi et al. in [3] in the context of incomplete factorization and sparse approximate inverse preconditioners. If a robust and efficient solver is sought then a sparse direct method is the most practical choice. Over the last decade, significant progress has been made in developing parallel direct methods for solving sparse linear systems, due in particular to advances made in both the combinatorial analysis of Gaussian elimination process, and on the design of parallel block solvers optimized for high-performance computers [2,27]. However, direct methods will still fail to solve very large three-dimensional problems, due to the potentially huge memory requirements for these cases. Iterative methods can be used to overcome this memory problem. However, a better solution can be found, combining advantages of both iterative and direct methods. For example, a popular approach is domain decomposition where one splits the computational domain into smaller subdomains and then uses a direct solver inside each subdomain coupled with an iterative solver on the interfaces (artificial boundaries) between subdomains. This approach is adopted in this work.

Domain decomposition methods are flexible and powerful techniques for the parallel numerical solution of systems of partial differential equations. Concerning their application to time-harmonic wave propagation problems, the simplest algorithm was proposed by Després [13] for solving the Helmholtz equation and then extended and generalized for the time-harmonic Maxwell equations in [14,11,1]. The analysis of a larger class of Schwarz algorithms has been performed recently in [16] where optimized transmission conditions are used. The latter extends the idea of the most general, optimized interface conditions designed for the Helmholtz problem in [23]. Our ultimate objective is the design and application of optimized Schwarz algorithms in conjunction with discontinuous Galerkin methods. The first step in this direction is understanding and analyzing classical overlapping and non-overlapping Schwarz algorithms in the discrete framework of these methods. To

our knowledge, except in Helluy [25], where such an algorithm is applied to a discretization of the first-order time-harmonic Maxwell equations by an upwind finite volume method, no other attempts for higher order discontinuous Galerkin methods or different kind of fluxes can be found in the literature.

A classical domain decomposition strategy is adopted in this study which takes the form of a Schwarz algorithm where Després type conditions [14] are imposed at the interfaces between neighboring subdomains. These conditions actually translate into a continuity condition for the incoming characteristic variables in the case of the first-order Maxwell system. A similar approach (using Robin transmission conditions) but applied to a second-order form of the Maxwell system, and in conjuction with a non-conforming finite element discretization, is presented in [34,42]. The main reason which led us to adopt the first-order formulation of the Maxwell equations, as pointed out in [16], stems from the fact that one can use the properties of the underlying time-domain problem, which is hyperbolic, to derive all possible classes of interface conditions. The use of higher order (*i.e.* optimized) transmission conditions is the natural prolongation of the present work and this has already been done in the two-dimensional case in [16] using a finite volume method formulated on a quadrangular grid. A second motivation lies in the added flexibility with regards to the choice of the numerical flux (*i.e.* centered or upwind scheme) given that we regard a discontinuous Galerkin formulation as a natural route to the design of a high order, compact stencil finite volume method. Finally, from the practical point of view and for what concerns the discontinuous Galerkin formulation on tetrahedral meshes, the software developments realized in this study build upon our former works on the numerical solution of the time-domain Maxwell equations [21] using the first-order system. However, beside the above mentioned motivations and advantages, it is clear that the use of the first-order form of the Maxwell equations leads to an algebraic system of equations with twice the number of unknowns compared to the discrete system associated with a second-order wave form. From the computational point of view, this can put a severe constraint on the required processing and memory capacities, especially in the context of a discontinuous Galerkin formulation, which further emphasize that parallel processing is a mandatory path. Nevertheless, it is also worthwhile to note that the matrix operator of the discrete system associated with a second-order wave form has a larger bandwidth and in general a worse conditioning than that resulting from the discretization of the first-order system.

A multifrontal sparse direct solver is used at the subdomain level. The resulting domain decomposition strategy can be viewed as a hybrid iterative/direct solution method for the large, sparse and complex coefficients algebraic system resulting from the discretization of the time-harmonic Maxwell equations by a discontinuous Galerkin method.

The rest of this paper is organized as follows. In Section 2, we formulate the continuous boundary value problem to be solved. Then, in Section 3, the adopted Schwarz-type domain decomposition method is introduced. Section 4 is devoted to the discretization of the global and domain decomposed boundary value problems. A well-posedness result for a perturbed discrete problem which generalizes the idea of [6] to higher order discontinuous Galerkin methods is established. Finally, in Section 5, numerical strategies for solving local problems as well as parallel computing aspects are discussed and experimental results are presented. Beside classical scattering test problems, we also consider a more challenging situation which consists in the propagation of a plane wave in a realistic geometric model of human head tissues.

2. Formulation of the continuous problem

The system of non-dimensioned time-harmonic Maxwell's equations can be written under the following form:

$$\begin{cases} \mathrm{i}\omega\varepsilon_r \boldsymbol{E} - \mathrm{curl}\boldsymbol{H} = -\boldsymbol{J}, \\ \mathrm{i}\omega\mu_r \boldsymbol{H} + \mathrm{curl}\boldsymbol{E} = 0, \end{cases} \tag{2.1}$$

where $\boldsymbol{E}$ and $\boldsymbol{H}$ are the unknown electric and magnetic fields and $\boldsymbol{J}$ is a known current source. The parameters ε_r and μ_r are respectively the complex-valued relative dielectric permittivity (integrating the electric conductivity) and the relative magnetic permeability; we consider here the case of linear isotropic media. The angular frequency of the problem is given by ω. Eq. (2.1) is solved in a bounded domain Ω. On the boundary $\partial\Omega = \Gamma_a \cup \Gamma_m$, the following boundary conditions are imposed:

$$\begin{aligned}&-\text{a perfect electric conductor (PEC) condition on } \Gamma_m : \boldsymbol{n}\times\boldsymbol{E}=0,\\ &-\text{a Silver-Müller (first-order absorbing condition) condition}\\ &\text{on } \Gamma_a : \boldsymbol{n}\times\boldsymbol{E}+\boldsymbol{n}\times(\boldsymbol{n}\times\boldsymbol{H})=\boldsymbol{n}\times\boldsymbol{E}^{\text{inc}}+\boldsymbol{n}\times(\boldsymbol{n}\times\boldsymbol{H}^{\text{inc}}).\end{aligned} \tag{2.2}$$

The vectors $\boldsymbol{E}^{\text{inc}}$ and $\boldsymbol{H}^{\text{inc}}$ represent the components of an incident electromagnetic wave and $\boldsymbol{n}$ denotes the unitary outward normal. Eqs. (2.1) and (2.2) can be further rewritten, assuming $\boldsymbol{J}$ equals to 0, under the following form:

$$\begin{cases} \mathrm{i}\omega G_0\boldsymbol{W}+G_x\partial_x\boldsymbol{W}+G_y\partial_y\boldsymbol{W}+G_z\partial_z\boldsymbol{W}=0 \quad \text{in } \Omega,\\ (M_{\Gamma_m}-G_{\boldsymbol{n}})\boldsymbol{W}=0 \quad \text{on } \Gamma_m,\\ (M_{\Gamma_a}-G_{\boldsymbol{n}})(\boldsymbol{W}-\boldsymbol{W}^{\text{inc}})=0 \quad \text{on } \Gamma_a. \end{cases} \tag{2.3}$$

where $\boldsymbol{W}=\begin{pmatrix}\boldsymbol{E}\\ \boldsymbol{H}\end{pmatrix}$ is the new unknown vector and $G_0=\begin{pmatrix}\varepsilon_r\mathrm{I}_3 & 0_{3\times3}\\ 0_{3\times3} & \mu_r\mathrm{I}_3\end{pmatrix}$. The terms I_3 and $0_{3\times3}$ denote respectively the identity matrix and a null matrix, of dimensions 3×3. The real part of G_0 is symmetric positive definite and its imaginary part, which appears for instance in the case of conductive materials, is symmetric negative. Denoting by $(\boldsymbol{e}^x,\boldsymbol{e}^y,\boldsymbol{e}^z)$ the canonical basis of $\mathbb{R}^3$, the matrices G_l with $l\in\{x,y,z\}$ are given by:

$$G_l=\begin{pmatrix}0_{3\times3} & N_{\boldsymbol{e}^l}\\ N_{\boldsymbol{e}^l}^t & 0_{3\times3}\end{pmatrix} \quad \text{where for a vector} \quad \boldsymbol{v}=\begin{pmatrix}v_x\\ v_y\\ v_z\end{pmatrix}, \quad N_{\boldsymbol{v}}=\begin{pmatrix}0 & v_z & -v_y\\ -v_z & 0 & v_x\\ v_y & -v_x & 0\end{pmatrix}.$$

In the following we denote by $G_{\boldsymbol{n}}$ the sum $G_x n_x+G_y n_y+G_z n_z$ and by $G_{\boldsymbol{n}}^+$ and $G_{\boldsymbol{n}}^-$ its positive and negative parts.[1] We also define $|G_{\boldsymbol{n}}|=G_{\boldsymbol{n}}^+-G_{\boldsymbol{n}}^-$. In order to take into account the boundary conditions, the matrices M_{Γ_m} and M_{Γ_a} are given by:

$$M_{\Gamma_m}=\begin{pmatrix}0_{3\times3} & N_{\boldsymbol{n}}\\ -N_{\boldsymbol{n}}^t & 0_{3\times3}\end{pmatrix} \quad \text{and} \quad M_{\Gamma_a}=|G_{\boldsymbol{n}}|.$$

3. A classical domain decomposition method

We consider now the problem (2.3). In order to ease the presentation we decompose the domain Ω into two overlapping or non-overlapping subdomains Ω_1 and Ω_2 but the extension of the formulation of the method to any number of subdomains is straightforward. We define $\Gamma_{12}=\partial\Omega_1\cap\Omega_2$ and $\Gamma_{21}=\partial\Omega_2\cap\Omega_1$. In the following we denote by $\boldsymbol{n}_{ij}$ the outward normal vector to the interface Γ_{ij} with i,j in $\{1,2\}$. We solve system (2.3) in both subdomains and we enforce on the subdomain interfaces the continuity of the incoming characteristic variables which provides a so-called *classical Schwarz algorithm* (see [16] for details). The classical Schwarz algorithm allows to compute the $(n+1)$th iterate of the solution from the nth iterate, starting from an arbitrary initial guess, by solving local problems and then exchanging information between artificial boundaries, called interfaces. This algorithm is given by:

$$\begin{cases} \mathrm{i}\omega\boldsymbol{W}^{1,n+1}+\sum\limits_{l\in\{x,y,z\}}G_l\partial_l\boldsymbol{W}^{1,n+1}=0 \quad \text{in } \Omega_1,\\ G_{\boldsymbol{n}_{12}}^-\boldsymbol{W}^{1,n+1}=G_{\boldsymbol{n}_{12}}^-\boldsymbol{W}^{2,n} \quad \text{on } \Gamma_{12},\\ +\text{Boundary conditions on } \partial\Omega_1\cap\partial\Omega,\\ \mathrm{i}\omega\boldsymbol{W}^{2,n+1}+\sum\limits_{l\in\{x,y,z\}}G_l\partial_l\boldsymbol{W}^{2,n+1}=0 \quad \text{in } \Omega_2,\\ G_{\boldsymbol{n}_{21}}^-\boldsymbol{W}^{2,n+1}=G_{\boldsymbol{n}_{21}}^-\boldsymbol{W}^{1,n} \quad \text{on } \Gamma_{21},\\ +\text{Boundary conditions on } \partial\Omega_2\cap\partial\Omega, \end{cases} \tag{3.1}$$

where subscripts denote components, and superscripts denote the subdomain number and the iteration count.

[1] If $T\Lambda T^{-1}$ is the eigenfactorization of $G_{\boldsymbol{n}}$ then $G_{\boldsymbol{n}}^{\pm}=T\Lambda^{\pm}T^{-1}$ where Λ^+ (resp. Λ^-) only gathers the positive (resp. negative) eigenvalues.

This algorithm has been analyzed in [16] and its convergence rate has been computed in the case of an infinite domain $\Omega = \mathbb{R}^3$.

4. Discretization

The subproblems of the Schwarz algorithm (3.1) are discretized using a discontinuous Galerkin formulation. In this section, we first introduce this discretization method in the one-domain case. Then, we state a well-posedness result for a perturbed discrete problem. Finally, we establish the discretization of the interface condition of algorithm (3.1) with respect to the adopted discontinuous Galerkin formulation.

Let Ω_h denote a discretization of the domain Ω into a union of conforming tetrahedral elements $\overline{\Omega}_h = \bigcup_{K\in\mathcal{T}_h} K$. We look for the approximate solution $\boldsymbol{W}_h = \begin{pmatrix} \boldsymbol{E}_h \\ \boldsymbol{H}_h \end{pmatrix}$ of (2.3) in $V_h \times V_h$ where the functional space V_h is defined by:

$$V_h = \{\boldsymbol{U} \in [L^2(\Omega)]^3 / \forall K \in \mathcal{T}_h, \boldsymbol{U}_{|K} \in \mathbb{P}_p(K)\}. \tag{4.1}$$

where $\mathbb{P}_p(K)$ denotes a space of vectors with polynomial components of degree at most p over the element K.

4.1. Discretization of the one-domain problem

Following the presentation of Ern and Guermond [18,19], the discontinuous Galerkin discretization of system (2.3) yields the formulation of the discrete problem:

$$\begin{cases} \text{Find } \boldsymbol{W}_h \text{ in } V_h \times V_h \text{ such that :} \\ \int_{\Omega_h} (\mathrm{i}\omega G_0 \boldsymbol{W}_h)^t \overline{\boldsymbol{V}}\,\mathrm{d}v + \sum_{K\in\mathcal{T}_h} \int_K \left(\sum_{l\in\{x,y,z\}} G_l \partial_l(\boldsymbol{W}_h) \right)^t \overline{\boldsymbol{V}}\,\mathrm{d}v \\ + \sum_{F\in\Gamma^m\cup\Gamma^a} \int_F \left(\tfrac{1}{2}(M_{F,K} - I_{FK} G_{\boldsymbol{n}_F})\boldsymbol{W}_h\right)^t \overline{\boldsymbol{V}}\,\mathrm{d}s \\ - \sum_{F\in\Gamma^0} \int_F (G_{\boldsymbol{n}_F}[\![\boldsymbol{W}_h]\!])^t \{\overline{\boldsymbol{V}}\}\,\mathrm{d}s + \sum_{F\in\Gamma^0} \int_F (S_F[\![\boldsymbol{W}_h]\!])^t [\![\overline{\boldsymbol{V}}]\!]\,\mathrm{d}s \\ = \sum_{F\in\Gamma^a} \int_F \left(\tfrac{1}{2}(M_{F,K} - I_{FK} G_{\boldsymbol{n}_F})\boldsymbol{W}^{\mathrm{inc}}\right)^t \overline{\boldsymbol{V}}\,\mathrm{d}s, \quad \forall \boldsymbol{V} \in V_h \times V_h, \end{cases} \tag{4.2}$$

where Γ^0, Γ^a and Γ^m, respectively denote the set of interior (triangular) faces, the set of faces on Γ_a and the set of faces on Γ_m. The unitary normal associated to the oriented face F is $\boldsymbol{n}_F$ and I_{FK} stands for the incidence matrix between oriented faces and elements whose entries are given by:

$$I_{FK} = \begin{cases} 0 & \text{if the face } F \text{ does not belong to element } K, \\ 1 & \text{if } F \in K \text{ and their orientations match,} \\ -1 & \text{if } F \in K \text{ and their orientations do not match.} \end{cases}$$

We also define respectively the jump and the average of a vector $\boldsymbol{V}$ of $V_h \times V_h$ on a face F shared by two elements K and $\widetilde{K}$:

$$[\![\boldsymbol{V}]\!] = I_{FK}\boldsymbol{V}_{|K} + I_{F\widetilde{K}}\boldsymbol{V}_{|\widetilde{K}} \quad \text{and} \quad \{\boldsymbol{V}\} = \frac{1}{2}\left(\boldsymbol{V}_{|K} + \boldsymbol{V}_{|\widetilde{K}}\right).$$

Finally, the matrix S_F, which is hermitian positive, allows to penalize the jump of a field or of some components of this field on the face F and the matrix $M_{F,K}$, to be defined later, insures the asymptotic consistency with the boundary conditions of the continuous problem.

Problem (4.2) is often interpreted in terms of local problems in each element K of $\mathcal{T}_h$ coupled by the introduction of an element boundary term called numerical flux (see also [18]). In this study, we consider two classical numerical fluxes, which lead to distinct definitions for matrices S_F and $M_{F,K}$:

– *a centered flux* (see [21] for the time-domain equivalent). In this case $S_F = 0$ for all the faces F and, for the boundary faces, we use:

$$M_{F,K} = \begin{cases} I_{FK} \begin{pmatrix} 0_{3\times 3} & N_{\boldsymbol{n}_F} \\ -N^t_{\boldsymbol{n}_F} & 0_{3\times 3} \end{pmatrix} & \text{if } F \in \Gamma^m, \\ |G_{\boldsymbol{n}_F}| \quad \text{if } F \in \Gamma^a. \end{cases} \tag{4.3}$$

– *an upwind flux* (see [38,18]). In this case:

$$S_F = \begin{pmatrix} \alpha^E_F N_{\boldsymbol{n}_F} N^t_{\boldsymbol{n}_F} & 0_{3\times 3} \\ 0_{3\times 3} & \alpha^H_F N^t_{\boldsymbol{n}_F} N_{\boldsymbol{n}_F} \end{pmatrix},$$
$$M_{F,K} = \begin{cases} \begin{pmatrix} \eta_F N_{\boldsymbol{n}_F} N^t_{\boldsymbol{n}_F} & I_{FK} N_{\boldsymbol{n}_F} \\ -I_{FK} N^t_{\boldsymbol{n}_F} & 0_{3\times 3} \end{pmatrix} & \text{if } \in \Gamma^m, \\ |G_{\boldsymbol{n}_F}| \quad \text{if } F \in \Gamma^a \end{cases} \tag{4.4}$$

with α^E_F, α^H_F and η_F equals to 1/2 for homogeneous media.

Remark 1. The formulation of the discontinuous Galerkin scheme above (in particular, the centered and upwind fluxes) actually applies to homogeneous materials. For describing the flux in the inhomogeneous case, let us define:

$$Z^K = \frac{1}{Y^K} = \sqrt{\frac{\mu_r}{\varepsilon_r}}, \quad Z^F = \frac{Z^K + \widetilde{Z^K}}{2} \quad \text{and} \quad Y^F = \frac{Y^K + \widetilde{Y^K}}{2}, \tag{4.5}$$

where $F = \overline{K} \cap \overline{\widetilde{K}}$. With these definitions, the discontinuous Galerkin scheme in the inhomogeneous case can be written formally as (4.2) but by modifying S_F as:

$$S_F = \frac{1}{2} \begin{pmatrix} \frac{1}{Z^F} N_{\boldsymbol{n}_F} N^t_{\boldsymbol{n}_F} & 0_{3\times 3} \\ 0_{3\times 3} & \frac{1}{Y^F} N^t_{\boldsymbol{n}_F} N_{\mathbf{n}_F} \end{pmatrix}, \tag{4.6}$$

and by using for the average, a weighted average $\{\cdot\}_F$ for each face F:

$$\{\mathbf{V}\}_F = \frac{1}{2} \left(\begin{pmatrix} \frac{\widetilde{Z^K}}{Z^F} & 0_{3\times 3} \\ 0_{3\times 3} & \frac{\widetilde{Y^K}}{Y^F} \end{pmatrix} \mathbf{V}_{|K} + \begin{pmatrix} \frac{Z^K}{Z^F} & 0_{3\times 3} \\ 0_{3\times 3} & \frac{Y^K}{Y^F} \end{pmatrix} \mathbf{V}_{|\widetilde{K}} \right). \tag{4.7}$$

In order to simplify the presentation in the following sections, we only retain the formulation adapted to homogeneous materials.

4.2. Comments on the discretization

A few works have considered the discretization of the time-harmonic Maxwell equations by a discontinuous Galerkin formulation combined to the numerical fluxes (4.3) and (4.4). Concerning the convergence properties of such discontinuous Galerkin formulations, the state-of-art is the following:

- when the discontinuous Galerkin method is combined to the upwind flux (4.4), convergence results have been obtained by Helluy and Dayma in [26] for a perturbed problem, *i.e.* replacing $i\omega$ by $i\omega + \nu$ with ν a strictly positive parameter. Their result states that, if the solution is sufficiently regular and if a polynomial approximation of order p is used in each element K, the L^2-norm error of the electromagnetic fields behaves as $h^{p+1/2}$ where h is the mesh parameter.
- The discontinuous Galerkin method combined to the centered flux (4.3) has been studied by Fezoui et al. in [21] for the time-domain Maxwell equations. In this case, the L^2-norm error of the electromagnetic fields behaves as h^p. This result should extend to the time-harmonic case however no convergence proofs are available so far.

The convergence of the discontinuous Galerkin methods considered here is studied numerically in the context of the two-dimensional time-harmonic Maxwell equations discretized on triangular meshes in [15].

Beside, we can study the solvability of the discrete problem in the case of a perturbed problem (we replace $i\omega$ by $i\omega + v$ with $v > 0$). Nonetheless, the numerical evaluation in Section 5 is performed in the unperturbed situation ($v = 0$). We recall here the proof already presented in [15]. In this setting and assuming homogeneous boundary conditions, the problem at hand can be simply written as:

$$\begin{cases} \text{Find } \boldsymbol{W}_h \quad \text{in } V_h \times V_h \text{ such that:} \\ a(\boldsymbol{W}_h, \boldsymbol{V}) + b(\boldsymbol{W}_h, \boldsymbol{V}) = 0, \quad \forall \boldsymbol{V} \in V_h \times V_h, \end{cases} \tag{4.8}$$

with, $\forall \boldsymbol{U}, \boldsymbol{V} \in V_h \times V_h$:

$$\begin{aligned} a(\boldsymbol{U}, \boldsymbol{V}) = & \int_{\Omega_h} ((i\omega + v) G_0 \boldsymbol{U})^t \overline{\boldsymbol{V}} \, dv + \sum_{F \in \Gamma^a} \int_F \left(\frac{1}{2} |G_{\boldsymbol{n}_F}| \boldsymbol{U} \right)^t \overline{\boldsymbol{V}} \, ds + \sum_{F \in \Gamma^m} \int_F \left(\frac{1}{2} M_{F,K} \boldsymbol{U} \right)^t \overline{\boldsymbol{V}} \, ds \\ & + \sum_{F \in \Gamma^0} \int_F (S_F [\![\boldsymbol{U}]\!])^t [\![\overline{\boldsymbol{V}}]\!]_F \, ds, \end{aligned} \tag{4.9}$$

and

$$b(\boldsymbol{U}, \boldsymbol{V}) = \sum_{K \in \mathcal{T}_h} \int_K \left(\sum_{l \in \{x,y,z\}} G_l \partial_l(\boldsymbol{U}) \right)^t \overline{\boldsymbol{V}} \, dv - \sum_{F \in \Gamma^a \cup \Gamma^m} \int_F \left(\frac{1}{2} I_{FK} G_{\boldsymbol{n}_F} \boldsymbol{U} \right)^t \overline{\boldsymbol{V}} ds - \sum_{F \in \Gamma^0} \int_F (G_{\boldsymbol{n}_F} [\![\boldsymbol{U}]\!])^t \{ \overline{\boldsymbol{V}} \} \, ds. \tag{4.10}$$

Then, we have the following result.

Proposition 1. *The solution of problem (4.8) is equal to zero.*

Proof. Let $\Re(G_0)$ and $\Im(G_0)$, respectively denote the real and imaginary parts of G_0. First, considering the fact that the matrices $|G_{\boldsymbol{n}_F}|$, S_F, $\Re(G_0)$ and $-\Im(G_0)$ are hermitian and denoting by $\mathcal{H}(\mathcal{M}_{\mathcal{F},\mathcal{K}})$ the hermitian part of $M_{F,K}$ for F in Γ^m, which is equal to $\begin{pmatrix} \eta_F N_{\boldsymbol{n}_F} N_{\boldsymbol{n}_F}^t & 0_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{pmatrix}$, one has:

$$\begin{aligned} \Re(a(\boldsymbol{W}_h, \boldsymbol{W}_h)) = & \int_{\Omega_h} ((v\Re(G_0) - \omega \Im(G_0)) \boldsymbol{W}_h)^t \overline{\boldsymbol{W}_h} \, dv + \sum_{F \in \Gamma^0} \int_F (S_F [\![\boldsymbol{W}_h]\!])^t [\![\overline{\boldsymbol{W}_h}]\!]_F \, ds \\ & + \sum_{F \in \Gamma^a} \int_F \left(\frac{1}{2} |G_{\boldsymbol{n}_F}| \boldsymbol{W}_h \right)^t \overline{\boldsymbol{W}_h} \, ds + \sum_{F \in \Gamma^m} \int_F \left(\frac{1}{2} \mathcal{H}(\mathcal{M}_{\mathcal{F},\mathcal{K}}) \boldsymbol{W} \right) \overline{\boldsymbol{W}}. \end{aligned} \tag{4.11}$$

Then, we rewrite using the corresponding Green identity an equivalent expression of the sesquilinear form b:

$$\begin{aligned} b(\boldsymbol{U}, \boldsymbol{V}) = & - \sum_{K \in \mathcal{T}_h} \left[\int_K \boldsymbol{U}^t \left(\sum_{l \in \{x,y,z\}} G_l \partial_l(\overline{\boldsymbol{V}}) \right) dv - \sum_{F \in \partial K} \int_F (I_{FK} G_{\boldsymbol{n}_F} \boldsymbol{U}_{|K})^t \overline{\boldsymbol{V}}_{|K} \, ds \right] \\ & - \sum_{F \in \Gamma^a \cup \Gamma^m} \int_F \left(\frac{1}{2} I_{FK} G_{\boldsymbol{n}_F} \boldsymbol{U} \right)^t \overline{\boldsymbol{V}} \, ds - \sum_{F \in \Gamma^0} \int_F (G_{\boldsymbol{n}_F} [\![\boldsymbol{U}]\!])^t \{ \overline{\boldsymbol{V}} \} \, ds, \quad \forall \boldsymbol{U}, \boldsymbol{V} \in V_h \times V_h. \end{aligned} \tag{4.12}$$

By noticing that on a face $F \in \Gamma^0$ separating two elements K and $\widetilde{K}$:

$$(G_{\boldsymbol{n}_F} \{ \boldsymbol{U} \})^t [\![\boldsymbol{V}]\!] + (G_{\boldsymbol{n}_F} [\![\boldsymbol{U}]\!])^t \{ \boldsymbol{V} \} = (I_{FK} G_{\boldsymbol{n}_F} \boldsymbol{U}_{|K})^t \boldsymbol{V}_{|K} + (I_{F\widetilde{K}} G_{\boldsymbol{n}_F} \boldsymbol{U}_{|\widetilde{K}})^t \boldsymbol{V}_{|\widetilde{K}},$$

which is in part due to the fact that $G_{\boldsymbol{n}_F}$ is hermitian, one deduces:

$$b(\boldsymbol{U},\boldsymbol{V}) = -\sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{U}^t\left(\sum_{l\in\{x,y,z\}} G_l\partial_l(\overline{\boldsymbol{V}})\right)\mathrm{d}v + \sum_{F\in\Gamma^a\cup\Gamma^m}\int_F\left(\frac{1}{2}I_{FK}G_{\boldsymbol{n}_F}\boldsymbol{U}\right)^t\overline{\boldsymbol{V}}\,\mathrm{d}s + \sum_{F\in\Gamma^0}\int_F (G_{\boldsymbol{n}_F}\{\boldsymbol{U}\})^t[\![\overline{\boldsymbol{V}}]\!]ds,\ \forall \boldsymbol{U},\boldsymbol{V}\in V_h\times V_h. \tag{4.13}$$

Thus, it is now straightforward to see that b is anti-hermitian and consequently:

$$\Re(a(\boldsymbol{W}_h,\boldsymbol{W}_h)+b(\boldsymbol{W}_h,\boldsymbol{W}_h)) = \int_{\Omega_h}((v\Re(G_0)-\omega\Im(G_0))\boldsymbol{W}_h)^t\overline{\boldsymbol{W}_h}\,\mathrm{d}v + \sum_{F\in\Gamma^0}\int_F (S_F[\![\boldsymbol{W}_h]\!])^t[\![\overline{\boldsymbol{W}_h}]\!]_F\,\mathrm{d}s + \sum_{F\in\Gamma^a}\int_F\left(\frac{1}{2}|G_{\boldsymbol{n}_F}|\boldsymbol{W}_h\right)^t\overline{\boldsymbol{W}_h}\,\mathrm{d}s + \sum_{F\in\Gamma^m}\int_F\left(\frac{1}{2}\mathcal{H}(\mathcal{M}_{\mathcal{F},\mathcal{K}})\boldsymbol{W}\right)\overline{\boldsymbol{W}},$$

From (4.8), $\Re(a(\boldsymbol{W}_h,\boldsymbol{W}_h)+b(\boldsymbol{W}_h,\boldsymbol{W}_h))$ is also equal to zero. As $v\Re(G_0)-\omega\Im(G_0)$ is positive definite and $|G_{\boldsymbol{n}_F}|$, S_F and $\mathcal{H}(\mathcal{M}_{\mathcal{F},\mathcal{K}})$ are positive, the vector field $\boldsymbol{W}_h$ is zero. □

4.3. Discretization of the domain decomposition algorithm

4.3.1. Discontinuous Galerkin formulation of the multi-domain problem

Let us now assume that the domain Ω is decomposed into N_s subdomains $\Omega=\bigcup_{i=1}^{N_s}\Omega_i$. A superscript i indicates that some notations are relative to the subdomain Ω_i and not to the whole domain Ω. Thus, we will refer to $\mathcal{T}_h^i$ and V_h^i with obvious definitions from those of $\mathcal{T}_h$ and V_h and we also define $\Gamma_m^i=\Gamma_m\cap\partial\Omega_i$, $\Gamma_a^i=\Gamma_a\cap\partial\Omega_i$ and $\Gamma_0^i=\Gamma_0\cap\partial\Omega_i$ with their corresponding sets of faces $\Gamma^{m,i}$, $\Gamma^{a,i}$ and $\Gamma^{0,i}$. Finally Γ^{ij} will denote the set of faces which belongs to $\Gamma_{ij}=\partial\Omega_i\cap\Omega_j$.

According to algorithm (3.1), the interface condition on Γ_{ij} writes as:

$$G_{\boldsymbol{n}_F}^-(\boldsymbol{W}_h^{i,n+1}-\boldsymbol{W}_h^{j,n})=0 \quad \text{for all} \quad F \text{ belonging to } \Gamma^{ij},$$

where $\boldsymbol{W}_h^{i,n+1}$ denotes the approximation of $\boldsymbol{W}^{i,n+1}$ for $i=1,2$. Thus, the discontinuous Galerkin discretization of a local problem of algorithm (3.1) can be written using (4.2), as the solution of the following problem:

$$\begin{cases} \text{Find } \boldsymbol{W}_h^{i,n+1} \quad \text{in } V_h^i\times V_h^i \text{ such that:} \\ \int_{\Omega_h^i}(\mathrm{i}\omega G_0\boldsymbol{W}_h^{i,n+1})^t\overline{\boldsymbol{V}}\,\mathrm{d}v + \sum_{K\in\mathcal{T}_h^i}\int_K\left(\sum_{l\in\{x,y,z\}} G_l\partial_l(\boldsymbol{W}_h^{i,n+1})\right)^t\overline{\boldsymbol{V}}\,\mathrm{d}v \\ + \sum_{F\in\Gamma^{m,i}}\int_F\left(\tfrac{1}{2}(M_{F,K}-I_{FK}G_{\boldsymbol{n}_F})\boldsymbol{W}_h^{i,n+1}\right)^t\overline{\boldsymbol{V}}\,\mathrm{d}s \\ + \sum_{F\in(\Gamma^{a,i}\cup\Gamma^{ij})}\int_F\left(I_{FK}G_{\boldsymbol{n}_F}^-\boldsymbol{W}_h^{i,n+1}\right)^t\overline{\boldsymbol{V}}\,\mathrm{d}s \\ - \sum_{F\in\Gamma^{0,i}}\int_F\left(G_{\boldsymbol{n}_F}[\![\boldsymbol{W}_h^{i,n+1}]\!]\right)^t\{\overline{\boldsymbol{V}}\}\,\mathrm{d}s \\ + \sum_{F\in\Gamma^{0,i}}\int_F\left(S_F[\![\boldsymbol{W}_h^{i,n+1}]\!]\right)^t[\![\overline{\boldsymbol{V}}]\!]\,\mathrm{d}s \\ = \sum_{F\in\Gamma^{a,i}}\int_F\left(I_{FK}G_{\boldsymbol{n}_F}^-\boldsymbol{W}^{\mathrm{inc}}\right)^t\overline{\boldsymbol{V}}\,\mathrm{d}s \\ + \sum_{F\in\Gamma^{ij}}\int_F(I_{FK}G_{\boldsymbol{n}_F}^-\boldsymbol{W}^{j,n})^t\overline{\boldsymbol{V}}\,\mathrm{d}s, \quad \forall\boldsymbol{V}\in V_h^i\times V_h^i. \end{cases} \tag{4.14}$$

4.3.2. Formulation of an interface system

In the two-domain case the Schwarz algorithm can be written formally as follows:

$$\begin{cases} \mathcal{L}\boldsymbol{W}^{1,n+1} = \boldsymbol{f}^1, \quad \text{in } \Omega_1, \\ \mathcal{B}_1(\boldsymbol{W}^{1,n+1}) = \Lambda^{1,n}, \quad \text{on } \Gamma_{12}, \\ +\text{Boundary conditions on } \partial\Omega_1 \cap \partial\Omega, \end{cases} \quad \begin{cases} \mathcal{L}\boldsymbol{W}^{2,n+1} = \boldsymbol{f}^2 \quad \text{in } \Omega_2, \\ \mathcal{B}_2(\boldsymbol{W}^{2,n+1}) = \Lambda^{2,n} \quad \text{on } \Gamma_{21}, \\ +\text{Boundary conditions on } \partial\Omega_2 \cap \partial\Omega, \end{cases} \tag{4.15}$$

and then:

$$\begin{cases} \Lambda^{1,n+1} = \mathcal{B}_1(\boldsymbol{W}^{2,n+1}) \quad \text{on } \Gamma_{12}, \\ \Lambda^{2,n+1} = \mathcal{B}_2(\boldsymbol{W}^{1,n+1}) \quad \text{on } \Gamma_{21}, \end{cases} \tag{4.16}$$

where $\mathcal{L}$ is a linear differential operator, $\boldsymbol{f}^{1,2}$ denotes right hand sides associated to $\Omega_{1,2}$ and, $\mathcal{B}_1$ and $\mathcal{B}_2$ are the interface operators. The Schwarz algorithm (4.15) and (4.16) can be rewritten as:

$$\begin{cases} \Lambda^{1,n+1} = \mathcal{B}_1(\boldsymbol{W}^2(\Lambda^{2,n}, \boldsymbol{f}^2)), \\ \Lambda^{2,n+1} = \mathcal{B}_2(\boldsymbol{W}^1(\Lambda^{1,n}, \boldsymbol{f}^1)), \end{cases}$$

where $\boldsymbol{W}^j = \boldsymbol{W}^j(\Lambda^j, \boldsymbol{f}^j)$ are the solution of the local problems. By linearity of the operators involved, an iteration of the Schwarz algorithm is equivalent to:

$$\boldsymbol{\lambda}^{n+1} = (\mathrm{Id} - \mathcal{T})\boldsymbol{\lambda}^n + \boldsymbol{d},$$

which is a fixed point iteration to solve the interface system:

$$\mathcal{T}\boldsymbol{\lambda} = \boldsymbol{d}, \tag{4.17}$$

where $\boldsymbol{\lambda} = (\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2)$. From the discrete point of view, the global problem on domain Ω can be written in the matrix form:

$$\begin{pmatrix} A_1 & 0 & R_1 & 0 \\ 0 & A_2 & 0 & R_2 \\ 0 & -B_2 & \mathrm{I} & 0 \\ -B_1 & 0 & 0 & \mathrm{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{W}_h^1 \\ \boldsymbol{W}_h^2 \\ \boldsymbol{\lambda}_h^1 \\ \boldsymbol{\lambda}_h^2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_h^1 \\ \boldsymbol{f}_h^2 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix},$$

where $A_{1,2}$ are local matrices coupling only internal unknowns, $R_{1,2}$ express the coupling between internal unknowns and interface unknowns and the subscript h denotes the discrete counterpart of a given quantity (*e.g.* $\boldsymbol{\lambda}_h^{1,2}$ are the discretized unknown vectors corresponding to $\boldsymbol{\lambda}^{1,2}$). The elimination of the internal unknowns $\boldsymbol{W}_h^{1,2}$ leads to the discrete counterpart of the interface problem (4.17), $\mathcal{T}_h\boldsymbol{\lambda}_h = \boldsymbol{d}_h$, with:

$$\mathcal{T}_h = \begin{pmatrix} \mathrm{I} & B_2A_2^{-1}R_2 \\ B_1A_1^{-1}R_1 & \mathrm{I} \end{pmatrix} \quad \text{and} \quad \boldsymbol{g}_h = \begin{pmatrix} B_2A_2^{-1}\boldsymbol{f}_h^2 \\ B_1A_1^{-1}\boldsymbol{f}_h^1 \end{pmatrix},$$

where $\mathcal{T}_h$ and $\boldsymbol{g}_h$ are the discretization of $\mathcal{T}$ and $\boldsymbol{d}$. This system is further solved by a Krylov subspace method as discussed in the following section.

5. Numerical and performance results

5.1. Implemented formulations and experimental testbed

For this study, the implementation of the discontinuous Galerkin formulations described in section (4.1) has been limited to a $\mathbb{P}_0$ approximation with the centered flux (4.3) (which is equivalent to a finite volume method which will be referred as DG-$\mathbb{P}_0$-c in the sequel) and a $\mathbb{P}_1$ approximation (*i.e.* a linear discontinuous Galerkin method) with either the centered flux (4.3) or the upwind flux (4.4) and nodal polynomial basis functions (respectively referred as DG-$\mathbb{P}_1$-c and DG-$\mathbb{P}_1$-u in the sequel).

For what concerns the implementation of the domain decomposition solver, the geometric partitioning of the underlying tetrahedral mesh is a (non-overlapping) element-wise partitioning, the separator between neighboring submeshes being a triangulated surface (*i.e.* a list of triangular faces). Then, for each face on a given artificial interface Γ_{ij} between neighboring submeshes Ω_i and Ω_j, the physical fields involved in the interface conditions of (4.15) and (4.16) are those defined in the two tetrahedra attached to the face.

Unless otherwise indicated, computations have been performed in 64 bit arithmetic. The experimental testbed is a cluster of AMD Opteron 2 GHz dual nodes with 2 GB of RAM memory, interconnected by a Gigabit Ethernet switch. The computer codes for the DG-$\mathbb{P}_0$ and DG-$\mathbb{P}_1$ methods have been programmed in Fortran and the parallelization relies on the MPI (Message Passing Interface). The implementation of the domain decomposition solver requires a partitioning of the underlying tetrahedral mesh which is obtained using the MeTiS graph partitioning tool [31].

5.2. Solution strategies

An unpreconditioned BiCGstab(ℓ) Krylov subspace method [41] is used for the solution of the interface system (4.17). After different tests for assessing the convergence of the method and the associated computation time, the parameter ℓ has been set to 6. This method is adapted to linear systems involving non-symmetric matrices with complex spectrum. The convergence of the iterative solution of the interface system is evaluated in terms of the euclidian norm of the residual normalized to the norm of the right-hand side vector. The corresponding linear threshold has been set to $\varepsilon_i = 10^{-6}$. Each iteration of this Krylov subspace method requires a certain number of matrix–vector products with the interface matrix of system (4.17). Within the domain decomposition framework of algorithm (3.1), such a matrix–vector product translates into the solution of the subdomain discrete problems (4.14). For this purpose, several strategies have been considered:

- a preconditioned restarted GMRES(m) [40] (with $m = 10$) or a preconditioned BiCGstab(ℓ) (with $\ell = 1$) method where the preconditioner is taken to be a LU factorization computed and stored in single precision arithmetic using the MUMPS multifrontal sparse direct solver [2], while the Krylov subspace method works on double precision arithmetic vectors. In both cases, the linear threshold has been set to $\varepsilon_i = 10^{-6}$. These solution strategies will be referred respectively as `DD-gmres` and `DD-bicgl`.
- a LU factorization where the L and U factors are computed and stored in single precision (32 bits) arithmetic and an iterative refinement procedure is applied to recover double precision arithmetic (64 bits). More precisely, assuming that the linear system is $Ax = b$, the iterative refinement procedure is as follows:

  $x \leftarrow 0$
  REPEAT
  $r \leftarrow b - Ax$ %residual evaluation step.
  Solve $Ly = r$
  Solve $Uz = y$
  $x \leftarrow x + z$ %updating step.
  UNTIL$\|r\| < \varepsilon_l$

  where the triangular solves $Ly = r$ and $Uz = y$ are performed using single precision arithmetic while the residual evaluation and updating step are computed in double precision arithmetic. In practice, we set $\varepsilon_l = 10^{-10}$ and a maximum of five iterations of the above procedure. In the sequel, this solution strategy will be referred as `DD-itref`.

These strategies have been selected with the aim to reduce the memory requirements for storing the L and U factors and thus allowing to tackle large problems. We note that such mixed-precision strategies have recently been considered in the linear algebra community essentially for performance issues [32,33] on modern high-performance processors. In these works, the mixing of single and double precision computations is performed in the context of an iterative refinement procedure. Here, the single precision L and U factors yield a very

accurate preconditioner and consequently, a few iterations of the preconditioned Krylov subspace methods are sufficient for solving the subdomain problems. In practice we use one iteration of BiCGstab and two iterations of GMRES.

In the following tables and figures:

- $L_{\min}$, $L_{\max}$ and L_{avg}, respectively denote the minimum, maximum and average length of an edge in a given tetrahedral mesh,
- N_s is the number of subdomains which is also the number of processes involved in a parallel simulation,
- 'CPU' is the CPU time which is evaluated on each process of a parallel simulation and, for this reason, we give both the minimum and maximum values of this quantity,
- 'REAL' is the real (or elapsed) time of a parallel simulation,
- 'RAM' is the memory requirement for storing the L and U factors which is evaluated on each process of a parallel simulation and, as for the 'CPU' quantity, we give both the minimum and maximum values of this quantity.

5.3. Diffraction of a plane wave by a PEC sphere

The first test problem that we consider is the diffraction of a plane wave by a perfectly conducting sphere with radius $R = 1$ m centered at the origin. The artificial boundary on which the first-order absorbing condition (2.2) applies is defined by a sphere with radius $R_a = 1.5$ m centered at the origin. The medium is considered homogeneous with ε_r and μ_r equal to one. The frequency of the incident plane wave is $F = 600$ MHz and $\omega = 2\pi F/F_0$ with $F_0 = 300$ MHz. Its polarization is such that:

$$\mathsf{k} = (0, 0, -k_z)^t, \quad \mathsf{E} = (E_x, 0, 0)^t \quad \text{and} \quad \mathsf{H} = (0, H_y, 0)^t.$$

Four tetrahedral meshes of increased resolution have been used and their characteristics are summarized in Table 1. Views of the triangulations in the plane $Z = 0.0$ m are given in Fig. 1. Note that the mesh with the finest resolution is such that $L_{\text{avg}} = \Lambda/11$ while this ratio is equal to 6 for the mesh with the coarsest resolution.

Numerical solutions are shown in Figs. 2 and 3 in the form of the contour lines in the plane $Z = 0.0$ m of the E_x and E_y components. Fig. 2(a) and (b) correspond to the analytical solution for this problem, expressed using Debye potentials [17]. Clearly, the solutions obtained on mesh M4 using the DG-$\mathbb{P}_0$-c method, and on mesh M1 using the DG-$\mathbb{P}_1$-c/DG-$\mathbb{P}_1$-u methods are in very good agreement with the reference result. Timing measures are given in Tables 2 (solution phase) and 3 (factorization phase).

5.4. Diffraction of a plane wave by a PEC cube

The test problem considered here consists in the diffraction of a plane wave by a perfectly conducting cube of side length $C = 1/3$ m centered at the origin. The artificial boundary on which the first-order absorbing condition (2.2) applies is defined by a unitary cube centered at the origin. The frequency of the incident plane wave is $F = 900$ MHz and its polarization is such that:

$$\mathsf{k} = (k_x, 0, 0)^t, \quad \mathsf{E} = (0, E_y, 0)^t \quad \text{and} \quad \mathsf{H} = (0, 0, H_z)^t.$$

Table 1
Diffraction of a plane wave by a PEC sphere, $F = 600$ MHz

Mesh	# Vertices	# Tetrahedra	$L_{\min}$ (m)	$L_{\max}$ (m)	L_{avg} (m)
M1	32,418	172,800	0.051990	0.152832	0.086657
M2	70,422	384,000	0.039267	0.118029	0.066279
M3	151,452	843,648	0.030206	0.091805	0.051038
M4	244,834	1,382,400	0.025665	0.078819	0.043431

Characteristics of the tetrahedral meshes ('#' refers to the number of).

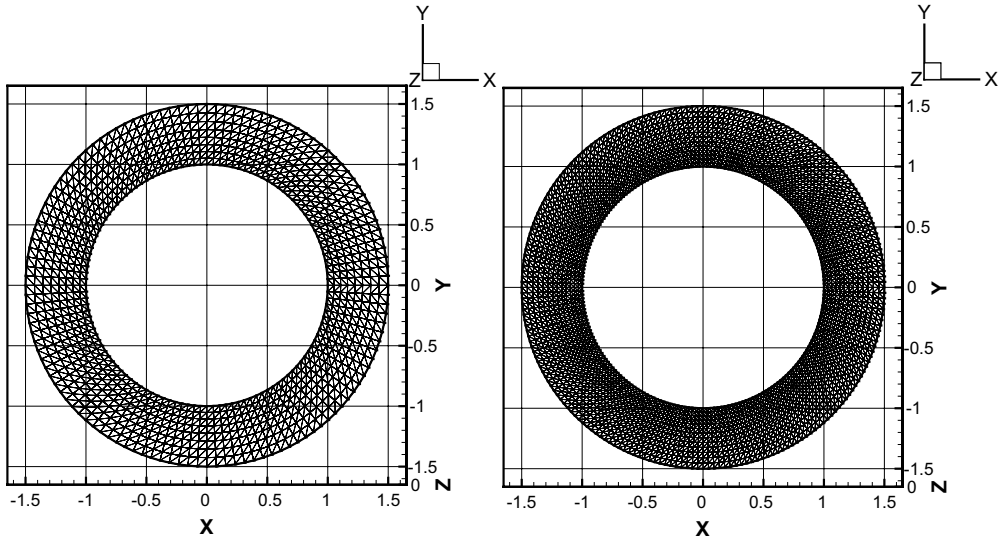


Fig. 1. Diffraction of a plane wave by a PEC sphere, $F = 600$ MHz. Triangulation in the plane $Z = 0.0$ m.

Five tetrahedral meshes have been used whose characteristics are summarized in Table 4 (see also Fig. 4(a)). The exterior domain is always the vacuum but two situations have been considered for the cube: either it is strictly a perfect conductor or it is coated by a dielectric material with $\varepsilon_r = 4.0$ (see also Fig. 4(b) for a view of the corresponding zone).

5.4.1. Propagation in vacuum

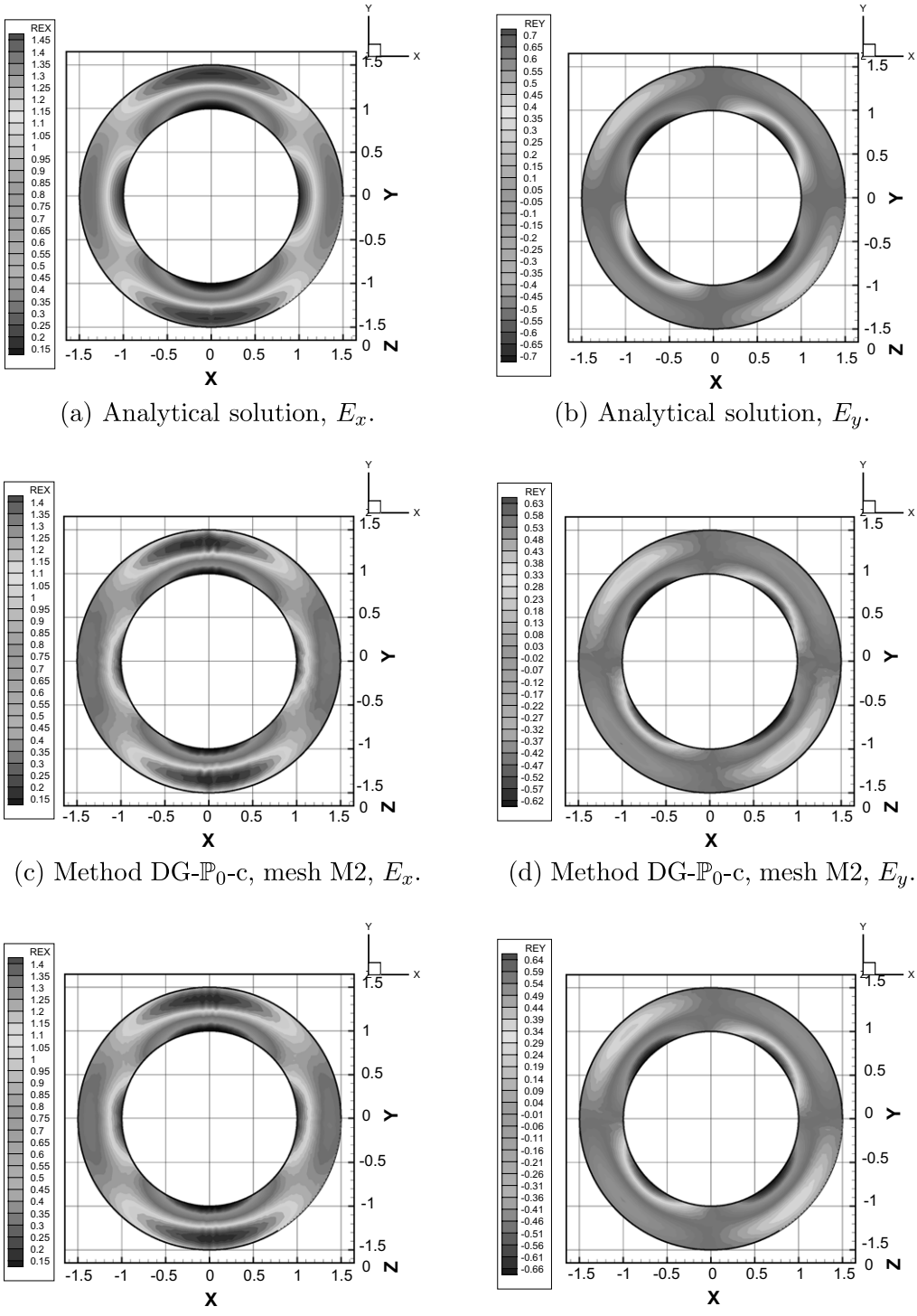
Numerical solutions are shown in Figs. 5 and 6 in the form of the contour lines in the plane $Z = 0.5$ m of the E_x and E_y components. One can note that the solution resulting from the DG-$\mathbb{P}_1$-c method applied on mesh M1 is very similar to the one obtained using the DG-$\mathbb{P}_0$-c method with mesh M4. Moreover, the former solution exhibits a better symmetry with regards to the distribution of the E_y component. Timing measures are given in Tables 5 (solution phase) and 6 (factorization phase).

5.4.2. Coated PEC cube

Numerical solutions are shown in Fig. 7 in the form of the contour lines in the plane $Z = 0.5$ m of the E_x and E_y components. This time, we only report on results obtained using the DG-$\mathbb{P}_0$-c method applied to mesh M5 and the DG-$\mathbb{P}_1$-c/DG-$\mathbb{P}_1$-u methods applied to mesh M2. One can note that the solution resulting from the DG-$\mathbb{P}_0$-c method does not exhibit all the peculiarities of the underlying wave propagation problem. This is made particularly clear on the distributions of the E_x component which suggest that the DG-$\mathbb{P}_0$-c might require a discretization mesh with an increased resolution. Timing measures are given in Tables 7 (solution phase) and 8 (factorization phase).

5.5. Discussion of the numerical and parallel performances

A first noticeable behaviour that can be emphasized is that the convergence of the proposed domain decomposition solver for a given approximation method is weakly dependent on the number of system unknowns and the granularity of the decomposition (*i.e.* the number of subdomains). For instance, in the case of the diffraction of a plane wave by a PEC sphere, when switching from mesh M2 to mesh M4 (see Table 1), the number of unknowns of the algebraic system associated to the DG-$\mathbb{P}_0$-c approximation method increases from $6 \times 384{,}000 = 2{,}304{,}000$ to $6 \times 1{,}382{,}400 = 8{,}294{,}400$ while the number of iter-



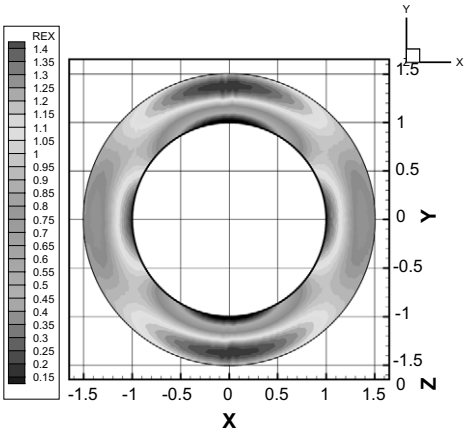
(a) Analytical solution, E_x. (b) Analytical solution, E_y.

(c) Method DG-$\mathbb{P}_0$-c, mesh M2, E_x. (d) Method DG-$\mathbb{P}_0$-c, mesh M2, E_y.

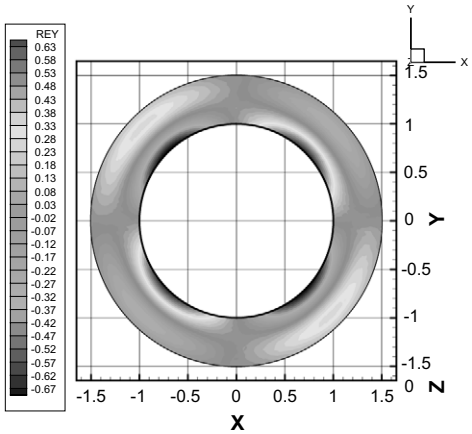
(e) Method DG-$\mathbb{P}_0$-c, mesh M3, E_x. (f) Method DG-$\mathbb{P}_0$-c, mesh M3, E_y.

Fig. 2. Diffraction of a plane wave by a PEC sphere, $F = 600$ MHz. Contour lines of E_x and E_y in the plane $Z = 0.0$ m.

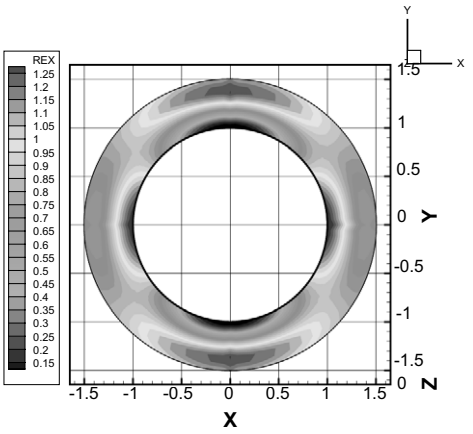
ations of the domain decomposition solver ranges from 8 to 10 for a number of subdomains N_s in the set $\{16, 32, 64\}$. Similarly, for the diffraction of a plane wave by a PEC cube, when switching from mesh M2 to mesh M4 (see Table 1), the number of unknowns of the algebraic system associated to the



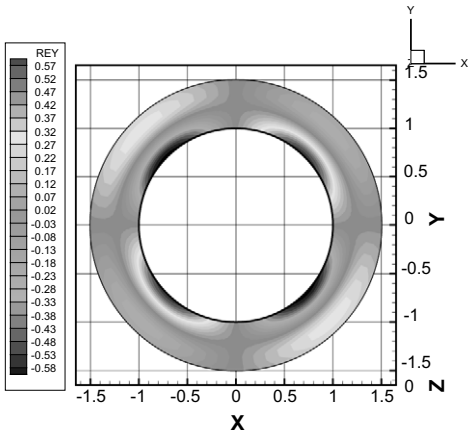
(a) Method DG-$\mathbb{P}_0$-c, mesh M4, E_x.



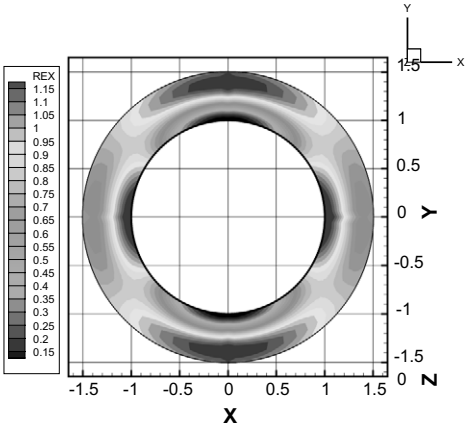
(b) Method DG-$\mathbb{P}_0$-c, mesh M4, E_y.



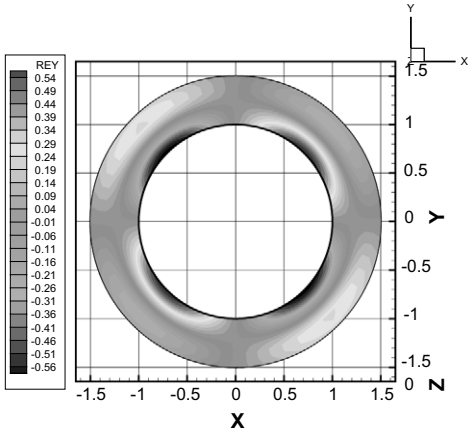
(c) Method DG-$\mathbb{P}_1$-c, mesh M1, E_x.



(d) Method DG-$\mathbb{P}_1$-c, mesh M1, E_y.



(e) Method DG-$\mathbb{P}_1$-u, mesh M1, E_x.



(f) Method DG-$\mathbb{P}_1$-u, mesh M1, E_y.

Fig. 3. Diffraction of a plane wave by a PEC sphere, $F = 600$ MHz. Contour lines of E_x and E_y in the plane $Z = 0.0$ m.

Table 2
Diffraction of a plane wave by a PEC sphere, $F = 600$ MHz

Mesh	Method	Strategy	N_s	# it	CPU (min/max)	REAL
M1	DG-$\mathbb{P}_1$-c	DD-bicgl	32	10	441 s/772 s	929 s
–	–	DD-gmres	–	10	227 s/271 s	442 s
–	–	DD-itref	–	9	197 s/300 s	480 s
M1	DG-$\mathbb{P}_1$-u	DD-bicgl	32	10	544 s/616 s	842 s
–	–	DD-gmres	–	9	259 s/284 s	464 s
–	–	DD-itref	–	9	170 s/200 s	344 s
M2	DG-$\mathbb{P}_0$-c	DD-bicgl	16	8	215 s/379 s	390 s
–	–	–	32	9	98 s/132 s	143 s
–	–	DD-gmres	16	8	110 s/139 s	143 s
–	–	–	32	9	46 s/58 s	68 s
–	–	DD-itref	16	8	215 s/379 s	390 s
–	–	–	32	9	101 s/159 s	172 s
M3	DG-$\mathbb{P}_0$-c	DD-bicgl	32	8	244 s/352 s	456 s
–	–	–	64	9	116 s/178 s	184 s
–	–	DD-gmres	32	8	121 s/164 s	249 s
–	–	–	64	9	56 s/87 s	98 s
–	–	DD-itref	32	8	116 s/197 s	256 s
–	–	–	64	9	53 s/98 s	111 s
M4	DG-$\mathbb{P}_0$-c	DD-bicgl	64	9	197 s/432 s	460 s
–	–	DD-gmres	–	10	109 s/173 s	211 s
–	–	DD-itref	–	9	101 s/193 s	233 s

Computation times (solution phase).

Table 3
Diffraction of a plane wave in vacuum by a PEC sphere, $F = 600$ MHz

Mesh	Method	N_s	CPU (min/max)	RAM (min/max)
M1	DG-$\mathbb{P}_1$-c	32	198 s/301 s	1217 MB/1457 MB
M1	DG-$\mathbb{P}_1$-u	32	211 s/329 s	1257 MB/1512 MB
M2	DG-$\mathbb{P}_0$-c	8	220 s/359 s	1365 MB/1679 MB
–	–	16	56 s/121 s	492 MB/733 MB
–	–	32	11 s/26 s	156 MB/249 MB
M3	–	32	69 s/185 s	586 MB/959 MB
–	–	64	17 s/52 s	210 MB/370 MB
M4	–	64	43 s/135 s	425 MB/737 MB

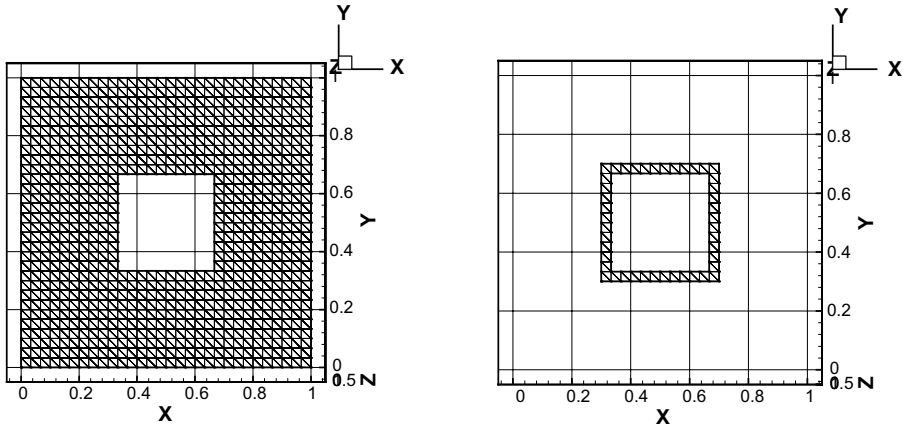
Computation times and memory requirement for storing the L and U factors.

Table 4
Diffraction of a plane wave by a PEC cube, $F = 900$ MHz

Mesh	# Vertices	# Tetrahedra	$L_{\min}$ (m)	$L_{\max}$ (m)	L_{avg} (m)
M1	9,136	46,704	0.05000	0.08660	0.06343
M2	29,062	156,000	0.03333	0.05773	0.04242
M3	67,590	373,632	0.02500	0.04330	0.03187
M4	129,276	725,424	0.02000	0.03464	0.02552
M5	220,122	1,248,000	0.01666	0.02886	0.02128

Characteristics of the tetrahedral meshes.

DG-$\mathbb{P}_0$-c approximation method increases from $6 \times 156{,}000 = 936{,}000$ to $6 \times 725{,}424 = 4{,}352{,}544$ while the number of iterations of the domain decomposition solver ranges from 6 to 9 for a number of subdomains N_s in the set $\{16, 32, 64\}$.



(a) Mesh example of the domain. (b) Cube coated by a dielectric material.

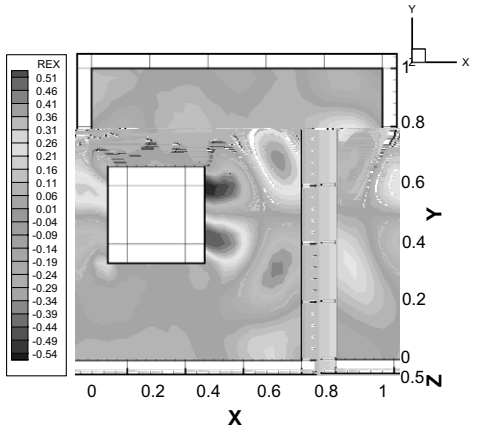
Fig. 4. Diffraction of a plane wave by a PEC cube, $F = 900$ MHz. Mesh M2: triangulation in the plane $Z = 0.5$ m.

Moreover, for a given mesh, the number of iterations increases when switching between the DG-$\mathbb{P}_0$ and DG-$\mathbb{P}_1$ approximation methods. This is mainly related to the fact that at the same time the number of system unknowns increases noticeably. For instance, when simulating the diffraction of a plane wave by a PEC cube using mesh M2 (see Table 4), the number of system unknowns is equal to $6 \times 156{,}000 = 936{,}000$ and $24 \times 156{,}000 = 3{,}744{,}000$, respectively for the DG-$\mathbb{P}_0$ and DG-$\mathbb{P}_1$ approximation methods. In the former case, the number of iterations is equal to 6 for $N_s = 16$ while in the latter case, it is equal to 9 (respectively 10) for $N_s = 32$ (respectively 64).

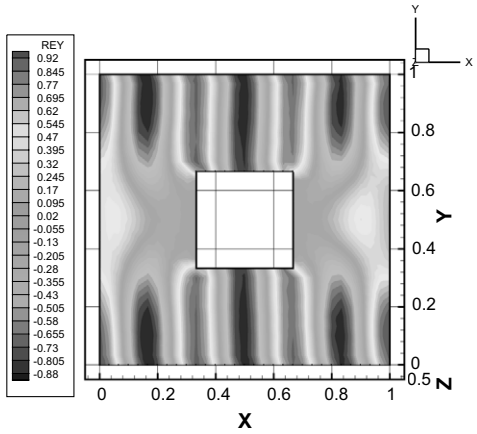
It is also worthwhile to note that:

- the convergence of the domain decomposition solver when combined to the DG-$\mathbb{P}_1$ approximation method seems insensitive to the type of scheme (*i.e.* centered or upwind) used for the evaluation of the numerical flux through internal faces.
- As expected, when the propagation media is more complex than a simple uniform (*i.e.* homogeneous) material, the convergence of the domain decomposition solver requires more iterations. Indeed, comparing the performances of the domain decomposition solver combined to the DG-$\mathbb{P}_1$ approximation method for the simply PEC and coated PEC cubes and using mesh M2, the number of iterations increases from 9 to 14. Note that for mesh M5 and the DG-$\mathbb{P}_0$-c approximation method, the number of system unknowns is equal to $6 \times 1{,}248{,}000 = 7{,}488{,}000$ which is exactly twice the number obtained for the DG-$\mathbb{P}_1$ approximation method using mesh M2. Despite this large increase of the number of system unknowns and the difference in the number of subdomains (respectively $N_s = 32$ and $N_s = 64$), the solution of the underlying algebraic systems requires essentially the same number of iterations.

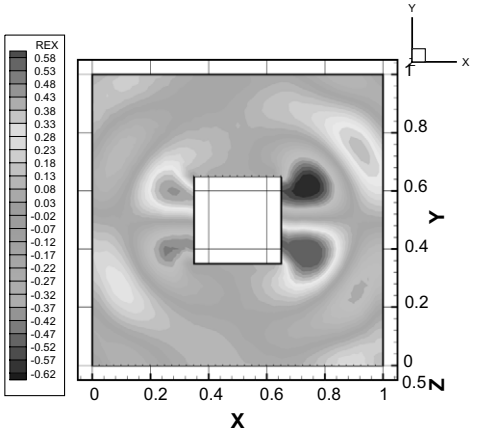
We evaluate the parallel performances of the proposed domain decomposition solver using two metrics: the ratio of the maximum of the per process CPU times to the REAL time which is referred as '%CPU' in the sequel and, the relative parallel speedup $S_{N_{s1}}^{N_{s2}}$ evaluated as the ratio of the elapsed time for N_{s1} subdomains to the elapsed time for N_{s2} subdomains. We mainly discuss the performance results of Table 2 but similar conclusions can be drawn based on the solution times given in Tables 5 and 7. First, we remark that, in the case of the diffraction of a plane wave by a PEC sphere, %CPU ranges from 58% to 97%. The lowest values of this metric are essentially obtained for the computations conducted with mesh M1 (see Table 1) which is due to the fact that the underlying mesh is too coarse and the communication cost induced by the BiCGstab(ℓ) method, which is applied to the solution of the interface system (4.17), prevails. Apart from this particular situation, one can note that super-linear parallel speedups are often obtained, For instance, for the same test case and computations based on mesh M2 (respectively M3), the



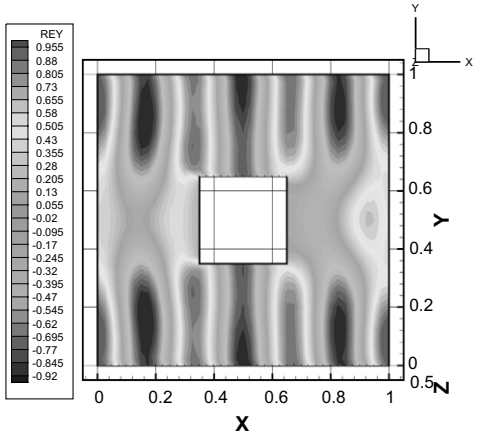
(a) Method DG-$\mathbb{P}_0$-c, mesh M2, E_x.



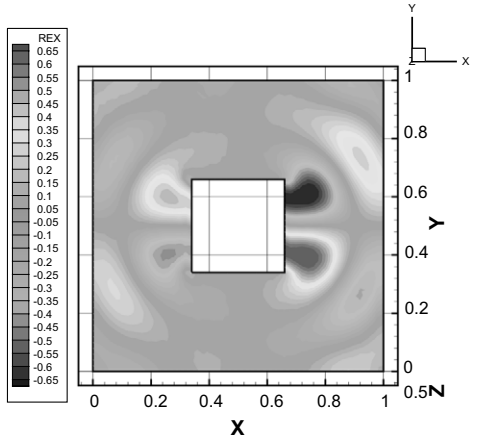
(b) Method DG-$\mathbb{P}_0$-c, mesh M2, E_y.



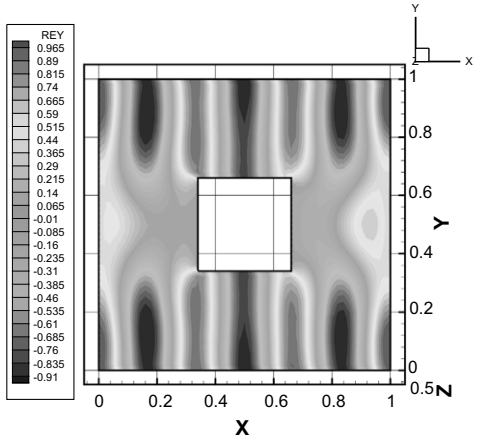
(c) Method DG-$\mathbb{P}_0$-c, mesh M3, E_x.



(d) Method DG-$\mathbb{P}_0$-c, mesh M3, E_y.



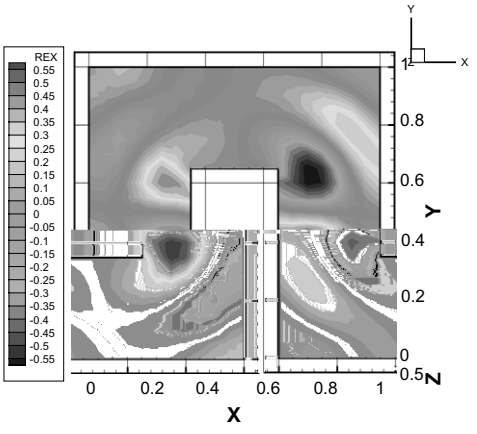
(e) Method DG-$\mathbb{P}_0$-c, mesh M4, E_x.



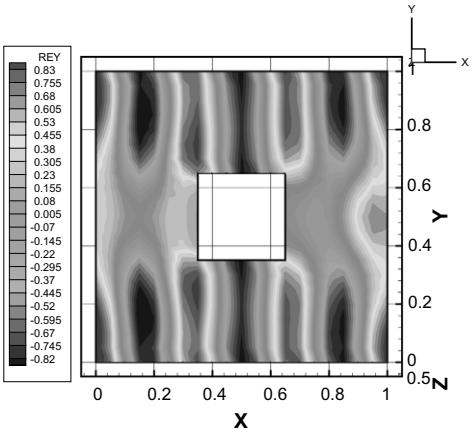
(f) Method DG-$\mathbb{P}_0$-c, mesh M4, E_y.

Fig. 5. Diffraction of a plane wave in vacuum by a PEC cube, $F = 900$ MHz. Contour lines of E_x and E_y in the plane $Z = 0.5$ m.

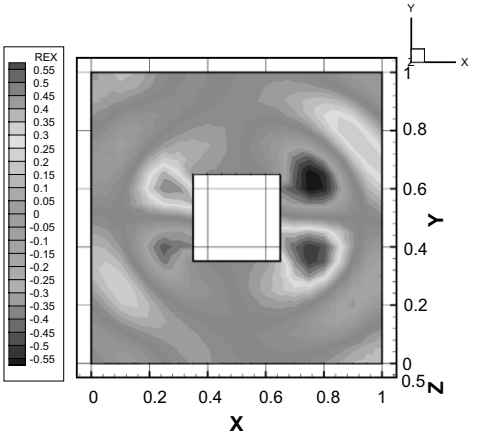
$S_{16}^{32} \in [2.1, 2.7]$ (respectively $S_{32}^{64} \in [2.3, 2.5]$). This behaviour essentially stems from the super-linear reduction of the cost of the local solves when increasing the number or subdomains for a constant global problem size.



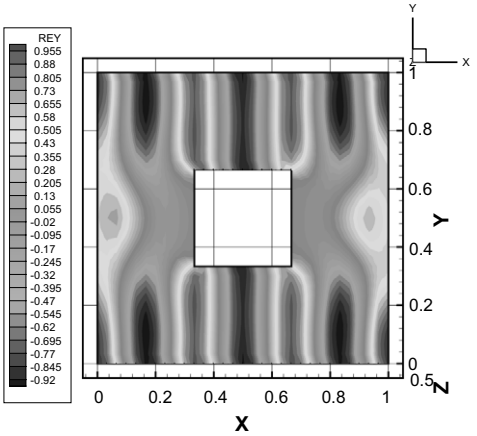
(a) Method DG-$\mathbb{P}_1$-c, mesh M1, E_x.



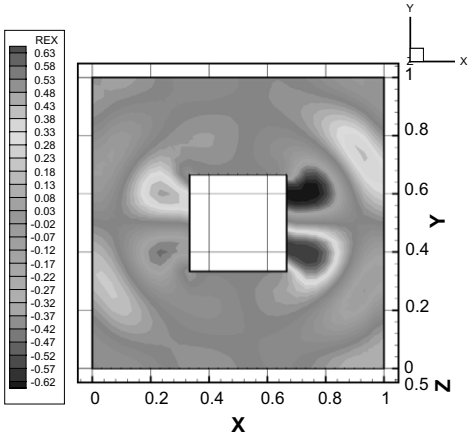
(b) Method DG-$\mathbb{P}_1$-c, mesh M1, E_y.



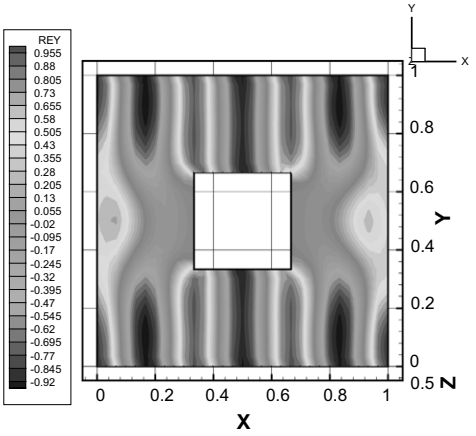
(c) Method DG-$\mathbb{P}_1$-c, mesh M2, E_x.



(d) Method DG-$\mathbb{P}_1$-c, mesh M2, E_y.



(e) Method DG-$\mathbb{P}_1$-u, mesh M2, E_x.



(f) Method DG-$\mathbb{P}_1$-u, mesh M2, E_y.

Fig. 6. Diffraction of a plane wave in vacuum by a PEC cube, $F = 900$ MHz. Contour lines of E_x and E_y in the plane $Z = 0.5$ m.

Table 5
Diffraction of a plane wave in vacuum by a PEC cube, $F = 900$ MHz

Mesh	Method	Strategy	N_s	# it	CPU (min/max)	REAL
M1	DG-$\mathbb{P}_1$-c	DD-bicgl	8	6	202 s/352 s	355 s
–	–	DD-gmres	–	6	106 s/118 s	124 s
–	–	DD-itref	–	6	102 s/130 s	136 s
M2	DG-$\mathbb{P}_1$-c	DD-bicgl	32	9	253 s/440 s	506 s
–	–	–	64	10	105 s/202 s	236 s
–	–	DD-gmres	32	9	115 s/151 s	207 s
–	–	–	64	10	60 s/82 s	117 s
–	–	DD-itref	32	9	108 s/152 s	168 s
–	–	–	64	10	47 s/72 s	91 s
M2	DG-$\mathbb{P}_1$-u	DD-bicgl	32	9	343 s/389 s	430 s
–	–	–	64	10	161 s/207 s	234 s
–	–	DD-gmres	32	9	170 s/204 s	258 s
–	–	–	64	11	90 s/116 s	137 s
–	–	DD-itref	32	9	114 s/131 s	174 s
–	–	–	64	10	51 s/69 s	94 s
M2	DG-$\mathbb{P}_0$-c	DD-bicgl	16	6	48 s/61 s	64 s
–	–	DD-gmres	–	6	26 s/32 s	35 s
–	–	DD-itref	–	6	20 s/27 s	31 s
M3	DG-$\mathbb{P}_0$-c	DD-bicgl	16	7	150 s/184 s	199 s
–	–	–	32	8	81 s/101 s	122 s
M4	DG-$\mathbb{P}_0$-c	DD-bicgl	16	7	345 s/395 s	452 s
–	–	–	32	8	161 s/224 s	238 s
–	–	–	64	9	87 s/108 s	120 s

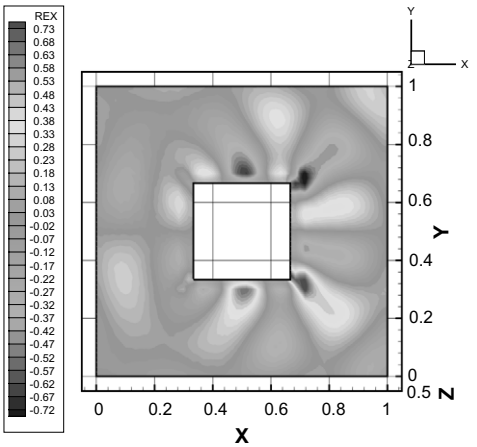
Computation times (solution phase).

Table 6
Diffraction of a plane wave in vacuum by a PEC cube, $F = 900$ MHz

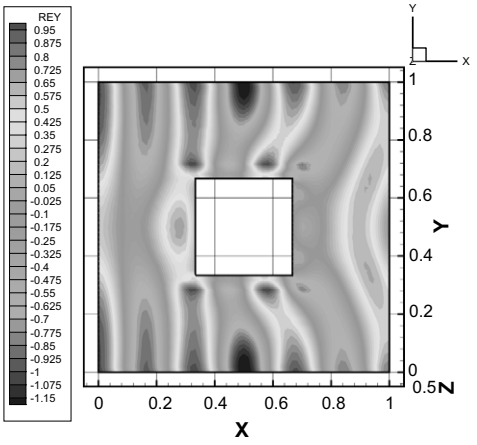
Mesh	Method	N_s	CPU (min/max)	RAM (min/max)
M1	DG-$\mathbb{P}_1$-c	8	82 s/129 s	735 MB/905 MB
M2	DG-$\mathbb{P}_1$-c	32	71 s/105 s	614 MB/728 MB
–	–	64	17 s/32 s	217 MB/295 MB
M2	DG-$\mathbb{P}_1$-u	32	88 s/143 s	710 MB/874 MB
–	–	64	20 s/36 s	243 MB/329 MB
M2	DG-$\mathbb{P}_0$-c	16	5 s/8 s	82 MB/105 MB
M3	–	16	23 s/36 s	285 MB/359 MB
–	–	32	7 s/10 s	104 MB/134 MB
M4	–	16	76 s/114 s	732 MB/884 MB
–	–	32	21 s/37 s	265 MB/359 MB
–	–	64	6 s/10 s	96 MB/129 MB

Computation times and memory requirement for storing the L and U factors.

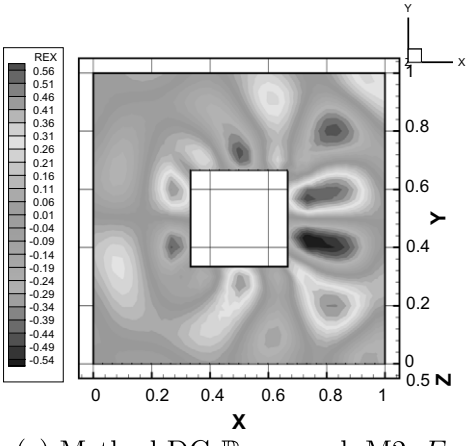
On the other hand, it is equally important to observe that although the MeTiS partitioning tool almost always yields well-balanced partitions (in the present case, the load balance is evaluated in terms of the local number of tetrahedra), there is a noticeable disparity in the required amount of RAM for storing the subdomain L and U factors, especially for large values of the number of subdomains N_s. As a matter of fact, the fill-in of the L and U factors is influenced by several factors among which, the presence in the original matrix of diagonal blocks related to physical boundaries (metallic wall, absorbing boundary) which in turn has effects on the numerical pivoting strategy. But, above all, the partitioning of a mesh (in practice, the adjacency graph associated to the mesh) using a tool such as MeTiS, is dictated by two main criteria, namely the minimization



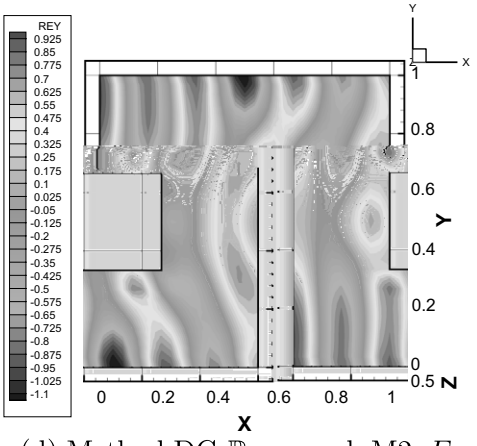
(a) Method DG-$\mathbb{P}_0$-c, mesh M5, E_x.



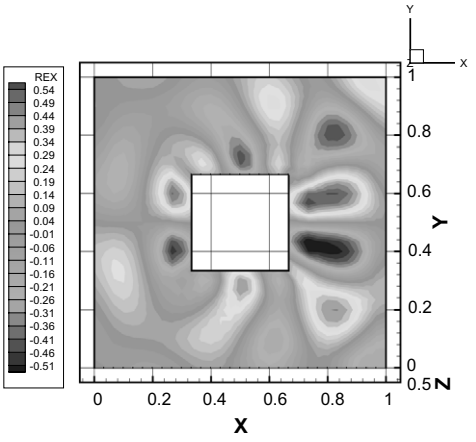
(b) Method DG-$\mathbb{P}_0$-c, mesh M5, E_y.



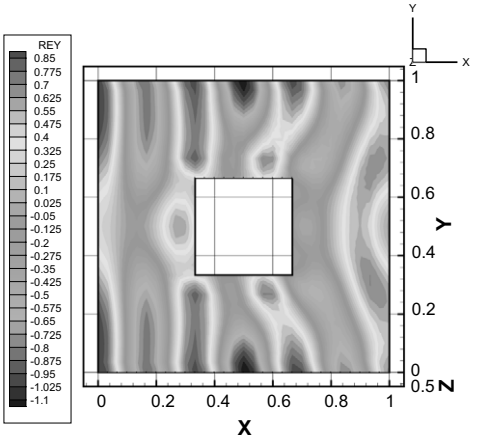
(c) Method DG-$\mathbb{P}_1$-c, mesh M2, E_x.



(d) Method DG-$\mathbb{P}_1$-c, mesh M2, E_y.



(e) Method DG-$\mathbb{P}_1$-u, mesh M2, E_x.



(f) Method DG-$\mathbb{P}_1$-u, mesh M2, E_y.

Fig. 7. Diffraction of a plane wave by a coated PEC cube, $F = 900$ MHz. Contour lines of E_x and E_y in the plane $Z = 0.5$ m.

Table 7
Diffraction of a plane wave by a coated PEC cube, $F = 900$ MHz

Mesh	Method	Strategy	N_s	# it	CPU (min/max)	REAL
M2	DG-$\mathbb{P}_1$-c	DD-bicgl	32	14	375 s/643 s	678 s
–	–	DD-gmres	–	14	198 s/236 s	256 s
–	–	DD-itref	–	14	173 s/247 s	271 s
M2	DG-$\mathbb{P}_1$-u	DD-bicgl	32	13	471 s/562 s	630 s
–	–	DD-gmres	–	13	245 s/290 s	317 s
–	–	DD-itref	–	13	156 s/184 s	215 s
M5	DG-$\mathbb{P}_0$-c	DD-bicgl	64	14	247 s/310 s	339 s
–	–	DD-gmres	–	14	130 s/164 s	185 s
–	–	DD-itref	–	14	98 s/133 s	157 s

Computation times (solution phase).

Table 8
Diffraction of a plane wave by a coated PEC cube, $F = 900$ MHz

Mesh	Method	N_s	CPU (min/max)	RAM (min/max)
M2	DG-$\mathbb{P}_1$-c	32	71 s/102 s	614 MB/728 MB
M2	DG-$\mathbb{P}_1$-u	32	87 s/143 s	710 MB/874 MB
M5	DG-$\mathbb{P}_0$-c	64	16 s/29 s	213 MB/299 MB

Computation times and memory requirement for storing the L and U factors.

of the subdomains separator and the achievement of a well-balanced computational load, while a balance of the fill-in is rarely an objective. In the present case, the main consequence of this load unbalance in the fill-in of the local L and U factors is a potentially large gap between the minimum and maximum CPU times for the subdomain triangular solves, which results in non-negligible idle times across processes between each iteration of the interface system solver.

5.6. A bioelectromagnetics application

We conclude this section of results with the application of the proposed numerical methodology to the simulation of a time-harmonic electromagnetic wave propagation problem in an irregularly shaped and heterogeneous medium. The problem under consideration is concerned with the propagation of a plane wave in realistic geometrical models of head tissues. It is a first step towards the development of a computational framework for the numerical dosimetry of electromagnetic fields radiated by mobile phones. Starting from MR images of the Visible Human 2.0 project [39], head tissues are segmented and the interfaces of a selected number of tissues (namely, the skin, the skull and the brain) are triangulated. Different strategies can be used in order to obtain a smooth and accurate segmentation of head tissues and interface triangulations as well. A first strategy consists in using a marching cube algorithm [35] which leads to huge triangulations of interfaces between segmented subdomains. These triangulations can then be regularized, refined and decimated in order to obtain reasonable surface meshes, for example using the YAMS [22] re-meshing tool. Another strategy consists in using a variant of Chew's algorithm [8], based on Delaunay triangulation restricted to the interface, which allows to control the size and aspect ratio of interfacial triangles [5]. Surface meshes of increased resolution resulting from such a procedure are presented in Fig. 8. Then, these triangulated surfaces together with a triangulation of the artificial boundary (absorbing boundary) of the overall computational domain, which is taken here to be a sphere, are used as inputs for the generation of volume meshes. In this study, the GHS3D tetrahedral mesh generator [24] is used to mesh volume domains between the various interfaces. Two tetrahedral meshes have been used whose characteristics are

summarized in Table 9. The frequency of the incident plane wave is $F = 1800$ MHz and its polarization is such that:

$$\mathsf{k} = (k_x, 0, 0)^t, \quad \mathsf{E} = (0, 0, E_z)^t \quad \text{and} \quad \mathsf{H} = (0, H_y, 0)^t.$$

Albeit this propagation problem clearly involves irregularly shaped domains and non-uniform tetrahedral meshes, it is yet a simplified configuration with regards to the simulations usually used in numerical dosimetry studies of human exposition to mobile phone radiation [4], for two reasons: a mobile phone geometrical model has not been taken into account in the present simulation setting and, the electromagnetic parameters of the materials are set to artificial values for the purpose of exemplifying the characteristics of the propagation of the plane wave in the head tissues (null conductivity, $\varepsilon_r = 4.0$ for the brain, $\varepsilon_r = 6.5$ for the cerebrospinal fluid, $\varepsilon_r = 1.5$ for the skull and $\varepsilon_r = 4.0$ for the skin). For the computations reported here, the methods DG-$\mathbb{P}_1$-c and DG-$\mathbb{P}_1$-u are used in conjunction with mesh M1 while method DG-$\mathbb{P}_0$-c is used with mesh M2. Moreover, this problem has also been simulated using a DGTD-$\mathbb{P}_1$-c (Discontinuous Galerkin Time-Domain) method [21] and the corresponding result will be considered here as the reference solution. The contour lines of E_z in various configurations are visualized in Figs. 10–13. As can be seen on these figures, on one hand, there is a good agreement between the results of the time-domain and time-harmonic computations and, on the other hand, the DG-$\mathbb{P}_1$ methods used with the coarsest mesh yield solutions which are closer to the reference computation than the one resulting from the DG-$\mathbb{P}_0$-c method applied on the finest geometrical model.

Performance results are given in Tables 10 and 11. In addition, the convergence curves for the iterative solution of the interface system (4.17) using the BiCGstab(ℓ) method are shown in Fig. 9. Firstly, we note that the iterative solution requires 3–4 times more iterations than the numbers observed for the previous test cases, which is the consequence of the increased complexity in both the underlying discretization and the propagation medium. Secondly, the parallel efficiency, evaluated using the %CPU ratio, ranges from 65% to 75%. Here again, the load unbalance in the fill-in of the local L and U factors is the main reason for this parallel performance drop.

Table 9
Propagation of a plane wave in a heterogeneous medium, $F = 1800$ MHz

Mesh	# Vertices	# Tetrahedra	$L_{\min}$ (m)	$L_{\max}$ (m)	L_{avg} (mm)
M1	60,590	361,848	0.00185	0.04537	0.01165
M2	309,599	1,853,832	0.00158	0.02476	0.00693

Characteristics of the tetrahedral meshes.

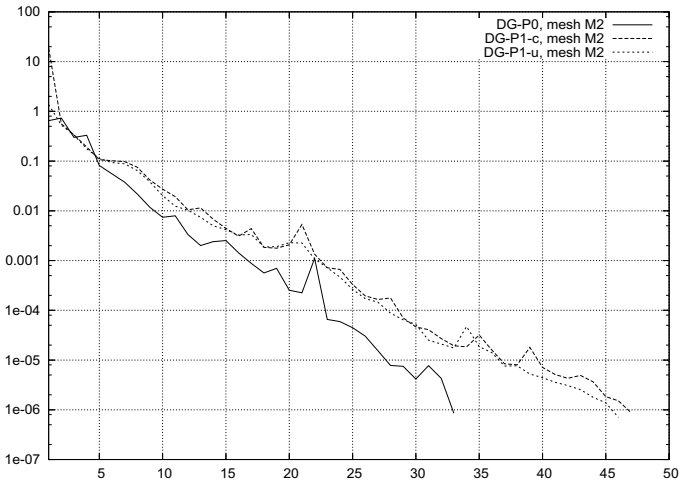
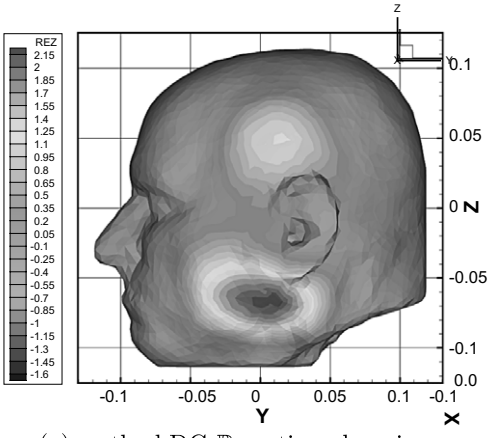
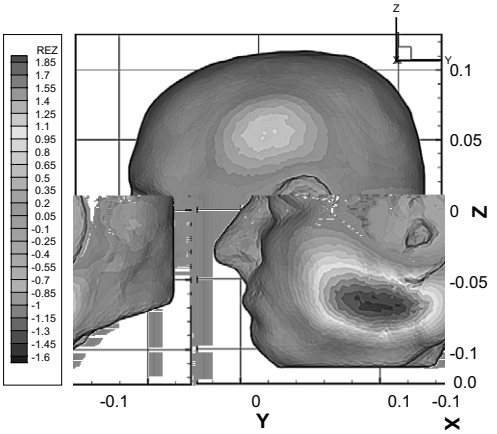


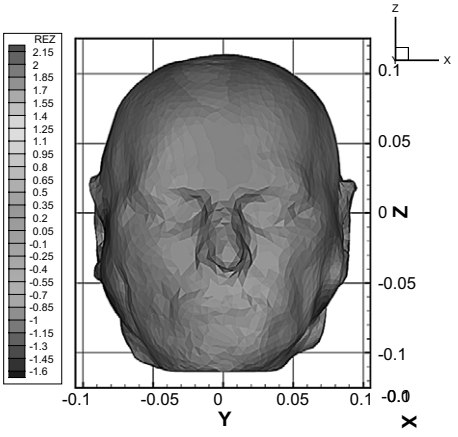
Fig. 9. Propagation of a plane wave in a heterogeneous medium, $F = 1800$ MHz. Iterative solution of the interface system.



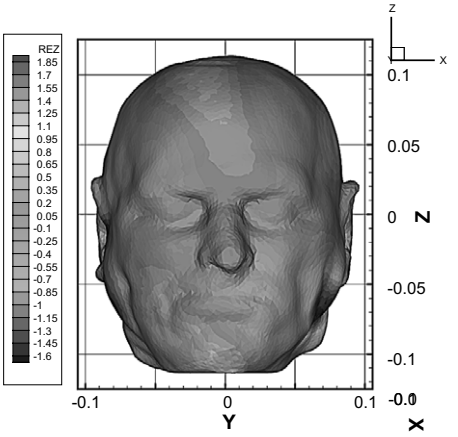
(a) method DG-$\mathbb{P}_1$-c, time-domain, skin, lateral view, mesh M1.



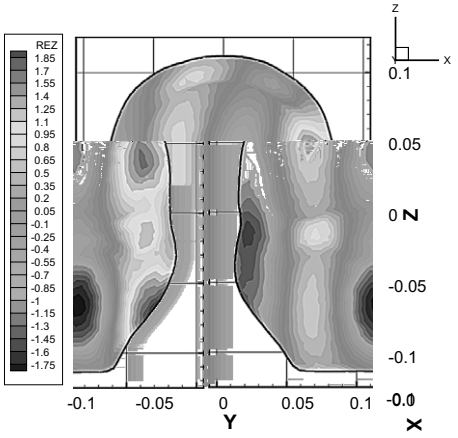
(b) method DG-$\mathbb{P}_0$-c, time-harmonic, skin, lateral view, mesh M2.



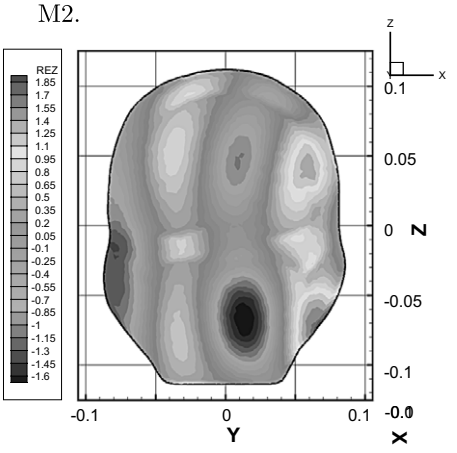
(c) method DG-$\mathbb{P}_1$-c, time-domain, skin, frontal view, mesh M1.



(d) method DG-$\mathbb{P}_0$-c, time-harmonic, skin, frontal view, mesh M2.

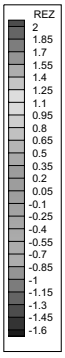


(e) method DG-$\mathbb{P}_1$-c, time-domain, view in the plane Y=0.0 m, mesh M1.

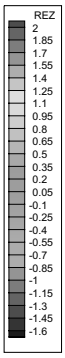


(f) method DG-$\mathbb{P}_0$-c, time-harmonic, view in the plane Y=0.0 m, mesh M2.

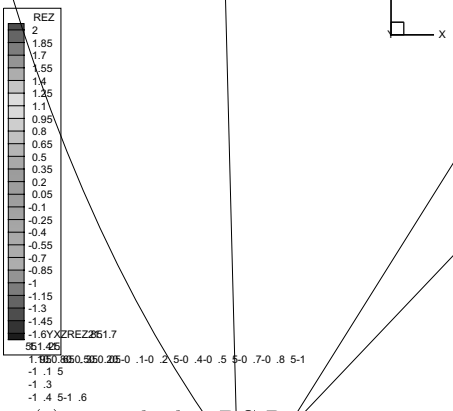
Fig. 10. Propagation of a plane wave in a heterogeneous medium, $F = 1800$ MHz. Contour lines of E_z.



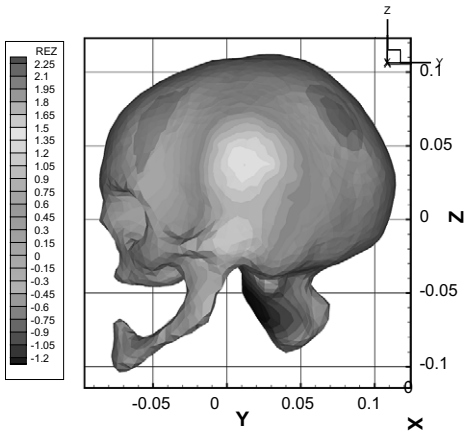
(a) method DG-$\mathbb{P}_1$-c, time-harmonic, skin, lateral view, mesh M1.



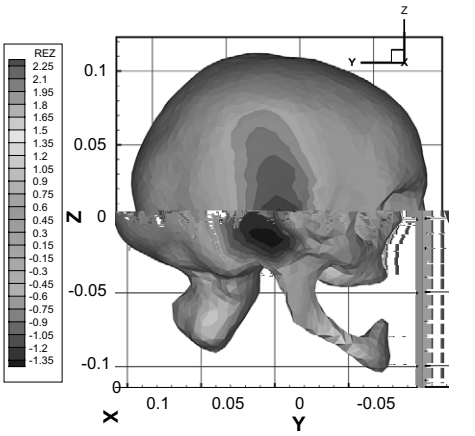
(b) method DG-$\mathbb{P}_1$-u, time-harmonic, skin, lateral view, mesh M1.



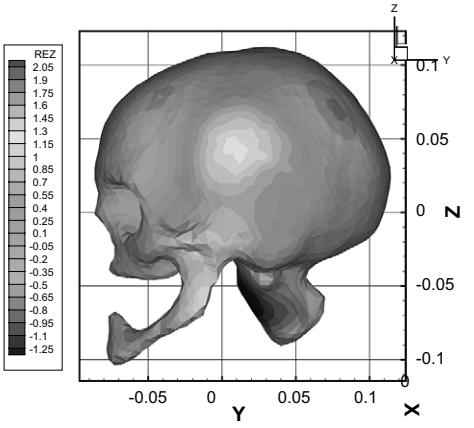
(c) method DG-$\mathbb{P}_1$-c, time-harmonic, skin, front



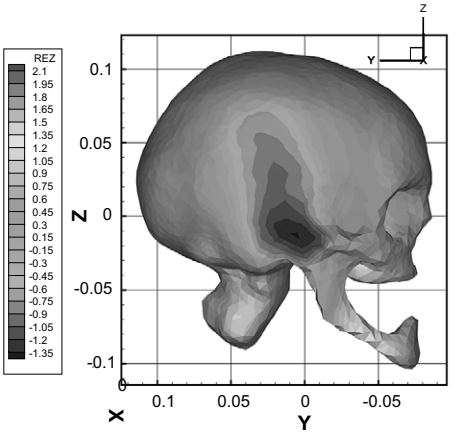
(a) method DG-$\mathbb{P}_1$-c, time-domain, skull, left side, mesh M1.



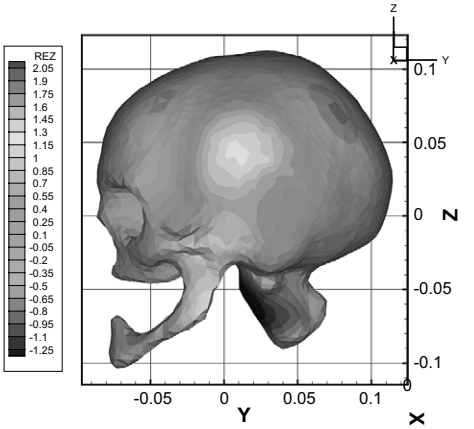
(b) method DG-$\mathbb{P}_1$-c, time-domain, skull, right side, mesh M1.



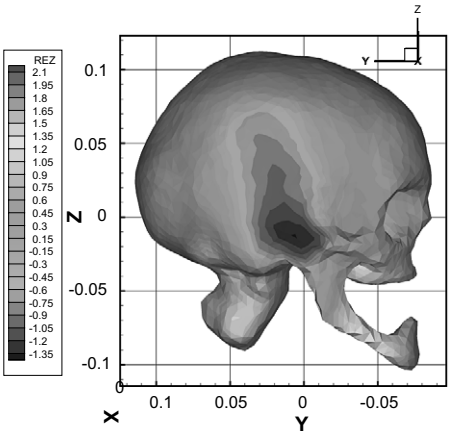
(c) method DG-$\mathbb{P}_1$-c, time-harmonic, skull, left side, mesh M1.



(d) method DG-$\mathbb{P}_1$-c, time-harmonic, skull, right side, mesh M1.

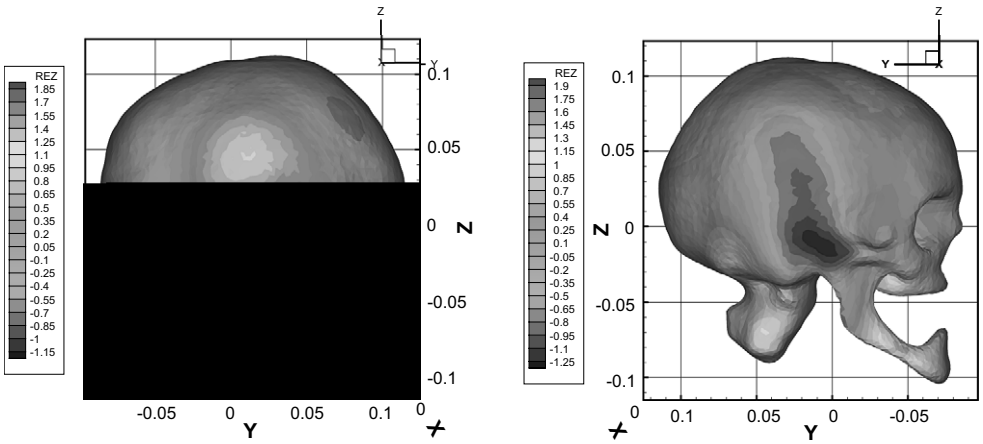


(e) method DG-$\mathbb{P}_1$-u, time-harmonic, skull, left side, mesh M1.



(f) method DG-$\mathbb{P}_1$-u, time-harmonic, skull, right side, mesh M1.

Fig. 12. Propagation of a plane wave in a heterogeneous medium, $F = 1800$ MHz. Contour lines of E_z.



(a) Skull, left side. (b) Skull, right side.

Fig. 13. Propagation of a plane wave in a heterogeneous medium, $F = 1800$ MHz. Contour lines of E_z: method DG-$\mathbb{P}_0$-c with mesh M2.

Table 10
Propagation of a plane wave in an heterogeneous medium, $F = 1800$ MHz

Mesh	Method	Strategy	N_s	# it	CPU (min/max)	REAL
M1	DG-$\mathbb{P}_1$-c	`DD-itref`	96	47	346 s/466 s	714 s
–	DG-$\mathbb{P}_1$-u	`DD-itref`	96	46	347 s/547 s	765 s
M2	DG-$\mathbb{P}_0$-c	`DD-itref`	96	33	228 s/322 s	428 s

Computation times (solution phase).

Table 11
Propagation of a plane wave in a heterogeneous medium, $F = 1800$ MHz

Mesh	Method	N_s	CPU (min/max)	RAM (min/max)	# dof
M1	DG-$\mathbb{P}_1$-c	96	64 s/125 s	640 MB/852 MB	8,684,352
–	DG-$\mathbb{P}_1$-u	96	80 s/134 s	633 MB/866 MB	8,684,352
M2	DG-$\mathbb{P}_0$-c	96	53 s/98 s	519 MB/684 MB	11,122,992

Computation times and memory requirement for storing the L and U factors.

5.7. Conclusion

We have presented a hybrid iterative/direct solution method for the large, sparse and complex coefficients algebraic systems resulting from the discretization of the time-harmonic Maxwell equations by discontinuous Galerkin methods. The discretization in space relies on an unstructured tetrahedral mesh and as a result, the proposed numerical methodology is particularly well suited to the simulation of wave propagation problems in irregularly shaped media. Moreover, the local nature of a discontinuous Galerkin formulation allows for a natural treatment of heterogeneous media. Numerical and performance results reported here, albeit promising, have also raised a weakness in the current implementation of the domain decomposition method in the fact that the fill-in of the local L and U factors is generally not well-balanced except for relatively simple problems (simply shaped domain, uniform mesh and homogeneous media). We believe that this drawback should be recurrent to almost all similar implementations of domain decomposition algorithms (*i.e.* based on exact factorization methods for the subdomain solves). This problem could be figure out by resorting to constrained level of fill-in subdomain solvers or/and by improving the quality of the mesh partitions (with regards to the

resulting fill-in unbalance). In addition to these directions, our future works will be towards the improvement of the numerical efficiency of the Schwarz-type algorithm adopted in this study thanks to the design of discrete optimized interface conditions [16] in the framework of our discontinuous Galerkin formulations on tetrahedral meshes.

References

[1] A. Alonso-Rodriguez, L. Gerardo-Giorda, New nonoverlapping domain decomposition methods for the harmonic Maxwell system, SIAM J. Sci. Comput. 28 (1) (2006) 102–122.
[2] P. Amestoy, I. Duff, J.-Y. L'Excellent, Multifrontal parallel distributed symmetric and unsymmetric solvers, Comput. Meth. App. Mech. Eng. 184 (2000).
[3] M. Benzi, J. Haws, M. Tuma, Preconditioning highly indefinite and nonsymmetric matrices, SIAM J. Sci. Comput. 22 (4) (2000) 1333–1353.
[4] P. Bernardi, M. Cavagnaro, S. Pisa, E. Piuzzi, Specific absorption rate and temperature increases in the head of a cellular phone user, IEEE Trans. Microwave Theory Tech. 48 (7) (2000) 1118–1126.
[5] J.-D. Boissonnat, S. Oudot, Provably good sampling and meshing of surfaces, Graph. Models 67 (5) (2005) 405–451.
[6] F. Bourdel, P. Mazet, P. Helluy, Resolution of the non-stationary or harmonic Maxwell equations by a discontinuous finite element method, in: 10th International Conference in Computing Methods in Applied Sciences and Engineering, Nova Science, Paris, 1992, pp. 1–18.
[7] A. Buffa, I. Perugia, Discontinuous Galerkin approximation of the Maxwell eigenproblem, SIAM J. Numer. Anal. 44 (5) (2006) 2198–2226.
[8] L. Chew, Guaranteed-quality mesh generation for curved surfaces, in: 9th Annual ACM Symposium Computational Geometry, ACM Press, 1993, pp. 274–280.
[9] B. Cockburn, G. Karniadakis, C. Shu (Eds.), Discontinuous Galerkin methods. Theory, computation and applications, Lecture Notes in Computational Science and Engineering, vol. 11, Springer-Verlag, 2000.
[10] B. Cockburn, C. Shu (Eds.), Special issue on discontinuous Galerkin methods, J. Sci. Comput, 22–23, Springer, 2005.
[11] P. Collino, G. Delbue, P. Joly, A. Piacentini, A new interface condition in the non-overlapping domain decomposition, Comput. Methods Appl. Mech. Eng. 148 (1997) 195–207.
[12] C. Dawson (Ed.), Special issue on discontinuous Galerkin methods, Comput. Meth. App. Mech. Eng., vol. 195, Elsevier, 2006.
[13] B. Després, Décomposition de domaine et problème de Helmholtz, C.R. Acad. Sci. Paris 1 (6) (1990) 313–316.
[14] B. Després, P. Joly, J. Roberts, A domain decomposition method for the harmonic Maxwell equations, in: Iterative Methods in Linear Algebra, North-Holland, Amsterdam, 1992, pp. 475–484.
[15] V. Dolean, H. Fol, S. Lanteri, R. Perrussel, Solution of the time-harmonic Maxwell equations using discontinuous Galerkin methods, J. Comput. Appl. Math. Available on line, doi:10.1016/j.cam.2007.05.026, in press.
[16] V. Dolean, L. Gerardo-Giorda, M. Gander, Optimized Schwarz methods for Maxwell equations, <http://hal.archives-ouvertes.fr/hal-00107263>, submitted for publication.
[17] M. Duruflé, Intégration numérique et éléments finis d'ordre élevé appliqués aux équations de Maxwell en régime harmonique, Thèse de doctorat en mathématiques Appliquées, Université Paris Dauphine, 2006.
[18] A. Ern, J.-L. Guermond, Discontinuous Galerkin methods for Friedrichs systems I. General theory, SIAM J. Numer. Anal. 44 (2) (2006) 753–778.
[19] A. Ern, J.-L. Guermond, Discontinuous Galerkin methods for Friedrichs systems II.Second-order elliptic PDE's, SIAM J. Numer. Anal. 44 (6) (2006) 2363–2388.
[20] H. Fahs, S. Lanteri, F. Rapetti, A hp-like discontinuous Galerkin method for solving the 2D time-domain maxwell's equations on non-conforming locally refined triangular meshes, Tech. Rep. RR-6162, INRIA, 2007. <http://hal.inria.fr/ inria-00140783/fr/>.
[21] L. Fezoui, S. Lanteri, S. Lohrengel, S. Piperno, Convergence and stability of a discontinuous Galerkin time-domain method for the 3D heterogeneous Maxwell equations on unstructured meshes, ESAIM: Math. Model. Numer. Anal. 39 (6) (2005) 1149–1176.
[22] P. Frey, YAMS: a fully automatic adaptive isotropic surface remeshing procedure, INRIA Research Report No. 4252, 2003.
[23] M. Gander, F. Magoulès, F. Nataf, Optimized Schwarz methods without overlap for the Helmholtz equation, SIAM J. Sci. Comput. 24 (1) (2002) 38–60.
[24] P.-L. George, F. Hecht, E. Saltel, Automatic mesh generator with specified boundary, Comput. Methods Appl. Mech. Eng. 92 (1991) 269–288.
[25] P. Helluy, Résolution numérique des équations de Maxwell harmoniques par une méthode d'éléments finis discontinus, Thèse en mathématiques appliquées, Ecole Nationale Supérieure de l'Aéronautique, 1994.
[26] P. Helluy, S. Dayma, Convergence d'une approximation discontinue des systèmes du premier ordre, C.R. Acad. Sci. Paris Sér. I Math. 319 (12) (1994) 1331–1335.
[27] P. Hénon, P. Ramet, J. Roman, PaStiX: a high-performance parallel direct solver for sparse symmetric definite systems, Parallel Comput. 28 (2002) 301–321.
[28] J. Hesthaven, T. Warburton, Nodal high-order methods on unstructured grids. I Time-domain solution of Maxwell's equations, J. Comput. Phys. 181 (1) (2002) 186–221.
[29] P. Houston, I. Perugia, A. Schneebeli, D. Schötzau, Interior penalty method for the indefinite time-harmonic Maxwell equations, Numer. Math. 100 (3) (2005) 485–518.

[30] P. Houston, I. Perugia, A. Schneebeli, D. Schötzau, Mixed discontinuous Galerkin approximation of the Maxwell operator: the indefinite case, ESAIM: Math. Model. Numer. Anal. 39 (4) (2005) 727–753.
[31] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM J. Sci. Comput. 20 (1) (1999) 359–392.
[32] J. Kurzak, J. Dongarra, Implementation of the mixed-precision in solving systems of linear equations on the CELL processor, Tech. Rep. UT-CS-06-580, University of Tennessee, 2006.
[33] J. Langou, J. Langou, P. Luszczek, J. Kurzak, A. Buttari, J. Dongarra, Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy, Technical Report UT-CS-06-574, University of Tennessee, 2006.
[34] S. Lee, M. Vouvakis, J.F. Lee, A non-overlaping domain decomposition method with non-matching grids for modeling large finite antenna arrays, J. Comput. Phys. 203 (2005) 1–21.
[35] W. Lorensen, H. Cline, Marching cubes: a high resolution 3D surface construction algorithm, in: Siggraph 87, vol. 21, 1987, pp. 163–170.
[36] P. Monk, Finite element methods for Maxwell's equations, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2003.
[37] I. Perugia, D. Schötzau, P. Monk, Stabilized interior penalty methods for the time-harmonic Maxwell equations, Comput. Methods Appl. Mech. Eng. 191 (41–42) (2002) 4675–4697.
[38] S. Piperno, L^2-stability of the upwind first order finite volume scheme for the Maxwell equations in two and three dimensions on arbitrary unstructured meshes, M2AN: Math. Model. Numer. Anal. 34 (1) (2000) 139–158.
[39] P. Ratiu, B. Hillen, J. Glaser, D.P. Jenkins, Visible human 2.0 – the next generation, in: J.D. West wood (Ed.), Medicine Meets Virtual Reality 11 – NextMed: Health Horizon, vol. 11, IOS Press, 2003, pp. 275–281.
[40] Y. Saad, H. Schultz, GMRES: Generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[41] G. Sleijpen, D. Fokkema, BiCGstab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum, Electron. Trans. Numer. Anal. 1 (1993) 11–32, electronic only.
[42] M. Vouvakis, Z. Cendes, J.F. Lee, A FEM domain decomposition method for photonic and electromagnetic band gap structures, IEEE Trans. Ant. Prop. 54 (2) (2006) 721–733.