A hybridizable discontinuous Galerkin method combined to a Schwarz algorithm for the solution of 3d time-harmonic Maxwell's equations

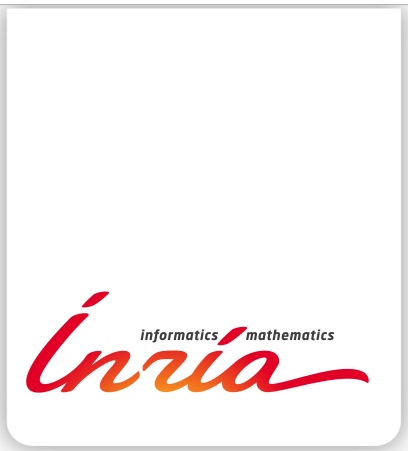
Liang Li, Stéphane Lanteri, Ronan Perrussel

A hybridizable discontinuous Galerkin method combined to a Schwarz algorithm for the solution of 3d time-harmonic Maxwell's equations

Liang Li, Stéphane Lanteri, Ronan Perrussel

RESEARCH REPORT
N° 8251
February 2013
Project-Teams Nachos

ISSN 0249-6399 ISRN INRIA/RR--8251--FR+ENG



A hybridizable discontinuous Galerkin method combined to a Schwarz algorithm for the solution of 3d time-harmonic Maxwell's equations

Liang Li*, Stéphane Lanteri†, Ronan Perrussel‡

Project-Teams Nachos

Research Report n° 8251 — February 2013 — 26 pages

Abstract: A Schwarz-type domain decomposition method is presented for the solution of the system of 3d time-harmonic Maxwell equations. We introduce a hybridizable discontinuous Galerkin (HDG) scheme for the discretization of the problem based on a tetrahedrization of the computational domain. The discrete system of the HDG method on each subdomain is solved by an optimized sparse direct (LU factorization) solver. The solution of the interface system in the domain decomposition framework is accelerated by a Krylov subspace method. The formulation and the implementation of the resulting DD-HDG (Domain Decomposed-Hybridizable Discontinuous Galerkin) method are detailed. Numerical results show that the resulting DD-HDG solution strategy has an optimal convergence rate and can save both CPU time and memory cost compared to a classical upwind flux-based DD-DG (Domain Decomposed-Discontinuous Galerkin) approach.

Key-words: Maxwell's equations, time-harmonic, discontinuous Galerkin, hybridizable method, domain decomposition, Schwarz algorithm

* School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, P. R. China, plum_liliang@uestc.edu.cn

† INRIA Sophia Antipolis - Méditerranée, Nachos project-team, France

‡ Université de Toulouse; CNRS; INPT, UPS; Laboratoire Plasma et Conversion d'Énergie (LAPLACE), Toulouse, France, perrussel@laplace.univ-tlse.fr

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Une méthode Galerkin discontinue hybride combinée à un algorithme de Schwarz pour la résolution des équations Maxwell 3d en régime harmonique

Résumé : On présente dans ce rapport une méthode par décomposition de domaine de type Schwarz pour la résolution du système d'équations de Naxwell 3d en régime harmonique. On introduit un schéma Galerkin discontinu hybride pour la discrétisation du problème en supposant que le domaine de calcul est triangulé par un maillage tétraédrique non-structuré. Le système discret obtenu dans chaque sous-domaine est résolu au moyen d'un solveur direct creaux (factorisation LU) optimisé. La résolution du système interface associé à l'algorithme de Schwarz repose sur une méthode de Krylov. La formulation et l'implémentation de la méthodologie numérique résultante sont détaillées. Les résutats numériques montrent que cette méthodologie a un taux de convergence optimal et permet de réduire notablement les coûts CPU et occupation mémoire comparativement à une méthodologie similaire basée sur un schéma Galerkin discontinu standard à flux décentré.

Mots-clés : équations de Maxwell, régime harmonique, Galerkin discontinu, méthode hybride, décomposition de domaine, algorithme de Schwarz

1 Introduction

During the last 10 years, discontinuous Galerkin (DG) methods have been extensively considered for obtaining approximate solution of Maxwell's equations, see [CLS04, DFFL10, FLLP05, HW02, HPS04]. Thanks to the discontinuity of the approximation, this kind of methods has many advantages, such as adaptivity to complex geometries through the use of unstructured possibly non-conforming meshes, easily obtained high order accuracy, *hp*-adaptivity and natural parallelism [HW08]. However, despite these advantages, DG methods have one main drawback particularly sensitive for stationary problems: the number of globally coupled degrees of freedom (DOFs) is much greater than the number of DOFs required by conforming finite element methods for the same accuracy. Consequently, DG methods are expensive in terms of both CPU time and memory consumption, especially for time-harmonic problems [DFLP08]. Hybridization of DG methods [CGL09] is devoted to address this issue while keeping all the advantages of DG methods. The design of such a hybridizable discontinuous Galerkin (HDG) method for the discretization of the system of 3d time-harmonic Maxwell's equations is one of the main objectives of the present work.

HDG methods introduce an additional *hybrid* variable on the faces of the elements, on which the definition of the local (element-wise) solutions is based. A so-called *conservativity condition* is imposed on the numerical trace, whose definition involved the hybrid variable, at the interface between neighboring elements [CGL09, LLP11a, NPC11]. As a result, HDG methods produce a linear system in terms of the DOFs of the additional hybrid variable only. In this way, the number of globally coupled DOFs is reduced. The local values of the electromagnetic fields can be obtained by solving local problems element-by-element. In this work, for the system of 3d time-harmonic Maxwell's equations, we propose a HDG formulation taking the tangential component of the magnetic field as the hybrid variable. We show that the reduced system of the hybrid variable has a wave-equation-like characterization, and the tangential components of the numerical traces for both electric and magnetic fields are single-valued. Moreover, numerical results seems to indicate that the approximate solutions for both $\mathbf{E}$ and $\mathbf{H}$ have optimal convergence orders.

Though the HDG method results in a smaller linear system than the one associated to a classical upwind flux-based DG method, the size of this system is often too large to be solved by a direct solver as soon as one consider realistic 3d problems. In addition, for very large-scale propagation problems, exploiting a multi-processor system is a mandatory path to reduce the solution time and have access to the required memory capacity. Unfortunately, it is also difficult to solve the resulting linear system with standard preconditioned iterative methods [EG11]. Therefore, as a second objective of the present work, we propose here to design a hybrid iterative-direct solution strategy for the HDG system, based on domain decomposition principles [SBG96]. Our previous works on domain decomposition methods for the solution of the time-harmonic Maxwell equations show promising results of Schwarz-type algorithms through many test problems in 2d and 3d [DLP08a, DLP08b]. A classical Schwarz algorithm which exchanges impedance data between subdomains is studied [DLP08a]. We consider here a similar algorithm and study its adaptation to the HDG discretization framework. We show that the interface system of the Schwarz algorithm is similar to the one obtain with the upwind flux-based DG discretization.

For solving the local sparse linear system within each subdomain in the Schwarz iteration, we adopt a mixed precision arithmetic methodology in order to further reduce the memory requirements and to tackle large problems. A multifrontal sparse direct solver [ADL00] is used to compute the LU factorization of the coefficient matrix in single precision, and then the L and U factors are employed as preconditioners for a Krylov subspace method which works on double precision DOFs vectors, or an iterative refinement procedure based on these two factors

is performed [BDK$^+$08] to retrieve double precision accuracy.

This paper is organized as follows. Section 2 defines the considered boundary value problem for the 3d time-harmonic Maxwell's equations and gives some notations. A Schwarz solution algorithm is also stated at the continuous level in this section. In Section 3 we propose a HDG formulation for the one-domain problem and discuss the properties of the reduced system. Section 4 presents the discretization of the domain decomposition algorithm. Numerical examples are given in section 5 with two objectives in mind: first, using a simple (model) problem, the proposed DD-HDG solution strategy is validated and its convergence properties are assessed; second, by considering more complex problems, we study the overall efficiency of the DD-HDG solution strategy and show how much one can gain in terms of CPU time and memory consumption with the new discretization. Finally, we draw some conclusions and state future works in section 6.

2 Problem statement and notations

2.1 Boundary value problem

The system of 3d time-harmonic Maxwell's equations is considered

$$\begin{cases} \mathrm{i}\omega\varepsilon_r\mathbf{E} - \mathbf{curl}\,\mathbf{H} = -\mathbf{J}, \text{ in } \Omega, \\ \mathrm{i}\omega\mu_r\mathbf{H} + \mathbf{curl}\,\mathbf{E} = 0, \text{ in } \Omega, \\ \mathbf{n}\times\mathbf{E} = 0, \text{ on } \Gamma_m, \\ \mathbf{n}\times\mathbf{E} + \mathbf{n}\times(\mathbf{n}\times\mathbf{H}) = \mathbf{n}\times\mathbf{E}^{\mathrm{inc}} + \mathbf{n}\times(\mathbf{n}\times\mathbf{H}^{\mathrm{inc}}), \text{ on } \Gamma_a, \end{cases} \tag{1}$$

where i is the imaginary unit, ω is the angular frequency, ε_r and μ_r are the relative permittivity and permeability, $\mathbf{n}$ is the outward unit normal vector, $\mathbf{E}$ and $\mathbf{H}$ are the electric and magnetic fields, $\mathbf{J}$ is a known current source, and $(\mathbf{E}^{\mathrm{inc}}, \mathbf{H}^{\mathrm{inc}})$ is a given incident electromagnetic wave. The boundary of the computational domain Ω is $\partial\Omega = \Gamma_m \cup \Gamma_a$ with $\Gamma_m \cap \Gamma_a = \emptyset$. The boundary condition on Γ_m indicates a metallic boundary condition (also called perfect electric conductor condition), while the second relation states a Silver-Müller (first order absorbing boundary) condition on Γ_a.

2.2 Notations

We consider a simplicial mesh $\mathcal{T}_h$ (consisting of tetrahedra) of the computational domain Ω. We denote by $\mathcal{F}_h^I$ the union of all interior interfaces of $\mathcal{T}_h$, by $\mathcal{F}_h^B$ the union of all the boundary interfaces of $\mathcal{T}_h$, and by $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^B$. For an interface $F = \overline{K}^+ \cap \overline{K}^- \in \mathcal{F}_h^I$, let $\mathbf{v}^\pm$ be the traces of $\mathbf{v}$ on F from the interior of $K^\pm$. On this face, we define *mean* (*average*) *values* $\{\cdot\}$ and *jumps* $[\![\cdot]\!]$ ($[\![[\cdot]]\!]$ only consider in Section 4) as follows

$$\begin{cases} \{\mathbf{v}\}_F = \dfrac{1}{2}(\mathbf{v}^+ + \mathbf{v}^-), \\ [\![\mathbf{v}]\!]_F = \mathbf{n}^+\times\mathbf{v}^+ + \mathbf{n}^-\times\mathbf{v}^-, \quad [\![[\mathbf{v}]]\!]_F = \mathbf{v}^+ - \mathbf{v}^-, \end{cases}$$

where $\mathbf{n}^\pm$ denotes the outward unit norm vector to $K^\pm$. For the boundary faces these expressions are modified as (assuming $F = \partial K^+ \cap \partial\Omega$)

$$\begin{cases} \{\mathbf{v}\}_F = \mathbf{v}^+, \\ [\![\mathbf{v}]\!]_F = \mathbf{n}^+\times\mathbf{v}^+, \quad [\![[\mathbf{v}]]\!]_F = 0, \end{cases}$$

since we assume $\mathbf{v}$ is single valued on the boundaries. Let $\mathbb{P}_p(D)$ denote the space of polynomial functions of degree at most p on a domain D. For any element $K \in \mathcal{T}_h$, let $\mathbf{V}^p(K) \equiv (\mathbb{P}_p(K))^3$. The discontinuous finite element spaces are then introduced as

$$\mathbf{V}_h^p = \left\{ \mathbf{v} \in (L^2(\Omega))^3 \mid \mathbf{v}|_K \in \mathbf{V}^p(K),\ \forall K \in \mathcal{T}_h \right\},$$

where $L^2(\Omega)$ is the space of square integrable functions on the domain Ω. We introduce a traced finite element space

$$\mathbf{M}_h^p = \left\{ \boldsymbol{\eta} \in (L^2(\mathcal{F}_h))^3 \mid \boldsymbol{\eta}|_F \in (\mathbb{P}_p(F))^3,\ (\boldsymbol{\eta} \cdot \mathbf{n})|_F = 0,\ \forall F \in \mathcal{F}_h \right\}. \tag{2}$$

Note that $\mathbf{M}_h^p$ consists of vector-valued functions whose normal component is zero on any face $F \in \mathcal{F}_h$. For two vectorial functions $\mathbf{u}$ and $\mathbf{v}$ in $(L^2(D))^3$, we denote $(\mathbf{u}, \mathbf{v})_D = \int_D \mathbf{u} \cdot \overline{\mathbf{v}}\, \mathrm{d}x$ provided D is a domain in $\mathbb{R}^3$, and we denote $\langle \mathbf{u}, \mathbf{v} \rangle_F = \int_F \mathbf{u} \cdot \overline{\mathbf{v}}\, \mathrm{d}s$ if F is a two-dimensional face. Accordingly, for the mesh $\mathcal{T}_h$ we have

$$(\cdot\,,\cdot)_{\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} (\cdot\,,\cdot)_K, \qquad \langle \cdot\,,\cdot \rangle_{\partial\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \langle \cdot\,,\cdot \rangle_{\partial K},$$
$$\langle \cdot\,,\cdot \rangle_{\mathcal{F}_h} = \sum_{F \in \mathcal{F}_h} \langle \cdot\,,\cdot \rangle_F, \qquad \langle \cdot\,,\cdot \rangle_{\Gamma_a} = \sum_{F \in \mathcal{F}_h \cap \Gamma_a} \langle \cdot\,,\cdot \rangle_F.$$

We set

$$\mathbf{v}^t = -\mathbf{n} \times (\mathbf{n} \times \mathbf{v}) \quad \text{and} \quad \mathbf{v}^n = \mathbf{n}(\mathbf{n} \cdot \mathbf{v}),$$

where $\mathbf{v}^t$ and $\mathbf{v}^n$ denote the tangential and normal components of $\mathbf{v}$ on a face of unit normal vector $\mathbf{n}$, and $\mathbf{v} = \mathbf{v}^t + \mathbf{v}^n$.

2.3 Continuous Schwarz algorithm

To simplify the presentation we assume a two-subdomain decomposition of the domain Ω into overlapping or non-overlapping subdomains Ω_1 and Ω_2. We set $\Gamma_{1,2} = \partial\Omega_1 \cap \Omega_2$ and $\Gamma_{2,1} = \partial\Omega_2 \cap \Omega_1$. By $\mathbf{n}_{i,j}$ $(i,j \in \{1,2\})$, we denote the outward unit normal vector to the interface $\Gamma_{i,j}$. The classical Schwarz algorithm in this setting is an iterative method where the $(n+1)$-th iterate is defined from the n-th iterate by solving the subdomain problems

$$\begin{cases} \mathrm{i}\omega\varepsilon_r \mathbf{E}^{(1),n+1} - \mathbf{curl}\, \mathbf{H}^{(1),n+1} = -\mathbf{J} \text{ in } \Omega_1, \\ \mathrm{i}\omega\mu_r \mathbf{H}^{(1),n+1} + \mathbf{curl}\, \mathbf{E}^{(1),n+1} = 0 \text{ in } \Omega_1, \\ \mathcal{B}_{\mathbf{n}_{1,2}}(\mathbf{E}^{(1),n+1}, \mathbf{H}^{(1),n+1}) = \mathcal{B}_{\mathbf{n}_{1,2}}(\mathbf{E}^{(2),n}, \mathbf{H}^{(2),n}), \\ \quad + \text{Boundary conditions on } \partial\Omega_1 \cap \partial\Omega, \end{cases}$$
$$\begin{cases} \mathrm{i}\omega\varepsilon_r \mathbf{E}^{(2),n+1} - \mathbf{curl}\, \mathbf{H}^{(2),n+1} = -\mathbf{J} \text{ in } \Omega_2, \\ \mathrm{i}\omega\mu_r \mathbf{H}^{(2),n+1} + \mathbf{curl}\, \mathbf{E}^{(2),n+1} = 0 \text{ in } \Omega_2, \\ \mathcal{B}_{\mathbf{n}_{2,1}}(\mathbf{E}^{(2),n+1}, \mathbf{H}^{(2),n+1}) = \mathcal{B}_{\mathbf{n}_{2,1}}(\mathbf{E}^{(1),n}, \mathbf{H}^{(1),n}), \\ \quad + \text{Boundary conditions on } \partial\Omega_2 \cap \partial\Omega, \end{cases} \tag{3}$$

where $(\mathbf{E}^{(i),n}, \mathbf{H}^{(i),n})$ denotes the electromagnetic field in the subdomain Ω_i and this local electromagnetic field should converge towards the true electromagnetic field in Ω_i as n goes to infinity. Moreover, $\mathcal{B}_{\mathbf{n}}$ denotes the transmission operator which, in the case of the classical Schwarz algorithm for the system of Maxwell's equations [DFLP08] writes as

$$\mathcal{B}_{\mathbf{n}}(\mathbf{E}, \mathbf{H}) = \mathbf{n} \times \frac{\mathbf{E}}{Z_r} + \mathbf{n} \times (\mathbf{n} \times \mathbf{H}) = \mathbf{n} \times \frac{\mathbf{E}}{Z_r} - \mathbf{H}^t, \tag{4}$$

with $Z_r = \sqrt{\dfrac{\mu_r}{\varepsilon_r}}$ being the impedance. We will comment later on the definition of the transmission operator in terms of the impedance condition.

3 Discretization of the one-domain problem

For sake of simplicity, we omit the volume source term $\mathbf{J}$ in what follows but it can be straightforwardly added.

3.1 Principles

The discontinuous Galerkin (DG) method seeks an approximate solution $(\mathbf{E}_h, \mathbf{H}_h)$ in the space $\mathbf{V}_h^p \times \mathbf{V}_h^p$ that should satisfy for all K in $\mathcal{T}_h$

$$\begin{cases} (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h, \mathbf{v})_K - (\mathbf{curl}\, \mathbf{H}_h, \mathbf{v})_K = 0, \ \forall \mathbf{v} \in \mathbf{V}^p(K), \\ (\mathrm{i}\omega\mu_r \mathbf{H}_h, \mathbf{v})_K + (\mathbf{curl}\, \mathbf{E}_h, \mathbf{v})_K = 0, \ \forall \mathbf{v} \in \mathbf{V}^p(K). \end{cases} \tag{5}$$

Applying appropriate Green's formulas and replacing the boundary terms by *numerical traces* $\widehat{\mathbf{E}}_h$ and $\widehat{\mathbf{H}}_h$, we obtain

$$\begin{cases} (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h, \mathbf{v})_K - (\mathbf{H}_h, \mathbf{curl}\, \mathbf{v})_K + \langle \widehat{\mathbf{H}}_h, \mathbf{n} \times \mathbf{v} \rangle_{\partial K} = 0, \ \forall \mathbf{v} \in \mathbf{V}^p(K), \\ (\mathrm{i}\omega\mu_r \mathbf{H}_h, \mathbf{v})_K + (\mathbf{E}_h, \mathbf{curl}\, \mathbf{v})_K - \langle \widehat{\mathbf{E}}_h, \mathbf{n} \times \mathbf{v} \rangle_{\partial K} = 0, \ \forall \mathbf{v} \in \mathbf{V}^p(K). \end{cases} \tag{6}$$

It is straightforward to verify that $\mathbf{n} \times \mathbf{v} = \mathbf{n} \times \mathbf{v}^t$ therefore we can rewrite (6) using numerical traces defined in terms of the tangential components $\widehat{\mathbf{E}}_h^t$ and $\widehat{\mathbf{H}}_h^t$

$$\begin{cases} (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h, \mathbf{v})_K - (\mathbf{H}_h, \mathbf{curl}\, \mathbf{v})_K + \langle \widehat{\mathbf{H}}_h^t, \mathbf{n} \times \mathbf{v} \rangle_{\partial K} = 0, \ \forall \mathbf{v} \in \mathbf{V}^p(K), \\ (\mathrm{i}\omega\mu_r \mathbf{H}_h, \mathbf{v})_K + (\mathbf{E}_h, \mathbf{curl}\, \mathbf{v})_K - \langle \widehat{\mathbf{E}}_h^t, \mathbf{n} \times \mathbf{v} \rangle_{\partial K} = 0, \ \forall \mathbf{v} \in \mathbf{V}^p(K). \end{cases} \tag{7}$$

In a classical DG method, we directly formulate the numerical traces to couple the traces of the local fields $\mathbf{E}_h$ and $\mathbf{H}_h$ between the neighboring elements

$$\widehat{\mathbf{H}}_h = \{\mathbf{H}_h\} + \alpha_E [\![\mathbf{E}_h]\!], \quad \widehat{\mathbf{E}}_h = \{\mathbf{E}_h\} + \alpha_H [\![\mathbf{H}_h]\!], \tag{8}$$

where α_E and α_H are positive penalty parameters. In the setting of a HDG method, we introduce a *hybrid variable* $\mathbf{\Lambda}_h$ that we define here as

$$\mathbf{\Lambda}_h := \widehat{\mathbf{H}}_h^t, \quad \forall F \in \mathcal{F}_h. \tag{9}$$

We now want to formulate the local fields in K through (7) assuming that $\mathbf{\Lambda}_h$ is known on all the faces of an element K. In order to achieve this, we consider a numerical trace $\widehat{\mathbf{E}}_h^t$ of the form

$$\widehat{\mathbf{E}}_h^t = \mathbf{E}_h^t + \tau_K \mathbf{n} \times (\mathbf{\Lambda}_h - \mathbf{H}_h^t) \text{ on } \partial K, \tag{10}$$

where τ_K is a local stabilization parameter. Note that we have used the fact that $\mathbf{n} \times \mathbf{H}_h^t = \mathbf{n} \times \mathbf{H}_h$. Adding the contributions of (7) over all elements and enforcing the continuity of the tangential component of $\widehat{\mathbf{E}}_h$, we can formulate a problem which is to find $(\mathbf{E}_h, \mathbf{H}_h, \mathbf{\Lambda}_h) \in \mathbf{V}_h^p \times \mathbf{V}_h^p \times \mathbf{M}_h^p$ such that for all $\mathbf{v} \in \mathbf{V}_h^p$ and $\boldsymbol{\eta} \in \mathbf{M}_h^p$

$$\begin{cases} (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h, \mathbf{v})_{\mathcal{T}_h} - (\mathbf{H}_h, \mathbf{curl}\, \mathbf{v})_{\mathcal{T}_h} + \langle \mathbf{\Lambda}_h, \mathbf{n} \times \mathbf{v} \rangle_{\partial \mathcal{T}_h} = 0, \\ (\mathrm{i}\omega\mu_r \mathbf{H}_h, \mathbf{v})_{\mathcal{T}_h} + (\mathbf{E}_h, \mathbf{curl}\, \mathbf{v})_{\mathcal{T}_h} - \langle \widehat{\mathbf{E}}_h^t, \mathbf{n} \times \mathbf{v} \rangle_{\partial \mathcal{T}_h} = 0, \\ \langle [\![\widehat{\mathbf{E}}_h^t]\!], \boldsymbol{\eta} \rangle_{\mathcal{F}_h} - \langle \mathbf{\Lambda}_h, \boldsymbol{\eta} \rangle_{\Gamma_a} = \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta} \rangle_{\Gamma_a}, \end{cases} \tag{11}$$

where the last equation is called the *conservativity condition*. With the definition (10) of $\widehat{\mathbf{E}}_h^t$, we employ again a Green formula with the second equation of (11), in order to get the following formulation, for all $\mathbf{v} \in \mathbf{V}_h^p$ and $\boldsymbol{\eta} \in \mathbf{M}_h^p$

$$\begin{cases} (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h, \mathbf{v})_{\mathcal{T}_h} - (\mathbf{H}_h, \mathbf{curl}\,\mathbf{v})_{\mathcal{T}_h} + \langle \boldsymbol{\Lambda}_h, \mathbf{n}\times\mathbf{v}\rangle_{\partial\mathcal{T}_h} = 0, \\ (\mathrm{i}\omega\mu_r \mathbf{H}_h, \mathbf{v})_{\mathcal{T}_h} + (\mathbf{curl}\,\mathbf{E}_h, \mathbf{v})_{\mathcal{T}_h} + \langle \tau\mathbf{n}\times(\mathbf{H}_h - \boldsymbol{\Lambda}_h), \mathbf{n}\times\mathbf{v}\rangle_{\partial\mathcal{T}_h} = 0, \\ \langle \mathbf{n}\times\mathbf{E}_h, \boldsymbol{\eta}\rangle_{\partial\mathcal{T}_h} + \langle \tau(\mathbf{H}_h^t - \boldsymbol{\Lambda}_h), \boldsymbol{\eta}\rangle_{\partial\mathcal{T}_h} - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a} = \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta}\rangle_{\Gamma_a}. \end{cases} \tag{12}$$

Note that we have used

$$\mathbf{n}\times\mathbf{v}^t = \mathbf{n}\times\mathbf{v} \text{ and } \mathbf{n}\times(\mathbf{n}\times\mathbf{v}^t) = -\mathbf{v}^t, \tag{13}$$

to obtain (12). Indeed, the last relation of (11) together with the definition (10) of $\widehat{\mathbf{E}}_h^t$ yields

$$\langle \mathbf{n}\times\mathbf{E}_h^t, \boldsymbol{\eta}\rangle_{\partial\mathcal{T}_h} + \langle \tau\mathbf{n}\times\left(\mathbf{n}\times(\mathbf{H}_h^t - \boldsymbol{\Lambda}_h)\right), \boldsymbol{\eta}\rangle_{\partial\mathcal{T}_h} - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a} = \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta}\rangle_{\Gamma_a},$$

which can be rewritten, using (13),

$$\langle \mathbf{n}\times\mathbf{E}_h, \boldsymbol{\eta}\rangle_{\partial\mathcal{T}_h} - \langle \tau(\boldsymbol{\Lambda}_h - \mathbf{H}_h^t), \boldsymbol{\eta}\rangle_{\partial\mathcal{T}_h} - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a} = \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta}\rangle_{\Gamma_a}.$$

Moreover, we note that the last relation of (11) for a boundary face F on Γ_a is equivalent to, using the fact that $[\![\widehat{\mathbf{E}}_h^t]\!] = \mathbf{n}\times\widehat{\mathbf{E}}_h^t$,

$$\langle \mathbf{n}\times\widehat{\mathbf{E}}_h^t, \boldsymbol{\eta}\rangle_F - \langle \widehat{\mathbf{H}}_h^t, \boldsymbol{\eta}\rangle_F = \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta}\rangle_F,$$

that is

$$\left\langle \left(\mathbf{n}\times\widehat{\mathbf{E}}_h + \mathbf{n}\times(\mathbf{n}\times\widehat{\mathbf{H}}_h)\right), \boldsymbol{\eta}\right\rangle_F = \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta}\rangle_F,$$

which is nothing else than the Silver-Müller boundary condition in (1) with the numerical traces $\widehat{\mathbf{E}}_h$ and $\widehat{\mathbf{H}}_h$.

Remark 3.1 *The equivalent numerical scheme for the Laplacian is called LDG-H [CGL09]. It also coincides with the scheme called HDGII in [NPC11].*

3.2 Characterization of the reduced problem

For any $\boldsymbol{\eta} \in \mathbf{M}_h$, we denote by $(\mathbf{E}_h^{\boldsymbol{\eta}}, \mathbf{H}_h^{\boldsymbol{\eta}})$ the approximate solutions whose restriction to an element K of $\mathcal{T}_h$ is the solution to the local problem given by, for all $\mathbf{v} \in \mathbf{V}_h^p(K)$

$$\begin{cases} (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h^{\boldsymbol{\eta}}, \mathbf{v})_K - (\mathbf{H}_h^{\boldsymbol{\eta}}, \mathbf{curl}\,\mathbf{v})_K + \langle \boldsymbol{\eta}, \mathbf{n}\times\mathbf{v}\rangle_{\partial K} = 0, \\ (\mathrm{i}\omega\mu_r \mathbf{H}_h^{\boldsymbol{\eta}}, \mathbf{v})_K + (\mathbf{curl}\,\mathbf{E}_h^{\boldsymbol{\eta}}, \mathbf{v})_K + \langle \tau\mathbf{n}\times(\mathbf{H}_h^{\boldsymbol{\eta}} - \boldsymbol{\eta}), \mathbf{n}\times\mathbf{v}\rangle_{\partial K} = 0, \end{cases} \tag{14}$$

We rewrite the reduced problem as

$$a_h(\boldsymbol{\Lambda}_h, \boldsymbol{\eta}) = b_h(\boldsymbol{\eta}),\ \forall \boldsymbol{\eta} \in \mathbf{M}_h^p, \tag{15}$$

with

$$\begin{cases} a_h(\boldsymbol{\Lambda}_h, \boldsymbol{\eta}) = \langle [\![\mathbf{n}\times\widehat{\mathbf{E}}^{\boldsymbol{\Lambda}_h}]\!], \eta\rangle_{\mathcal{F}_h} - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a}, \\ b_h(\boldsymbol{\eta}) = \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta}\rangle_{\Gamma_a}. \end{cases}$$

In order to give an explicit formulation for $a_h(\boldsymbol{\Lambda}_h, \boldsymbol{\eta})$ and analyze its properties, we follow the approach adopted in 2d [LLP11a]. Summing the contributions of (14) over all the elements of $\mathcal{T}_h$, we obtain the following formulations by recalling the definition of $[\![\cdot]\!]$

$$\begin{cases} \overline{(\mathrm{i}\omega\varepsilon_r \mathbf{E}_h^{\boldsymbol{\eta}}, \mathbf{v})_{\mathcal{T}_h}} - \overline{(\mathbf{H}_h^{\boldsymbol{\eta}}, \mathbf{curl}\,\mathbf{v})_{\mathcal{T}_h}} + \overline{\langle \boldsymbol{\eta}, [\![\mathbf{n}\times\mathbf{v}]\!]\rangle_{\mathcal{F}_h}} = 0, \\ (\mathrm{i}\omega\mu_r \mathbf{H}_h^{\boldsymbol{\eta}}, \mathbf{v})_{\mathcal{T}_h} + (\mathbf{curl}\,\mathbf{E}_h^{\boldsymbol{\eta}}, \mathbf{v})_{\mathcal{T}_h} + \langle [\![\tau\mathbf{n}\times(\mathbf{n}\times(\boldsymbol{\eta}-\mathbf{H}_h^{\boldsymbol{\eta}}))]\!], \mathbf{v}\rangle_{\mathcal{F}_h} = 0. \end{cases} \tag{16}$$

Note that the first relation of (16) is obtained by taking the summation in the complex conjugation of the first relation of (14). The sesquilinear form in (15) can now be obtained

$$\begin{aligned} a_h(\boldsymbol{\Lambda}_h, \boldsymbol{\eta}) &= \langle [\![\mathbf{n}\times\widehat{\mathbf{E}}_h^{\boldsymbol{\Lambda}_h}]\!], \boldsymbol{\eta}\rangle_{\mathcal{F}_h} - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a} \\ &= \langle [\![\mathbf{n}\times\mathbf{E}_h^{\boldsymbol{\Lambda}_h}]\!], \boldsymbol{\eta}\rangle_{\mathcal{F}_h} + \langle [\![\mathbf{n}\times(\widehat{\mathbf{E}}_h^{\boldsymbol{\Lambda}_h} - \mathbf{E}_h^{\boldsymbol{\Lambda}_h})]\!], \boldsymbol{\eta}\rangle_{\mathcal{F}_h} - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a} \\ &= (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h^{\boldsymbol{\Lambda}_h}, \mathbf{E}_h^{\boldsymbol{\eta}})_{\mathcal{T}_h} + (\mathbf{curl}\,\mathbf{E}_h^{\boldsymbol{\Lambda}_h}, \mathbf{H}_h^{\boldsymbol{\eta}})_{\mathcal{T}_h} + \langle [\![\mathbf{n}\times(\widehat{\mathbf{E}}_h^{\boldsymbol{\Lambda}_h} - \mathbf{E}_h^{\boldsymbol{\Lambda}_h})]\!], \boldsymbol{\eta}\rangle_{\mathcal{F}_h} \\ &\qquad - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a}, \end{aligned}$$

by the first relation of (16), taking $\mathbf{v} = \mathbf{E}_h^{\boldsymbol{\Lambda}_h}$, and

$$\begin{aligned} a_h(\boldsymbol{\Lambda}_h, \boldsymbol{\eta}) = (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h^{\boldsymbol{\Lambda}_h}, \mathbf{E}_h^{\boldsymbol{\eta}})_{\mathcal{T}_h} &- (\mathrm{i}\omega\mu_r \mathbf{H}_h^{\boldsymbol{\Lambda}_h}, \mathbf{H}_h^{\boldsymbol{\eta}})_{\mathcal{T}_h} \\ &+ \langle [\![\tau\mathbf{n}\times(\mathbf{n}\times(\boldsymbol{\Lambda} - \mathbf{H}_h^{\boldsymbol{\Lambda}_h}))]\!], (\boldsymbol{\eta} - \mathbf{H}_h^{\boldsymbol{\eta}})\rangle_{\mathcal{F}_h} \\ &- \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a} \end{aligned}$$

by the second relation of (16) and considering the definition (10) of $\widehat{\mathbf{E}}_h$. Finally we obtain

$$\begin{aligned} a_h(\boldsymbol{\Lambda}_h, \boldsymbol{\eta}) = (\mathrm{i}\omega\varepsilon_r \mathbf{E}_h^{\boldsymbol{\Lambda}_h}, \mathbf{E}_h^{\boldsymbol{\eta}})_{\mathcal{T}_h} &- (\mathrm{i}\omega\mu_r \mathbf{H}_h^{\boldsymbol{\Lambda}_h}, \mathbf{H}_h^{\boldsymbol{\eta}})_{\mathcal{T}_h} \\ &- \langle \tau\mathbf{n}\times(\boldsymbol{\Lambda}_h - \mathbf{H}_h^{\boldsymbol{\Lambda}_h}), \mathbf{n}\times(\boldsymbol{\eta} - \mathbf{H}_h^{\boldsymbol{\eta}})\rangle_{\partial\mathcal{T}_h} - \langle \boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a}. \end{aligned} \tag{17}$$

When ε_r and μ_r are real-valued, we can infer that the corresponding coefficient matrix K is complex-valued symmetric and all the eigenvalues lie in the left half-plane of the complex plane, because of the third and fourth terms in (17). Moreover, the first two terms define the imaginary part of K if ε_r, μ_r and τ are real-valued. Note that K can be straightforwardly computed in a classical finite-element way. This matrix K has usual characteristics of the discretization of the wave equation: it is symmetric but indefinite as soon as ω is sufficiently large. A probably more classical formulation is obtained if we multiply (17) by $\mathrm{i}\omega$ in order to obtain

$$\begin{aligned} \mathrm{i}\omega a_h(\boldsymbol{\Lambda}_h, \boldsymbol{\eta}) = (\mu_r(-\mathrm{i}\omega\mathbf{H}_h^{\boldsymbol{\Lambda}_h}), (-\mathrm{i}\omega\mathbf{H}_h^{\boldsymbol{\eta}}))_{\mathcal{T}_h} &- \omega^2(\varepsilon_r\mathbf{E}_h^{\boldsymbol{\Lambda}_h}, \mathbf{E}_h^{\boldsymbol{\eta}})_{\mathcal{T}_h} \\ &- \mathrm{i}\omega\langle\tau(\boldsymbol{\Lambda}_h - \mathbf{H}_h^{\boldsymbol{\Lambda}_h}), (\boldsymbol{\eta} - \mathbf{H}_h^{\boldsymbol{\eta}})\rangle_{\partial\mathcal{T}_h} - \mathrm{i}\omega\langle\boldsymbol{\Lambda}_h, \boldsymbol{\eta}\rangle_{\Gamma_a}. \end{aligned}$$

3.3 About the numerical fluxes

The conservativity condition holds on all the interior faces, which means $\langle [\![\widehat{\mathbf{E}}_h^t]\!], \boldsymbol{\eta}\rangle_{\mathcal{F}_h^I} = 0$. From the choice of approximation spaces and if the approximation order p is uniform, we can infer that $[\![\widehat{\mathbf{E}}_h^t]\!] = 0$ on every interior face of a conforming mesh. Substituting $\widehat{\mathbf{E}}_h$ with the expression (10), we have

$$\begin{aligned} [\![\mathbf{E}_h^t + \tau\mathbf{n}\times(\boldsymbol{\Lambda}_h - \mathbf{H}_h^t)]\!] &= [\![\mathbf{E}_h^t]\!] + \tau_{K^+}\mathbf{H}_h^{t+} + \tau_{K^-}\mathbf{H}_h^{t-} - (\tau_{K^+} + \tau_{K^-})\boldsymbol{\Lambda}_h \\ &= 0 \text{ on } \mathcal{F}_h^I. \end{aligned}$$

Solving for $\mathbf{\Lambda}_h$, we obtain (assuming $\tau_{K^+} + \tau_{K^-} \neq 0$)

$$\widehat{\mathbf{H}}_h^t = \mathbf{\Lambda}_h = \frac{1}{\tau_{K^+} + \tau_{K^-}}(\tau_{K^+}\mathbf{H}_h^{t+} + \tau_{K^-}\mathbf{H}_h^{t-}) + \frac{1}{\tau_{K^+} + \tau_{K^-}}[\![\mathbf{E}_h^t]\!] \text{ on } F. \tag{18}$$

Substituting (18) into (10), and taking the tangential components on both sides, we have

$$\begin{aligned}\widehat{\mathbf{E}}_h^t = \widehat{\mathbf{E}}_h^{t+} = \widehat{\mathbf{E}}_h^{t-} =& \frac{1}{\tau_{K^+} + \tau_{K^-}}(\tau_{K^-}\mathbf{E}_h^{t+} + \tau_{K^+}\mathbf{E}_h^{t-}) \\ &+ \frac{\tau_{K^+}\tau_{K^-}}{\tau_{K^+} + \tau_{K^-}}[\![\mathbf{H}_h^t]\!] \text{ on } F.\end{aligned} \tag{19}$$

Thus, the tangential components of the numerical traces for both $\widehat{\mathbf{E}}_h$ and $\widehat{\mathbf{H}}_h$ fields are single-valued.

Remark 3.2 *If we take τ uniformly equal to 1, the numerical traces coincide with the upwind flux-based DG formulation considered in [DFLP08]. Moreover, equipped only with Silver-Müller boundary conditions, both approaches provide the same discrete solution. For a more general upwind flux-based DG formulation, i.e. adapted to inhomogeneous media as proposed in [HW02], the numerical traces coincide with the HDG method by the simple choice $\tau = Z_r$.*

Assuming the interpolation order is p on every element, we obtain the number of the globally coupled degrees of freedom

$$\begin{aligned}\textit{Upwind flux-based DG formulation} &: \quad (p+1)(p+2)(p+3)N_e, \\ \textit{HDG formulation} &: \quad (p+1)(p+2)N_f,\end{aligned}$$

with N_e be the number of elements and N_f the number of faces. For a simplicial mesh, $N_f \approx 2N_e$, thus the ratio of the globally coupled degrees of freedom of HDG method over DG method is roughly $2/(p+3)$.

4 Discretization of the multi-domain problem

In this section we formulate a HDG-based discrete variant of the Schwarz domain decomposition algorithm introduced in section 2.3. We first make some preliminary remarks concerning the definition of the interface (or transmission) conditions.

4.1 Preamble

As a preliminary remark, note that an objective of the HDG scheme is that the hybrid variable and the numerical trace satisfy on each interior face of the computational domain the following conditions

$$[\![\mathbf{n} \times \widehat{\mathbf{E}}_h]\!] = 0 \quad \text{and} \quad [\![[\![\mathbf{\Lambda}_h]\!]]\!] = 0. \tag{20}$$

Recall that $\mathbf{n} \times \widehat{\mathbf{E}}_h = \mathbf{n} \times \widehat{\mathbf{E}}_h^t$ and $\widehat{\mathbf{E}}_h^t = -\mathbf{n} \times (\mathbf{n} \times \widehat{\mathbf{E}}_h)$. In a given subdomain Ω_i, the hybrid variable $\mathbf{\Lambda}_h^{(i)}$ is by definition single-valued on each face and thus for any face in the interior of the subdomain, the condition $[\![[\![\mathbf{\Lambda}_h^{(i)}]\!]]\!] = 0$ is automatically satisfied. However at the interface between two subdomains Ω_i and Ω_j, the hybrid variables is *a priori* double-valued and the condition $[\![[\![\mathbf{\Lambda}_h]\!]]\!] = \mathbf{\Lambda}_h^{(i)} - \mathbf{\Lambda}_h^{(j)} = 0$ has to be explicitly enforced.

Consequently, on any face of an interface between two subdomains Ω_1 and Ω_2, we have to enforce (20) but we can equivalently consider

$$[\![\mathbf{n} \times \widehat{\mathbf{E}}_h]\!] + z_1[\![[\![\mathbf{\Lambda}_h]\!]]\!] = 0 \quad \text{and} \quad [\![\mathbf{n} \times \widehat{\mathbf{E}}_h]\!] + z_2[\![[\![\mathbf{\Lambda}_h]\!]]\!] = 0, \tag{21}$$

with z_1 and z_2 such that $z_1 - z_2 \neq 0$ (more complex conditions could also be considered especially in the framework of optimized Schwarz [DLP08b], but the only important thing is that (21) is *equivalent* to (20) on the interface between subdomains). In the case of the Schwarz algorithm (3) based on the transmission operator in terms of the impedance condition (4), and using the definitions $\widehat{\mathbf{E}}_h$ and $\widehat{\mathbf{H}}_h^t$ (and again the fact that $\mathbf{n} \times \widehat{\mathbf{E}}_h^t = \mathbf{n} \times \widehat{\mathbf{E}}_h$), we want to enforce

$$\left\{ \begin{array}{r} \mathbf{n}_{12} \times (\mathbf{E}_h^{(1)} + \tau \mathbf{n}_{12} \times (\mathbf{\Lambda}_h^{(1)} - \mathbf{H}_h^{t,(1)})) - Z_r^{(1)} \mathbf{\Lambda}_h^{(1)} = \\ \mathbf{n}_{12} \times (\mathbf{E}_h^{(2)} + \tau \mathbf{n}_{21} \times (\mathbf{\Lambda}_h^{(2)} - \mathbf{H}_h^{t,(2)})) - Z_r^{(1)} \mathbf{\Lambda}_h^{(2)}, \\ \mathbf{n}_{21} \times (\mathbf{E}_h^{(2)} + \tau \mathbf{n}_{21} \times (\mathbf{\Lambda}_h^{(2)} - \mathbf{H}_h^{t,(2)})) - Z_r^{(2)} \mathbf{\Lambda}_h^{(2)} = \\ \mathbf{n}_{21} \times (\mathbf{E}_h^{(1)} + \tau \mathbf{n}_{12} \times (\mathbf{\Lambda}_h^{(1)} - \mathbf{H}_h^{t,(1)})) - Z_r^{(2)} \mathbf{\Lambda}_h^{(1)}, \end{array} \right. \tag{22}$$

or after simplifications

$$\left\{ \begin{array}{r} \mathbf{n}_{12} \times \mathbf{E}_h^{(1)} + \tau(\mathbf{H}_h^{t,(1)} - \mathbf{\Lambda}_h^{(1)}) - Z_r^{(1)} \mathbf{\Lambda}_h^{(1)} = \\ \mathbf{n}_{12} \times \mathbf{E}_h^{(2)} - \tau(\mathbf{H}_h^{t,(2)} - \mathbf{\Lambda}_h^{(2)}) - Z_r^{(1)} \mathbf{\Lambda}_h^{(2)}, \\ \mathbf{n}_{21} \times \mathbf{E}_h^{(2)} + \tau(\mathbf{H}_h^{t,(2)} - \mathbf{\Lambda}_h^{(2)}) - Z_r^{(2)} \mathbf{\Lambda}_h^{(2)} = \\ \mathbf{n}_{21} \times \mathbf{E}_h^{(1)} - \tau(\mathbf{H}_h^{t,(1)} - \mathbf{\Lambda}_h^{(1)}) - Z_r^{(2)} \mathbf{\Lambda}_h^{(1)}. \end{array} \right. \tag{23}$$

If we choose as definition $[\![[\mathbf{\Lambda}_h]\!] = \mathbf{\Lambda}_h^{(1)} - \mathbf{\Lambda}_h^{(2)}$ and remark that $\mathbf{n}_{12} = -\mathbf{n}_{21}$, the conditions (23) are equivalent to

$$[\![[\mathbf{n} \times \widehat{\mathbf{E}}_h]\!] - Z_r^{(1)} [\![[\mathbf{\Lambda}_h]\!] = 0 \quad \text{and} \quad [\![[\mathbf{n} \times \widehat{\mathbf{E}}_h]\!] + Z_r^{(2)} [\![[\mathbf{\Lambda}_h]\!] = 0. \tag{24}$$

It is, as soon as $Z_r^{(1)} + Z_r^{(2)} \neq 0$, equivalent to (20) on the interface between subdomains Ω_1 and Ω_2. In the Schwarz iterative algorithm at iteration $(n+1)$, the transmission conditions will translate as

$$\left\{ \begin{array}{r} \mathbf{n}_{12} \times \mathbf{E}_h^{(1),n+1} + \tau(\mathbf{H}_h^{t,(1),n+1} - \mathbf{\Lambda}_h^{(1),n+1}) - Z_r^{(1)} \mathbf{\Lambda}_h^{(1),n+1} = \\ \mathbf{n}_{12} \times \mathbf{E}_h^{(2),n} - \tau(\mathbf{H}_h^{t,(2),n} - \mathbf{\Lambda}_h^{(2),n}) - Z_r^{(1)} \mathbf{\Lambda}_h^{(2),n}, \\ \mathbf{n}_{21} \times \mathbf{E}_h^{(2),n+1} + \tau(\mathbf{H}_h^{t,(2),n+1} - \mathbf{\Lambda}_h^{(2),n+1}) - Z_r^{(2)} \mathbf{\Lambda}_h^{(2),n+1} = \\ \mathbf{n}_{21} \times \mathbf{E}_h^{(1),n} - \tau(\mathbf{H}_h^{t,(1),n} - \mathbf{\Lambda}_h^{(1),n}) - Z_r^{(2)} \mathbf{\Lambda}_h^{(1),n}. \end{array} \right. \tag{25}$$

Moreover, in the HDG scheme, interface conditions are explicitly enforced independently of the volume condition which is more convenient for the formulation of the domain decomposition algorithm in the discrete case compared to a classical DG scheme.

4.2 HDG 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 related to the subdomain Ω_i. Thus, $\mathcal{T}_h^{(i)}$, $\mathbf{V}_h^{(i)}$, $\partial\mathcal{T}_h^{(i)}$ and $\mathbf{M}_h^{(i)}$ can be defined from those of $\mathcal{T}_h$, $\mathbf{V}_h$, $\partial\mathcal{T}_h$ and $\mathbf{M}_h$. We also define $\Gamma_m^{(i)} = \Gamma_m \cap \partial\Omega_i$ and $\Gamma_a^{(i)} = \Gamma_a \cap \partial\Omega_i$. Finally, $\Gamma_{i,j}$ will denote the interface $\partial\Omega_i \cap \overline{\Omega_j}$.

Discretization of the Schwarz algorithm (3) by the HDG scheme (12) integrates conditions (25) and leads to the following problem which is to find $(\mathbf{E}_h^{(i),n+1}, \mathbf{H}_h^{(i),n+1}, \mathbf{\Lambda}_h^{(i),n+1})$ $(n = 1, 2, \cdots)$

until convergence, satisfying for all $\mathbf{v} \in \mathbf{V}_h^{(i)}$ and $\boldsymbol{\eta} \in \mathbf{M}_h^{(i)}$,

$$
\left\{
\begin{array}{rcl}
(\mathrm{i}\omega\varepsilon_r \mathbf{E}_h^{(i),n+1}, \mathbf{v})_{\mathcal{T}_h^{(i)}} & - & (\mathbf{H}_h^{(i),n+1}, \mathbf{curl}\, \mathbf{v})_{\mathcal{T}_h^{(i)}} \\
& + & \langle \mathbf{\Lambda}_h^{(i),n+1}, \mathbf{n} \times \mathbf{v} \rangle_{\partial \mathcal{T}_h^{(i)}} = 0, \\
(\mathrm{i}\omega\mu_r \mathbf{H}_h^{(i),n+1}, \mathbf{v})_{\mathcal{T}_h^{(i)}} & + & (\mathbf{curl}\, \mathbf{E}_h^{(i),n+1}, \mathbf{v})_{\mathcal{T}_h^{(i)}} \\
& + & \langle \tau \mathbf{n} \times (\mathbf{H}_h^{(i),n+1} - \mathbf{\Lambda}_h^{(i),n+1}), \mathbf{n} \times \mathbf{v} \rangle_{\partial \mathcal{T}_h^{(i)}} \\
& = & 0, \\
\langle \mathbf{n} \times \mathbf{E}_h^{(i),n+1}, \boldsymbol{\eta} \rangle_{\partial \mathcal{T}_h^{(i)}} & + & \langle \tau \left(\mathbf{H}_h^{t,(i),n+1} - \mathbf{\Lambda}_h^{(i),n+1} \right), \boldsymbol{\eta} \rangle_{\partial \mathcal{T}_h^{(i)}} \\
& - & \langle \mathbf{\Lambda}_h^{(i),n+1}, \boldsymbol{\eta} \rangle_{\Gamma_a^{(i)}} \\
& - & \sum_j \langle Z_r^{(i)} \mathbf{\Lambda}_h^{(i),n+1}, \boldsymbol{\eta} \rangle_{\Gamma_{i,j}} = \\
\sum_j \langle \mathbf{n} \times \mathbf{E}_h^{(j),n}, \boldsymbol{\eta} \rangle_{\Gamma_{i,j}} & - & \sum_j \langle \tau \left(\mathbf{H}_h^{t,(j),n} - \mathbf{\Lambda}_h^{(j),n} \right), \boldsymbol{\eta} \rangle_{\Gamma_{i,j}} \\
& + & \langle \mathbf{g}^{\mathrm{inc}}, \boldsymbol{\eta} \rangle_{\Gamma_a^{(i)}} - \sum_j \langle Z_r^{(i)} \mathbf{\Lambda}_h^{(j),n}, \boldsymbol{\eta} \rangle_{\Gamma_{i,j}}.
\end{array}
\right. \tag{26}
$$

For simplicity, we have dropped the superscript p which represents the interpolation order.

4.3 Formulation of the interface system

For sake of simplicity we consider a problem with two domains Ω_1 and Ω_2. The reduced linear system for the solution on the whole domain Ω, issued from the sesquilinear form (17), can take the following form

$$
\begin{pmatrix}
K_{ii}^{(1)} & 0 & K_{ig}^{(1)} \\
0 & K_{ii}^{(2)} & K_{ig}^{(2)} \\
K_{gi}^{(1)} & K_{gi}^{(2)} & K_{gg}^{(1)} + K_{gg}^{(2)}
\end{pmatrix}
\begin{pmatrix}
\Lambda_{h,i}^{(1)} \\ \Lambda_{h,i}^{(2)} \\ \Lambda_{h,g}
\end{pmatrix}
=
\begin{pmatrix}
b_{h,i}^{(1)} \\ b_{h,i}^{(2)} \\ b_{h,g}^{(1)} + b_{h,g}^{(2)}
\end{pmatrix},
\tag{27}
$$

where

$$
\begin{pmatrix}
K_{ii}^{(l)} & K_{ig}^{(l)} \\
K_{gi}^{(l)} & K_{gg}^{(l)}
\end{pmatrix},
$$

denotes the matrix assembled on Ω_l for l in $\{1, 2\}$. The subscript g indicates that the concerned DOFs are on $\Gamma_{1,2}$, and the subscript i that DOFs are in Ω_1 or Ω_2 but not on $\Gamma_{1,2}$. Following the lines of the work [MRS04], we can deduce the following proposition.

Proposition 4.1 *Let $A^{(1)}$ and $A^{(2)}$ be any matrices such that $A^{(1)} + A^{(2)}$ is invertible. There*

is only one pair of Lagrange multipliers $(\Sigma^{(1)}, \Sigma^{(2)})$ *such that the coupled problem*

$$\begin{pmatrix} K_{ii}^{(1)} & K_{ig}^{(1)} \\ K_{gi}^{(1)} & K_{gg}^{(1)} + A^{(1)} \end{pmatrix} \begin{pmatrix} \Lambda_{h,i}^{(1)} \\ \Lambda_{h,g}^{(1)} \end{pmatrix} = \begin{pmatrix} b_{h,i}^{(1)} \\ b_{h,g}^{(1)} + \Sigma^{(1)} \end{pmatrix},$$

$$\begin{pmatrix} K_{ii}^{(2)} & K_{ig}^{(2)} \\ K_{gi}^{(2)} & K_{gg}^{(2)} + A^{(2)} \end{pmatrix} \begin{pmatrix} \Lambda_{h,i}^{(2)} \\ \Lambda_{h,g}^{(2)} \end{pmatrix} = \begin{pmatrix} b_{h,i}^{(2)} \\ b_{h,g}^{(2)} + \Sigma^{(2)} \end{pmatrix}, \tag{28}$$

$$\begin{aligned} \Sigma^{(1)} + \Sigma^{(2)} - (A^{(1)} + A^{(2)})\Lambda_{h,g}^{(1)} = 0, \\ \Sigma^{(1)} + \Sigma^{(2)} - (A^{(1)} + A^{(2)})\Lambda_{h,g}^{(2)} = 0, \end{aligned}$$

is equivalent to problem (27).

From (28), we are led to solve the following linear system for the interface variables $\Sigma^{(1)}$ and $\Sigma^{(2)}$

$$\begin{pmatrix} I & I - A^{(12)}(S^{(2)} + A^{(2)})^{-1} \\ I - A^{(12)}(S^{(1)} + A^{(1)})^{-1} & I \end{pmatrix} \begin{pmatrix} \Sigma^{(1)} \\ \Sigma^{(2)} \end{pmatrix} = \begin{pmatrix} A^{(12)}(S^{(2)} + A^{(2)})^{-1} c_{h,g}^{(2)} \\ A^{(12)}(S^{(1)} + A^{(1)})^{-1} c_{h,g}^{(1)} \end{pmatrix}, \tag{29}$$

with $A^{(12)} = A^{(1)} + A^{(2)}$ and where

$$S^{(l)} = K_{gg}^{(l)} - K_{gi}^{(l)}(K_{ii}^{(l)})^{-1}K_{ig}^{(l)} \text{ and } c_{h,g}^{(l)} = b_{h,g}^{(l)} - K_{gi}^{(l)}(K_{ii}^{(l)})^{-1}b_{h,i}^{(l)}, \quad l \in \{1, 2\}.$$

In the present case, the matrix $A^{(l)}$ is assembled by computing the following term

$$(A^{(l)})_{i,j} = -Z_r^{(l)} \langle \boldsymbol{\eta}_j, \boldsymbol{\eta}_i \rangle_{\Gamma_{1,2}},$$

where $(\boldsymbol{\eta}_i)_i$ is a basis of $\mathbf{M}_h^p$ restricted to $\Gamma_{1,2}$. Thus $A^{(1)} + A^{(2)}$ is invertible if and only if $Z_r^{(1)} + Z_r^{(2)} \neq 0$, which is obviously the case here.

The interface system (29) is then solved by a Krylov subspace method which is discussed in the following section.

4.4 Algorithmic aspects

4.4.1 Matrix-vector product

As mentioned above, one has to solve the interface system (29) by a Krylov subspace method. For this, we need to know how to efficiently perform the following matrix-vector product

$$\begin{pmatrix} I & I - A^{(12)}(S^{(2)} + A^{(2)})^{-1} \\ I - A^{(12)}(S^{(1)} + A^{(1)})^{-1} & I \end{pmatrix} \begin{pmatrix} \Sigma^{(1)} \\ \Sigma^{(2)} \end{pmatrix} = \begin{pmatrix} W^{(1)} \\ W^{(2)} \end{pmatrix}.$$

In practice, this could be done as follows

1. For $l = 1, 2$, *solve independently*

$$\begin{pmatrix} K_{ii}^{(l)} & K_{ig}^{(l)} \\ K_{gi}^{(l)} & K_{gg}^{(l)} + A^{(l)} \end{pmatrix} \begin{pmatrix} \Lambda_{h,i}^{(l)} \\ \Lambda_{h,g}^{(l)} \end{pmatrix} = \begin{pmatrix} 0 \\ \Sigma^{(l)} \end{pmatrix}.$$

Note that these systems are solved *only* for obtaining the hybrid variable in each subdomain.

2. For $l = 1, 2$, set $Y^{(l)} = \Lambda_{h,g}^{(l)}$ and *compute independently* $X^{(l)} = \Sigma^{(l)} - A^{(l)}Y^{(l)}$.
3. *Communicate* $X^{(1)}$, $Y^{(1)}$ to subdomain 2 and $X^{(2)}$, $Y^{(2)}$ to subdomain 1.
4. *Compute independently*

$$W^{(1)} = \Sigma^{(1)} + X^{(2)} - A^{(1)}Y^{(2)},$$
$$W^{(2)} = \Sigma^{(2)} + X^{(1)} - A^{(2)}Y^{(1)}.$$

4.4.2 Definition of the right-hand side

Before starting the Krylov subspace method, we need to define the right hand-side of the interface system (29). This can be achieved as follows

1. For $l = 1, 2$, *solve independently*

$$\begin{pmatrix} K_{ii}^{(l)} & K_{ig}^{(l)} \\ K_{gi}^{(l)} & K_{gg}^{(l)} + A^{(l)} \end{pmatrix} \begin{pmatrix} \Lambda_{h,i}^{(l)} \\ \Lambda_{h,g}^{(l)} \end{pmatrix} = \begin{pmatrix} b_{h,i}^{(l)} \\ b_{h,g}^{(l)} \end{pmatrix},$$

2. For $l = 1, 2$, set $Y^{(l)} = \Lambda_{h,g}^{(l)} = (S^{(l)} + A^{(l)})^{-1}(b_{h,g}^{(l)} - K_{gi}^{(l)}(K_{ii}^{(l)})^{-1}b_{h,i}^{(l)})$ and *compute independently* $X^{(l)} = A^{(l)}Y^{(l)}$.
3. *Communicate* $X^{(1)}$, $Y^{(1)}$ to subdomain 2 and $X^{(2)}$, $Y^{(2)}$ to subdomain 1.
4. *Compute independently* the part of the right hand-side F corresponding to each subdomain

$$F^{(1)} = X^{(2)} + A^{(1)}Y^{(2)},$$
$$F^{(2)} = X^{(1)} + A^{(2)}Y^{(1)}.$$

5 Numerical and performance results

In this section we present numerical examples with two objectives in mind: first, using a simple (model) problem, the proposed DD-HDG solution strategy is validated and its convergence properties are assessed; second, by considering more complex problems, we study the overall efficiency of the DD-HDG solution strategy and show how much one can gain in terms of CPU time and memory consumption with the new discretization. Comparisons are made with the DD-DG solution strategy presented in [DLP08a].

5.1 Solution strategies

The interface system (28) is solved by an unpreconditioned BiCGStab(l) [SF93] Krylov subspace method with $l = 6$. The threshold of the stopping criteria for the BiCGStab(l) iteration is set to be 10^{-6}. Each iteration of the Krylov subspace method requires several matrix-vector products with the interface matrix of system (28). As described in subsection 4.4.1. each matrix-vector product is translated into the solution of the subdomain problems. A mixed precision arithmetic method is employed for the solution to the discretized reduced system. A LU factorization is computed and stored by MUMPS [ADL00] (a multifrontal sparse direct solver) in single precision arithmetic (32 bit). With this factorization we can consider three local solution strategies

- DD-bicgl, a BiCGStab(l) (with $l = 1$) method which works on double precision vectors, preconditioned by the above LU factorization. The linear threshold is set to $\varepsilon_i = 10^{-6}$.

- DD-gmres, a GMRES(m) [SS86] (with $m = 10$) method which works on double precision vectors with the same stopping criteria as DD-bicgl, preconditioned by the above LU factorization.
- DD-itref, an iterative refinement procedure used to recover double precision arithmetic [DLP08a].

Otherwise stated, the DD-itref strategy is used by default. Moreover the numerical simulations have been performed using a Fortran 90 computer code implementing the HDG and upwind flux-based DG methods in the 3d case using a $\mathbb{P}_1$ approximation of the components of the electromagnetic field within each tetrahedron of the mesh $\mathcal{T}_h$. The parallel implementation of the DD-DG and DD-HDG solution strategies is based on a SPMD strategy combining a partitioning of the computational domain with a message passing programming based on the MPI standard. The adjacency graph associated to the tetrahedral mesh is partitioned using the MeTiS tool [KK99]. In the following tables, the number of subdomains in the domain partitioning is denoted by N_s which is also the number of parallel processes (i.e. each subdomain is mapped to one processing unit). Moreover, "#iter" refers to the required number of iterations of the Krylov method used for the solution of the interface system (28), "CPU (min/max)" gives the minimum and maximum values of the per-process CPU time and "REAL" is the total elapsed time. Otherwise stated, parallel simulations with the DD-DG and DD-HDG solution strategies have been performed on a cluster operated by the INRIA Sophia Antipolis - Méditerranée research center which consists of AMD Opteron 6174 nodes (48×2.2GHz cores per node and 128GB per node) interconnected by a high performance Infiniband network.

5.2 Plane wave propagation in vacuum

We first consider a model problem consisting in the propagation of a plane wave in the vacuum. The computational domain is chosen to be the unit cube $\Omega =]-0.5; 0.5[^3$ and the Silver Müller absorbing boundary condition is imposed on the whole $\partial\Omega$. The electromagnetic parameters ε_r and μ_r are set to be 1 everywhere, and the angular frequency is $\omega = 2\pi$. Finally, the penalty parameter τ is set to be 1.

A series of regular tetrahedral meshes are employed, which divide the unit cube into $(N_x - 1) \times (N_y - 1) \times (N_z - 1)$ little cubes, where N_x, N_y and N_z are the number of grid points on the edges of the unit cube, and then each little cube is divided into 6 tetrahedrons. Table 1 summarizes the error and convergence behaviors of both the HDG-$\mathbb{P}_1$ and HDG-$\mathbb{P}_2$ methods. For these results we have employed a Matlab implementation of the HDG-$\mathbb{P}_1$ and HDG-$\mathbb{P}_2$ methods with nodal Lagrange basis expansions on a tetrahedral element. We also note that these results were presented for the first time in [LLP11b] which was our very preliminary contribution related to the design of HDG methods for the solution of the 3d time-harmonic Maxwell equations. The corresponding simulations are run in sequential mode and the native sparse direct solver of Matlab is used. In Table 1, "Mesh size" denotes the edge length of the tetrahedrons on the edge of the unit cube (we set $N_x = N_y = N_z$ in the construction of the tetrahedral meshes). We observe that the asymptotic convergence orders of the approximate solutions for both $\mathbf{E}$ and $\mathbf{H}$ are optimal. We now switch to results obtained using the Fortran 90 computer code implementing the HDG and upwind flux-based DG methods combined to the Schwarz domain decomposition solution algorithm. Table 2 summarizes the characteristics of the two finer meshes used to test the performances of the domain decomposition solver. In Table 2, $N = N_x = N_y = N_z$ denotes the number of the equally distributed grid points on each edge of the unit cube; N_v, N_t and N_f respectively denote the number of vertices, tetrahedra and faces of the meshes. In Table 3, we compare the CPU time and memory consumption required for the storage of the LU factors using

Mesh size	$\|\mathbf{E}-\mathbf{E}_h\|_2$		$\|\mathbf{H}-\mathbf{H}_h\|_2$	
	Error	Order	Error	Order
	HDG-$\mathbb{P}_1$			
1/2	2.27e-01	-	2.35e-01	-
1/4	6.02e-02	1.9	6.68e-02	1.8
1/8	1.54e-02	2.0	1.78e-02	1.9
	HDG-$\mathbb{P}_2$			
1/2	3.13e-02	-	3.36e-02	-
1/4	4.00e-03	3.0	4.44e-03	2.9
1/8	4.93e-04	3.0	5.53e-04	3.0

Table 1: Propagation of a plane wave in vacuum: numerical convergence of the HDG-$\mathbb{P}_1$ and HDG-$\mathbb{P}_2$ methods.

Mesh	N	N_v	N_t	N_f
M1	17	4,913	24,576	50,688
M2	25	15,625	82,944	169,344

Table 2: Propagation of a plane wave in vacuum: characteristics of tetrahedral meshes.

the HDG-$\mathbb{P}_1$ method (denoted by HDG) and the upwind flux-based DG-$\mathbb{P}_1$ method (denoted by UF-DG). We observe notable gains resulting from the new HDG discretization in both CPU time and memory requirement. Thanks to the reduction of the number of degrees of freedom, with the HDG method we can deal with larger problems. In Table 4, we compare the performances

Mesh	N_s	CPU (min/max)		RAM (min/max)	
		HDG	UF-DG	HDG	UF-DG
M1	16	2.7s/2.8s	26.1s/26.7s	39MB/40MB	157MB/157MB
-	32	0.8s/0.8s	5.8s/5.9s	15MB/15MB	60MB/60MB
M2	32	7.3s/8.0s	108.7s/112.1s	81MB/85MB	329MB/329MB

Table 3: Propagation of a plane wave in vacuum: comparisons between DD-HDG and upwind flux-based DD-DG solution strategies based on computation times and memory requirement for storing the L and U factors.

of the Schwarz method with different discretization schemes. Using the HDG method and the upwind flux-based DG method we have identical accuracy of the approximate solutions on the same mesh with the same domain partitioning. Again, the HDG method outperforms the upwind flux-based DG method.

5.3 Scattering of a plane wave by a PEC cube

The test problem considered here consists in the scattering of a plane wave by a perfectly electrically conducting (PEC) cube of side length $C = 1/3$m centered at the origin. The artificial boundary on which the Silver-Müller absorbing condition applies is defined by a unitary cube centered at the origin. The frequency of the incident plane wave is $\omega = 6\pi$ and its polarization is such that

$$\mathbf{k} = \begin{pmatrix} k_x \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} 0 \\ E_y \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{H} = \begin{pmatrix} 0 \\ 0 \\ H_z \end{pmatrix}.$$

Mesh	N_s	Local solver	#iter	CPU (min/max) HDG	CPU (min/max) UF-DG	REAL HDG	REAL UF-DG
M1	16	DD-bicgl	7	38.9s/39.2s	103.2s/104.9s	40.3s	107.1s
	-	DD-gmres	7	20.5s/20.8s	53.9s/54.2s	21.4s	55.3s
	-	DD-itref	7	13.2s/13.5s	34.9s/35.3s	13.9s	36.2s
	32	DD-bicgl	9	24.4s/24.8s	54.3s/55.2s	25.6s	55.6s
	-	DD-gmres	9	13.0s/13.3s	28.5s/28.7s	13.7s	29.5s
	-	DD-itref	9	7.3s/7.6s	18.2s/18.4s	7.9s	19.0s
M2	32	DD-bicgl	9	89.8s/91.2s	261.5s/267.7s	94.0s	272.8s
	-	DD-gmres	9	47.2s/47.9s	137.0s/137.6s	49.5s	140.3s
	-	DD-itref	9	32.4s/33.8s	87.2s/88.4s	35.2s	90.4s

Table 4: Propagation of a plane wave in vacuum: comparisons between DD-HDG and upwind flux-based DD-DG solution strategies based on memory consumption and computing times.

We make us of a non-uniform tetrahedral mesh consisting of 179,311 vertices and 1,005,932 tetrahedra (see Figure 1 left for a view of the triangulation in a selected cut plane). Figure 1 right shows the contour lines of the real part of the E_y component. Performance figures are summarized in Tables 5 and 6. We note that the maximum RAM requirement for computing and storing the L and U factors in the case of the DD-HDG strategy is about 4.2 times lower than that of the DD-DG strategy for the 96 subdomain decomposition. However we also observe that, for both strategies, there is a rather large difference between the minimum and maximum RAM requirements (44.2 % increase for the DD-DG strategy and 33.6 % increase for the DD-HDG strategy) despite a perfectly balanced decomposition of the mesh elements using the MeTiS tool. When switching from 96 to 192 subdomains, the maximum RAM requirement of the DD-HDG strategy is reduced by a factor approaching 2.5 (while the maximum of the per subdomain factorization time decreases by a factor 3.4). Finally, an almost perfect parallel speedup is achieved for the DD-HDG strategy which is, in the case of 96 subdomains, 8.5 faster than the DD-DG strategy.

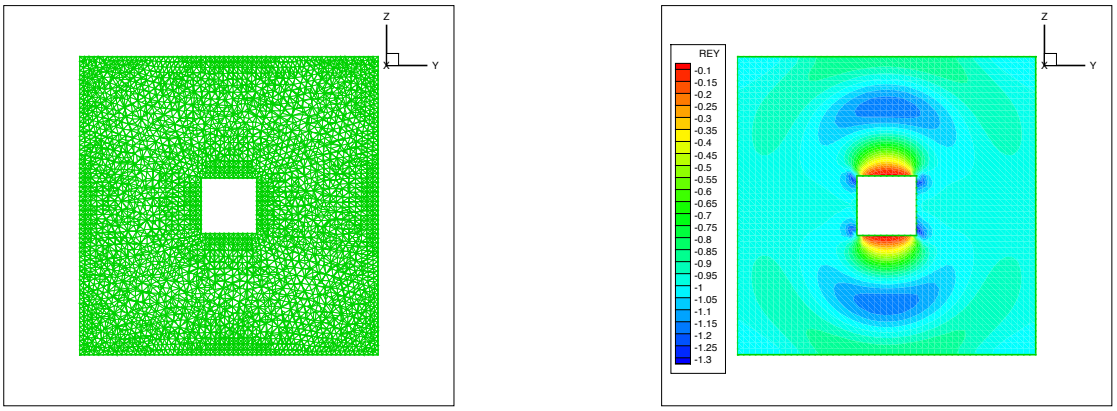


Figure 1: Scattering of a plane wave by a PEC cube. Discretization mesh (left) and contour lines of the real part of E_y (right) in a selected cut plane ($x = 0.5$).

Method	N_s	CPU (min/max)	RAM (min/max)
UF-DG	96	743s/746s	3616MB/5161MB
HDG	96	47s/48s	906MB/1211MB
-	192	13s/14s	349MB/488MB

Table 5: Scattering of a plane wave by a PEC cube. Computing times and memory requirement for storing the L and U factors.

Method	N_s	#iter	CPU (min/max)	TOTAL
UF-DG	96	14	1117s/1120s	1131s
HDG	96	15	128s/130s	133s
-	192	16	64s/65s	67s

Table 6: Scattering of a plane wave by a PEC cube. Computing times for the solution.

5.4 Scattering of a plane wave by a coated PEC cube

The test problem here is similar to these described in Subsection 5.3 except that the PEC cube is coated by a dielectric material. We make use of a uniform tetrahedral mesh consisting of 131,922 vertices and 744,000 tetrahedra. In order to model the coating of the cube we consider that a thickness of elements corresponds to a material with $\epsilon_r = 4.0$. Contour lines of the E_x and E_y components in a selected cut plane are visualized in Figure 2 (bottom figures). Performance figures are summarized in Tables 7 and 8. We observe a superlinear parallel speedup of the factorization phase when switching from 64 to 128 subdomains in the DD-HDG solution strategy. However this is not the case for the solution phase. Indeed a very poor acceleration is obtained which is explained here by an unfavorable computation to communication ratio in the case $N_s = 128$.

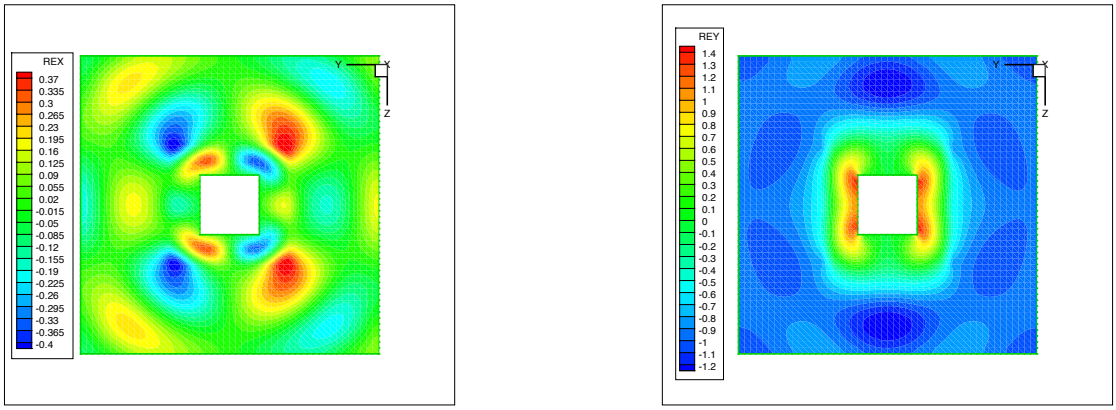


Figure 2: Scattering of a plane wave by a coated PEC cube. Contour lines of the real part of E_x (bottom left) and E_y (bottom right) in a selected cut plane ($x = 0.5$).

5.5 Scattering of a plane wave by an aircraft

The problem under consideration is the simulation of the scattering of a plane wave by an aircraft geometry. We consider several values of the angular frequency of the incident field: $\omega = 4/3\pi$,

Method	N_s	CPU (min/max)	RAM (min/max)
UF-DG	128	94s/144s	1115MB/1628MB
HDG	64	44s/71s	1126MB/1479MB
-	128	13s/19s	436MB/588MB

Table 7: Scattering of a plane wave by a coated PEC cube. Computing times and memory requirement for storing the L and U factors.

Method	N_s	#iter	CPU (min/max)	TOTAL
UF-DG	128	24	572s/643s	661s
HDG	64	21	249s/274s	284s
-	128	23	195s/216s	229s

Table 8: Scattering of a plane wave by a coated PEC cube. Computing times for the solution.

$\omega = 8/3\pi$ and $\omega = 16/3\pi$ (in the following we often refer to the corresponding frequency F with respective values 200, 400 and 800MHz). The computational domain is artificially bounded by a parallelepiped box on which the Silver-Müller condition is imposed. The underlying tetrahedral mesh consists of 619,759 vertices and 3,483,551 tetrahedra. This problem has been solved using the HDG-$\mathbb{P}_1$ method for the discretization of the subdomain problems in the Schwarz algorithm. Indeed, it has not been possible to use the DG-$\mathbb{P}_1$ method because of the memory capacity required for storing the L and U factors associated to the factorization of the subdomain problems in that case, at least for the number of subdomains considered here *i.e.* 288. The triangulation of the surface of the aircraft is shown on the top part of Figure 3 while the contour lines of $|\mathbf{E}|$ on the aircraft surface for an incident plane wave with frequency $F = 400$MHz are shown in Figure 4. The convergence histories of the DD-HDG solution strategy for a decomposition of the mesh in 288 subdomains are shown in Figure 5 for each of the considered values of the frequency F.

Performance results are summarized in Tables 9 and 10. The first of these tables is concerned with the factorization of the subdomain problems. Although the partitioning of the tetrahedral mesh using the MeTiS tool [KK99] yields perfectly balanced subdomains in terms of the number of elements, we again note that there is a large difference between the minimum and maximum memory requirements for constructing and storing the L and U factors. These results show that, as the frequency of the incident wave increases, the interface system (28) is solved in less iterations of the BiCGStab(l) method. This problem which involves a total of 83,605,224 unknowns for the components of the electromagnetic field $(\mathbf{E}, \mathbf{H})$ [1], and 42,343,638 unknowns for the components of the hybrid variable $\mathbf{\Lambda}_h$ [2] is solved in about 8min on 288 cores when $F = 200$MHz, and in less than 2min when $F = 800$MHz.

5.6 Exposure of head tissues to a plane wave

We now present the application of the proposed numerical methodology to the simulation of a time-harmonic electromagnetic wave propagation problem in an irregularly shaped and hetero-

[1] This number is the total number of degrees of freedom for the electromagnetic field which is obtained when discretizing the problem using the UF-DG-$\mathbb{P}_1$ method and a single domain, and is thus computed as $6 \times 4 \times$ #elements (*i.e.* 6 components for the electromagnetic field and 4 degrees of freedom per component, for each tetrahedron).

[2] This number is the total number of degrees of freedom for the hybrid variable which is obtained when discretizing the problem using the HDG-$\mathbb{P}_1$ method and a single domain, and is thus computed as $2 \times 3 \times$ #faces (*i.e.* 2 components for the hybrid variable and 3 degrees of freedom per component, for each face).

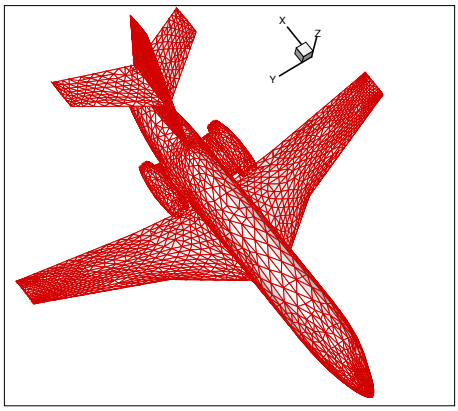


Figure 3: Scattering of a plane wave by an aircraft. Triangulation of the aircraft surface.

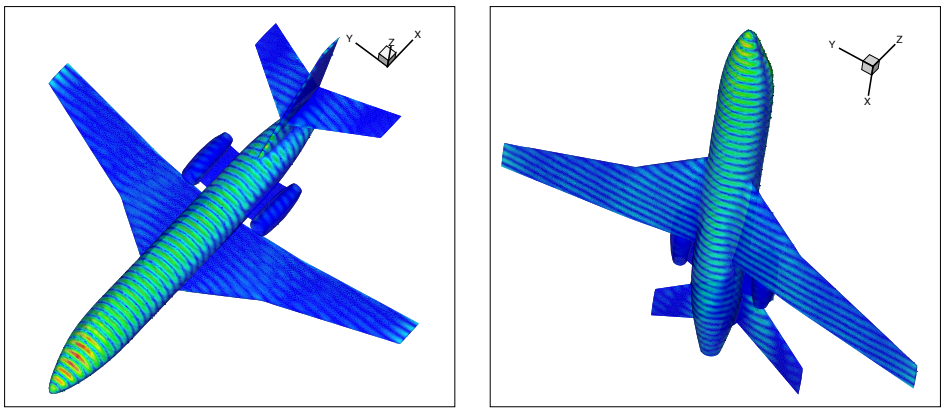


Figure 4: Scattering of a plane wave by an aircraft. Contour lines of $|\mathbf{E}|$ on the aircraft surface (F=400 MHz).

Method	N_s	CPU (min/max)	RAM (min/max)
HDG	288	38s/66s	993MB/1514MB

Table 9: Scattering of a plane wave by an aircraft. Computing times and memory requirement for storing the L and U factors.

Frequency	Method	N_s	#iter	CPU (min/max)	TOTAL
200MHz	HDG	288	17	269s/455s	484s
400MHz	-	-	10	143s/232s	243s
800MHz	-	-	4	59s/96s	98s

Table 10: Scattering of a plane wave by an aircraft. Computing times for the solution.

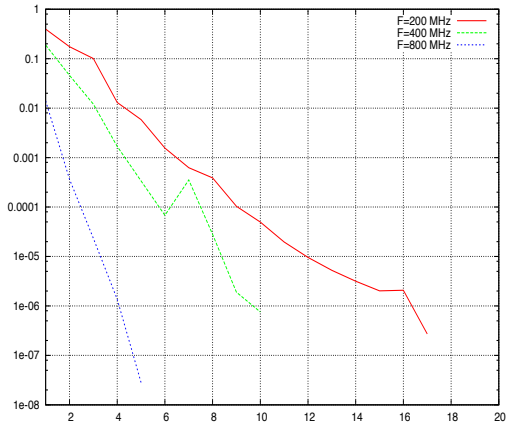


Figure 5: Scattering of a plane wave by an aircraft. Convergence history for 288 subdomains using the DD-HDG solution strategy.

geneous medium. The problem under consideration is concerned with the propagation of a plane wave in realistic geometrical models of head tissues. We refer to [DLP08a] for a detailed description of the problem setting and in particular of the geometrical modeling approach considered for constructing realistic tetrahedral meshes of the tissues starting from medical images. Surface meshes of the skin and the skull are visualized in Figure 6. The underlying tetrahedral mesh that has been used for the simulations contains 309,599 vertices and 1,853,832 elements. The frequency of the incident plane wave is $F = 1800$MHz and its polarization is such that:

$$\mathbf{k} = \begin{pmatrix} k_x \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} 0 \\ 0 \\ E_z \end{pmatrix} \quad \text{and} \quad \mathbf{H} = \begin{pmatrix} 0 \\ H_y \\ 0 \end{pmatrix}.$$

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 [BCPP00], 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).

Parallel simulations have been performed on the PlaFRIM (Plateforme Fédérative pour la Recherche en Informatique et Mathématiques) system operated by the INRIA Bordeaux - Sud-Ouest research center. The cluster that we have used consists of nodes with Nehalem Intel Xeon X5550 processors (8×2.66GHz cores per node and 24GB per node), interconnected by a high performance Infiniband network. Contour lines of the real part of E_z are visualized in Figures 7 and 8. Performance results are summarized in Tables 11 and 12. Because of the heterogeneity of the propagation medium, the required number of iterations for the Krylov methods to obtain the convergence of the interface system (28) are notably higher than those observed for the previous test problems. Similar remarks can be made for what concerns the memory consumption for

construction and storing the L and U factors of the subdomain problems. for the 256 subdomain decomposition of the mesh the maximum of the per subdomain memory consumption for the DD-HDG solution strategy is about 4 times lower than that of the DD-DG solution strategy. We observe a linear parallel speedup between the 128 and 256 subdomains simulations with the DD-HDG strategy, and in the case of 256 subdomains, the computing time of the DD-HDG solution strategy is about 4.9 times lower than that of the DD-DG solution strategy.

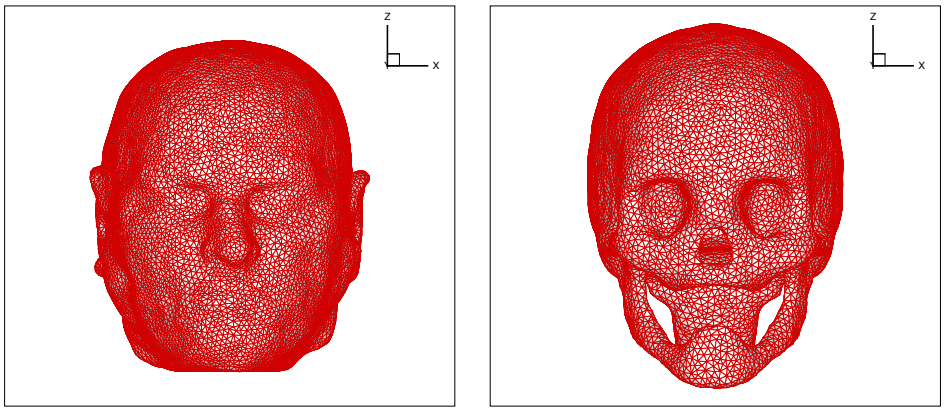


Figure 6: Exposure of head tissues to a plane wave. Triangulations of the skin and the skull.

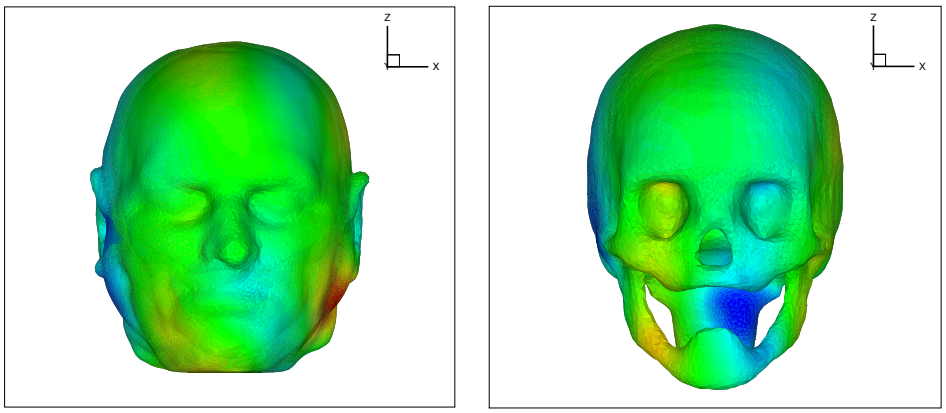


Figure 7: Exposure of head tissues to a plane wave. Contour lines of real part of E_z.

5.7 Scattering of multiple spheres in a waveguide

We conclude this series of numerical results by considering a configuration to assess the potentialities of the proposed DD-HDG solution strategy in treating very large-size problems. The proposed test problem is a L-shape waveguide-like structure including a set of 8 perfectly conducting spheres aligned with the geometry (see Figure 9). The spheres are identical with radius $R = 0.06$m. The distance between a sphere and the lateral walls is set to 0.1m. All the faces of the L-shape structure are assumed perfectly conducting at the exception of the face defined by $x = -0.42$m and $y = -0.58$m. The tetrahedral mesh that has been used for the simulations

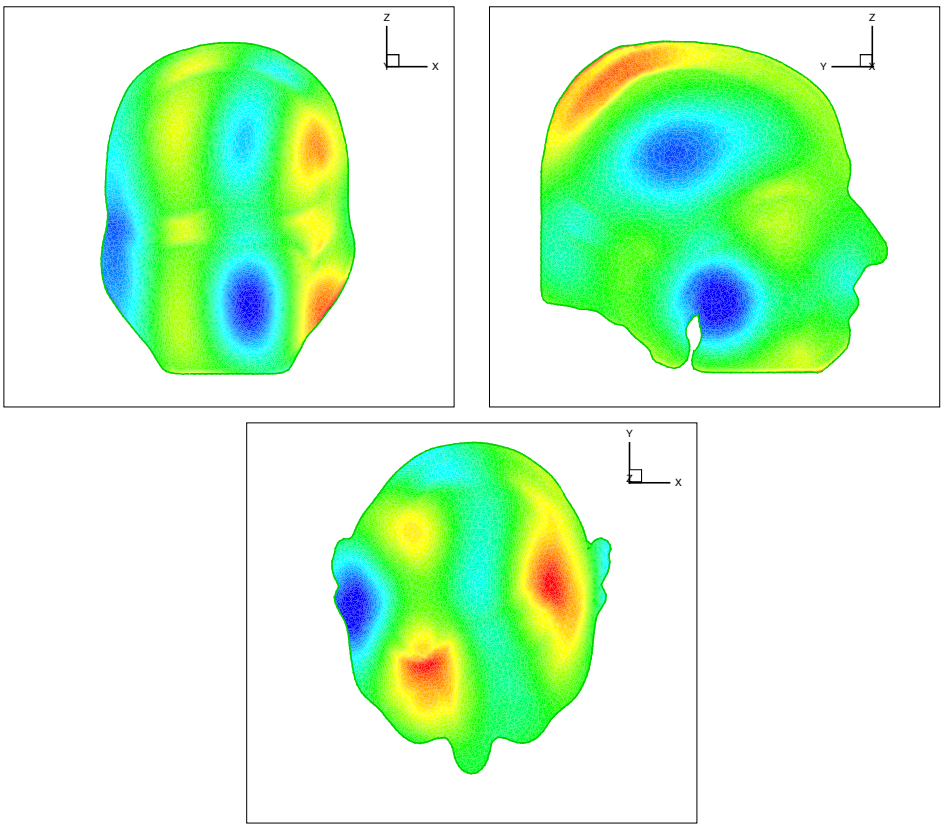


Figure 8: Exposure of head tissues to a plane wave. Contour lines of real part of E_z in selected planes.

Method	N_s	CPU (min/max)	RAM (min/max)
UF-DG	256	207s/210s	2174MB/3178MB
HDG	128	48s/49s	1435MB/2004MB
-	256	14s/15s	566MB/784MB

Table 11: Exposure of head tissues to a plane wave. Computing times and memory requirement for storing the L and U factors.

Method	N_s	#iter	CPU (min/max)	TOTAL
UF-DG	256	68	760s/764s	766s
HDG	128	63	317s/319s	321s
-	256	71	156s/157s	158s

Table 12: Exposure of head tissues to a plane wave. Computing times for the solution.

contains 802,532 vertices and 4,495,769 elements. The corresponding total numbers of degrees of freedom for the components of the electromagnetic field $(\mathbf{E}, \mathbf{H})$ and for the components of the hybrid variable $\mathbf{\Lambda}_h$ are respectively 107,898,456 and 54,635,364. The frequency of the incident plane wave is $F = 10$GHz and its polarization is similar to these proposed in the human head example.

This problem has been solved using the HDG-$\mathbb{P}_1$ method for the discretization of the subdomain problems in the Schwarz algorithm. Indeed, it has not been possible to use the DG-$\mathbb{P}_1$ method because of the memory capacity required for storing the L and U factors associated to the factorization of the subdomain problems in that case, at least for the number of subdomains considered here *i.e.* 288. Performance results are summarized in Tables 13 and 14. We could have expected that the high frequency of the incident wave would have facilitated the solution of the interface system (28) if we refer to the convergence histories obtained for the test problem of the scattering by an aircraft (see Figure 5), however the waveguide-like structure of the considered geometry seems to stiffen the problem. Despite this difficulty, the proposed DD-HDG solution strategy is able to solve the problem in about 52min (including the factorization time) which is a rather encouraging performance result for such a large system size.

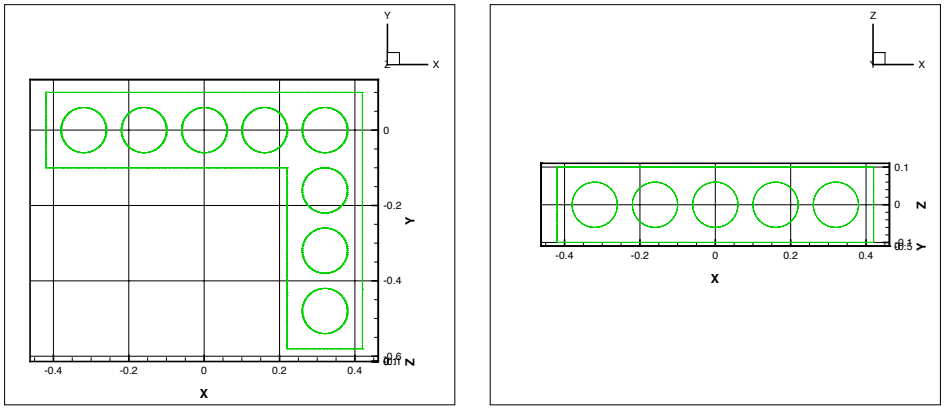


Figure 9: Scattering of multiple spheres in a waveguide. Geometrical setting

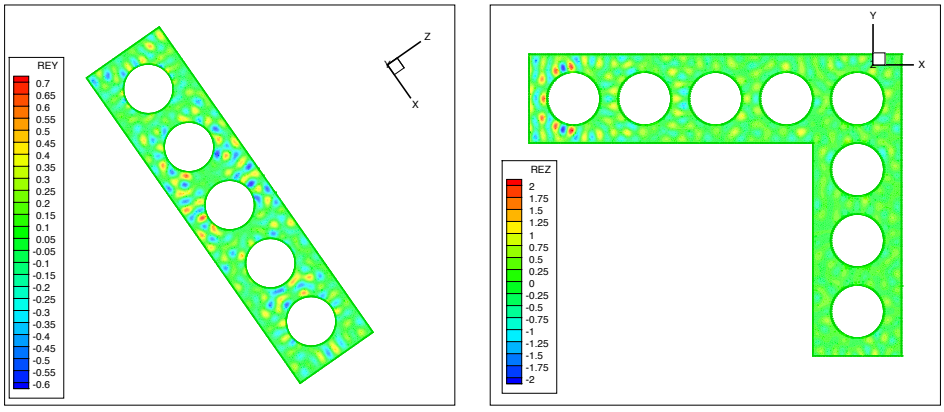


Figure 10: Scattering of multiple spheres in a waveguide. Contour lines of the real part of E_y (left) and E_z (right) in selected cut planes.

Method	N_s	CPU (min/max)	RAM (min/max)
HDG	288	85s/406s	1463MB/2093MB

Table 13: Scattering of multiple spheres in a waveguide. Computing times and memory requirement for storing the L and U factors.

Method	N_s	#iter	CPU (min/max)	TOTAL
HDG	288	63	2514s/2669s	2714s

Table 14: Scattering of multiple spheres in a waveguide. Computing times for the resolution.

6 Conclusions

We have presented a hybrid iterative/direct solution strategy for the large, sparse and complex coefficients algebraic systems resulting from the discretization of the three-dimensional time-harmonic Maxwell equations. This solution strategy relies on a Schwarz-type domain decomposition method making use of natural transmission conditions at the interface between neighboring subdomains. Within each subdomain, the discretization of the time-harmonic Maxwell equations relies on a new hybridizable discontinuous Galerkin scheme formulated on an unstructured tetrahedral mesh. 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 the hybridizable discontinuous Galerkin formulation allows for a natural treatment of heterogeneous media. Moreover, the hybridizable discontinuous Galerkin scheme allows for a dratsic reduction of the number of globally coupled degrees of freedom (within each sudomain) as compared to a classical discontinuous Galerkin scheme such as the one adopted in our previous study [DLP08a]. Numerical results have shown that the resulting DD-HDG solution strategy has an optimal convergence rate and can save both CPU time and memory cost compared to a classical upwind flux-based DD-DG approach. Our future efforts 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 [DGGG09] in the framework of a hybridizable discontinuous Galerkin formulation.

References

[ADL00] P. Amestoy, I. Duff, and J.Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Comput. Meth. Appl. Mech. Eng.*, 184:501–520, 2000.

[BCPP00] P. Bernardi, M. Cavagnaro, S. Pisa, and E. Piuzzi. Specific absorption rate and temperature increases in the head of a cellular phone user. *IEEE Trans. Microwave Theory Tech.*, 48(7):1118–1126, 2000.

[BDK+08] A. Buttari, J. Dongarra, J. Kurzak, P. Luszczek, and S. Tomov. Using mixed precision for sparse matrix computations to enhance the performance while achieving 64-bit accuracy. *ACM Trans. Math. Softw.*, 34(4):17:1–17:22, 2008.

[CGL09] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. *SIAM J. Numer. Anal.*, 47(2):1319–1365, 2009.

[CLS04] B. Cockburn, F.Y. Li, and C.W. Shu. Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. *J. Comput. Phys.*, 194(2):588–610, 2004.

[DFFL10] V. Dolean, H. Fahs, L. Fezoui, and S. Lanteri. Locally implicit discontinuous Galerkin method for time domain electromagnetics. *J. Comput. Phys.*, 229(2):512–526, 2010.

[DFLP08] V. Dolean, H. Fol, S. Lanteri, and R. Perrussel. Solution of the time-harmonic Maxwell equations using discontinuous Galerkin methods. *J. Comput. Appl. Math.*, 218:435–445, 2008.

[DGGG09] V. Dolean, M.J. Gander, and L. Gerardo-Giorda. Optimized Schwarz methods for Maxwell equations. *SIAM J. Scient. Comp.*, 31(3):2193–2213, 2009.

[DLP08a] V. Dolea, S. Lanteri, and R. Perrussel. A domain decomposition method for solving the three-dimensional time-harmonic Maxwell equations discretized by discontinuous Galerkin methods. *J. Comput. Phys.*, 227(3):2044–2072, 2008.

[DLP08b] V. Dolean, S. Lanteri, and R. Perrussel. Optimized schwarz algorithms for solving time-harmonic Maxwell's equations discretized by a discontinuous galerkin method. *IEEE Trans. Magn.*, 44(6):954–957, 2008.

[EG11] O.G. Ernst and M.J. Gander. *Why it is difficult to solve Helmholtz problems with classical iterative methods*, volume 83 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin Heidelberg, 2011.

[FLLP05] L. Fezoui, S. Lanteri, S. Lohrengel, and S. Piperno. Convergence and stability of a discontinuous Galerkin time-domain method for the 3d heterogeneous Maxwell equations on unstructured meshes. *ESAIM: Math. Model. and Numer. Anal.*, 39(6):1149–1176, 2005.

[HPS04] P. Houston, I. Perugia, and D. Schötzau. Mixed discontinuous Galerkin appproximation of the Maxwell's equations. *SIAM J. Numer. Anal.*, 42(1):434–459, 2004.

[HW02] J.S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids: I. time-domain solution of maxwell's equations. *J. Comput. Phys.*, 181(1):186–221, 2002.

[HW08] Jan S. Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods - algorithms, analysis, and applications*. Springer, New York, 2008.

[KK99] G. Karypis and V. Kumar. Parallel multilevel k-way partition scheme for irregular graphs. *SIAM Review*, 41(2):278–300, 1999.

[LLP11a] L. Li, S. Lanteri, and R. Perrussel. A hybridizable discontinuous Galerkin method for 2d time-harmonic Maxwell's equations. Technical Report 7649, INRIA, 2011.

[LLP11b] L. Li, S. Lanteri, and R. Perrussel. A hybridizable discontinuous Galerkin method for solving 3D time-harmonic Maxwell's equations. In A. Cangiani, R.L. Davidchack, E. Georgoulis, A.N. Gorban, J. Levesley, and M.V. Tretyakov, editors, *The European Numerical Mathematics and Advanced Applications Conference (ENUMATH 2011)*, Lecture Notes in Computational Science and Engineering, Leicester, United Kingdom, 2011. Springer. to appear.

[MRS04] F. Magoulès, F.X. Roux, and S. Salmon. Optimal discrete transmission conditions for a nonoveralpping domain decomposition method for the Helmholtz equation. *SIAM J. Sci. Comput.*, 25(5):1497–1515, 2004.

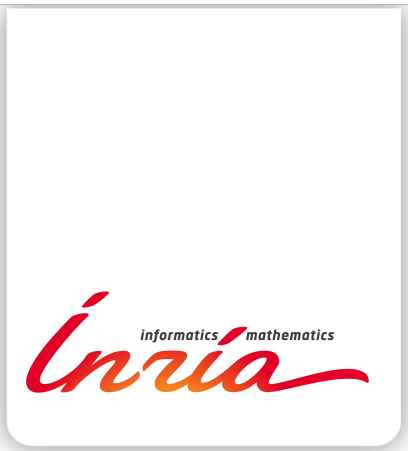
[NPC11] N.C. Nguyen, J. Peraire, and B. Cockburn. Hybridizable discontinuous Galerkin methods for the time-harmonic Maxwell's equations. *J. Comput. Phys.*, 230(19):7151–7175, 2011.

[SBG96] B. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition and parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press, 1996.

[SF93] L.G. Sleijpen and D.R. Fokkema. BiCGStab(l) for linear equations involving unsymmetric matrics with complex spectrum. *Electron. Trans. Numer. Anal.*, 1:11–32, 1993.

[SS86] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7:856–869, 1986.

Contents



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399