Non-Spurious Spectral Like Element Methods for Maxwell's equations

Gary Cohen, Marc Duruflé

Non Spurious Spectral-Like Element Methods for Maxwell's Equations

Gary Cohen[1] and Marc Duruflé[1]

INRIA, Domaine de Voluceau, Rocquencourt - BP 105, 78153 Le Chesnay Cedex, France

Abstract. To be written.

1 Introduction

For a long time, Maxwell's equations were mainly solved in the time-harmonic domain. The evolution of radar techniques showed the limit of this formulation which can only treat monochromatic sources. Both engineers and researchers were then motivated to use equations in the time domain which can take into account large frequency sources in one resolution. The first and most popular approximation of Maxwell's equations in the time domain was provides by the Yee's scheme [28], commonly called FDTD (Finite Difference in the Time Domain) by enginees, which is basically a centered second order finite difference approximation of Maxwell's equation.

Although easy to implement, FDTD has some difficulties to treat complex geometries. In fact, the staircase approximation of curved boundary can produce spurious reflections which can substantially pollute the solution. On the other hand, finite element methods (FETD) have the major drawback of producing a n-diagonal (n can grow up to several tens in 3D) mass matrix which must be inverted at each time-step, which is a serious handicap for FETD versus FDTD whose mass matrix is the identity matrix. This mass matrix does not present any difficulty to time harmonic problems, for which even the stiffness matrix must be inverted. For this reason, industry was reluctant to use FETD for a long time and FDTD remains the reference for Maxwell's equations in the time domain for 40 years!

The mass lumping technique is an efficient alternative to mass matrix inversion. However, this technique was well known for lower order continuous (or H^1) elements but not obvious for higher-order approximations. A first step towards a general mass lumping technique was made by Hennart et al. [19,20] and independently by Young [29] which proposed to use Gauss-Lobatto quadrature formulas to get mass lumping for continuous quadrilateral or hexahedral finite elements. Besides mass lumping, these formulas ensure to keep the order induced by the finite element approximation [3]. First introduced for ODE or parabolic problems, this technique was extended to the wave equation by Cohen et al. [11] and renamed spectral element methods [22]. The non trivial extension of this technique to triangular and tetrahedral elements was later realized by Cohen et al [10] for triangles up to third order and Mulder et al. [23] for higher-order triangles and tetrahedra.

The problem of the mass matrix inversion was solved for the wave equation and remained a challenging problem for Maxwell's equations. A first try was done by Haugazeau et al. [18] but this approach remains restricted to first-order approximation. A second and natural step was to extend spectral element techniques to edge (or $H(curl)$) elements. This was done by Cohen et al. for orthogonal meshes for the first family [12] and for any mesh for the second family of edge elements of any order [13]. The extension to triangular and tetrahedral meshes was realized by Elmkies et al. [16,17] but lead to efficient approximations up to the second-order elements.

Due to the storage of the stiffness matrix, even by using mass lumping techniques, FETD remained much more expensive than FDTD in terms of storage and, to a lesser extent, in computational time. This ultimate problem was solved by using a mixed $H(curl) - L^2$ formulation of Maxwell's equations based on $H(curl)$-conform definition of the $curl$ operator in both spaces [13]. This technique provides a local definition of the stiffness matrices which induces a substantial gain of storage. It was later extended to acoustics [6] and linear elastodynamics [7]. A detailed presentation of all these techniques can be found in [5].

Unfortunately, although H^1 spectral elements and H^1 and $H(curl)$ triangular and tetrahedral elements behave quite well for any mesh, $H(curl)$ spectral elements present important parasitic

waves for very distorted meshes, which are often used in industrial problems. For this reason, discontinuous Galerkin methods appeared as an efficient alternative for Maxwell's equations. First introduced by Hesthaven [21] for tetrahedra, this approach was adapted by Cohen et al. [9] to the spectral element point of view, which provided a low-storage as well as fast method to solve Maxwell's equations. This approach seemed to deal better with parasitic waves but eigenvalues considerations showed that such waves were however present in this method. All these remarks motivated us to discuss the numerical dissipation terms which can attenuate parasitic waves. This discussion is the new part of our survey.

Our paper is divided into four parts. In a first section, we present the continuous formulations of the Maxwell's equations and different approaches for its approximation. In the second section, we discuss the parasitic modes through an eigenvalue analysis. In the third section, we introduce dissipative jump terms to get rid of parasitic modes. The last section is devoted to the approximation of the time-harmonic problem by the methods described in the first section.

2 Different Approximations of the Problem

2.1 Formulations of the Continuous Problem

In this paper, we are interested in solving the so-called lossy Maxwell's equations in anisotropic which read

Find $(\boldsymbol{E}, \boldsymbol{H})$: $\Omega\times]0,T[\rightarrow I\!R^3$ such that

$$\underline{\underline{\varepsilon}}\frac{\partial \boldsymbol{E}}{\partial t} - \nabla\times\boldsymbol{H} + \underline{\underline{\sigma}}\boldsymbol{E} = -\boldsymbol{J}, \tag{1a}$$

$$\underline{\underline{\mu}}\frac{\partial \boldsymbol{H}}{\partial t} + \nabla\times\boldsymbol{E} = 0, \tag{1b}$$

where $\boldsymbol{J}$ is a given function of time and space which can model a pulse or a plane wave. $\underline{\underline{\varepsilon}}$, $\underline{\underline{\sigma}}$ and $\underline{\underline{\mu}}$ are the permittivity, conductivity and permeability matrices which can take into account anisotropic media.

To these equations, we add homogeneous initial conditions on $(\boldsymbol{E}, \boldsymbol{H})$ and the perfectly conducting condition $\boldsymbol{E}\times\boldsymbol{n} = 0$ on $\partial\Omega$.

By plugging equation (1b) in (1a), this system can be expressed as the following unique equation in $\boldsymbol{E}$:

$$\underline{\underline{\varepsilon}}\frac{\partial^2 \boldsymbol{E}}{\partial t^2} - \nabla\times\left(\underline{\underline{\mu}}^{-1}\left(\nabla\times\boldsymbol{E}\right)\right) + \underline{\underline{\sigma}}\frac{\partial \boldsymbol{E}}{\partial t} = -\boldsymbol{J'} \tag{2}$$

where $\boldsymbol{J'} = \partial\boldsymbol{J}/\partial t$.

2.2 General Issues

Some General Principles In this section, we introduce some basic principle which will be used for the different approximations of the Maxwell's problem descibed below.

1. Let us first define the fundamental space for Maxwell's equations:

$$H_0(\boldsymbol{curl}, \Omega) = \left\{\boldsymbol{v}_h \in \left[L^2(\Omega)\right]^3 \text{ such that } \nabla\times\boldsymbol{v}_h \in \left[L^2(\Omega)\right]^3\right\} \tag{3}$$

   Functions of this functional space are characterized by the continuity of their tangential component.

2. For any quadrilateral or hexahedral mesh, basis functions are defined on the *reference element* $\widehat{K}$ which is the unit square in 2D and the unit cube in 3D. These functions (generally called hat functions because indicated by a hat) are functions of $\widehat{\boldsymbol{x}} \in \widehat{K}$. Then, a basis functions in any element K_j of a mesh $\mathcal{Q}$ is obtained by a transform based on $\boldsymbol{F}_j$ such that $\boldsymbol{F}_j(\widehat{K}) = K_j$. Obviously, this function depends on $\boldsymbol{x} = \boldsymbol{F}_j(\widehat{\boldsymbol{x}})$.
   In our different approximations, we use the $H(curl)$-conform transform which maps a hat vector basis function $\widehat{\boldsymbol{\varphi}}$ onto a basis function $\boldsymbol{\varphi}$ on K_j as follows:

$$\boldsymbol{\varphi} \circ \boldsymbol{F}_j = DF_i^{*-1} \widehat{\boldsymbol{\varphi}},$$

   where DF_i^{*-1} is the inverse of the transposed Jacobian matrix of $\boldsymbol{F}_j$.
   This transform is called $H(curl)$-conform because it ensures the continuity of the tangential component of $\boldsymbol{\varphi}$ all over the mesh.

3. Another important point in the following, is the use of mass lumping. This property is obtained by defining interpolation points for the basis functions (which are basically Lagrange interpolation polynomials) as quadrature points and by computing mass integrals by the corresponding quadrature formula. In the case of Maxwell's equations, this technique provides *block-diagonal* mass matrices.

A General Framework for Edge Elements The edge element methods are based on the following variational formulation of problem (2):

Find $\boldsymbol{E} \in L^2(0,T; H_0(\boldsymbol{curl}, \Omega))$ such that

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} \int_\Omega \underline{\underline{\varepsilon}} \boldsymbol{E} \cdot \boldsymbol{\varphi} \, \mathrm{d}\boldsymbol{x} - \int_\Omega \underline{\underline{\mu}}^{-1} (\nabla \times \boldsymbol{E}) \cdot \nabla \times \boldsymbol{\varphi} \, \mathrm{d}\boldsymbol{x} + \frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \underline{\underline{\sigma}} \boldsymbol{E} \cdot \boldsymbol{\varphi} \, \mathrm{d}\boldsymbol{x} = -\int_\Omega \boldsymbol{J'} \cdot \boldsymbol{\varphi} \, \mathrm{d}\boldsymbol{x}, \qquad (4)$$

$\forall \boldsymbol{\varphi} \in H_0(\boldsymbol{curl}, \Omega)$.

Then, we construct an edge element approximation based on this variational formulation which reads

Find $\boldsymbol{E}_h \in L^2(0,T; \boldsymbol{V}_h^r)$ such that

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} \int_\Omega \underline{\underline{\varepsilon}} \boldsymbol{E}_h \cdot \boldsymbol{\varphi}_h \, \mathrm{d}\boldsymbol{x} - \int_\Omega \underline{\underline{\mu}}^{-1} (\nabla \times \boldsymbol{E}_h) \cdot \nabla \times \boldsymbol{\varphi}_h \, \mathrm{d}\boldsymbol{x} + \frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \underline{\underline{\sigma}} \boldsymbol{E}_h \cdot \boldsymbol{\varphi}_h \, \mathrm{d}\boldsymbol{x} = -\int_\Omega \boldsymbol{J'} \cdot \boldsymbol{\varphi}_h \, \mathrm{d}\boldsymbol{x}, \quad (5)$$

$\forall \boldsymbol{\varphi}_h \in \boldsymbol{V}_h^r$.

2.3 A First Family of Edge Elements

For the first family of edge elements (defined in [24]), we have,

$$\boldsymbol{V}_h^r = \{\boldsymbol{v}_h \in H_0(\boldsymbol{curl}, \Omega) \text{ such that } \forall K_j \in \mathcal{Q},\ DF_i^* \boldsymbol{v}_{h|_{K_j}} \circ \boldsymbol{F}_j \in Q_{r-1,r,r} \times Q_{r,r-1,r} \times Q_{r,r,r-1}\}, \quad (6)$$

where

$$Q_{r_1,r_2,r_3} = \left\{ p(\boldsymbol{x}) = \sum_{\ell=0}^{r_1} \sum_{m=0}^{r_2} \sum_{n=0}^{r_3} a_{\ell,m,n}\, x_1^\ell x_2^m x_3^n,\ a_{\ell,m,n} \in I\!R \right\}. \qquad (7)$$

For spectral edge elements, the basis functions on $\widehat{K}$ for this family are constructed from 1D Lagrange interpolation polynomials based on Gauss and Gauss-Lobatto quadrature points (denoted $(\widehat{\xi}_\ell^G,\ \ell = 1,..,r)$ and $(\widehat{\xi}_\ell^{GL},\ \ell = 1,..,r+1)$ respectively). For each set of points, the corresponding polynomials are denoted $\widehat{g}_\ell$ and $\widehat{l}_\ell$ respectively. For instance, the function of $Q_{r-1,r,r} \times Q_{r,r-1,r} \times Q_{r,r,r-1}$ corresponding to the point $\left(\widehat{\xi}_\ell^G, \widehat{\xi}_m^{GL}, \widehat{\xi}_n^{GL}\right) \in \widehat{K}$ reads

$$\widehat{\boldsymbol{\varphi}}_{\ell,m,n}(\widehat{x}_1, \widehat{x}_2, \widehat{x}_3) = \widehat{g}_\ell(\widehat{x}_1) \widehat{l}_m(\widehat{x}_2) \widehat{l}_n(\widehat{x}_3) \boldsymbol{e}_1,$$

where $\boldsymbol{e}_1$ is the first basis vector of $\mathbb{R}^3$. In the same way, functions corresponding to points with a Gauss coordinate as second (resp. third) coordinate, are supported by $\boldsymbol{e}_2$ (resp. $\boldsymbol{e}_3$).

For any $\widehat{g}_\ell$ and any $\widehat{\xi}_m^G$ and for any $\widehat{l}_\ell$ and any $\widehat{\xi}_m^{GL}$, we have $\widehat{g}_\ell(\widehat{\xi}_m^G) = \delta_{\ell m}$ and $\widehat{l}_\ell(\widehat{\xi}_m^{GL}) = \delta_{\ell m}$, where $\delta_{\ell m}$ is the Kronecker symbol.

The corresponding degrees of freedom are given in Fig. 1.

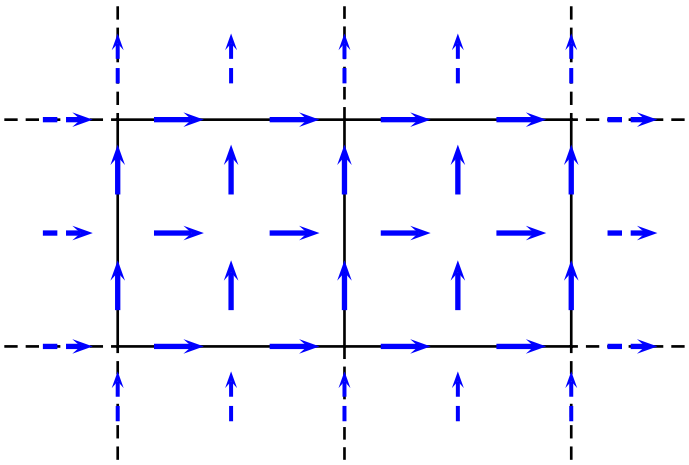

Fig. 1. The degrees of freedom of the first family of edge elements for $\boldsymbol{E}$ on faces of cubes when $r = 2$

2.4 A Second Family of Edge Elements

The definition of $\boldsymbol{V}_h^r$ for the second family of edge elements is much simpler. It reads

$$\boldsymbol{V}_h^r = \{\boldsymbol{v}_h \in H_0(\boldsymbol{curl}, \Omega) \text{ such that } \forall K_j \in \mathcal{Q},\ DF_i^* \boldsymbol{v}_{h_{|_{K_j}}} \circ \boldsymbol{F}_j \in (Q_r)^3\}, \tag{8}$$

where $Q_r = Q_{r,r,r}$, $Q_{r,r,r}$ defined as in (7).

For spectral edge elements, the basis functions on $\widehat{K}$ for this family are constructed from 1D Lagrange interpolation polynomials based on Gauss-Lobatto quadrature points. For any point ℓ, the corresponding polynomial is denoted $\widehat{\varphi}_\ell$. For instance, the function of $(Q_r)^3$ corresponding to the point $\left(\widehat{\xi}_\ell^{GL}, \widehat{\xi}_m^{GL}, \widehat{\xi}_n^{GL}\right) \in \widehat{K}$ reads

$$\widehat{\boldsymbol{\varphi}}_{\ell,m,n}^p(\widehat{x}_1, \widehat{x}_2, \widehat{x}_3) = \widehat{\varphi}_\ell(\widehat{x}_1)\widehat{l}_\varphi(\widehat{x}_2)\widehat{l}_\varphi(\widehat{x}_3)\boldsymbol{e}_p.$$

Moreover, we have:

$$\widehat{\boldsymbol{\varphi}}_{\ell,m,n}^p(\widehat{\xi}_i^{GL}, \widehat{\xi}_j^{GL}, \widehat{\xi}_k^{GL}) = \delta_{\ell i}\delta_{mj}\delta_{nk}\boldsymbol{e}_p.$$

The corresponding degrees of freedom are given in Fig. 2.

This method provides a block-diagonal mass matrix on any mesh. This structure is obtained by numbering the degrees of freedom n_i around a point i sequentially and by using a Gauss-Lobatto quadrature formula for computing the mass (and also stiffness) integrals. Therefore, the size of a block is equal to n_i.

2.5 Mixed Formulation

Although the above method solves the problem of mass lumping, which is important in terms of computational time, the storage of the stiffness matrix on non-orthogonal meshes remains substantial. This storage is still a drawback of FETD, even mass-lumped, versus FDTD. A way to overcome this problem is to use a mixed formulation.

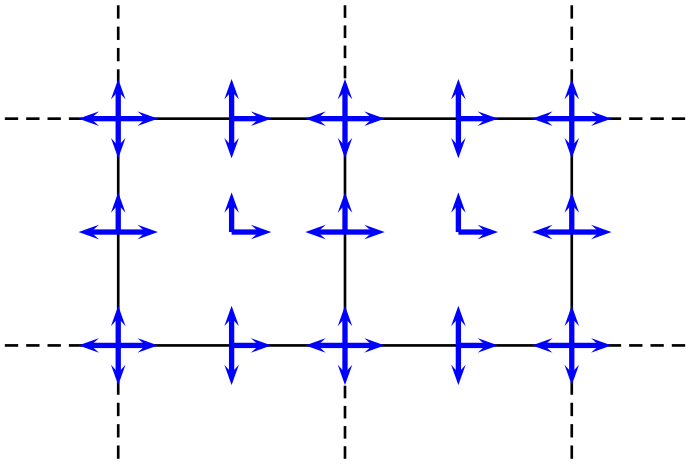

Fig. 2. The degrees of freedom of the second family of edge elements for $\boldsymbol{E}$ on faces of cubes when $r = 2$

In order to write this new approach, we use equations (1a)-(1b) with the following variational formulation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \underline{\underline{\varepsilon}}\boldsymbol{E}\cdot\boldsymbol{\varphi}\,\mathrm{d}\boldsymbol{x} - \int_\Omega \boldsymbol{H}\cdot\nabla\times\boldsymbol{\varphi}\,\mathrm{d}\boldsymbol{x} + \int_\Omega \underline{\underline{\sigma}}\boldsymbol{E}\cdot\boldsymbol{\varphi}\,\mathrm{d}\boldsymbol{x} = -\int_\Omega \boldsymbol{J}\cdot\boldsymbol{\varphi}\,\mathrm{d}\boldsymbol{x},\ \forall\boldsymbol{\varphi}\in H_0(\boldsymbol{curl},\Omega), \quad (9a)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \underline{\underline{\mu}}\boldsymbol{H}\cdot\boldsymbol{\psi}\,\mathrm{d}\boldsymbol{x} + \int_\Omega \nabla\times\boldsymbol{E}\cdot\boldsymbol{\psi}\,\mathrm{d}\boldsymbol{x} = 0,\ \forall\boldsymbol{\psi}\in\left[L^2(\Omega)\right]^3, \quad (9b)$$

which provides the approximate problem

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \underline{\underline{\varepsilon}}\boldsymbol{E}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x} - \int_\Omega \boldsymbol{H}_h\cdot\nabla\times\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x} + \int_\Omega \underline{\underline{\sigma}}\boldsymbol{E}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x} = -\int_\Omega \boldsymbol{J}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x},\ \forall\boldsymbol{\varphi}_h\in\boldsymbol{V}_h^r, \quad (10a)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \underline{\underline{\mu}}\boldsymbol{H}_h\cdot\boldsymbol{\psi}_h\,\mathrm{d}\boldsymbol{x} + \int_\Omega \boldsymbol{E}_h\cdot\nabla\times\boldsymbol{\psi}_h\,\mathrm{d}\boldsymbol{x} = 0,\ \forall\boldsymbol{\psi}_h\in\boldsymbol{W}_h^r, \quad (10b)$$

where

$$\boldsymbol{W}_h^r = \{\boldsymbol{v}_h\in\left[L^2(\Omega)\right]^3 \text{ such that } \forall K_j\in\mathcal{Q},\ DF_i^*\boldsymbol{v}_{h_{|K_j}}\circ\boldsymbol{F}_j\in(Q_r)^3\}. \quad (11)$$

Basis functions of $\boldsymbol{W}_h^r$ are constructed exactly in the same way as in section 2.4 but they are discontinuous. In fact, this discontinuous character makes useless the use of the $H(curl)$-conform transform which ensures the tangential continuity. Its presence in $\boldsymbol{W}_h^r$ is an artifact to provide an important gain of storage based on the following identity:

$$\forall K_j\in\mathcal{Q},\ \forall\boldsymbol{\varphi}_h\in\boldsymbol{V}_h^r,\ \forall\boldsymbol{\psi}_h\in\boldsymbol{W}_h^r,\ \int_{K_j}\nabla\times\boldsymbol{\varphi}_h\cdot\boldsymbol{\psi}_h\,\mathrm{d}\boldsymbol{x} = \mathrm{sign}(J_j)\int_{\widehat{K}}\widehat{\nabla}\times\widehat{\boldsymbol{\varphi}}_h\cdot\widehat{\boldsymbol{\psi}}_h\,\mathrm{d}\widehat{\boldsymbol{x}}. \quad (12)$$

where $\widehat{\nabla}$ is the ∇ operator in $\widehat{\boldsymbol{x}}$ coordinates, $J_j = \det DF_i$ and $\widehat{\boldsymbol{\varphi}}_h = DF_i^*\boldsymbol{\varphi}_h|_{K_j}\circ\boldsymbol{F}_j$.

Identity (12) means that the knowledge of the stiffness matrix on $\widehat{K}$ implies its knowledge on any element K_j (modulo one sign per element). In other words, if one assemblies the stiffness matrix at each time-step (which is a low-cost operation, in particular for higher-order approximations), the storage of the stiffness matrix is reduced to its storage over $\widehat{K}$ (which is peanuts!), the storage of one sign per element and the storage of the mass matrix of (10b) which is a 3×3 block-diagonal matrix. Of course, this induces a huge gain of storage, as indicated by the left curve of Fig. 3. Besides this gain of storage, due to orthogonality properties of the basis functions over $\widehat{K}$, this approach also provides a substantial gain of storage, as one can see in the right curve of Fig 3.

By using a leapfrog scheme in time, (10a)-(10b) can be written as follows in its discrete form

$$B_\varepsilon\frac{\mathbf{E^{n+1}}-\mathbf{E^{n-1}}}{\Delta t} + R_h\,\mathbf{H^{n+1/2}} + B_\sigma\frac{\mathbf{E^{n+1}}+\mathbf{E^{n}}}{2} = 0, \quad (13a)$$

$$B_\mu\frac{\mathbf{H^{n+1/2}}-\mathbf{H^{n-1/2}}}{\Delta t} + R_h^*\,\mathbf{E^{n}} = 0, \quad (13b)$$

where B_ε, B_σ are $n_i \times n_i$ block-diagonal mass matrices, B_μ is a 3×3 block-diagonal mass matrix and R_h is a stiffness matrix which needs almost no storage.

A detailed description of the above methods can be found in [5].

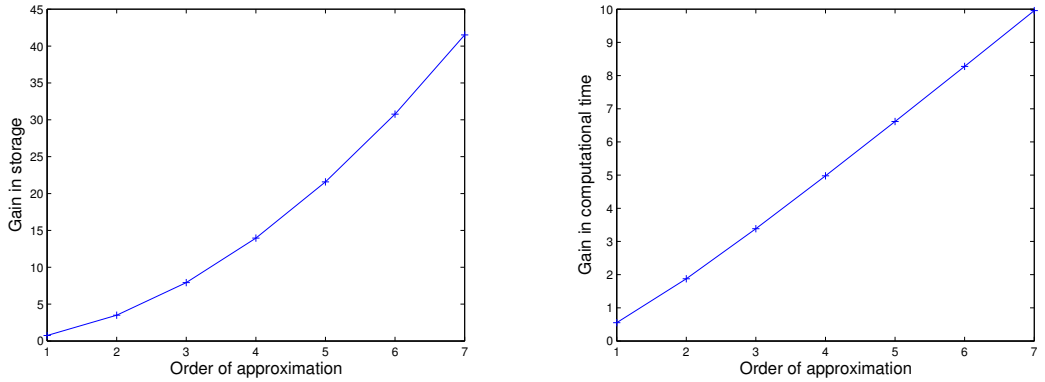

Fig. 3. Storage (*left*) and CPU time (*right*) ratios for spectral elements and mixed spectral elements for the same number of d.o.f

Remarks:

1. The mixed formulation was also applied to the acoustics equation [6] and the linear elastodynamics system [7].
2. A proof of the equivalence between the approximations given in (4) and in (10a)-(10b) is given in Pernet's thesis [26]. This proof is a powerful generalization of the equivalence theorem given in [6].
3. This mixed formulation can also be applied to the first family of edge elements. Although inefficient in terms of mass lumping on non-orthogonal meshes for this family, this formulation provides a fast algorithm of matrix-vector product very useful for the time-harmonic problem, as discussed in section ???.
4. Another approximation of (9a)-(9b) for the first family of edge elements, in which $\left[L^2(\Omega)\right]^3$ is replaced by $H(\mathrm{div}, \Omega)$, is given in [12]. This approach leads to a variational generalization of the Yee's scheme at any order.

2.6 Discontinuous Galerkin Methods

Description of the Method In a discontinuous Galerkin method (DGM), the solution is sought in $\left[L^2(\Omega)\right]^3$. So, one cannot express a continuous variational formulation. The approximate problem reads as follows

For each $K_j \in \mathcal{Q}$, we set:

$$\begin{aligned}\int_{K_j} \varepsilon \frac{\partial \mathbf{E}}{\partial t} \cdot \boldsymbol{\psi}\, dx \;=& -\int_{K_j} \sigma \mathbf{E} \cdot \boldsymbol{\psi}\, dx \int_{K_j} \nabla \times \mathbf{H} \cdot \boldsymbol{\psi}\, dx \;-\; \int_{K_j} \boldsymbol{J} \cdot \boldsymbol{\psi}\, dx \\ &+ \alpha \int_{\partial K_j} [\![\mathbf{H} \times \mathbf{n}]\!]_{\partial K_j}^{K_j} \cdot \boldsymbol{\psi}\, ds + \beta \int_{\partial K_j} [\![\mathbf{n} \times \mathbf{E}]\!]_{\partial K_j}^{K_j} \cdot (\mathbf{n} \times \boldsymbol{\psi})\, ds,\end{aligned} \tag{14}$$

$$\begin{aligned}\int_{K_j} \mu \frac{\partial \mathbf{H}}{\partial t} \cdot \boldsymbol{\phi}\, dx =& -\int_{K_j} \nabla \times \mathbf{E} \cdot \boldsymbol{\phi}\, dx + \gamma \int_{\partial K_j} [\![\mathbf{E} \times \mathbf{n}]\!]_{\partial K_j}^{K_j} \cdot \boldsymbol{\phi}\, ds \\ &+ \delta \int_{\partial K_j} [\![\mathbf{n} \times \mathbf{H}]\!]_{\partial K_j}^{K_j} \cdot (\mathbf{n} \times \boldsymbol{\phi})\, ds,\end{aligned} \tag{15}$$

where α, β, γ, δ are positive constants, $\boldsymbol{\phi} \in \boldsymbol{W}_h^r$, $\boldsymbol{\psi} \in \boldsymbol{W}_h^r$ and $[\![\mathbf{v}]\!]_{\partial K_j}^{K_j}$ is the jump of a vector-valued function $\mathbf{v}$ across the boundary of K_j.

Basis functions of $\boldsymbol{W}_h^r$ are constructed exactly in the same way as in section 2.5. Now, the discontinuous character of these basis functions allows us to use either Gauss-Lobatto or Gauss points as interpolation and quadrature points. A dispersion analysis of the schemes obtain by using each set of points shows a better accuracy for Gauss points which actually provide an exact value of the integrals. However, since some of the points are located on the faces of the unit cube, Gauss-Lobatto points give an immediate definition of the jumps whereas Gauss points require an extrapolation. However, due to orthogonality properties, this extrapolation is 1D and do not induce a substantial additional cost.

Discrete Energy Analysis If we compute the discrete energy of (14)-(15) (when $\sigma = 0$ and $\boldsymbol{J} = 0$)

$$\mathcal{E}(t) = \sum_{K_i \subset \Omega} \left\{ \int_{K_i} (\underline{\underline{\epsilon}} \boldsymbol{E}_{K_i}) \cdot \boldsymbol{E}_{K_i} dx + \int_{K_i} (\underline{\underline{\mu}} \boldsymbol{H}_{K_i}) \cdot \boldsymbol{H}_{K_i} dx \right\},$$

one can show that, for $-\alpha = \gamma = \frac{1}{2}$, $\alpha \geq 0$ and $\delta \geq 0$, we get

$$\frac{\mathrm{d}\mathcal{E}}{\mathrm{d}t}(t) = \sum_{K_i \cap K_j} \{-\beta \| [\![\boldsymbol{n} \times \boldsymbol{E}]\!]_{K_i \cap K_j} \|_\Gamma^2 - \delta \| [\![\boldsymbol{n} \times \boldsymbol{H}]\!]_{K_i \cap K_j} \|_\Gamma^2 \} \tag{16}$$

which provides a decreasing energy and so, a disspative aproximation of the problem. On the other hand, if we set $\alpha = 0$ and $\delta = 0$, we get a conservative approximation.

Now, the discrete form of this approximation with a leapfrog scheme in time reads

$$B_\varepsilon \frac{\boldsymbol{E}^{n+1} - \boldsymbol{E}^n}{\Delta t} + B_\sigma \frac{\boldsymbol{E}^{n+1} + \boldsymbol{E}^n}{2} + R_h \boldsymbol{H}^{n+1/2} + \alpha\, S_h \boldsymbol{H}^{n+1/2} + \beta\, D_h \boldsymbol{E}^n + \boldsymbol{J}^{n+1/2} = 0, \tag{17}$$

$$B_\mu \frac{\boldsymbol{H}^{n+1/2} - \boldsymbol{H}^{n-1/2}}{\Delta t} + R_h \boldsymbol{E}^n + \gamma\, S_h^* \boldsymbol{E}^n + \delta\, D_h^* \boldsymbol{H}^{n+1/2} = 0, \tag{18}$$

where B_ε, B_σ and B_μ ar 3×3 block diagonal mass matrices, R_h is a stiffness matrix which needs to be stored on the unit element, S_h is a jump matrix which needs to be stored on the unit element and D_h is a jump matrix which needs to be stored (its storage is equivalent to that of a mass matrix). Moreover, the jump terms in which D_h appear need to be uncentered in order to keep the scheme explicit, which introduces some numerical dissipation. However, this is not too much troublesome since these two terms are themselves dissipative. So, in order to reduce the storage and to avoid numerical dissipation, it seems a priori better to set $\beta = \delta = 0$. This was our first choice, as described in details in [9,26].

Remark: Plane wave analysis and error estimates can be found in [26]. Error analysis shows that, due to the presence of DF_i which is not constant on non-regular meshes, a Q_r approximation is of order $r - 1$. In particular, this implies that the first-order approximation is not consistent on non-regular meshes. This theoretical result was confirmed by numerical experiments.

2.7 Numerical Results

Both DGM and the second family of edge elements on slightly non-regular meshes provided performant results in 2D and 3D in terms of storage and CPU time. In particular comparisons with FDTD showed its efficiency [8]. Such meshes are easy to construct in 2D but not always possible to get in 3D. In fact, for complex geometries, it is difficult to obtain a purely hexahedral mesh. A palliative to this difficulty is to produce a hexahedral mesh from a tetrahedral mesh by splitting each tetrahedron into four hexahedra, as shown in Fig 4. Unfortunately, this technique leads to very distorted meshes which produces very important parasitic waves.

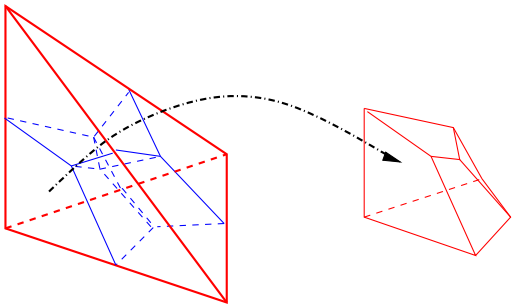

Fig. 4. Splitting a tetrahedron into four hexahedra

3 Eigenvalue Analysis

Since one can write a spectral decomposition the solution of Maxwell's equations, the study of the eigenvalues problem related to the Maxwell's equations in their continuous or discrete form can exhibit parasitic modes.

In [14], the problem of spurious modes for nodal elements is dealt. The author suggests a weighted regularization to avoid these spurious modes. In [1], the second family is shown as spectrally non-correct in the case of rectangular meshes. The authors prooved that "hp" elements based on the first family for rectangular meshes provide a spurious-free method. I. Perugia and Al [2] treats the Maxwell eigenvalue for several Discontinuous Galerkin formulations : interior penalty IP, non-symmetric interior penalty NIP and Local Discontinuous Galerkin LDG, which is the formulation we use. The authors conclude that the formulation is spurious free for first and second family on tetrahedra, for first family on hexahedra. Nevertheless, they think that the method has spurious modes for the second family. Our study will focus on the second family, in order to give a new enlightement.

Our eigenvalue analysis is developped on two test cases which are unit square and cubic cavities meshed by using split triangles or tetrahedra as shown in Fig. 5 with a Q_3 approximation. For these two cavities, we compute the eigenvalues in the range of frequencies contained in the solution.

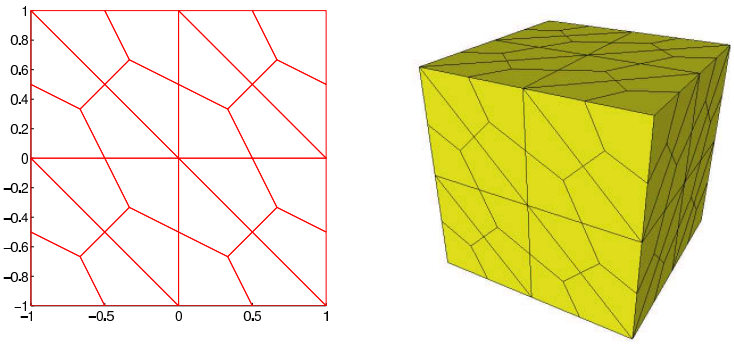

Fig. 5. Test meshed cavities in 2D and 3D

3.1 Discontinuous Galerkin Methods

For DGM without numerical dissipation (i.e. $\beta = \delta = 0$), the behaviour of the eigenvalues is somehow strange. In 2D, we only get one parasitic eigenvalue but a huge amount of such eigenvalues appear in 3D (Fig. 6). If we compare with the second family of edge elements (Fig. 9), we get about 25 times more parasitic eigenvalues in 3D!

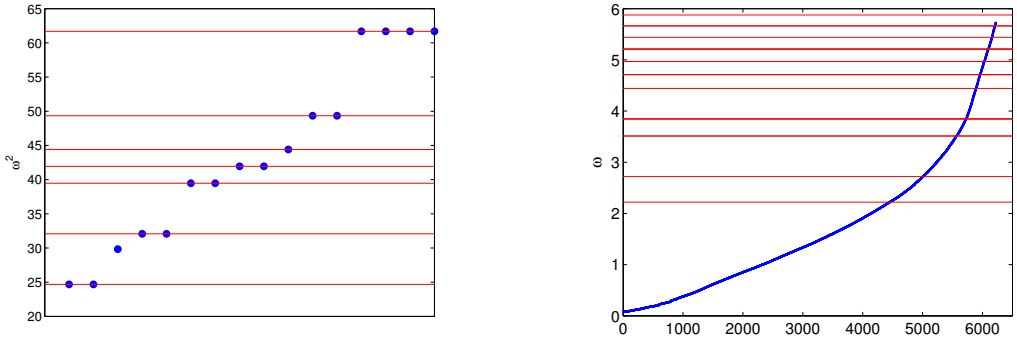

Fig. 6. Eigenvalues for the 2D (left) and 3D (right) cavities computed with DGM. Physical eigenvalues are indicated by red lines

An efficient way to get rid of these parasitic eigenvalues is to introduce some numerical dissipation (i.e. to set $\beta > 0$ or/and $\delta > 0$). In this way, the parasitic eigenvalues become all complex with a negative imaginary part. This last property implies that all the parasitic waves become evanescent. In Fig. 7, we show the eigenvalues obtained by setting $\beta = 0.1$ then $= 0.5$ and $\delta = 0$ in 3D. One can notice that for both values of the parameter, a lot of parasitic eigenvalues are not too much affected by the change of parameter.

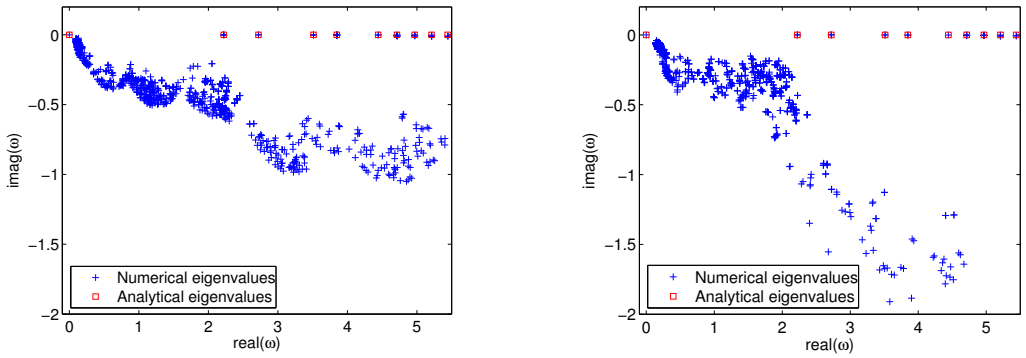

Fig. 7. Eigenvalues in the complex plane for $\beta = 0.1$ (left) and $\beta = 0.5$ (right) computed with DGM

The effect of numerical dissipation on the solution can be seen in Fig. 8, in which show snapshots of the solution (on the $x - y$ plane) in unit cubic cavity after a wave travel of 30 wavelengths.

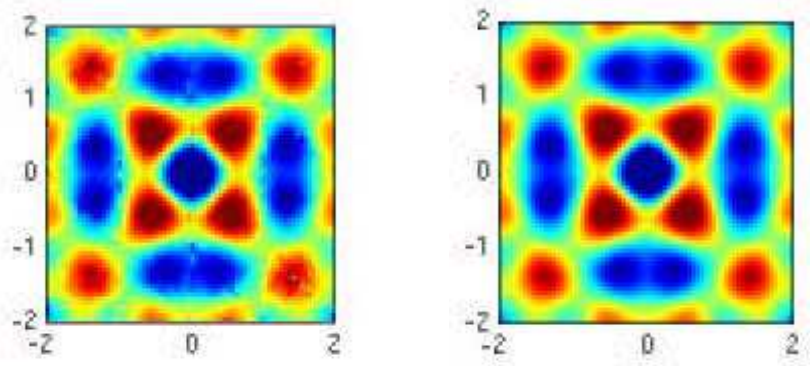

Fig. 8. Snapshots of the solution (on the $x - y$ plane) in unit cubic cavity after a wave travel of 30 wavelengths for $\beta = 0$ (left) and $\beta = 0.1$ (right) computed with DGM

3.2 Second Family of Edge Elements

In the 2D and the 3D case, edge elements present a substantial number of parasitic eigenvalues, even more important in 3D, as shown in Fig. 8. These parasitic eigenvalues confirm the presence of parasitic waves in the solution. A priori, these parasitic eigenvalues seem to make this family of edge elements inefficient. However, this problem can be overcome in a similar way as DGM, by adding some numerical dissipation to this approximation. These dissipative terms can be introduced in (10a)-(10b) as follows

$$\sum_{K_j\in\mathcal{Q}}\left\{\frac{\mathrm{d}}{\mathrm{d}t}\int_{K_j}\underline{\underline{\varepsilon}}\boldsymbol{E}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x}-\int_{K_j}\boldsymbol{H}_h\cdot\nabla\times\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x}+\int_{K_j}\underline{\underline{\sigma}}\boldsymbol{E}_h\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x}\right.$$
$$\left.+\,\beta'\int_{\partial K_j}[\![\mathbf{E}\cdot\mathbf{n}]\!]_{\partial K_j}^{K_j}(\boldsymbol{\psi}\cdot\mathbf{n})\,ds+\int_{K_j}\boldsymbol{J}\cdot\boldsymbol{\varphi}_h\,\mathrm{d}\boldsymbol{x}\right\}=0,\ \forall\boldsymbol{\varphi}_h\in\boldsymbol{V}_h^r, \tag{19a}$$

$$\sum_{K_j\in\mathcal{Q}}\left\{\frac{\mathrm{d}}{\mathrm{d}t}\int_{K_j}\underline{\underline{\mu}}\boldsymbol{H}_h\cdot\boldsymbol{\psi}_h\,\mathrm{d}\boldsymbol{x}+\int_{K_j}\boldsymbol{E}_h\cdot\nabla\times\boldsymbol{\psi}_h\,\mathrm{d}\boldsymbol{x}\right.$$
$$\left.+\ \delta'\int_{\partial K_j}[\![\mathbf{n}\times\mathbf{H}]\!]_{\partial K_j}^{K_j}\cdot(\mathbf{n}\times\boldsymbol{\psi}_h)\,ds\right\}=0,\ \forall\boldsymbol{\psi}_h\in\boldsymbol{W}_h^r. \tag{19b}$$

As for (14)-(15), one can write the following energy identity:

$$\frac{\mathrm{d}\mathcal{E}}{\mathrm{d}t}(t)=\sum_{K_i\cap K_j}\{-\beta'\|[\![\boldsymbol{n}\cdot\boldsymbol{E}]\!]_{K_i\cap K_j}\|_\Gamma^2-\delta'\|[\![\boldsymbol{n}\times\boldsymbol{H}]\!]_{K_i\cap K_j}\|_\Gamma^2\} \tag{20}$$

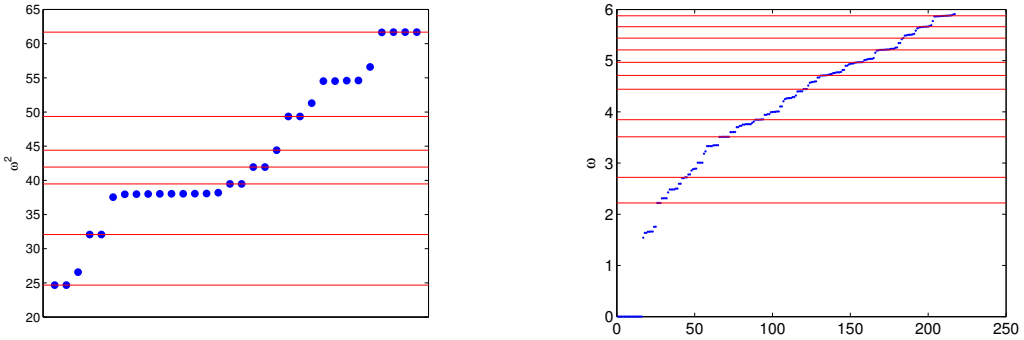

Fig. 9. Eigenvalues for the 2D (left) and 3D (right) cavities computed with edge elements. Physical eigenvalues are indicated by red lines

The use of the above dissipative terms provides the same effect on the eigenvalues as shown in Fig. 10 and 11.

Remarks:

1. The dissipative term in (19a) is not equal to zero because only the tangent components of $\boldsymbol{E}$ are continuous.
2. The dissipative term in (19b) is similar to this of DGM but was not yet tested.
3. Both for DGM and edge elements, the dissipation was tested in cavities for very long times (even for time-harmonic problems) for Q_3 and higher-order approximations. For all these experiments, we did not notice substantial effect on the amplitudes of the solutions.

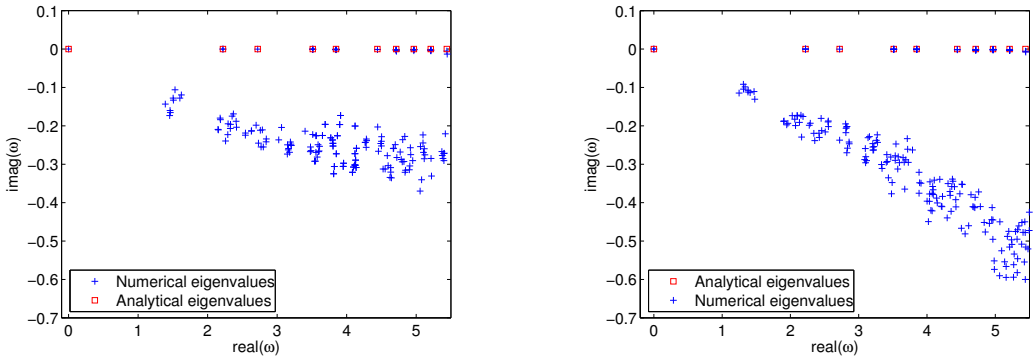

Fig. 10. Eigenvalues in the complex plane for $\beta' = 0.1$ (left) and $\beta' = 0.5$ (right) and $\delta' = 0$ computed with the second family of edge elements

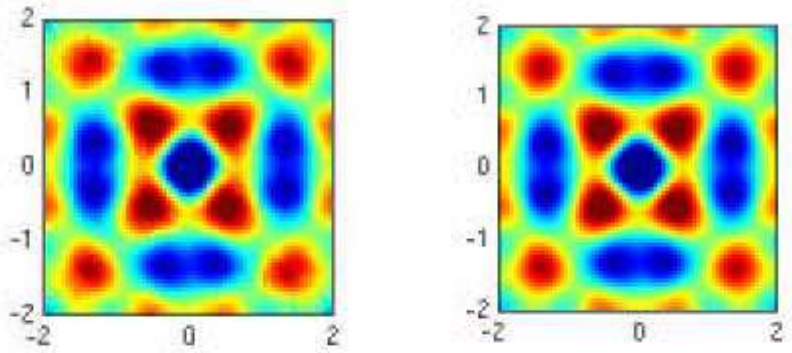

Fig. 11. Snapshots of the solution (on the $x - y$ plane) in unit cubic cavity after a wave travel of 30 wavelengths for $\beta' = 0$ (left) and $\beta' = 0.1$ (right) and $\delta' = 0$ computed with the second family of edge elements

3.3 The Case of the First Family

The first family of edge elements does not provide mass-lumping on non-orthogonal meshes. However, this family is very useful for time-harmonic problem, as we shall see in the next section. Its efficiency is based on a fast matrix-vector product but also on the fact that, for some quadrature rules, this family is free of spurious modes

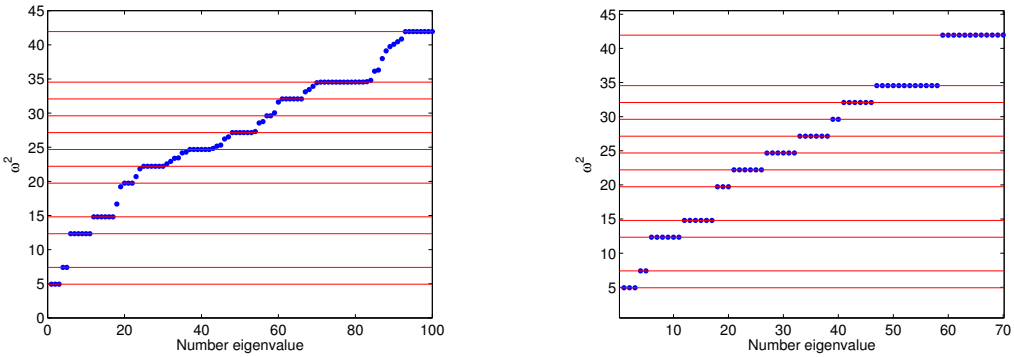

Fig. 12. Eigenvalues for the 3D first family of edge elements with Gauss (left) and Gauss-Lobatto (right) quadrature rules for the stiffness matrix. Physical eigenvalues are indicated by red lines

In fact, due to the structure of its degrees of freedom, both Gauss and Gauss-Lobatto quadrature rules are suitable to this family. For the mass matrices, Gauss points lead to non invertible mass matrices whereas Gauss-Lobatto quadrature rules behave well. The stiffness matrices can also be computed by both quadrature rules. However, as shown in Fig. 12, the Gauss-Lobatto rules leads to a spurious-free approximation.

4 Time-Harmonic Problems

In this section, we only study the first family, which is spurious free and which needs less storage than other methods to reach the same accuracy.

4.1 Fast Matrix-Vector Product

After applying the Fourier transform in time and discretizing the variational formulation (5), we get the following linear system:

$$-\omega^2 \, M_h \, E_h \, + \, K_h \, E_h \, = \, F_h$$

The mass matrix M_h and stiffness matrix K_h can be written as

$$(M_h)_{j,k} \, = \, \int_\Omega \, \varepsilon \varphi_j \cdot \varphi_k$$

$$(K_h)_{j,k} \, = \, \int_\Omega \, \mu^{-1} \, \nabla \times \, \varphi_j \cdot \nabla \times \, \varphi_k$$

After a change of variables, the integrands are computed over the unit cube $\hat{K}$ as follows

$$(M_h)_{j,k} \, = \, \int_{\hat{K}} \, J_i \, DF_i^{-1} \, \varepsilon \, DF_i^{-t} \hat{\varphi}_j \cdot \hat{\varphi}_k$$

$$(K_h)_{j,k} \, = \, \int_{\hat{K}} \, \frac{1}{J_i} DF_i^* \, \mu^{-1} \, DF_i \, \hat{\nabla} \times \, \hat{\varphi}_j \cdot \hat{\nabla} \times \hat{\varphi}_k$$

We have supposed here that the jacobian J_i is always positive. This can be obtained by a permutation on the local numbers of the vertices in each hexahedron.

These two matrices are computed by using Gauss-Lobatto quadrature rules. We need then to compute the following 3x3 symmetric matrices $(A_h)_{k,k}$ and $(B_h)_{k,k}$ at each quadrature point ξ_k^{GL}:

$$(A_h)_{k,k} \, = \, \omega_k^{GL} J_i \, DF_i^{-1} \, \varepsilon \, DF_i^{-t} (\hat{\xi}_k^{GL})$$

$$(B_h)_{k,k} \, = \, \frac{\omega_k^{GL}}{J_i} \, DF_i^* \, \mu^{-1} \, DF_i (\hat{\xi}_k^{GL})$$

Matrices A_h and B_h are block-diagonal matrices, with 3x3 symmetric blocks. An "interpolation" operator $\hat{C}$ is defined by

$$\hat{C}_{j,k} \, = \, \hat{\varphi}_j (\hat{\xi}_k^{GL})$$

In the same way, a discrete "curl" $\hat{R}$ operator is defined by

$$\hat{R}_{j,k} \, = \, \hat{\nabla} \times \hat{\varphi}_j^{GL} (\hat{\xi}_k^{GL}),$$

where functions $\hat{\varphi}_j^{GL}$ are basis functions associated to Gauss-Lobatto points. The reader can notice that matrix $\hat{R}$ is the same as the stiffness matrix introduced for the mixed formulation.

Theorem 1 *With the previous notations, we get the following factorizations:*

$$M_h \, = \, \hat{C} \, A_h \, \hat{C}^*$$
$$K_h \, = \, \hat{C} \, \hat{R} \, B_h \, \hat{R}^* \, \hat{C}^*$$

Proof: Let us write the elementary mass matrix by using quadrature rules:

$$(M_h)_{j,k} \;=\; \sum_{m=1}^{(r+1)^3} (A_h)_{m,m}\, \hat{\varphi}_j(\hat{\xi}_m^{GL}) \cdot \hat{\varphi}_k(\hat{\xi}_m^{GL})$$

If we denote a matrix F_h :

$$(F_h)_{j,m} \;=\; (A_h)_{m,m}\, \hat{\varphi}_j(\hat{\xi}_m^{GL})$$

we have a classical matrix-matrix product :

$$M_h \;=\; F_h \hat{C}^*$$

The matrix F_h can be easily decomposed as

$$F_h \;=\; \hat{C}\, A_h$$

The factorization of the elementary mass matrix is proven. In the same way, we write the elementary stiffness matrix :

$$(K_h)_{j,k} \;=\; \sum_{m=1}^{(r+1)^3} (B_h)_{m,m}\, \hat{\nabla} \times \hat{\varphi}_j(\hat{\xi}_m^{GL}) \cdot \hat{\nabla} \times \hat{\varphi}_k(\hat{\xi}_m^{GL})$$

As for the mass matrix, we have the following factorization

$$K_h \;=\; \hat{S}\, B_h\, \hat{S}^*$$

where matrix $\hat{S}$ is defined by

$$\hat{S} \;=\; \nabla \times \hat{\varphi}_j(\hat{\xi}_k^{GL})$$

$\hat{S}$ is different from $\hat{R}$, because we take basis functions of the first family instead of basis functions associated to Gauss-Lobatto points. But we claim the identity (21).

$$\hat{S} \;=\; \hat{C}\, \hat{R} \tag{21}$$

Because of polynomial inclusion ($Q_{r-1,r,r} \times Q_{r,r-1,r} \times Q_{r,r,r-1} \subset Q_r^3$), we have:

$$\hat{\varphi}_j(\hat{x}) \;=\; \sum_{m=1}^{3(r+1)^3} \hat{C}_{j,m}\, \hat{\varphi}_m^{GL}(\hat{x})$$

By taking the curl of this expression, and for all $\hat{x}$ equal to a Gauss-Lobatto point, we get:

$$\hat{\nabla} \times \hat{\varphi}_j(\hat{\xi}_k^{GL}) \;=\; \sum_{m=1}^{3(r+1)^3} \hat{C}_{j,m}\, \hat{\nabla} \times \hat{\varphi}_m^{GL}(\hat{\xi}_k^{GL})$$

By definition of the operators, this can be rewritten as

$$\hat{S}_{j,k} \;=\; \sum_{m=1}^{3(r+1)^3} \hat{C}_{j,m} \hat{R}_{m,k}$$

Identity (21) is proven, as well as the factorization of the elementary stiffness matrix.

Complexity of the matrix-vector product $(-\omega^2\, M_h + K_h) X_h$ is in $O(r^4)$. Moreover, the number of operations is equal to : $((r+1)^3\,[24(r+1) + 12\,r\, + 30])N_e$, where N_e represents the number of hexahedra of the mesh. The required storage is equal to $12\,(r+1)^3\, N_e$ coefficients, which is equivalent to four vectors if r is large enough. In order to compare the different orders of approximation, we compute the two quantities :

$$\frac{\text{Computational cost}}{\text{Number of d.o.f.s}}$$

$$\frac{\text{Storage cost}}{\text{Number of d.o.f.s}}$$

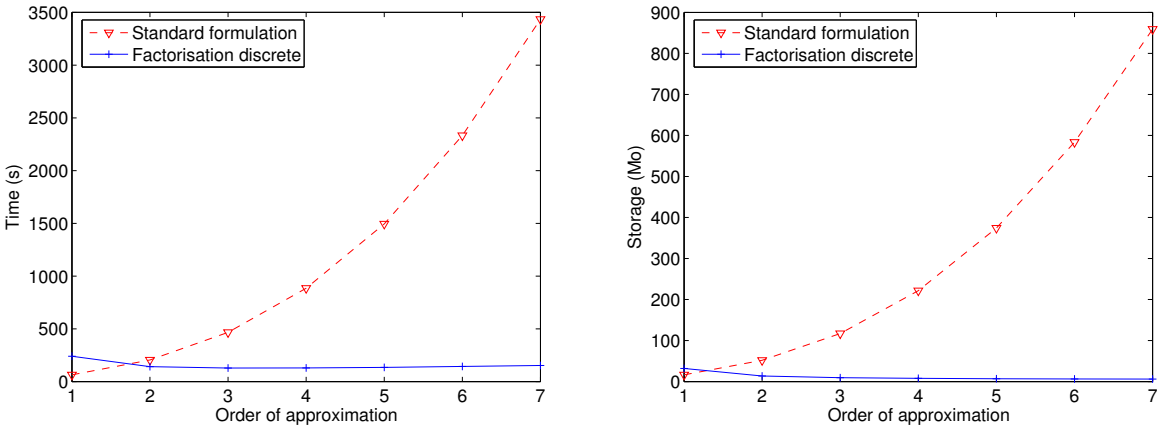

Fig. 13. At left, computational time according to the order of approximation, at right storage required. A standard matrix-vector product (by storing the full matrix) and the fast matrix-vector are compared

These quantites are displayed in Fig. 13 As we can see in this figure, the fast matrix-vector product is more efficient than standard matrix-vector product for order greater or equal to 3, and leads also to a lower storage than standard method for order greater or equal to 2.

Remark: The factorization is also true if an almost-"exact" integration is used ($k+1$ Gauss points instead of $k+1$ Gauss-Lobatto points). In this case, $\hat{R}$ has the same sparsity pattern, but $\hat{C}$ is a full matrix. Fortunatelaty, thanks to tensorization of basis functions on the hexahedron, the triple sum induced by the matrix-vector product $\hat{C}X$ can be decomposed into three single sums. By doing that, we use an implicit factorization: $\hat{C} = \hat{C}_1\,\hat{C}_2\,\hat{C}_3$, where the intermediaries matrices $\hat{C}_1, \hat{C}_2$ and $\hat{C}_3$ are sparse. A fast matrix-vector is obtained, but it is slower than the fast matrix-vector product obtained with approximate integration. More precisely, it is 67 % slower, if we use exact integration. This difference was confirmed numerically.

4.2 Preconditioning Technique

The iterative solver used is an iterative solver which only works for complex symmetric matrices. It is called BICGCR (BIConjugate Gradient Conjugate Residual method) and is described in [4]. Numerical experiments show the superiority of this solver over classical COCG, GMRES or BICGSTAB [15]. Nevertheless, the number of iterations can be very large for high frequency problems with hetereogeneities, and for fine meshes. In order to be efficient for the linear system, this basic iterative solver needs to be accelerated by a preconditioning technique.

The main idea of the preconditioning technique used in this article, is to consider damped Maxwell equations by taking ε' complex of the form :

$$\varepsilon' = \varepsilon\,(\theta + \mathrm{i}\eta)$$

Damping is obtained under the assumption $\eta > 0$. The preconditioner will be constructed on the damped Maxwell equations, while non-damped Maxwell equations are solved.

The effect of damping is to "localize" the inverse of the matrix so that the incomplete factorization works fine. In Table 1, we put the number of iterations of BICGCR preconditioned by ILUT, according to η and different values of the threshold. The incomplete factorization is called ILUT [27], but we only store U factor, considering that the L factor can be recovered by symmetry. We can see, that if you don't put a damping, the incomplete factorization crashes very fast, and there is no gain in storage. The gain in storage increases with damping, but also the number of iterations. The parameters choosen will be equal to $(1, 1)$ for all other numerical experiments on the paper.

In order to have a low storage, we use a subdivided $Q1$ mesh to compute a $Q1$ matrix. The procedure can be summarized by the figure 14. There is an exact matching between degrees of freedom (d.o.f.s) of the high-order mesh and d.o.f.s of low-order provided a diagonal scaling. This diagonal scaling is due to the transform DF_i^{*-1} incorporated in the definition of the basis functions.

Threshold	$1e-4$	$1e-3$	0.01	0.05	0.08	0.1
$\theta = 1\ \eta = 0$	$30/370\ Mo$	$\infty/350\ Mo$	$\infty/340\ Mo$	$\infty/326\ Mo$	$\infty/318\ Mo$	$\infty/314\ Mo$
$\theta = 1\ \eta = 0.5$	$55/299\ Mo$	$55/242\ Mo$	$55/149\ Mo$	$82/74\ Mo$	$116/55 Mo$	$145/47 Mo$
$\theta = 1\ \eta = 1$	$97/244\ Mo$	$97/197\ Mo$	$99/108\ Mo$	$110/53\ Mo$	$133/40 Mo$	$155/34 Mo$

Table 1. Number of iterations of BICGCR for a perfectly conducting sphere, and memory used for factorization. The preconditioning used is incomplete factorization on the finite element matrix Q_1

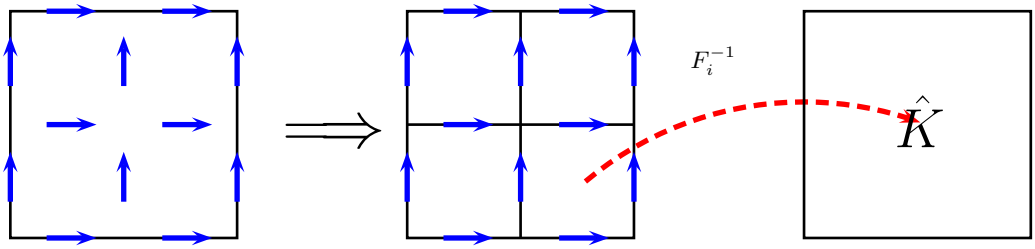

Fig. 14. Subdivided Q_1 mesh of a elementary quadrilateral Q_2 transformation F_i to go from unit square to a small square of the subdivided mesh

4.3 Cost of the Matrix-Vector Product. Comparison with Tetrahedra

In this section, we numerically compare hexahedral and tetrahedral elements for the first family. The results are summarized in Fig. 15. The **mesh** and the matrix are main components of the

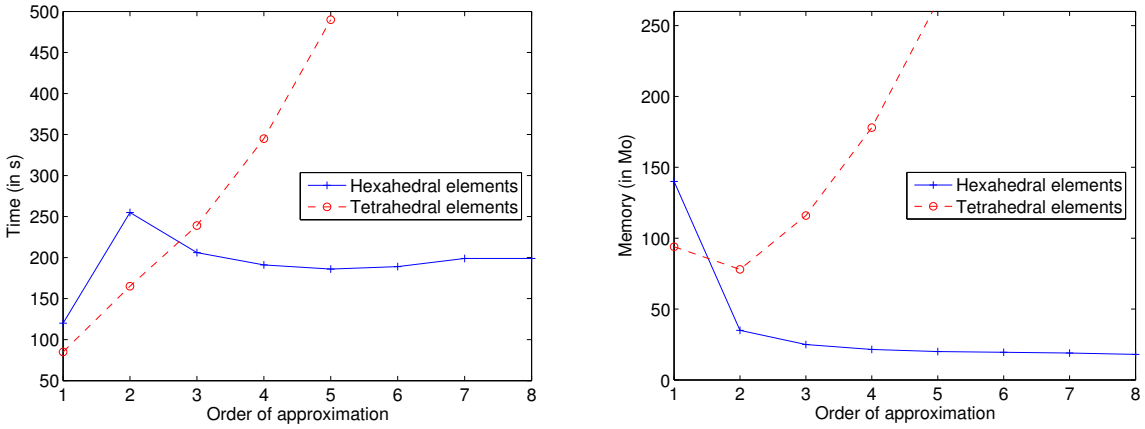

Fig. 15. At left, time for 1 000 iterations of COCG on a test-case of 200 000 d.o.f.s. At right, storage required before iterative solving (iterations vectors are not included)

storage requirement. For Q_1 and R_1, the mesh represents the main part. As we can see, the storage needed decreases for hexahedra, whereas it dramatically increases for tetrahedra. The hexahedral elements provide a matrix-vector product which becomes faster than for tetrahedral elements for

an order of approximation greater than 3. In practice, the use of high order for tetrahedra of the first family is quite difficult because of this high storage.

5 Numerical Results

5.1 A Dielectric Spherical Torus

In this experiment, we treat a metallic sphere coated by a dielectric layer (Fig. 16) in time domain. The radius of the sphere is equal to 0.5m whereas the width of the dielectric is 0.25m. The relative permittivity ε_r in the dielectric is equal to 10. This sphere is illuminated by a Gaussian plane wave of frequency 0.5 GHz.

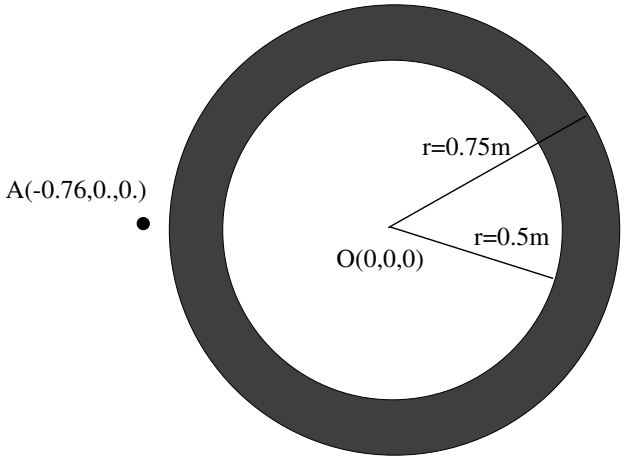

Fig. 16. The dielectric spherical torus

In Fig. 17, we give te he comparison of the value of the electric field at the point (-0.76,0) computed by FDTD (10 and 20 points per wavelength) and our method with a mesh composed of 16984 hexahedra (obtained by split tetrahedra adapted to the wavelength) for Q_3 and Q_5 approximations. The FDTD with 20 points/λ requires 1100s of CPU versus 300s for the Q_3 DG approximation and 10 times more storage for a very bad accuracy in long times.

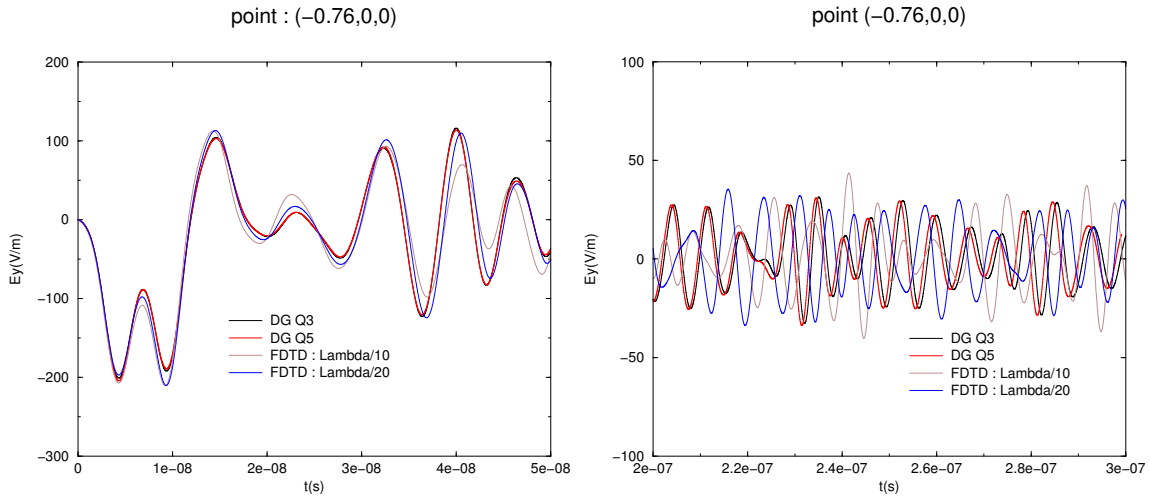

Fig. 17. E_y component of the electric field at a point of the domain after propagation across 10 wavelengths *(left)* and 120 wavelengths *(right)*

5.2 The Cobra Cavity

The scattering of a cobra cavity in time-harmonic domain is studied (cf. Fig 18). Its length is 3m and a plane wave of frequency 1 GHz illuminates its open end. If we want to obtain an error less

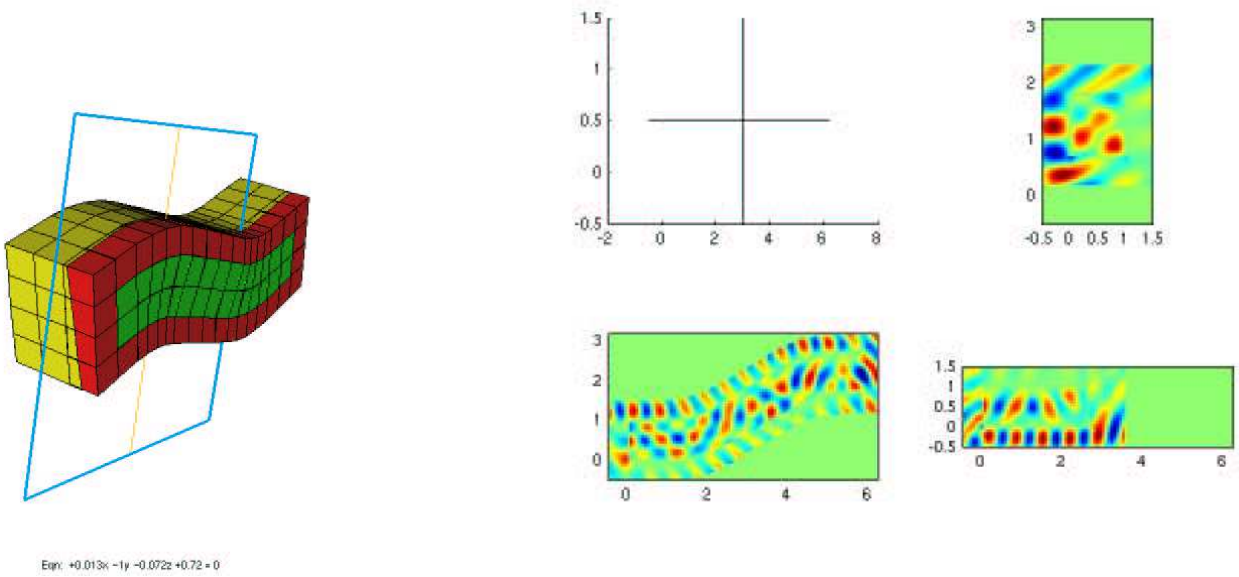

Fig. 18. At left mesh of the cavity, at right real part of x-component of electric field

than 0.5 dB for the RCS (see Fig. 19), we get results of table 2. For a Q_6 approximation, we need

Order	Number d.o.f.s	Error on the RCS	BICGCR	ILUT(0.05)
Q_4	412 000	0.45 dB	14 039 s 34 800 (47 Mo)	2 247 s 1 900 (391 Mo)
Q_6	187 000	0.4 dB	12 096 s 41 500 (22 Mo)	846 s 1 700 (161 Mo)

Table 2. Efficiency of first family on cobra cavity

187 000 d.o.f.s, the computational time is 846s, if we use the ILUT preconditioner. The iterative solver converged in 1700 iterations and needs 161 Mo to run.

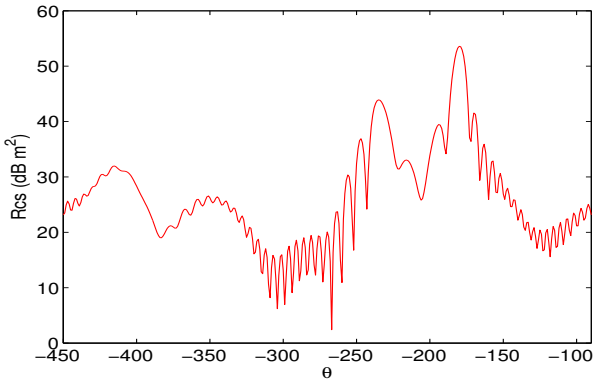

Fig. 19. Radar cross section for the cobra cavity

5.3 Scattering by an Airplane

This last experiment illustrates the possibilities of the methods in terms of storage and CPU time. We consider an airplane of length 11m in the time domain, illuminated by a time-harmonic plane wave of frequency 0.75 GHz. The whole mesh (obtained by split tetrahedra) contains about 78 000 elements and a Q_4 DG approximation was used, which leads to about 30 000 000 of degrees of freedom. The dissipation coefficient β defined in (14) is equal to 0.1. This experiment requires a storage of 0.8 Go and the wave used 25 hours of CPU on a monoprocessor computer (2.4 Go and 3.2 GHz) to reach the end of the plane. In Fig 20, we show the triangular surfacic mesh (before splitting) and a snapshot at the final time. We validated the accuracy of the solution by a comparison with a Q_5 approximation on the same mesh.

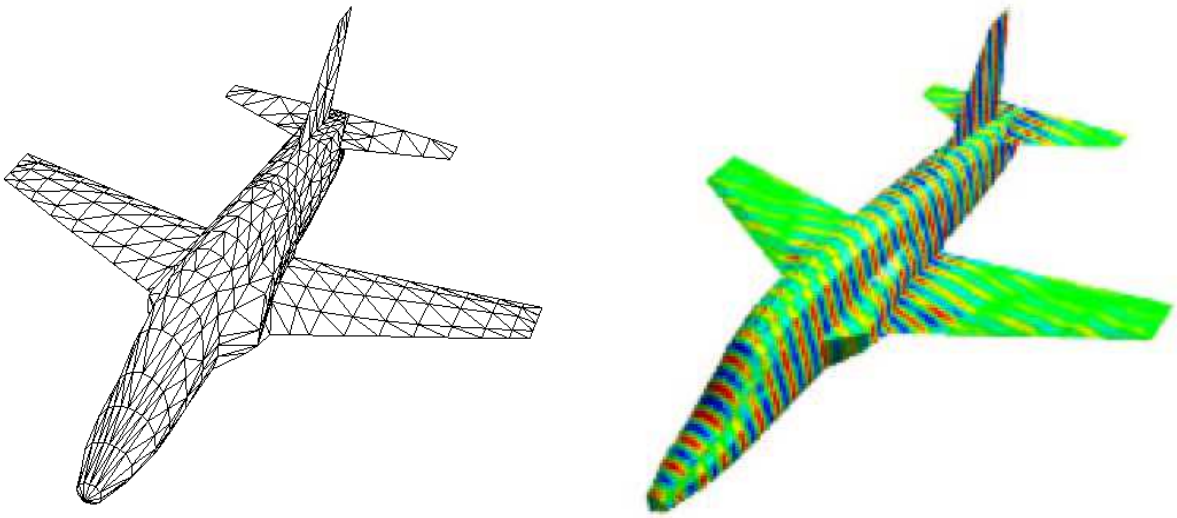

Fig. 20. The plane mesh *(left)* and the final snapshot *(right)*

6 Conclusion

Mixed spectral approximations provide fast an low-storage algorithms both in the time domain and the time-harmonic domain, provided some numerical dissipation. Their advantage over tetrahedral methods is obvious. On the other hand, time domain approximation can efficiently compete with FDTD methods. However, the use of split tetrahedra for meshing remains a handicap for the performance of the method. Some alternative meshing stratetegies are under study.

References

1. D. Boffi, M. Costabel, M. Dauge, L. Demkowicz, *Discrete compactness for the hp version of rectangular edge finite elements*, SIAM Journal on Numerical Analysis, Vol. 44, No. 3, pp. 979-1004, 2006.
2. A. Buffa, I. Perugia, *Discontinuous Galerkin approximation of the Maxwell eigenproblem*, SIAM Numer. Anal., to appear.
3. P. G. Ciarlet, *The finite element method for elliptic problems*, North-Holland, exercises 4.1.4 and following, 1987.
4. M. Clemens, T. Weiland, *Iterative methods for the solution of very large complex symmetric linear systems of equations in electrodynamics*, Technische Hochschule Darmstadt, 2002.
5. G. Cohen, *Higher Order Numerical Methods for Transient Wave Equations* , Springer, 2001.
6. G. Cohen, S. Fauqueux, *Mixed finite elements with mass-lumping for the transient wave equation*, J. Comp. Acous. **8** (1), pp. 171-188, 2000.
7. G. Cohen, S. Fauqueux, *Mixed spectral finite elements for the linear elasticity system in unbounded domains*, SIAM J. Sci. Comput., **26** (3), pp. 864-884, 2005.

8. G. Cohen, X. Ferrieres, S Pernet, *High Spatial Order Finite Element Method to Solve Maxwell's Equations in Time Domain*, IEEE Trans on Antennas and Propagation, **53** (9), pp. 2889-2899, 2005.
9. G. Cohen, X. Ferrieres, S. Pernet, *A Spatial High-Order Hexahedral Discontinuous Galerkin Methods to Solve Maxwell's Equations in Time Domain*, J. Comp. Phys., **217**, pp. 340-363 , 2006.
10. G. Cohen, P. Joly, J. E. Roberts, N. Tordjman, *Higher order triangular finite elements with mass lumping for the wave equation.*, SIAM J. on Numer. Anal., **38** (6), pp. 2047-2078, 2001.
11. G. Cohen, P. Joly, N. Tordjman, *Higher-order finite elements with mass lumping for the 1-D wave equation*, Finite elements in analysis and design, **17** (3-4), pp. 329-336, 1994.
12. G. Cohen, P. Monk, *Gauss point mass lumping schemes for Maxwell's equations*, NMPDE Journal **14** (1), pp. 63-88, 1998.
13. G. Cohen, P. Monk, *Mur-Nédélec finite element schemes for Maxwell's equations*, Comp. Meth. in Appl. Mech. Eng., **169** (3-4), pp. 197-217, 1999.
14. M. Costabel, M. Dauge *Computation of resonance frequencies for Maxwell equations in non smooth domains*, Lecture notes in computational science and engineering, Vol. 31, Springer, 2003
15. M. Duruflé, *Integration numérique et éléments finis d'ordre élevé appliqués aux équations de Maxwell en régime harmonique*, PhD thesis, U. de Paris-Dauphine, 2006.
16. A. Elmkies, P. Joly, *Éléments finis d'arête et condensation de masse pour les équations de Maxwell : le cas 2D*, C.R.A.S., Math., **324**, série I, pp. 1287-1293, 1997.
17. A. Elmkies, P. Joly, *Éléments finis d'arête et condensation de masse pour les équations de Maxwell : le cas de la dimension 3*, C.R.A.S., Math., **325**, série I, pp. 1217-1222, 1997.
18. Y. Haugazeau, P. Lacoste,, *Condensation de la matrice masse pour les éléments finis mixtes de H(rot)*, C. R. Acad. Sci. Paris Sér. I Math., **316** (5), pp. 509–512, 1993.
19. J.-P. Hennart, E. Sainz, M. Villegas, *On the efficient use of the finite element method in static neutron diffusion calculations*, Computational Methods in Nuclear Engineering, **1**, pp. 3–87, 1979.
20. J.-P. Hennart, *Topics in finite element discretization of parabolic evolution problems*, Lecture Notes in Mathematics, Springer-Verlag, **909**, pp. 185–199, 1982.
21. J. S. Hesthaven, T. Warburton, *High-order nodal methods on unstructured grids. I. Time-domain solution of Maxwell's equations*, J. Comput. Phys. **181**, pp. 1–34, 2002.
22. Y. Maday, A. T. Patera, *Spectral element methods for the imcompressible Navier-Stokes equations*, State of the Art Survey in Computational Mechanics, ed. A. K. Noor, pp. 71-143, 1989.
23. W. A. Mulder, *Higher-order mass-lumped finite elements for the wave equation*, J. Comput. Acoustics, **9** (2), pp. 671-680, 2001.
24. J.-C. Nédélec, *Mixed finite elements in $\mathbb{R}^3$*, Numer.Math., **35** (3), pp. 315-341, 1980.
25. J.-C. Nédélec, *A new family of mixed finite elements in $\mathbb{R}^3$*, Numer.Math., **50** (1), pp. 57-81, 1986.
26. S. Pernet, *Étude de méthodes d'ordre élevé pour résoudre les équations de Maxwell dans le domaine temporel. Application à la détection et à la compatibilité électromagnétique*, PhD thesis, U. de Paris-Dauphine, 2004.
27. Y. Saad, *Iterative methods for sparse linear systems*, Series in Computer Science, 1996.
28. K. Yee, *Numerical solutions of initial boundary value problems involving Maxwell's equations in isotropic media*, IEEE Trans. on Antennas and Propagation AP-16, pp. 302-307, 1966.
29. L. C. Young, *An efficient finite element method for reservoir simulation*, Proc. of the 53rd Annual Fall Technical Conference and Exhibition of the Society of Petroleum Engineers of AIME, Houston, Texas Oct. 1-3, 1978.