

High-order discontinuous Galerkin method for time-domain electromagnetics on non-conforming hybrid meshes

Hassan Fahs*

GPEM Group, Department of Materials & Structures, Nantes University, IFSTTAR, Route de Bouaye, 44344 Bouguenais Cedex, France

Abstract

We present a high-order discontinuous Galerkin (DG) method for solving the time-dependent Maxwell equations on non-conforming hybrid meshes. The hybrid mesh combines unstructured tetrahedra for the discretization of irregularly shaped objects with a hexahedral mesh for the rest of the computational domain. The transition between tetrahedra and hexahedra is completely non-conform, that is, no pyramidal or prismatic elements are introduced to link these elements. Within each mesh element, the electromagnetic field components are approximated by a arbitrary order nodal polynomial and a centered approximation is used for the evaluation of numerical fluxes at inter-element boundaries. The time integration of the associated semi-discrete equations is achieved by a fourth-order leap-frog scheme. The method is described and discussed, including algorithm formulation, stability, and practical implementation issues such as the hybrid mesh generation and the computation of flux matrices with cubature rules. We illustrate the performance of the proposed method on several two- and three-dimensional examples involving comparisons with DG methods on single element-type meshes. The results show that the use of non-conforming hybrid meshes in DG methods allows for a notable reduction in computing time without sacrificing accuracy.

Keywords: Maxwell's equations, discontinuous Galerkin method, hybrid meshes, non-conformal meshes

1. Introduction

Nowadays, a variety of modeling strategies exist for the computer simulation of electromagnetic (EM) wave propagations in the time-domain. In particular, the ability to deal with complex geometries has been extensively studied in the context of unstructured methods as alternatives to the well established FDTD method [53]. In spite of its versatility, the conventional FDTD method suffers from serious accuracy degradation when modeling curved objects and small geometrical features. This is due to the commonly used staircase approximation (Cartesian grid) which introduces large discretization errors, even when the grid size is very small. In contrast, unstructured solvers such as the finite element time domain (FETD) [43] or the finite volume time domain (FVTD) [44] methods can easily handle complex boundaries by dealing with unstructured tetrahedral meshes. Both methods are more expensive than the FDTD method in terms of computing time and memory requirement. This motivates the development of hybrid solution strategies, in which the computational efficiency of the Cartesian grid is combined with the geometric flexibility of the unstructured mesh. The FDTD is then used in large homogeneous volumes of the computational domain while the FVTD or FETD methods are employed in small regions near small scale features. For instance, hybrid FETD-FDTD methods have been developed [1, 55], but they suffer from spurious reflections and late-time instabilities, which limit their usefulness. A stable hybrid FETD-FDTD method has been described in [47]. To achieve the time-stability, the scheme requires an implicit time integration in the unstructured elements but also a transition layer of pyramids between the FDTD cubes and tetrahedra to ensure the mesh conformity. Low order hybrid FDTD-FVTD methods are proposed in [23, 31]. These methods require the use of an intermediate layer of hexahedra between the Cartesian grid and the tetrahedral mesh. Each cell in the layer is treated by the FVTD method,

*Corresponding author

Email address: Hassan.Fahs@gmail.com, Hassan.Fahs@ifsttar.fr (Hassan Fahs)

which offers a correspondence between a quadrilateral face and two triangles in terms of the exchange of fluxes. Again, weak instabilities have been reported for long-time simulations.

The last few years have seen an increasing interest in so-called discontinuous Galerkin time domain (DGTD) methods. The DG methods are a class of FE methods based on completely discontinuous piecewise polynomial spaces for the numerical solution and the test functions. To obtain highly accurate and stable DG methods, suitable numerical fluxes need to be designed over elemental interfaces. We refer to the lecture notes [10] and the textbook [38] for details and history about DG methods. In particular, DG methods can easily handle elements of various types and shapes [32], irregular non-conforming meshes [15, 28], and even locally varying polynomial degree [24], and hence offer great flexibility in the mesh design. They also lead to (block-) diagonal mass matrices and therefore yield fully explicit, inherently parallel methods when coupled with explicit time schemes. Furthermore, DG methods are flexible with regards to the choice of the time-stepping scheme [27]. The DG methods have now acquired a level of maturity and have successfully penetrated several scientific and technological communities following to their adaptation to increasingly complex modeling contexts [7, 29, 49]. It is worthwhile to note that the DG method has also been adopted for the first time in a commercial software as the time domain alternative of a very well known EM wave simulation tool [51]. While DG methods on tetrahedral meshes have been extensively studied [15, 29, 37], their development on hexahedral meshes has received relatively less attention [11]. One reason is that tetrahedral meshing is highly automated while high quality hexahedral meshing of complex geometries commonly requires user intervention and is labor intensive. This motivates the development of DG methods on conforming hybrid meshes consisting of tetrahedra and hexahedra with transition layers of prisms or pyramids [5, 50]. Despite these advances, the exploitation of non-conforming hybrid meshes in the context of a DG-based numerical methodology applied to wave propagations, has been the subject of a very few studies so far. A 2D contribution in this context is the work of Durochat et al. [19, 20] who proposed a DG method formulated on hybrid rectangular-triangular meshes for Maxwell's equations.

We are concerned here with the design of a high-order nodal DGTD method for solving Maxwell's equations on non-conforming hybrid meshes in 2D and 3D. The proposed method is based on centered numerical fluxes and a fourth-order leap-frog time integration scheme. The hybrid mesh adopted here combines tetrahedral elements for the discretization of irregularly shaped objects with hexahedral elements to fill the rest of the domain. Thus, the resulting mesh is non-conformal in the sense that hanging nodes are allowed and no additional element types (e.g., pyramids or prisms) are needed to join tetrahedra with hexahedra. The non-conformal connection between these elements is ensured thanks to the correct computation of flux terms by using efficient cubature rules over polygons. The motivation behind this work stems partly from the following observations: first, the final mesh of an object is made of separately constructed meshes and a conforming assembling is a very hard task. A good example to illustrate this point is the interior penalty DGTD method formulated on non-conforming tetrahedral meshes presented in [15] for the EM simulation of 3D integrated circuit packaging. For the complicated geometries involved in such applications, the construction of a conforming mesh is a very hard task. Instead, a subdomain-based meshing approach is particularly attractive and allowing meshes with different element-types is highly desirable in this context. Second, regarding the computational effort, it turns out to be more efficient to compute on hexahedral meshes instead of tetrahedral meshes to reach a desired error level. Therefore, the performance of the DG method can be increased by combining tetrahedra with hexahedra. The outline of the paper is as follows. In Section 2 we briefly recall the basic features of DGTD discretizations based on totally centered numerical fluxes and formulated on single element-type meshes composed of triangles or quadrilaterals in 2D and tetrahedra or hexahedra in 3D. Then, we formulate the DG method on non-conforming hybrid triangular-quadrilateral or tetrahedral-hexahedral meshes in Section 3. Subsequently, we analyze the stability of the resulting algorithm and establish a sufficient CFL-like condition. Next, in Section 4, we discuss some practical implementation issues including the generation of non-conforming hybrid meshes and the computation of the flux matrices at hybrid interfaces between adjacent elements of different types. Finally, we present numerical results for several test-cases in Section 5, and draw some conclusions of the study in Section 6.

2. Governing equations and DGTD formulation

We consider the time-dependent Maxwell equations for linear, isotropic, non-dispersive media. The electric field \mathbf{E} and the magnetic field \mathbf{H} verify

$$\varepsilon(\mathbf{x})\partial_t\mathbf{E}(\mathbf{x}, t) = \nabla \times \mathbf{H}(\mathbf{x}, t), \quad \mu(\mathbf{x})\partial_t\mathbf{H}(\mathbf{x}, t) = -\nabla \times \mathbf{E}(\mathbf{x}, t), \quad (1)$$

where ε and μ are respectively the permittivity and the permeability of the medium. These equations are posed on a bounded domain Ω of \mathbb{R}^d , $d \geq 2$. At the domain boundary $\partial\Omega$, we apply either perfect electric conductor (PEC) conditions ($\mathbf{n} \times \mathbf{E} = 0$) or absorbing boundary conditions ($\mathbf{n} \times \mathbf{E} + c\mu\mathbf{n} \times (\mathbf{n} \times \mathbf{H}) = 0$). Here \mathbf{n} denotes the unit outward normal to $\partial\Omega$ and $c = (\varepsilon\mu)^{-1/2}$ is the speed of propagation.

We assume we dispose of a regular or irregular partition \mathcal{T}_h of Ω into n_e non-overlapping elements κ_i , which can be triangles or quadrilaterals in 2D and tetrahedra or hexahedra in 3D. Thus a $(d-1)$ -dimensional face of each element κ_i in \mathcal{T}_h is allowed to contain hanging nodes. We shall suppose that each $\kappa_i \in \mathcal{T}_h$ is the image of a fixed master element κ_r , under a smooth bijective mapping $\Psi_{\kappa_i} : \kappa_r \ni \boldsymbol{\xi} \rightarrow \mathbf{x} \in \kappa_i$, with nonsingular Jacobian $\mathbf{J}_{\kappa_i} = \partial\mathbf{x}/\partial\boldsymbol{\xi}$, that is, $\kappa_i = \Psi_{\kappa_i}(\kappa_r)$ for all $\kappa_i \in \mathcal{T}_h$, where κ_r is either the open unit simplex $\mathcal{T}^d = \{\boldsymbol{\xi} = (\xi_1, \dots, \xi_d) \in \mathbb{R}^d : \xi_i \geq 0, \sum_{i=1}^d \xi_i \leq 1\}$ or the open unit hypercube $\mathcal{D}^d = \{\boldsymbol{\xi} = (\xi_1, \dots, \xi_d) \in \mathbb{R}^d : 0 \leq \xi_i \leq 1\}$ in \mathbb{R}^d . Within this construction we allow for elements κ_i with curved edges or faces. On the master element κ_r , with $\boldsymbol{\xi} = (\xi_1, \dots, \xi_d) \in \kappa_r$ and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$, we define spaces of polynomials of total degree $\ell \geq 1$ as follows

$$\begin{aligned} \mathbb{Q}_\ell(\mathcal{D}^d) &= \text{span}\{\boldsymbol{\xi}^\boldsymbol{\alpha} : 0 \leq \alpha_i \leq \ell, 1 \leq i \leq d\}, \\ \mathbb{P}_\ell(\mathcal{T}^d) &= \text{span}\{\boldsymbol{\xi}^\boldsymbol{\alpha} : 0 \leq |\boldsymbol{\alpha}| \leq \ell\}. \end{aligned}$$

The dimension of each of these spaces depends on the spatial dimension d and on the degree ℓ and is given by

$$N(\ell, d) = \begin{cases} (\ell + 1)^d & \text{if } \kappa_r = \mathcal{D}^d, \\ (\ell + d)!/\ell!d! & \text{if } \kappa_r = \mathcal{T}^d. \end{cases}$$

As a particular basis for $\mathbb{Q}_\ell(\mathcal{D}^d)$ and $\mathbb{P}_\ell(\mathcal{T}^d)$ consider the fundamental polynomials $\{L_k^{(\ell)}\}_{k=1}^{N_\ell}$ of multivariate Lagrange interpolation, corresponding to a set of nodal points $\mathcal{S}_\ell = \{\eta_j, j = 1, \dots, N_\ell\}$, such that $L_k^{(\ell)}(\eta_j) = \delta_{jk}$ for $\eta_j \in \mathcal{S}_\ell$. The distribution of the nodes η_j is a key issue for the properties of the interpolation, especially for very high-order approximations. For the unit simplex \mathcal{T}^d , we choose the nodal set derived from the electrostatics principle [35, 36], while the Gauss-Legendre-Lobatto points are used for the hypercube \mathcal{D}^d .

To each $\kappa_i \in \mathcal{T}_h$ we assign an integer $\ell_i \geq 1$ that is the local polynomial degree inside κ_i ; collecting the ℓ_i and Ψ_{κ_i} in the vectors $\boldsymbol{\ell} = \{\ell_i : \kappa_i \in \mathcal{T}_h\}$ and $\boldsymbol{\Psi} = \{\Psi_{\kappa_i} : \kappa_i \in \mathcal{T}_h\}$, respectively, we introduce the finite element space

$$\begin{aligned} V_p(\Omega, \mathcal{T}_h, \boldsymbol{\Psi}) &= \left\{ \mathbf{u} \in [L^2(\Omega)]^d : \mathbf{u}|_{\kappa_i} \circ \Psi_{\kappa_i} \in [\mathbb{Q}_{\ell_i}(\kappa_r)]^d \text{ if } \Psi_{\kappa_i}^{-1}(\kappa_i) = \mathcal{D}^d, \right. \\ &\quad \left. \text{and } \mathbf{u}|_{\kappa_i} \circ \Psi_{\kappa_i} \in [\mathbb{P}_{\ell_i}(\kappa_r)]^d \text{ if } \Psi_{\kappa_i}^{-1}(\kappa_i) = \mathcal{T}^d, \quad \forall \kappa_i \in \mathcal{T}_h \right\}, \end{aligned}$$

where $[L^2(\Omega)]^d$ is the space of square-integrable functions on Ω . For two distinct elements κ_i and κ_k in \mathcal{T}_h , the intersection $a_{ik} = \kappa_i \cap \kappa_k$ is a $(d-1)$ -dimensional face in \mathcal{T}_h , with unitary normal vector \mathbf{n}_{ik} , oriented from κ_i towards κ_k . For the boundary faces, the index k corresponds to a fictitious element outside the domain. Finally, we denote by \mathcal{F}^I and \mathcal{F}^B the union of all interior and boundary faces of \mathcal{T}_h , respectively.

The derivation of the DG scheme for the simulation of electromagnetic wave propagation on purely simplicial meshes has already been published in previous works [24, 30]. Therefore, we avoid the repetition of a comprehensive derivation and refer the reader to these works for details. In brief, the electric and magnetic fields are approximated inside each finite element κ_i by a linear combination of basis functions $\varphi_{ij} = L_j^{(\ell_i)} \circ \Psi_{\kappa_i}^{-1}$ of degree ℓ_i with support κ_i and with time-dependent coefficient functions $E_{ij}(t)$ and $H_{ij}(t)$ as follows

$$\mathbf{E}_i(\mathbf{x}, t) = \sum_{1 \leq j \leq N_i} E_{ij}(t)\varphi_{ij}(\mathbf{x}), \quad \mathbf{H}_i(\mathbf{x}, t) = \sum_{1 \leq j \leq N_i} H_{ij}(t)\varphi_{ij}(\mathbf{x}). \quad (2)$$

Here, the index j indicates the j th basis function and $N_i = N(\ell_i, d)$ is the number of degrees of freedom inside κ_i . We denote by $\bar{\mathbf{E}}_i$ and $\bar{\mathbf{H}}_i$, respectively, the column vectors $(E_{ij})_{1 \leq j \leq N_i}$ and $(H_{ij})_{1 \leq j \leq N_i}$. As usual for DG schemes, the Maxwell system, Eq. (1), is multiplied by a test function $\varphi \in \text{span}\{\varphi_{ij}, 1 \leq j \leq N_i\}$ and integrated over each κ_i . After integration by parts, inserting the DG approximation, Eq. (2), and after applying a centered numerical flux in the boundary integrals, the semi-discrete formulation of the scheme in the physical element κ_i reads as

$$\begin{aligned} \mathbf{M}_i^\varepsilon \partial_t \bar{\mathbf{E}}_i &= \mathbf{K}_i \bar{\mathbf{H}}_i - \sum_{a_{ik} \in \mathcal{F}^I} \mathbf{S}_{ik} \bar{\mathbf{H}}_k - \sum_{a_{ik} \in \mathcal{F}^B} \mathbf{S}_{ik} \bar{\mathbf{H}}_k, \\ \mathbf{M}_i^\mu \partial_t \bar{\mathbf{H}}_i &= -\mathbf{K}_i \bar{\mathbf{E}}_i + \sum_{a_{ik} \in \mathcal{F}^I} \mathbf{S}_{ik} \bar{\mathbf{E}}_k + \sum_{a_{ik} \in \mathcal{F}^B} \mathbf{S}_{ik} \bar{\mathbf{E}}_k, \end{aligned} \quad (3)$$

where the mass matrices \mathbf{M}_i^γ (γ stands for ε or μ), the stiffness matrix \mathbf{K}_i and the interface (or flux) matrix \mathbf{S}_{ik} write

$$(\mathbf{M}_i^\gamma)_{jl} = \gamma_i \int_{\kappa_i} \varphi_{ij} \cdot \varphi_{il}, \quad (\mathbf{K}_i)_{jl} = \frac{1}{2} \int_{\kappa_i} \varphi_{ij} \cdot \nabla \times \varphi_{il} + \varphi_{il} \cdot \nabla \times \varphi_{ij}, \quad (\mathbf{S}_{ik})_{jl} = \frac{1}{2} \int_{a_{ik}} \varphi_{ij} \cdot (\varphi_{kl} \times \mathbf{n}_{ik}). \quad (4)$$

These matrices involve volume and surface integrals which are evaluated numerically over the master element, κ_r , using the mapping Ψ_{κ_i} . For straight-sided simplices and parallelepipedic elements, all integrals can be computed using affine mappings. Since the Jacobian of the affine mapping is constant, the matrices in Eq. (4) can be precomputed and stored for κ_r in advance of the main calculation once and for all. Straight-sided convex quadrilateral and hexahedral elements are expressed by isoparametric bilinear and trilinear mappings [40, 41]. These mappings are not linear and the Jacobian is not constant over a quadrilateral and hexahedron anymore. Thus, all metric terms relating to Ψ_{κ_i} are evaluated, and need to be stored for each node in the elements. Finally, we also allow for curved elements which are usually present inside \mathcal{T}_h (e.g., around a boundary layer or a curved material interface) or in the vicinity of $\partial\Omega$. These are introduced in order to avoid geometrical errors and to achieve optimal convergence for high-order polynomials. In our implementation we use quadratic interpolation of curved boundaries, and define isoparametric mappings based on that approximation of the surface [4, 6, 26]. For such mappings, the Jacobians are not constants and all matrices should be computed and stored for each curved element.

Once the local mapping Ψ_{κ_i} is introduced for each κ_i , the integrals in Eq. (4) are evaluated using cubature formulas which are able to accurately integrate high-order polynomials over the master element κ_r . A cubature is a set of N^c 2D or 3D points $\{\lambda_k^c\}_{k=1}^{k=N^c}$ with N^c associated weights $\{\omega_k^c\}_{k=1}^{k=N^c}$, where the number of points N^c depends on the desired maximum order of polynomial required to be accurately integrated. Surveys of cubature formulas were performed in [12, 14] and an on-line database containing many of the best known rules is described in [13]. Here, we adopt the efficient high-order symmetric rules proposed in [16–18, 54, 56] for hypercubes and simplices. Finally, edge integrals are calculated using Gauss-Legendre quadrature. We evaluate

$$\begin{aligned} (\mathbf{M}_i^\gamma)_{jl} &= \gamma_i \sum_{k=1}^{k=N^c} \omega_k^c \left[L_j^{(\ell_i)} L_l^{(\ell_i)} \|\mathbf{J}_{\kappa_i}\| \right] \Big|_{\lambda_k^c}, \\ (\mathbf{K}_i)_{jl} &= \frac{1}{2} \sum_{k=1}^{k=N^c} \omega_k^c \left[L_j^{(\ell_i)} (\mathbf{j}_{\kappa_i})_{jl} \nabla \times L_l^{(\ell_i)} + L_l^{(\ell_i)} (\mathbf{j}_{\kappa_i})_{jl} \nabla \times L_j^{(\ell_i)} \right] \Big|_{\lambda_k^c}, \\ (\mathbf{S}_{ik})_{jl} &= \frac{1}{2} \sum_{k=1}^{k=N^c} \omega_k^c \left[L_j^{(\ell_i)} L_l^{(\ell_i)} \|\nabla \mathbf{J}_{\kappa_i}\| \right] \Big|_{\lambda_k^c}, \end{aligned} \quad (5)$$

where each term in the brackets of the right hand sides are evaluated at the cubature points, $\|\mathbf{J}_{\kappa_i}\|$ is the determinant of the Jacobian matrix, and $\mathbf{j}_{\kappa_i} = \|\mathbf{J}_{\kappa_i}\| \mathbf{J}_{\kappa_i}^{-1}$.

The set of local system of ordinary differential equations for each κ_i , Eq. (3), can be formally transformed in a global system. To this end, we suppose that all electric (resp. magnetic) unknowns are gathered in a column vector \mathbb{E} (resp. \mathbb{H}) of size $N_g = \sum_{i=1}^{n_e} N_i$. Then system (3) can be rewritten as

$$\begin{aligned} \mathbf{M}^\varepsilon \partial_t \mathbb{E} &= \mathbf{KH} - \mathbf{AH} - \mathbf{BH}, \\ \mathbf{M}^\mu \partial_t \mathbb{H} &= -\mathbf{KE} + \mathbf{AE} - \mathbf{BE}, \end{aligned} \quad (6)$$

where we have the following definitions and properties: (a) $\mathbb{M}^\varepsilon, \mathbb{M}^\mu$ and \mathbb{K} are $N_g \times N_g$ block diagonal matrices with diagonal blocks equal to $\mathbb{M}_i^\varepsilon, \mathbb{M}_i^\mu$, and \mathbb{K}_i respectively. \mathbb{M}^ε and \mathbb{M}^μ are symmetric positive definite matrices, and \mathbb{K} is a symmetric matrix; (b) \mathbb{A} is also a $N_g \times N_g$ symmetric block sparse matrix, whose non-zero blocks are equal to \mathbb{S}_{ik} when $a_{ik} \in \mathcal{F}^I$; (c) \mathbb{B} is a $N_g \times N_g$ skew-symmetric block diagonal matrix, whose non-zero blocks are equal to \mathbb{S}_{ik} when $a_{ik} \in \mathcal{F}^B$. Let $\mathbb{S} = \mathbb{K} - \mathbb{A} - \mathbb{B}$; the system (6) rewrites as

$$\begin{aligned} \mathbb{M}^\varepsilon \partial_t \mathbb{E} &= \mathbb{S} \mathbb{H}, \\ \mathbb{M}^\mu \partial_t \mathbb{H} &= -\mathbb{S}^\dagger \mathbb{E}. \end{aligned} \quad (7)$$

where the symbol \dagger stands for the matrix transpose. To define a fully discrete scheme, we divide the time interval $[0, T]$ into M_t uniform subintervals such that $0 = t_0 < t_1 < \dots < t_{M_t} = T$, where $t_n = n\Delta t$ and Δt is the fixed time step. Time integration of the semi-discrete system, Eq. (7), is done using a fourth-order explicit leap-frog scheme. Let us denote $\mathbb{E}^n = \mathbb{E}(t_n)$ and $\mathbb{H}^{n+1/2} = \mathbb{H}(t_{n+1/2})$, the fully discrete central DGTD method writes

$$\begin{cases} \mathbb{M}^\varepsilon \frac{\mathbb{E}^{n+1} - \mathbb{E}^n}{\Delta t} = \mathbb{G} \mathbb{H}^{n+1/2}, \\ \mathbb{M}^\mu \frac{\mathbb{H}^{n+1/2} - \mathbb{H}^{n-1/2}}{\Delta t} = -\mathbb{G}^\dagger \mathbb{E}^{n+1}, \end{cases} \quad (8)$$

where $\mathbb{G} = \mathbb{S} \left(\mathbb{I} - \frac{\Delta t^2}{24} (\mathbb{M}^\mu)^{-1} \mathbb{S}^\dagger (\mathbb{M}^\varepsilon)^{-1} \mathbb{S} \right)$ and \mathbb{I} is the identity matrix. The resulting DGTD method is analyzed in [25, 30] on purely simplicial meshes where it is shown that the method is non-dissipative, conserves a discrete form of the EM energy and is stable under the CFL-like condition

$$\Delta t \leq 2 / \| (\mathbb{M}^\varepsilon)^{-1/2} \mathbb{G} (\mathbb{M}^\mu)^{-1/2} \|,$$

where $\| \cdot \|$ denotes any matrix norm, and the matrix $(\mathbb{M}^\gamma)^{-1/2}$ is the inverse square root of \mathbb{M}^γ (γ stands for ε or μ).

3. DGTD method on hybrid tetrahedral-hexahedral meshes

3.1. The numerical scheme

The set of elements κ_i of the mesh is now assumed to be partitioned into two subsets: one made of the tetrahedral (triangular in 2D) elements and the other one gathering the hexahedral (quadrilateral in 2D) elements. In the following, these two subsets are respectively referred as \mathcal{S}_τ and \mathcal{S}_h . There is no need of a particular assumption on the connectivity of the two subsets. In order to further describe this scheme, we introduce additional definitions. First, the problem unknowns are reordered as

$$\mathbb{E} = \begin{pmatrix} \mathbb{E}_\tau \\ \mathbb{E}_h \end{pmatrix} \quad \text{and} \quad \mathbb{H} = \begin{pmatrix} \mathbb{H}_\tau \\ \mathbb{H}_h \end{pmatrix},$$

where subvectors with a τ subscript (resp., a h subscript) are associated to the elements of the set \mathcal{S}_τ (resp., the set \mathcal{S}_h). We deduce from this partitioning of the unknown vectors the following decompositions of the system matrices

$$\mathbb{M}^\varepsilon = \begin{pmatrix} \mathbb{M}_\tau^\varepsilon & 0 \\ 0 & \mathbb{M}_h^\varepsilon \end{pmatrix}, \quad \mathbb{M}^\mu = \begin{pmatrix} \mathbb{M}_\tau^\mu & 0 \\ 0 & \mathbb{M}_h^\mu \end{pmatrix}, \quad \mathbb{K} = \begin{pmatrix} \mathbb{K}_\tau & 0 \\ 0 & \mathbb{K}_h \end{pmatrix}, \quad \mathbb{B} = \begin{pmatrix} \mathbb{B}_\tau & 0 \\ 0 & \mathbb{B}_h \end{pmatrix}.$$

where $\mathbb{M}_{\tau/h}^\varepsilon$ and $\mathbb{M}_{\tau/h}^\mu$ are symmetric positive definite matrices, $\mathbb{K}_{\tau/h}$ are symmetric matrices and $\mathbb{B}_{\tau/h}$ are skew-symmetric matrices. The matrix \mathbb{A} which involves the interface matrices \mathbb{S}_{ik} is decomposed as

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_{\tau\tau} & \mathbb{A}_{\tau h} \\ \mathbb{A}_{h\tau} & \mathbb{A}_{hh} \end{pmatrix},$$

where $\mathbb{A}_{\tau\tau}$ and \mathbb{A}_{hh} are symmetric matrices, and $\mathbb{A}_{\tau h} = \mathbb{A}_{h\tau}^\dagger$. Introducing the two matrices $\mathbb{S}_\tau = \mathbb{K}_\tau - \mathbb{A}_{\tau\tau} - \mathbb{B}_\tau$ and $\mathbb{S}_h = \mathbb{K}_h - \mathbb{A}_{hh} - \mathbb{B}_h$, the global system of ordinary differential equations (7) can be split into two systems

$$\begin{cases} \mathbb{M}_\tau^\varepsilon \partial_t \mathbb{E}_\tau &= \mathbb{S}_\tau \mathbb{H}_\tau - \mathbb{A}_{\tau h} \mathbb{H}_h, \\ \mathbb{M}_\tau^\mu \partial_t \mathbb{H}_\tau &= -\mathbb{S}_\tau^\dagger \mathbb{E}_\tau + \mathbb{A}_{\tau h} \mathbb{E}_h, \end{cases} \quad (9)$$

$$\begin{cases} \mathbb{M}_h^\varepsilon \partial_t \mathbb{E}_h &= \mathbb{S}_h \mathbb{H}_h - \mathbb{A}_{h\tau} \mathbb{H}_\tau, \\ \mathbb{M}_h^\mu \partial_t \mathbb{H}_h &= -\mathbb{S}_h^\dagger \mathbb{E}_h + \mathbb{A}_{h\tau} \mathbb{E}_\tau. \end{cases}$$

Then, the proposed DGTD scheme on hybrid meshes can be written as

$$\begin{cases} \mathbb{M}_\tau^\varepsilon \left(\frac{\mathbb{E}_\tau^{n+1} - \mathbb{E}_\tau^n}{\Delta t} \right) &= \mathbb{G}_\tau \mathbb{H}_\tau^{n+\frac{1}{2}} - \mathbb{G}_{\tau h} \mathbb{H}_h^{n+\frac{1}{2}}, \\ \mathbb{M}_\tau^\mu \left(\frac{\mathbb{H}_\tau^{n+\frac{1}{2}} - \mathbb{H}_\tau^{n-\frac{1}{2}}}{\Delta t} \right) &= -\mathbb{G}_\tau^\dagger \mathbb{E}_\tau^{n+1} + \mathbb{G}_{\tau h} \mathbb{E}_h^{n+1}, \end{cases} \quad (10a)$$

$$\begin{cases} \mathbb{M}_h^\varepsilon \left(\frac{\mathbb{E}_h^{n+1} - \mathbb{E}_h^n}{\Delta t} \right) &= \mathbb{G}_h \mathbb{H}_h^{n+\frac{1}{2}} - \mathbb{G}_{h\tau} \mathbb{H}_\tau^{n+\frac{1}{2}}, \\ \mathbb{M}_h^\mu \left(\frac{\mathbb{H}_h^{n+\frac{1}{2}} - \mathbb{H}_h^{n-\frac{1}{2}}}{\Delta t} \right) &= -\mathbb{G}_h^\dagger \mathbb{E}_h^{n+1} + \mathbb{G}_{h\tau} \mathbb{E}_\tau^{n+1}, \end{cases} \quad (10b)$$

where

$$\mathbb{G}_\tau = \mathbb{S}_\tau \left(\mathbb{I} - \frac{\Delta t^2}{24} (\mathbb{M}_\tau^\mu)^{-1} \mathbb{S}_\tau^\dagger (\mathbb{M}_\tau^\varepsilon)^{-1} \mathbb{S}_\tau \right), \quad \mathbb{G}_{\tau h} = \mathbb{A}_{\tau h} \left(\mathbb{I} - \frac{\Delta t^2}{24} (\mathbb{M}_h^\mu)^{-1} \mathbb{A}_{\tau h}^\dagger (\mathbb{M}_\tau^\varepsilon)^{-1} \mathbb{A}_{\tau h} \right),$$

$$\mathbb{G}_h = \mathbb{S}_h \left(\mathbb{I} - \frac{\Delta t^2}{24} (\mathbb{M}_h^\mu)^{-1} \mathbb{S}_h^\dagger (\mathbb{M}_h^\varepsilon)^{-1} \mathbb{S}_h \right), \quad \mathbb{G}_{h\tau} = \mathbb{G}_{\tau h}^\dagger.$$

The DGTD method on triangular-quadrilateral meshes or tetrahedral-hexahedral meshes, Eq. (10), is referred to as DGTD- $\mathbb{P}_p\mathbb{Q}_q$ in the following. If the mesh contains fully triangular or tetrahedral elements (resp. quadrilateral or hexahedral elements), then Eq. (10a) (resp. Eq. (10b)) is referred to as DGTD- \mathbb{P}_p method (resp. DGTD- \mathbb{Q}_q method).

3.2. Energy conservation

We define the following discrete electromagnetic energy

$$\mathcal{E}^n = \frac{1}{2} \left[(\mathbb{E}_\tau^n)^\dagger \mathbb{M}_\tau^\varepsilon \mathbb{E}_\tau^n + (\mathbb{H}_\tau^{n-\frac{1}{2}})^\dagger \mathbb{M}_\tau^\mu \mathbb{H}_\tau^{n-\frac{1}{2}} + (\mathbb{E}_h^n)^\dagger \mathbb{M}_h^\varepsilon \mathbb{E}_h^n + (\mathbb{H}_h^{n-\frac{1}{2}})^\dagger \mathbb{M}_h^\mu \mathbb{H}_h^{n-\frac{1}{2}} \right]. \quad (11)$$

In the following, we shall verify that \mathcal{E}^n is exactly conserved in absence of absorbing boundary conditions, i.e., $\Delta \mathcal{E} := \mathcal{E}^{n+1} - \mathcal{E}^n = 0$.

Lemma 1. *Using the hybrid DGTD- $\mathbb{P}_p\mathbb{Q}_q$ method, Eq. (10), for solving the Maxwell system, Eq. (1), with PEC boundary conditions only, the energy \mathcal{E}^n is exactly conserved.*

Proof. We denote by $\mathbb{E}_{\tau/h}^{n+\frac{1}{2}} = (\mathbb{E}_{\tau/h}^n + \mathbb{E}_{\tau/h}^{n+1})/2$. The variation of the energy \mathcal{E}^n through one time step is given by

$$\begin{aligned} \Delta \mathcal{E} &= (\mathbb{E}_\tau^{n+\frac{1}{2}})^\dagger \mathbb{M}_\tau^\varepsilon [\mathbb{E}_\tau^{n+1} - \mathbb{E}_\tau^n] + \frac{1}{2} (\mathbb{H}_\tau^{n+\frac{1}{2}})^\dagger \mathbb{M}_\tau^\mu [\mathbb{H}_\tau^{n+\frac{1}{2}} - \mathbb{H}_\tau^{n-\frac{1}{2}}] + \\ &\quad (\mathbb{E}_h^{n+\frac{1}{2}})^\dagger \mathbb{M}_h^\varepsilon [\mathbb{E}_h^{n+1} - \mathbb{E}_h^n] + \frac{1}{2} (\mathbb{H}_h^{n+\frac{1}{2}})^\dagger \mathbb{M}_h^\mu [\mathbb{H}_h^{n+\frac{1}{2}} - \mathbb{H}_h^{n-\frac{1}{2}}] \\ &= \Delta t (\mathbb{E}_\tau^{n+\frac{1}{2}})^\dagger \mathbb{G}_\tau \mathbb{H}_\tau^{n+\frac{1}{2}} - \Delta t (\mathbb{E}_\tau^{n+\frac{1}{2}})^\dagger \mathbb{G}_{\tau h} \mathbb{H}_h^{n+\frac{1}{2}} - \Delta t (\mathbb{H}_\tau^{n+\frac{1}{2}})^\dagger \mathbb{G}_\tau^\dagger \mathbb{E}_\tau^{n+\frac{1}{2}} + \Delta t (\mathbb{H}_\tau^{n+\frac{1}{2}})^\dagger \mathbb{G}_{\tau h} \mathbb{E}_h^{n+\frac{1}{2}} + \\ &\quad \Delta t (\mathbb{E}_h^{n+\frac{1}{2}})^\dagger \mathbb{G}_h \mathbb{H}_h^{n+\frac{1}{2}} - \Delta t (\mathbb{E}_h^{n+\frac{1}{2}})^\dagger \mathbb{G}_{h\tau} \mathbb{H}_\tau^{n+\frac{1}{2}} - \Delta t (\mathbb{H}_h^{n+\frac{1}{2}})^\dagger \mathbb{G}_h^\dagger \mathbb{E}_h^{n+\frac{1}{2}} + \Delta t (\mathbb{H}_h^{n+\frac{1}{2}})^\dagger \mathbb{G}_{h\tau} \mathbb{E}_\tau^{n+\frac{1}{2}} \\ &= \Delta t (\mathbb{E}_\tau^{n+\frac{1}{2}})^\dagger [\mathbb{G}_\tau - \mathbb{G}_{\tau h}] \mathbb{H}_\tau^{n+\frac{1}{2}} - \Delta t (\mathbb{E}_\tau^{n+\frac{1}{2}})^\dagger [\mathbb{G}_{\tau h} - \mathbb{G}_{h\tau}^\dagger] \mathbb{H}_h^{n+\frac{1}{2}} + \\ &\quad \Delta t (\mathbb{E}_h^{n+\frac{1}{2}})^\dagger [\mathbb{G}_h - \mathbb{G}_{h\tau}] \mathbb{H}_h^{n+\frac{1}{2}} - \Delta t (\mathbb{E}_h^{n+\frac{1}{2}})^\dagger [\mathbb{G}_{h\tau} - \mathbb{G}_{\tau h}^\dagger] \mathbb{H}_\tau^{n+\frac{1}{2}}, \end{aligned}$$

which completes the proof. \square

3.3. Stability analysis

We shall prove that the electromagnetic energy is a positive definite quadratic form of all unknowns under a CFL-like condition on the time step Δt .

Lemma 2. *For the hybrid DGTD- $\mathbb{P}_p\mathbb{Q}_k$ method, Eq. (10), the electromagnetic energy, \mathcal{E}^n , is a positive definite quadratic form of all numerical unknowns provided that*

$$\Delta t \leq 2 \min\left(\frac{1}{\beta_\tau + \beta_{\tau h}}, \frac{1}{\beta_h + \beta_{\tau h}}\right) \quad \text{with} \quad \begin{cases} \beta_\tau = \|(\mathbb{M}_\tau^\varepsilon)^{-\frac{1}{2}} \mathbb{G}_\tau (\mathbb{M}_\tau^\mu)^{-\frac{1}{2}}\|, \\ \beta_h = \|(\mathbb{M}_h^\varepsilon)^{-\frac{1}{2}} \mathbb{G}_h (\mathbb{M}_h^\mu)^{-\frac{1}{2}}\|, \\ \beta_{\tau h} = \|(\mathbb{M}_\tau^\mu)^{-\frac{1}{2}} \mathbb{G}_{\tau h} (\mathbb{M}_h^\varepsilon)^{-\frac{1}{2}}\|, \end{cases} \quad (12)$$

where $\|\cdot\|$ denotes any matrix norm, and the matrix $(\mathbb{M}_{\tau/h}^\gamma)^{-\frac{1}{2}}$ is the inverse square root of $\mathbb{M}_{\tau/h}^\gamma$ (γ stands for ε or μ).

Proof. Using the second relations of Eq. (10a) and Eq. (10b), the energy \mathcal{E}^n , Eq. (11) can be written as

$$\begin{aligned} \mathcal{E}^n &= \frac{1}{2} \left[(\mathbb{E}_\tau^n)^\dagger \mathbb{M}_\tau^\varepsilon \mathbb{E}_\tau^n + (\mathbb{H}_\tau^{n-\frac{1}{2}})^\dagger \mathbb{M}_\tau^\mu \mathbb{H}_\tau^{n-\frac{1}{2}} - \Delta t (\mathbb{H}_\tau^{n-\frac{1}{2}})^\dagger \mathbb{G}_\tau^\dagger \mathbb{E}_\tau^n - \Delta t (\mathbb{H}_\tau^{n-\frac{1}{2}})^\dagger \mathbb{G}_{\tau h} \mathbb{E}_h^n \right. \\ &\quad \left. + (\mathbb{E}_h^n)^\dagger \mathbb{M}_h^\varepsilon \mathbb{E}_h^n + (\mathbb{H}_h^{n-\frac{1}{2}})^\dagger \mathbb{M}_h^\mu \mathbb{H}_h^{n-\frac{1}{2}} - \Delta t (\mathbb{H}_h^{n-\frac{1}{2}})^\dagger \mathbb{G}_h^\dagger \mathbb{E}_h^n - \Delta t (\mathbb{H}_h^{n-\frac{1}{2}})^\dagger \mathbb{G}_{h\tau} \mathbb{E}_\tau^n \right]. \end{aligned}$$

The mass matrices $\mathbb{M}_{\tau/h}^\varepsilon$ and $\mathbb{M}_{\tau/h}^\mu$ are symmetric positive definite and we can construct their square root (also symmetric positive definite) denoted by $(\mathbb{M}_{\tau/h}^\varepsilon)^{\frac{1}{2}}$ and $(\mathbb{M}_{\tau/h}^\mu)^{\frac{1}{2}}$, respectively. We deduce from Eq. (11) that

$$\begin{aligned} \mathcal{E}^n &\geq \frac{1}{2} \left[\|(\mathbb{M}_\tau^\varepsilon)^{\frac{1}{2}} \mathbb{E}_\tau^n\|^2 + \|(\mathbb{M}_\tau^\mu)^{\frac{1}{2}} \mathbb{H}_\tau^{n-\frac{1}{2}}\|^2 - \Delta t \beta_\tau \|(\mathbb{M}_\tau^\mu)^{\frac{1}{2}} \mathbb{H}_\tau^{n-\frac{1}{2}}\| \|(\mathbb{M}_\tau^\varepsilon)^{\frac{1}{2}} \mathbb{E}_\tau^n\| - \Delta t \beta_{\tau h} \|(\mathbb{M}_\tau^\mu)^{\frac{1}{2}} \mathbb{H}_\tau^{n-\frac{1}{2}}\| \|(\mathbb{M}_h^\varepsilon)^{\frac{1}{2}} \mathbb{E}_h^n\| \right. \\ &\quad \left. + \|(\mathbb{M}_h^\varepsilon)^{\frac{1}{2}} \mathbb{E}_h^n\|^2 + \|(\mathbb{M}_h^\mu)^{\frac{1}{2}} \mathbb{H}_h^{n-\frac{1}{2}}\|^2 - \Delta t \beta_h \|(\mathbb{M}_h^\mu)^{\frac{1}{2}} \mathbb{H}_h^{n-\frac{1}{2}}\| \|(\mathbb{M}_h^\varepsilon)^{\frac{1}{2}} \mathbb{E}_h^n\| - \Delta t \beta_{h\tau} \|(\mathbb{M}_h^\mu)^{\frac{1}{2}} \mathbb{H}_h^{n-\frac{1}{2}}\| \|(\mathbb{M}_\tau^\varepsilon)^{\frac{1}{2}} \mathbb{E}_\tau^n\| \right], \end{aligned}$$

where β_τ, β_h and $\beta_{\tau h}$ are defined in Eq. (12), and $\beta_{h\tau} = \|(\mathbb{M}_h^\mu)^{-\frac{1}{2}} \mathbb{G}_{h\tau} (\mathbb{M}_\tau^\varepsilon)^{-\frac{1}{2}}\|$. Then estimate $ab \leq (a^2 + b^2)/2$ leads to

$$\begin{aligned} \mathcal{E}^n &\geq \frac{1}{2} \left[\|(\mathbb{M}_\tau^\varepsilon)^{\frac{1}{2}} \mathbb{E}_\tau^n\|^2 + \|(\mathbb{M}_\tau^\mu)^{\frac{1}{2}} \mathbb{H}_\tau^{n-\frac{1}{2}}\|^2 - \beta_\tau \frac{\Delta t}{2} \left(\|(\mathbb{M}_\tau^\mu)^{\frac{1}{2}} \mathbb{H}_\tau^{n-\frac{1}{2}}\|^2 + \|(\mathbb{M}_\tau^\varepsilon)^{\frac{1}{2}} \mathbb{E}_\tau^n\|^2 \right) \right. \\ &\quad \left. - \beta_{\tau h} \frac{\Delta t}{2} \left(\|(\mathbb{M}_\tau^\mu)^{\frac{1}{2}} \mathbb{H}_\tau^{n-\frac{1}{2}}\|^2 + \|(\mathbb{M}_h^\varepsilon)^{\frac{1}{2}} \mathbb{E}_h^n\|^2 \right) \right. \\ &\quad \left. + \|(\mathbb{M}_h^\varepsilon)^{\frac{1}{2}} \mathbb{E}_h^n\|^2 + \|(\mathbb{M}_h^\mu)^{\frac{1}{2}} \mathbb{H}_h^{n-\frac{1}{2}}\|^2 - \beta_h \frac{\Delta t}{2} \left(\|(\mathbb{M}_h^\mu)^{\frac{1}{2}} \mathbb{H}_h^{n-\frac{1}{2}}\|^2 + \|(\mathbb{M}_h^\varepsilon)^{\frac{1}{2}} \mathbb{E}_h^n\|^2 \right) \right. \\ &\quad \left. - \beta_{h\tau} \frac{\Delta t}{2} \left(\|(\mathbb{M}_h^\mu)^{\frac{1}{2}} \mathbb{H}_h^{n-\frac{1}{2}}\|^2 + \|(\mathbb{M}_\tau^\varepsilon)^{\frac{1}{2}} \mathbb{E}_\tau^n\|^2 \right) \right]. \end{aligned}$$

We then sum up the lower bounds for the \mathcal{E}^n to obtain

$$\begin{aligned} \mathcal{E}^n &\geq \frac{1}{2} \left(1 - \Delta t \frac{\beta_\tau + \beta_{h\tau}}{2} \right) \|(\mathbb{M}_\tau^\varepsilon)^{\frac{1}{2}} \mathbb{E}_\tau^n\|^2 + \frac{1}{2} \left(1 - \Delta t \frac{\beta_\tau + \beta_{\tau h}}{2} \right) \|(\mathbb{M}_\tau^\mu)^{\frac{1}{2}} \mathbb{H}_\tau^{n-\frac{1}{2}}\|^2 + \\ &\quad \frac{1}{2} \left(1 - \Delta t \frac{\beta_h + \beta_{\tau h}}{2} \right) \|(\mathbb{M}_h^\varepsilon)^{\frac{1}{2}} \mathbb{E}_h^n\|^2 + \frac{1}{2} \left(1 - \Delta t \frac{\beta_h + \beta_{h\tau}}{2} \right) \|(\mathbb{M}_h^\mu)^{\frac{1}{2}} \mathbb{H}_h^{n-\frac{1}{2}}\|^2. \end{aligned}$$

Since the parameters ε and μ are piecewise constant and $\mathbb{G}_{\tau h} = \mathbb{G}_{h\tau}^\dagger$, we have that $\beta_{h\tau} = \beta_{\tau h}$. Then, under the condition proposed in Lemma 2, the energy \mathcal{E}^n is a positive definite quadratic form of the numerical unknowns $\mathbb{E}_{\tau/h}^n$ and $\mathbb{H}_{\tau/h}^{n-\frac{1}{2}}$, and Eq. (12) states a sufficient condition for the stability of the hybrid DGTD- $\mathbb{P}_p\mathbb{Q}_q$ method. \square

4. Implementation issues

For generality, we shall limit the discussions to the 3D case and regard the 2D case as a natural simplification.

4.1. Nonconforming hybrid mesh generation

A simple way for generating a non-conforming hybrid tetrahedral-hexahedral mesh without pyramidal or prismatic elements is based on element-type conversion. This indirect approach starts with the generation of a conforming or non-conforming mesh with only one type of elements (pure tetrahedra or pure hexahedra). It is partially converted to hexahedral or tetrahedral mesh by applying a conventional subdivision scheme. For instance, a tetrahedron can be split into four hexahedra [8] while a hexahedron can be decomposed into five or six tetrahedra [2]. The resulting hybrid mesh is non-conformal where the hybrid interface between a tetrahedron and a hexahedron is either a triangle or a quadrilateral, see Figure 1. Although this approach avoids some difficulties, it rapidly increases the number of elements and tends to introduce badly shaped elements which may deteriorate their numerical properties. We will not pursue this approach further here, and will propose an alternative method.

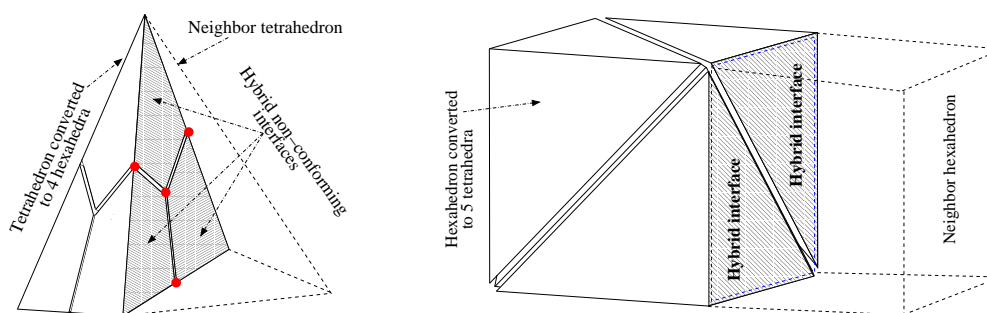


Figure 1: Examples of hybrid interfaces when the hybrid tetrahedral-hexahedral mesh is generated using the element-type conversion approach. As shown in the left, if a tetrahedron is converted to hexahedra, the hybrid interface is an entire quadrilateral face of the hexahedron, where three of its four vertices are hanging nodes (filled circles). When a hexahedron is converted to tetrahedra, the hybrid interface is an entire triangular face of the tetrahedron. In this case, none of the three vertices of the hybrid interface is a hanging node.

In this work, we consider a more general strategy which consists of the following steps: first, the computational domain Ω is decomposed into subdomains Ω_i such that $\bar{\Omega} = \bigcup_i \bar{\Omega}_i$. Then, an individual mesh for each subdomain Ω_i is generated using a suitable meshing algorithm and the most appropriate element type. The resulting meshes should not share nodes, edges, faces, or cells. Finally, we combine these submeshes to obtain the non-conforming hybrid mesh of the whole domain Ω . This strategy of non-conforming hybrid mesh generation has several advantages over other approaches based on hybrid conforming mesh. For instance, it allows one (i) to avoid the intermediate layer of pyramids as transition elements between triangular and quadrilateral faces; (ii) to relax the node-to-node matching requirement at the interface between the meshes; (iii) to change or replace certain portions of the domain without changing the entire mesh. These features are particularly useful for parametric modelling of complex geometries where the object is built from several parts designed separately. Figure 2(a)-(b) illustrates examples of hybrid meshes generated by this strategy.

After combining the meshes, we need to find the hybrid non-conforming interfaces between triangular and quadrilateral faces in order to compute the flux matrices, see Figure 2(c). First, we locate the boundary faces of each subdomain mesh and store them in a list including their element and face location. Next, we use the fact that only the boundary faces of two subdomains that shared boundaries may participate in a partial face-face connection. Let us assume that two subdomains Ω_i and Ω_j share a boundary. For each of the boundary faces of the mesh of Ω_i , we perform two tests: the first is an intersection test with every member of the boundary face list of the mesh of Ω_j . This just indicates whether or not an intersection exists. The second is a find-intersection test that involves computing the set of intersections between triangular and quadrilateral faces. To perform these intersection tests, we use the class `Intersection` provided by the Eberly's Wild Magic library [22]. For the reader's convenience, we briefly describe the general idea of this class. The test-intersection is based on the *method of separating axes* [34], a method for determining whether or not two convex objects are intersecting. A test for nonintersection of two convex objects is simply stated: if there exists a line for which the intervals of projection of the two objects onto that line do not intersect, then

the objects do not intersect. Such a line is called a *separating axis*. For a pair of convex polygons in 3D, only a finite set of direction vectors needs to be considered for separation tests. The polygons normals are potential separating directions. If either normal direction separates the polygons, then no intersection occurs. However, if neither normal direction separates the polygons, then two possibilities exist for the remaining potential separating directions. The first case is that the polygons are coplanar and the remaining potential separating directions are those vectors in the common plane and perpendicular to the polygon edges. The second case is that the polygon planes are not parallel and that each polygon is split by the plane of the other polygon. The remaining potential separating directions are the cross products of two edges, one from each polygon. In summary, for the intersection between a triangular and a quadrilateral face in 3D, a maximum of 19 separating directions need to be tested in the worst case. Given a line, it is straightforward to project the vertices of a polygon onto the line and compute a bounding interval for those projections. The two intervals obtained from the two polygons are easily compared for overlap. If an intersection exist, we perform a find-intersection query using Boolean operations on polygons [39, 46]. Here the intersection between a triangular and a quadrilateral face (both lies in the same plane) is a convex polygon with at most seven vertices. Let Q be a quadrilateral and T be a triangle with vertices (v_1, v_2, v_3) that are counterclockwise ordered. We define half-planes, π_{ij} , $1 \leq i < j \leq 3$, bounded by the lines $(v_i v_j)$ and containing the third vertex of the triangle T . The intersection between T and Q is defined as $T \cap Q = ((Q \cap \pi_{12}) \cap \pi_{23}) \cap \pi_{13}$. The problem is reduced to find the intersection between a polygon and a half-plane which is relatively straightforward. For a more complete exposition of the theory of intersection tests, we refer the reader to the books [48, Chap. 7 & 11]-[21, Chap. 15].

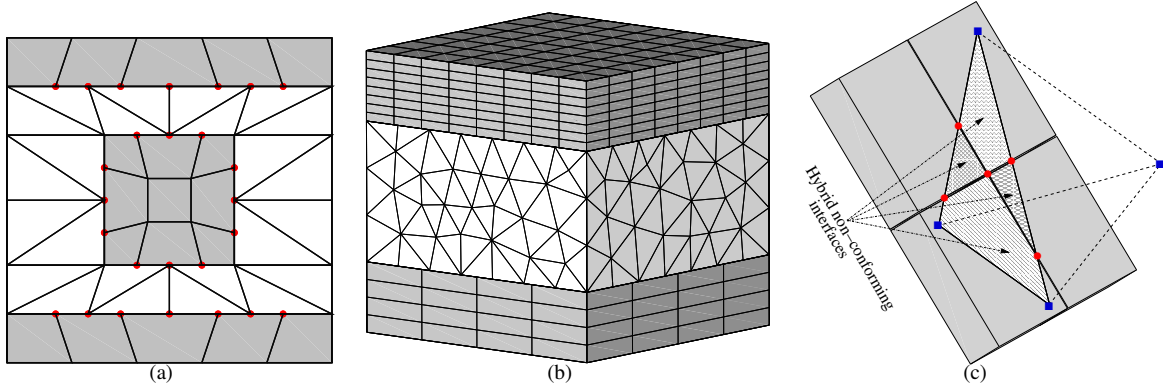


Figure 2: Examples of non-conformal hybrid meshes generated using the combining meshes strategy. In (a) we show a triangular-quadrilateral mesh where the filled circles stand for hanging nodes. A 2D hybrid interface is a line segment with at least one of its end-points is a hanging node. In (b) we show a 3D tetrahedral-hexahedral mesh. An example of 3D hybrid interfaces is illustrated in (c) where we show the intersection between a tetrahedron and four hexahedra. Each hybrid interface is a convex polygon with at most seven vertices, where at least one of them is a hanging node (filled circle).

4.2. Non-conforming hybrid interface matrix calculation

A key ingredient for the implementation of the DG method on hybrid meshes is the computation of the matrix $\mathbb{A}_{\tau h}$ involving integrals over a hybrid interface. We concentrate here on the evaluation of such integrals. In fact, the other matrices in Eq. (9) which involve integrals over an element or an interface between two elements of the same type are evaluated as specified in Eq. (5). We just have to utilize the correct set of basis functions corresponding to the particular element type.

Let $a_{ik} = \tau_i \cap h_k$ be a hybrid interface between a tetrahedron τ_i and a hexahedron h_k . Here a_{ik} is a n -gon with $n = 3, 4, 5, 6, 7$, that is, a_{ik} is a convex polygon with n edges (see Figure 3). Let also \mathcal{R} be a regular polygon in \mathbb{R}^2 and $\Upsilon : \mathcal{R} \ni \xi \rightarrow \mathbf{x} \in a_{ik} \subset \mathbb{R}^3$ be an explicit parametrization of a_{ik} . The matrix $\mathbb{A}_{\tau h}$ is block diagonal, whose non-zero blocks are equal to the interface matrix S_{ik} when a_{ik} is a hybrid interface. The matrix S_{ik} is defined as

$$(S_{ik})_{jl} = \int_{a_{ik}} \theta_{ij}(\mathbf{x}) \theta_{kl}(\mathbf{x}) d\mathbf{x} = \int_{\mathcal{R}} (L_j^{(p_i)} \circ \Psi_{\tau_i}^{-1} \circ \Upsilon(\xi)) (L_l^{(q_k)} \circ \Psi_{h_k}^{-1} \circ \Upsilon(\xi)) \|\nabla \Upsilon(\xi)\| d\xi \quad (13)$$

where $\theta_{ij} \in \mathbb{P}_{p_i}(\tau_i)$ and $\vartheta_{kl} \in \mathbb{Q}_{q_k}(h_k)$ are the basis functions of degrees p_i and q_k in the elements τ_i and h_k , respectively. The integrals in Eq. (13) are calculated numerically and stored for each hybrid interface a_{ik} . To this end, two approaches are possible

1. Partitioning the interface a_{ik} (a n -gon) into n triangles and then performing cubature rules over each triangle [17, 54]. For n -gons with $n \geq 4$, we use the centroid of a_{ik} to partition it into triangles. The matrix S_{ik} is computed with the following formula

$$(S_{ik})_{jl} = \int_{a_{ik}} \theta_{ij}(\mathbf{x}) \vartheta_{kl}(\mathbf{x}) d\mathbf{x} = \sum_{s=1}^{s=n} \int_{\Delta_s} \theta_{ij}(\mathbf{x}) \vartheta_{kl}(\mathbf{x}) d\mathbf{x}, \quad (14)$$

where Δ_s is a triangle formed by two successive vertices of a_{ik} and its centroid as illustrated in Figure 4(a). The parametrization of Δ_s can be defined as $\mathbf{x} = \Upsilon(\boldsymbol{\xi}) = \sum_{i=1}^3 \phi_i(\boldsymbol{\xi}) v_i$, where $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \mathcal{T}^2$, ϕ_i are the barycentric coordinates in \mathcal{T}^2 and v_i denotes the three vertices of Δ_s . To compute the last integral in Eq. (14), we lay down N_c cubature points $\lambda_n \in \mathcal{T}^2$ and weights ω_n . This is achieved by

$$\int_{\Delta_s} \theta_{ij}(\mathbf{x}) \vartheta_{kl}(\mathbf{x}) d\mathbf{x} = \sum_{n=1}^{N_c} L_j^{(p_i)}(\lambda_n^r) L_l^{(q_k)}(\lambda_n^h) \omega_n \|\nabla \Upsilon(\lambda_n)\|, \quad (15)$$

where $\lambda_n^r = \Psi_{\tau_i}^{-1} \circ \Upsilon(\lambda_n) \in \mathcal{T}^3$ and $\lambda_n^h = \Psi_{h_k}^{-1} \circ \Upsilon(\lambda_n) \in \mathcal{D}^3$.

2. An alternative is to use cubature rules over polygons. However, the design and development of such rules has not yet reached a mature stage. To our knowledge this approach has not been used previously in the context of DG methods. Recently, Mousavi et al. [42] presented an algorithm to build symmetric cubature rules for arbitrary regular polygons. This algorithm combines elements of group theory and numerical optimization and was originally proposed by Wandzura and Xiao [54] for the construction of cubature rules over triangles. Let \mathcal{P}^n be a reference regular n -gon with vertices $p_m = (\cos 2\pi m/n, \sin 2\pi m/n)$ for $m = 1, 2, \dots, n$ in the $\boldsymbol{\xi}$ -coordinate system, see Figure 4(b). The parametrization Υ from \mathcal{P}^n to a_{ik} is obtained via the isoparametric mapping $\mathbf{x} = \Upsilon(\boldsymbol{\xi}) = \sum_{i=1}^n \phi_i(\boldsymbol{\xi}) v_i$, where v_i are the vertices of the n -gon a_{ik} and ϕ_i are the Laplace shape functions at point $p \in \mathcal{P}^n$ defined as [52]

$$\phi_i(\boldsymbol{\xi}) = \frac{\alpha_i(\boldsymbol{\xi})}{\sum_{j=1}^n \alpha_j(\boldsymbol{\xi})}, \quad \alpha_j(\boldsymbol{\xi}) = \frac{(1 - |\boldsymbol{\xi}|^2) \sin^3(\pi/n) \cos(\pi/n)}{A_{i-1} A_i}, \quad \boldsymbol{\xi} = (\xi_1, \xi_2) \in \mathcal{P}^n,$$

where $\alpha_i(\boldsymbol{\xi})$ is the Laplace weight function, A_i is the area of the triangle $[p, p_i, p_{i+1}]$ and A_{i-1} is the area of the triangle $[p, p_{i-1}, p_i]$. The Laplace shape functions are non-negative, interpolate nodal data and form a partition of unity. They are also linear on the polygon boundaries which ensures that the parametrization $\Upsilon(\boldsymbol{\xi})$ is an affine mapping. Once $\Upsilon(\boldsymbol{\xi})$ is given explicitly, the integral in Eq. (13) is evaluated by laying down N_c cubature points on \mathcal{P}^n and proceed as in Eq. (15).

We compare the two approaches by evaluating the integral of $x^6 y^6 z^6$ over the regular pentagon, hexagon and heptagon. The relative errors between exact and approximate values are shown in Figure 5 as a function of the number of cubature points. We observe that for a relative error of 10^{-8} the polygonal cubatures require 3 to 5 times fewer points than the partitioning approach. Hence, for the numerical results given in Section 5, we adopt the second approach since it is more practical and requires the least number of cubature points for a given degree of accuracy.

5. Numerical results

5.1. Two-dimensional examples

We consider here the 2D TM_z -polarized Maxwell equations, that is, we solve Eq. (1) for the fields $\mathbf{H} = (H_x, H_y, 0)$ and $\mathbf{E} = (0, 0, E_z)$. Three examples with analytical solutions are studied in detail to illustrate the accuracy and performance of the DG method on hybrid triangular-quadrilateral meshes. The exact time-domain solutions of these problems can be found in [24].

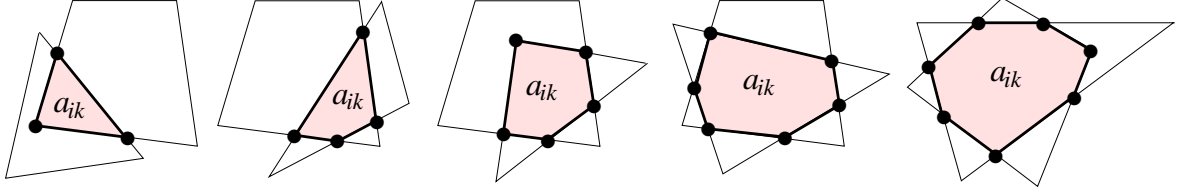


Figure 3: Five possible configurations for the hybrid interface a_{ik} between a quadrilateral face and a triangular face in 3D.

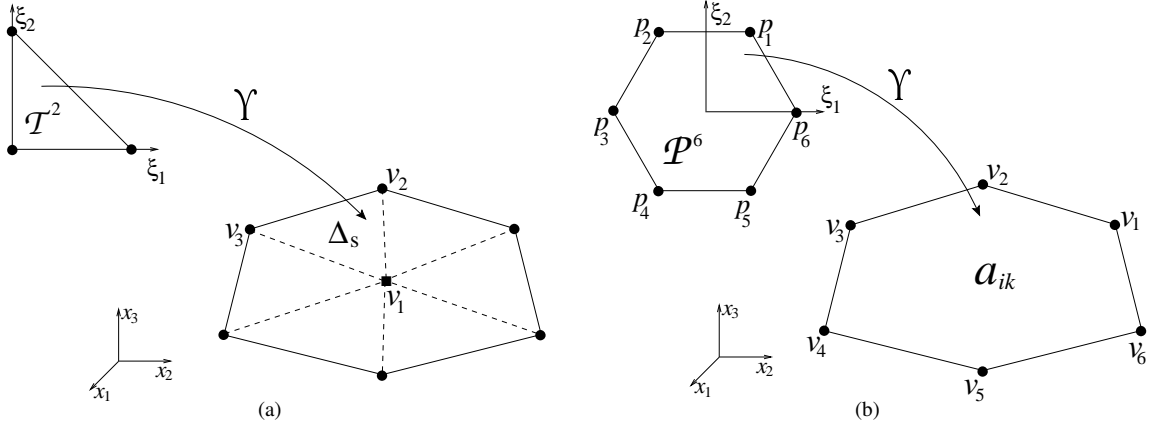


Figure 4: Two numerical integration schemes used for the evaluation of integrals over a polygon a_{ik} . In (a) we show a scheme based on the partition of a_{ik} into triangles Δ_s . The scheme illustrated in (b) consists of using cubature rules over regular polygons.

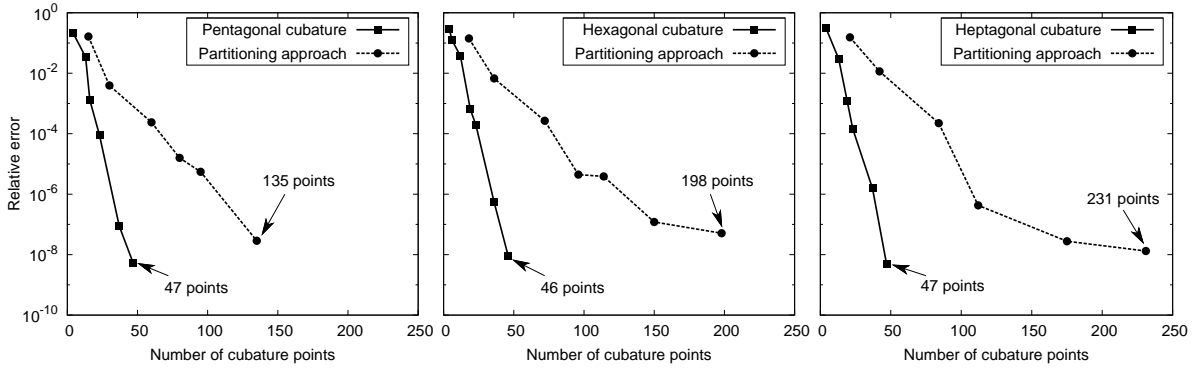


Figure 5: Convergence curves of two numerical integration approaches used to evaluate the integral of $x^6 y^6 z^6$ over the regular pentagon (left), hexagon (center) and heptagon (right).

5.1.1. Eigenmode in a PEC cavity

In the first example, we propose to validate the implementation of the proposed DGTD- $\mathbb{P}_p\mathbb{Q}_q$ method on non-conforming hybrid meshes by performing a numerical convergence test. To this end, we consider the propagation of an eigenmode which is a standing wave of frequency $f = 212$ MHz and wavelength of 1.4 m in a unitary vacuum-filled square cavity with PEC walls. The computations are performed on three successively refined hybrid meshes, named MS_i for $i = 1, 2, 3$. Each mesh MS_i consists of $5 \times 2^{2i-1}$ triangles and $3 \times 2^{2i-1}$ quadrilaterals, as shown in Figure 6. We present results with interpolation orders, p and q , up to degree 8 such that $p = q$. For each interpolation order, we use the maximum time step, Δt , allowed by the stability condition of the hybrid method. In Figure 7 (left) we illustrate the h -convergence graphs as a function of the total number of degrees of freedom (#DOF). As expected, for smooth solutions the convergence rate behaves as $\mathcal{O}(\Delta t^4 + h^p)$ even for non-conforming hybrid meshes. Also shown in Figure 7 (right) is the p -convergence rate in linear-log scale, observing an exponential decay of the error as $\exp(-s_p p)$, with decay constant $s_p = \log[E(p-1, q-1)/E(p, q)] \approx 2$, where $E(p, q)$ is the L^2 error when the interpolation orders

p and q are used. By inspecting the results in Figure 7, we observe that the gain in accuracy is notable when refining the mesh or increasing the interpolation orders. Furthermore, to achieve a certain accuracy level, the DG methods require less #DOF for higher interpolation orders, as typically found with other DG methods based on fully simplicial meshes [25, 30].

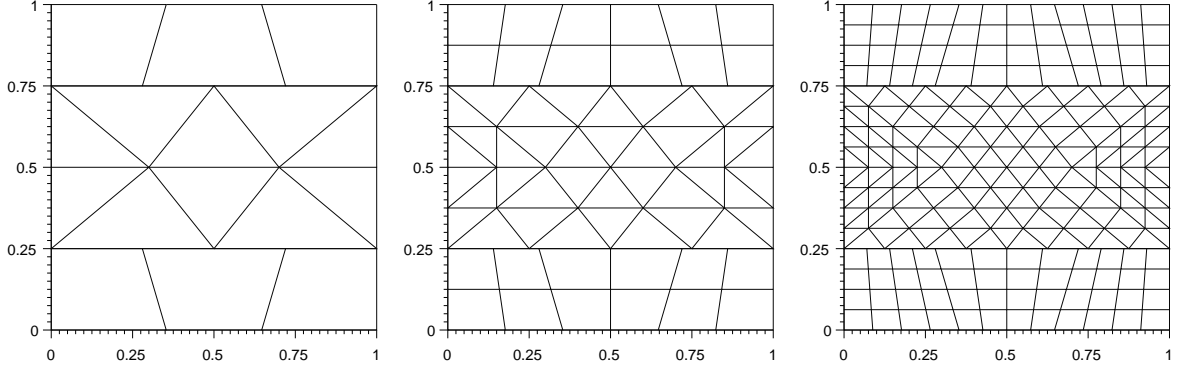


Figure 6: Sequences of three successively refined hybrid meshes used for assessing convergence of the eigenmode problem.

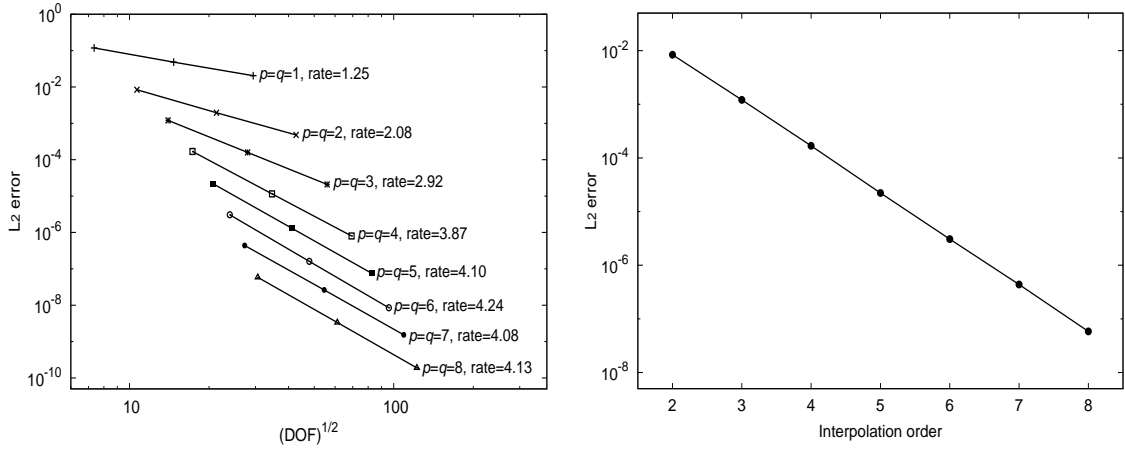


Figure 7: On the left is shown the L^2 -error for the EM field at time $t = 10$ ns as a function of the mesh resolution $h = (\text{DOF})^{1/2}$, for different values of the interpolation orders p and q with $p = q$. As expected, the h -convergence rate behaves as $O(\Delta t^4 + h^p)$ for all orders. On the right, we plot the slope of the error for the mesh MS_1 , confirming the spectral convergence of the hybrid $\text{DGTD-}\mathbb{P}_p\mathbb{Q}_q$ method.

5.1.2. Plane-wave scattering by a dielectric circular cylinder

As an illustration of the capability of the hybrid DG method to handle materials, let us consider plane wave scattering by a penetrable circular cylinder with a radius of 0.6 m consisting of a dielectric nonmagnetic material with a relative permittivity $\varepsilon_r = 12.0$. The frequency of the incident wave is $f = 300$ MHz and the wavelength is $\lambda = 0.288$ m. The surrounding medium is assumed to be vacuum. The computational domain Ω is bounded by a square of side length 3.2 m centered at $(0, 0)$. A Silver-Müller absorbing condition is applied on the boundary of the square. We compare the performance of the $\text{DGTD-}\mathbb{P}_p\mathbb{Q}_q$ method using a non-conforming hybrid mesh with the $\text{DGTD-}\mathbb{P}_p$ method on a fully triangular mesh, see Figure 8. The hybrid mesh is generated by combining submeshes of three subdomains, $\Omega_1 = [-0.35, 0.35]^2 \text{ m}^2$ (inside the cylinder), $\Omega_2 = [-0.7, 0.7]^2 \text{ m}^2 \setminus \Omega_1$ (close to cylinder and completely enclosing it) and $\Omega_3 = [-1.6, 1.6]^2 \text{ m}^2 \setminus (\Omega_1 \cup \Omega_2)$ such that $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$. The regions Ω_1 and Ω_2 are filled with regular quadrilaterals while Ω_3 is filled with unstructured triangles. The global hybrid mesh has 1888 triangles, 1204 quadrilaterals, 5044 conforming interfaces, 320 non-conforming hybrid interfaces and 120 non-conforming interfaces between quadrilaterals. The average edge length of this mesh is of 0.16λ . The triangular mesh is

obtained by replacing the quadrilateral elements of the hybrid mesh by unstructured triangles. The resulting triangular mesh is conformal and consists of 4376 triangles and 6626 interfaces with an average edge length of 0.18λ . For the hybrid DGTD- $\mathbb{P}_p\mathbb{Q}_q$ method, the interpolation order, q , is a pair $q = (q_1, q_2)$, where q_1 and q_2 are the interpolation orders used in the smallest and biggest quadrilaterals. The final simulation time has been set to $T = 33.33$ ns which corresponds to 10 periods of the incident wave. The results are listed in Table 1 in terms of accuracy and computational cost to reach the final simulation time for different p and q . Inspecting these results, we observe that for a fixed p , the hybrid method is 1.4 to 1.9 times more accurate and requires 1.2 to 2.1 times less computing time compared to the triangular method. On Figure 9 we compare the x -wise 1D distributions of the real part of the discrete Fourier transform of E_z and H_y components. We observe from this figure that, the main error of DG solutions arises at the material interfaces where the solution loses smoothness and the results confirm the ability of the hybrid DG method to accurately model problems with discontinuous coefficients and solutions. Finally, contour plots of the E_z component at time $t = T$ for numerical simulations performed with the DG method on the hybrid and the triangular meshes, are illustrated in Figure 10 for $p = 2$ and $q = (2, 2)$. We observe very good agreement between the solutions obtained by both DG methods and the exact solution even with relatively low grid resolutions of six points per λ . The hybrid method shows slightly better results than the triangular method.

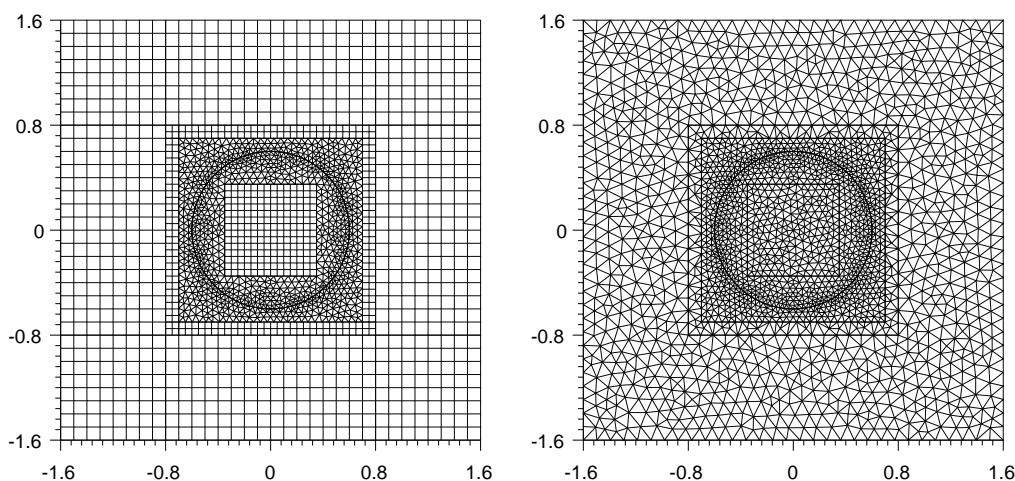


Figure 8: We show the non-conforming hybrid triangular-quadrilateral mesh (left) and the conforming fully triangular mesh (right), used for computing scattering by a dielectric cylinder of size 0.6 m.

Table 1: Comparison between the DGTD- $\mathbb{P}_p\mathbb{Q}_q$ method on a hybrid triangular-quadrilateral mesh and the DGTD- \mathbb{P}_p method on a fully triangular mesh. For the hybrid method, the interpolation order q is a pair (q_1, q_2) where q_1 and q_2 are the interpolation orders used in the smallest and biggest quadrilaterals, respectively. We give the maximum Δt allowed by the stability condition, the L^2 -errors at time $t = 33.33$ ns, the #DOF and the computing time (CPU) in seconds to achieve the final time period.

Method	Δt (ns)	L^2 -error	#DOF	CPU (s)
DGTD- \mathbb{P}_2	3.648E-02	2.394E-01	26256	201
DGTD- \mathbb{P}_3	2.132E-02	3.931E-02	43760	581
DGTD- \mathbb{P}_4	1.684E-02	6.532E-03	65640	1108
DGTD- \mathbb{P}_5	1.403E-02	1.083E-03	91896	1886
DGTD- \mathbb{P}_6	1.038E-02	1.785E-04	122528	3488
DGTD- $\mathbb{P}_2\mathbb{Q}_{(2,2)}$	3.648E-02	1.692E-01	22164	165
DGTD- $\mathbb{P}_3\mathbb{Q}_{(3,2)}$	2.346E-02	2.398E-02	32768	379
DGTD- $\mathbb{P}_4\mathbb{Q}_{(3,2)}$	1.841E-02	5.105E-03	42208	592
DGTD- $\mathbb{P}_5\mathbb{Q}_{(4,3)}$	1.523E-02	5.784E-04	62836	1162
DGTD- $\mathbb{P}_6\mathbb{Q}_{(4,3)}$	1.216E-02	1.484E-04	76052	1671

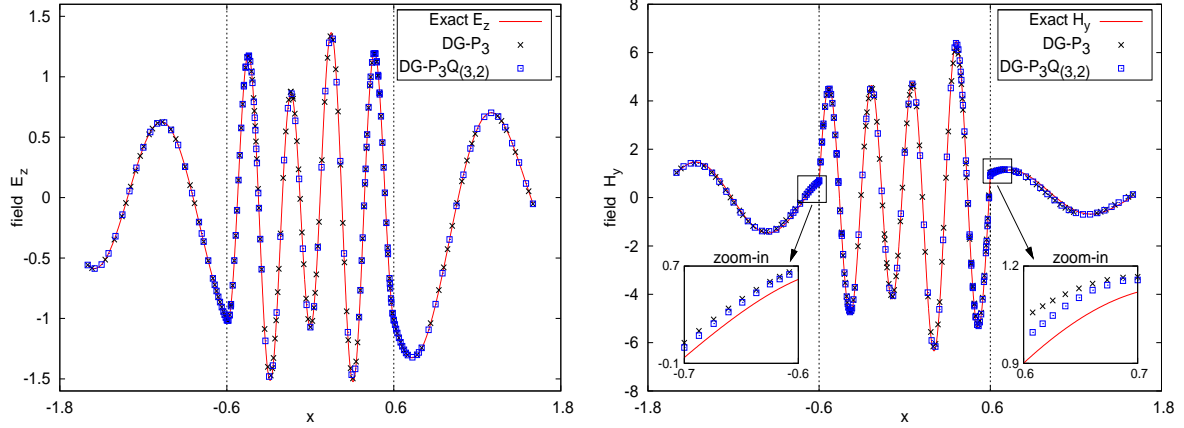


Figure 9: Plots of the E_z (left) and H_y (right) components along the axis $y = 0$ at time $t = 33.33$ ns for the scattering problem. We compare the exact solution (solid line) with solutions obtained using the hybrid DGTD- $\mathbb{P}_3\mathbb{Q}_{(3,2)}$ (square) and triangular DGTD- \mathbb{P}_3 (cross) methods. The vertical dotted lines at $x = \pm 0.6$ represent the material interfaces which distinguish the dielectric region, $|x| \leq 0.6$, with $\varepsilon_r = 12.0$ from the vacuum regions, $0.6 < |x| < 1.6$, with $\varepsilon_r = 1$. On the right figure we show zoomed-in views illustrating that main error arises at the material interfaces where the solution loses smoothness.

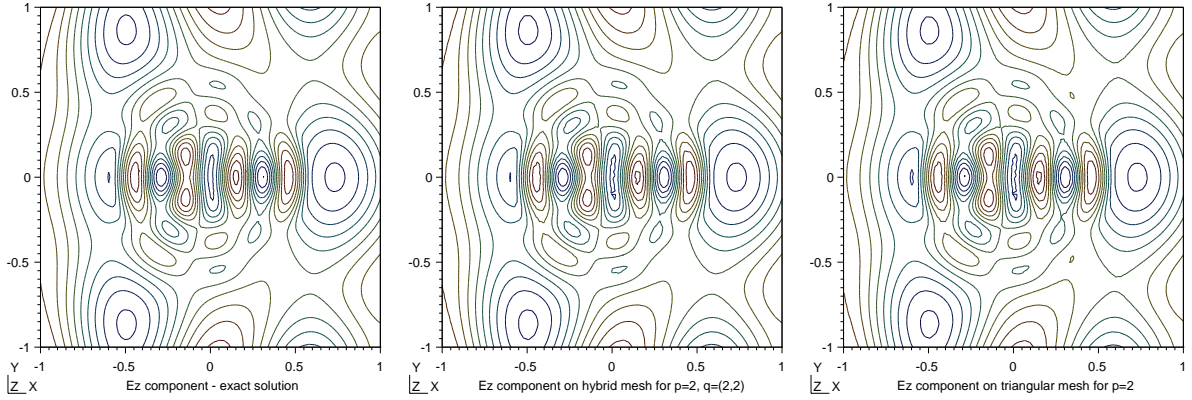


Figure 10: Contour lines of the E_z component at time $t = 33.33$ ns for the exact solution (left) and for solutions resulting from the hybrid DGTD- $\mathbb{P}_2\mathbb{Q}_{(2,2)}$ (center) and triangular DGTD- \mathbb{P}_2 (right) methods. Zoomed-in region to show more details. In all the plots, we use 13 equally spaced contours from $E_z = -1.5166$ to $E_z = 1.5299$.

5.1.3. Standing wave in a wedge-shaped PEC resonator

As an example with geometric singularities, we consider the propagation of a standing wave of frequency $f = 1293$ MHz and wavelength $\lambda = 0.232$ m inside a domain bounded by the curves $y = \tan(7\pi/4)x$, $x^2 + y^2 = 1/4$ and the x -axis. The non-convex nature of the outer boundary renders some components of the solution highly singular, making it a challenging problem. We compare approximate solutions obtained using the DGTD- \mathbb{P}_p method formulated on triangular meshes with those resulting from the DGTD- $\mathbb{P}_p\mathbb{Q}_q$ method formulated on hybrid non-conforming meshes. The triangular mesh consists of 6280 elements and 9531 interfaces with an average edge length of 0.057λ . The hybrid mesh contains 1745 triangles, 1820 quadrilaterals, 6228 conforming interfaces and 280 non-conforming interfaces. The global hybrid mesh has an average edge length of 0.082λ . Moreover, we can see that the hybrid mesh is locally refined in the vicinity of the re-entrant corner located at the origin. The resulting meshes are depicted in Figure 11. The simulation time has been set to $T = 38.5$ ns which corresponds to 50 periods. Results are summarized in Table 2 in terms of accuracy and computational costs. Here, the hybrid DG method produces numerical solutions that are one order of magnitude more accurate than those delivered by the triangular DG method. Figure 12 shows the contour lines of the H_x component at $t = T$ for numerical solutions obtained using the DGTD- \mathbb{P}_4 and DGTD- $\mathbb{P}_2\mathbb{Q}_4$ method. One can clearly observe a good qualitative agreement between the exact solution and the hybrid DGTD- $\mathbb{P}_2\mathbb{Q}_4$ solutions. The triangular DGTD- \mathbb{P}_4 solution produces some oscillations near the re-entrant corner due to bad resolution. This

is confirmed in Figure 13 where we plot the x -wise 1D distributions of the H_x and H_y components along the line $y = 5x + 0.025$ passing nearby the corner singularity. Inspecting Table 2 and Figure 13, we observe that, for the DGTD- \mathbb{P}_p method, the error decreases slowly when increasing the order from $p = 2$ to $p = 4$. In contrast, the hybrid method offers faster convergence when increasing the order in the quadrilateral elements from $q = 2$ to $q = 4$. More specifically, the global error depends on the global regularity of the exact solution which motivates the use of lower polynomial order together with mesh refinement near singular points in order to recover standard convergence rates.

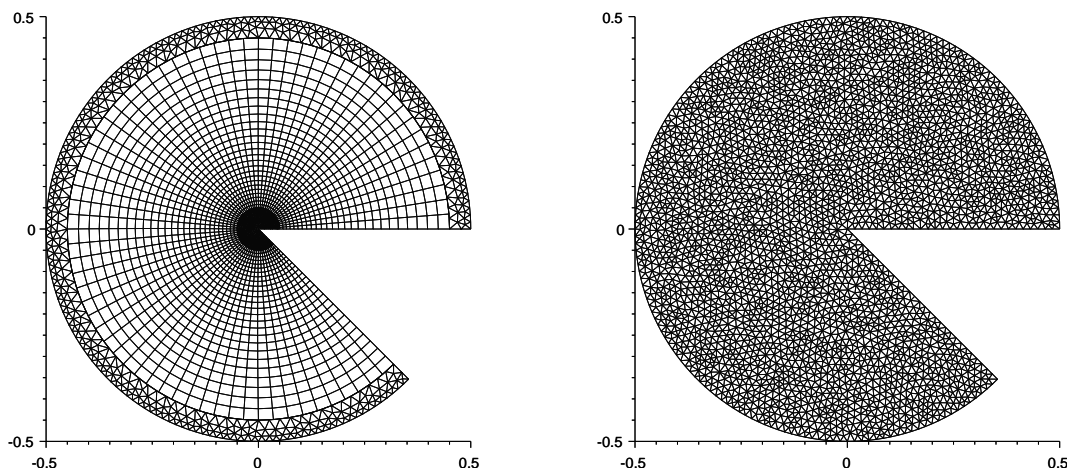


Figure 11: We show the non-conforming, locally refined, hybrid triangular-quadrilateral mesh (left) and the conforming fully triangular mesh (right), used in the simulation of the wedge-shaped resonator. The hybrid mesh is refined near the re-entrant corner using triangular elements.

Table 2: Comparison between the hybrid DGTD- $\mathbb{P}_p\mathbb{Q}_q$ method and the triangular DGTD- \mathbb{P}_p method. We give the maximum Δt allowed by the stability condition, the L^2 -errors at time $t = 38.5$ ns, the #DOF and the computing time (CPU) in seconds to achieve the final time period.

Method	Δt (ns)	L^2 -error	#DOF	CPU (s)
DGTD- \mathbb{P}_2	1.570E-02	5.482E-02	37680	1604
DGTD- \mathbb{P}_4	7.246E-03	2.381E-02	94200	9832
DGTD- $\mathbb{P}_2\mathbb{Q}_2$	1.724E-02	9.875E-03	26850	1211
DGTD- $\mathbb{P}_2\mathbb{Q}_4$	8.701E-03	1.043E-03	55970	6539

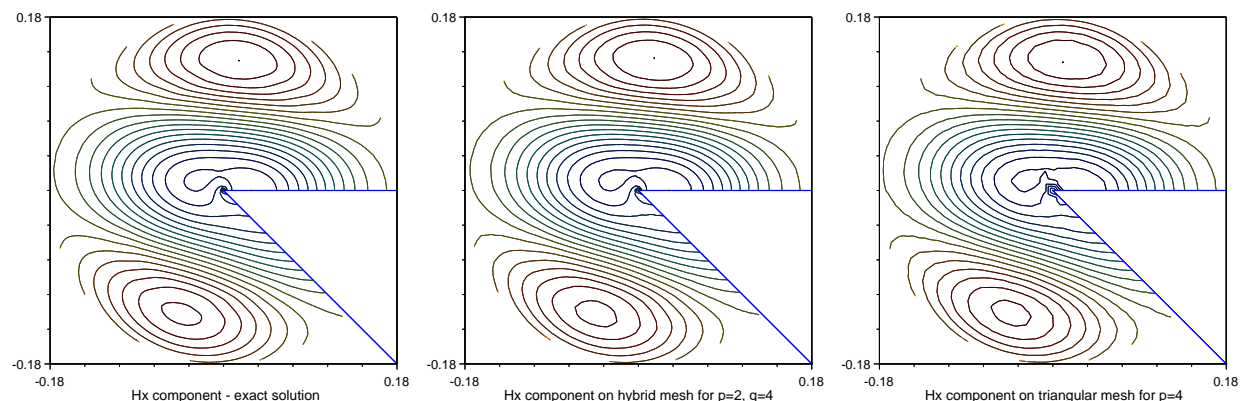


Figure 12: Contour lines of the H_x component at time $t = 38.5$ ns for the exact solution (left) and for solutions resulting from the hybrid DGTD- $\mathbb{P}_2\mathbb{Q}_4$ (center) and triangular DGTD- \mathbb{P}_4 (right) methods. Zoomed-in region to show more details. In all the plots, we use 20 equally spaced contours from $H_x = -0.440$ to $H_x = 0.402$.

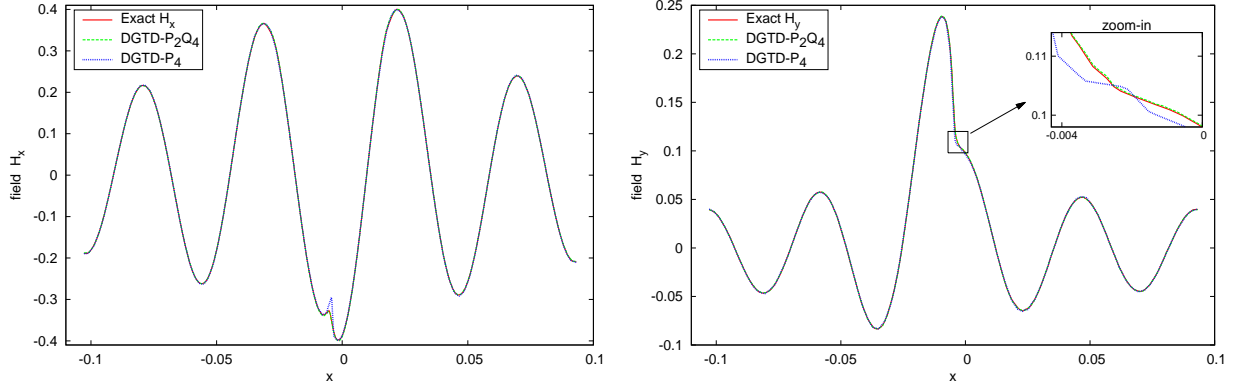


Figure 13: Plots of the H_x (left) and H_y (right) components along the line $y = 5x + 0.025$ at time $t = 38.5$ ns for the wedge-shaped cavity. We compare the exact solution (solid line) with solutions obtained using the DGTD- $\mathbb{P}_2\mathbb{Q}_4$ (dashed line) and DGTD- \mathbb{P}_4 (dotted line) methods.

5.2. Three-dimensional examples

5.2.1. Eigenmode in a spherical cavity

As a first verification of the general 3D framework, let us consider the propagation of the $(0, 1, 1)$ -mode inside a PEC spherical cavity of unit radius. The resonant frequency is 131 MHz and the wavelength is $\lambda = 2.29$ m. The exact time-domain solution of this problem can be found in [3]. We compare the accuracy and efficiency of the DG method using hexahedral, tetrahedral and hybrid meshes. First, a fully hexahedral mesh consisting of 6413 hexahedra and 19602 quadrilateral faces is generated. Then, the fully tetrahedral mesh is obtained by considering the hexahedral mesh and splitting each hexahedron to six tetrahedra. The resulting tetrahedral mesh has 38478 tetrahedra and 77682 triangular faces. Finally, the hybrid mesh contains 13068 tetrahedra, 216 regular hexahedra, 26136 triangular faces, 540 quadrilateral faces and 2916 hybrid non-conforming interfaces. These meshes are illustrated in Figure 14. The average edge lengths of the hexahedral, tetrahedral and hybrid meshes are respectively 0.0387λ , 0.0463λ and 0.0676λ which correspond to 26, 22 and 15 points per wavelength. We report on results obtained by the hexahedral DGTD- \mathbb{Q}_q , tetrahedral DGTD- \mathbb{P}_p and hybrid DGTD- $\mathbb{P}_p\mathbb{Q}_q$ methods with interpolation orders p and q ranging from $p = 1$ up to $p = 5$ with $p = q$. The simulation time is fixed to $T = 76$ ns which corresponds to 10 periods. Table 3 summarizes the values of time steps used in the simulation. In Figure 15 (left) we show the L^2 error as a function of the interpolation order. Inspecting these results we observe several things. First, the gain in accuracy is notable when increasing the interpolation order for all methods. Here, the exponential decay of the error as $\exp(-2p)$ is observed for all cases. Second, the tetrahedral method is 3.9 to 7.9 times more accurate than the hexahedral method when increasing the interpolation order from $p = 1$ to $p = 5$. The results reveal that the accuracy of the hybrid method is almost identical to that of the tetrahedral method. Figure 15 (center) shows the ratio between the #DOF required by the hexahedral, tetrahedral and hybrid methods to achieve the final simulation time. The results indicate that the hybrid method is able to reduce the #DOF for given p and q . For instance, with $p = q = 5$, the L^2 error is 7.1×10^{-5} for the hybrid mesh, 4.7×10^{-4} for the hexahedral method and 6.0×10^{-5} for the tetrahedral mesh. With the hybrid mesh, the #DOF is reduced by a factor of 1.8 compared to hexahedral mesh and by factor of 2.8 compared to tetrahedral mesh. Note that, the higher #DOF required by the tetrahedral mesh stems from the fact that the number of elements in the tetrahedral mesh is 6 times higher than those in the hexahedral mesh. Finally, the comparison in terms of computing times is displayed in Figure 15 (right). It can be observed that the reduction in the #DOF is translated into a corresponding reduction in computational cost. Here, the hybrid method allows for a reduction in computing time by a factor ranging from 2.4 to 8.0 compared to hexahedral method and by a factor of 3.5 compared to tetrahedral method. The lower computing times of the hybrid method partially stems from its larger time steps employed in hybrid mesh compared to those used in tetrahedral and hexahedral meshes.

5.2.2. Exposure of human head tissues to a localized source radiation

As a final example, we consider a more challenging problem which is concerned with the propagation of an EM wave emitted by a localized source in a realistic geometrical model of head tissues. Starting from magnetic resonance

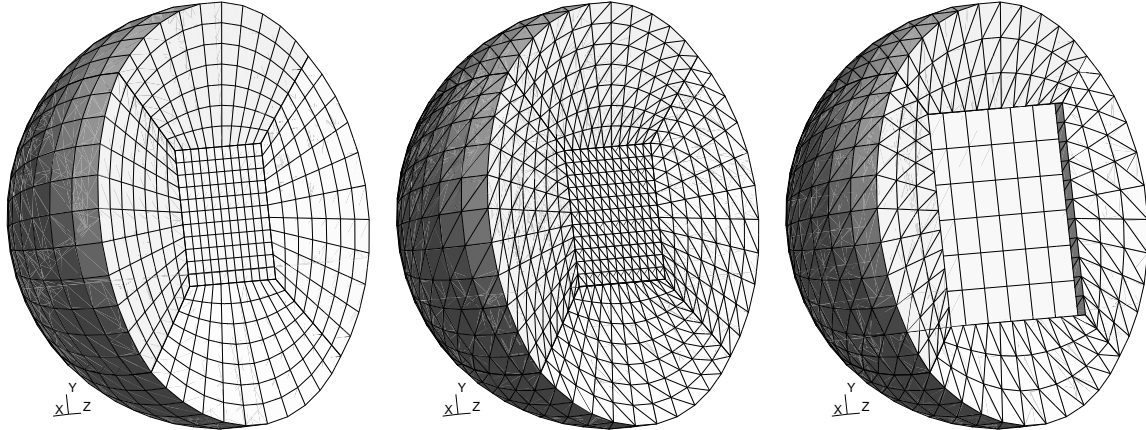


Figure 14: Cut-away view of the hexahedral mesh (left), the tetrahedral mesh (center) and the hybrid tetrahedral-hexahedral mesh (right), used in the simulation of the spherical cavity.

Table 3: The time steps allowed by the stability conditions of the DG method using hexahedral, tetrahedral and hybrid meshes for different interpolation orders.

	$p = q = 1$	$p = q = 2$	$p = q = 3$	$p = q = 4$	$p = q = 5$
Hexahedra	1.970E-02	9.850E-03	5.910E-03	4.728E-03	3.546E-03
Tetrahedra	2.357E-02	1.178E-02	7.072E-03	5.658E-03	4.243E-03
Hybrid	3.178E-02	1.589E-02	9.535E-03	7.628E-03	5.721E-03

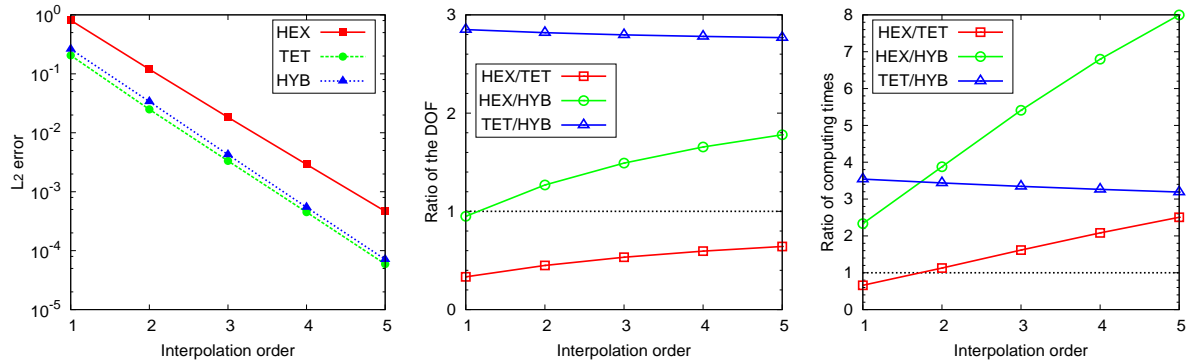


Figure 15: In the left we show the error at the final simulation time as a function of the interpolation order, obtained using DG methods on tetrahedral (TET), hexahedral (HEX) and hybrid (HYB) meshes. The ratio between the number of degrees of freedom and the ratio between the computing times are illustrated in the center and right panels, respectively.

images of the Visible Human 2.0 project [45], head tissues are segmented and the interfaces of a selected number of tissues are triangulated. Different strategies can be used in order to obtain a smooth and accurate segmentation of head tissues and interface triangulations as well. The strategy adopted in this example consists in using a variant of Chew's algorithm [9], based on Delaunay triangulation restricted to the interface, which allows to control the size and aspect ratio of interface triangles. Example of triangulations of the skin, skull and brain are shown in Figure 16. Then, these triangulated surfaces are used as inputs for the generation of volume meshes of the tissues. Finally, the GHS3D tetrahedral mesh generator [33] is used to mesh the volume domains between the various interfaces. The exterior of the head must also be meshed, up to a certain distance from the skin. The computational domain is here artificially bounded by a surface, denoted by Γ_a , on which the absorbing boundary condition is imposed. Overall, the constructed geometrical model considered here consists of four tissues (skin, skull, CSF - Cerebro Spinal Fluid and brain). The characteristics of the tissues are summarized in Table 4 where the values of the relative permittivity ϵ_r , and the conductivity σ , correspond to a frequency $F = 1800$ MHz and have been obtained from a special purpose online data

base. For all tissues, the relative permeability, μ_r , is set to 1. Given this tetrahedral mesh of the head tissues, the hybrid tetrahedral-hexahedral mesh is built in the following way (see the left panel of Figure 17): (a) the artificial boundary Γ_a is defined as a plane-parallel surface; (b) a second plane-parallel surface, denoted by Γ_0 , lying close to the head and completely enclosing it, is introduced; (c) the volume space between Γ_a and Γ_0 (vacuum space) is discretized using a regular hexahedral mesh; (d) the volume space between Γ_0 and the skin is discretized with unstructured tetrahedral elements. The global hybrid mesh consists of 197691 tetrahedra, 7000 hexahedra, 394397 triangular interfaces, 21900 quadrilateral interfaces and 5124 hybrid non-conforming interfaces. The minimum, maximum and average edge lengths of the hybrid mesh are equal to 1.12 mm, 43.14 mm and 17.26 mm, respectively. To compare the performance of the DG methods on hybrid and tetrahedral meshes, a fully tetrahedral mesh is generated by considering the hybrid mesh and replacing the hexahedral part by unstructured tetrahedral elements. We stress that the resulting tetrahedral mesh is conforming i.e., the same triangulated surface Γ_0 is used for the generation of tetrahedral elements inside and outside Γ_0 , as shown in Figure 17 (right). The global tetrahedral mesh consists of 350581 tetrahedra and 709395 interfaces. The minimum, maximum and average edge lengths of the tetrahedral mesh are equal to 0.49 mm, 44.32 mm and 13.98 mm, respectively. Finally, a dipolar type source is localized near the right ear of the head yielding a current of the form $\mathbf{J}_z(\mathbf{x}, t) = \delta(\mathbf{x} - \mathbf{x}_d)f(t)$, where $f(t)$ is a sinusoidally varying temporal signal and \mathbf{x}_d is the localization point of the source. The physical simulation time has been fixed to $T = 2.78$ ns which corresponds to five periods of the temporal signal. A discrete Fourier transform of the components of the electric field is computed during the last period of the simulation. For this problem, we consider using the DGTD- \mathbb{P}_2 method on the tetrahedral mesh and the DGTD- $\mathbb{P}_2\mathbb{Q}_2$ and DGTD- $\mathbb{P}_2\mathbb{Q}_1$ methods on the hybrid mesh. The time steps used in the simulations are $\Delta t = 2.728\text{E-}04$ ns for the DGTD- \mathbb{P}_2 method, $\Delta t = 6.655\text{E-}04$ ns for the DGTD- $\mathbb{P}_2\mathbb{Q}_2$ method and $\Delta t = 1.012\text{E-}03$ ns for the DGTD- $\mathbb{P}_2\mathbb{Q}_1$ method. These values correspond to the maximum Δt allowed by the stability conditions of the DG methods. Contour lines of the modulus of the electric field, $|\mathbf{E}|$, on selected planes of the skin for the approximate solutions resulting from the hybrid DGTD- $\mathbb{P}_2\mathbb{Q}_2$ and tetrahedral DGTD- \mathbb{P}_2 methods are visualized in Figure 18. Visually both methods provide almost the same solutions except slight amplitude differences around the right ear of the head. For the sake of completeness, we compare in Figure 19 the time evolution of the E_z and H_y components at a selected point in the center of the head. The comparisons show a visually perfect match between all solutions and no numerical artifacts due to the non-conformity of the mesh is observed. In Figure 19 we also plot the differences between the DGTD- \mathbb{P}_2 solution and the hybrid solutions, showing that the errors are relatively very small (less than 0.1% for DGTD- $\mathbb{P}_2\mathbb{Q}_1$ and less than 0.01% for DGTD- $\mathbb{P}_2\mathbb{Q}_2$). Finally, we give some informations on the simulation times required by both schemes. On a workstation equipped with an Intel Xeon CPU E5-2650 @2.00 GHz and 64 GB of RAM, the DGTD- \mathbb{P}_2 method requires 27 h 47 min for a total of 10183 time steps, the DGTD- $\mathbb{P}_2\mathbb{Q}_2$ method requires 7 h 35 min for a total of 4177 time steps, and the DGTD- $\mathbb{P}_2\mathbb{Q}_1$ method requires 4 h 17 min for a total of 2746 time steps. We obtain that in this case the hybrid DG method is 3.7 to 6.6 times less costly than the tetrahedral DG method.

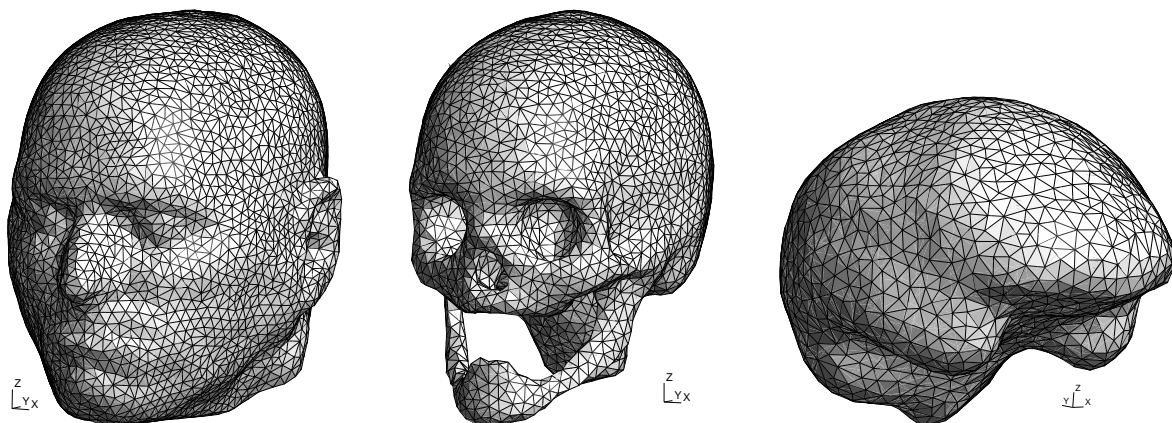


Figure 16: Surface mesh of the skin (left), the skull (center) and the brain (right).

Table 4: Electromagnetic characteristics of the selected head tissues at frequency $f = 1800$ MHz.

Tissue	ϵ_r	σ (S/m)	wavelength (mm)
Skin	38.87	1.18	26.73
Skull	15.56	0.43	42.25
CSF	67.20	2.92	20.33
Brain	43.55	1.15	25.26

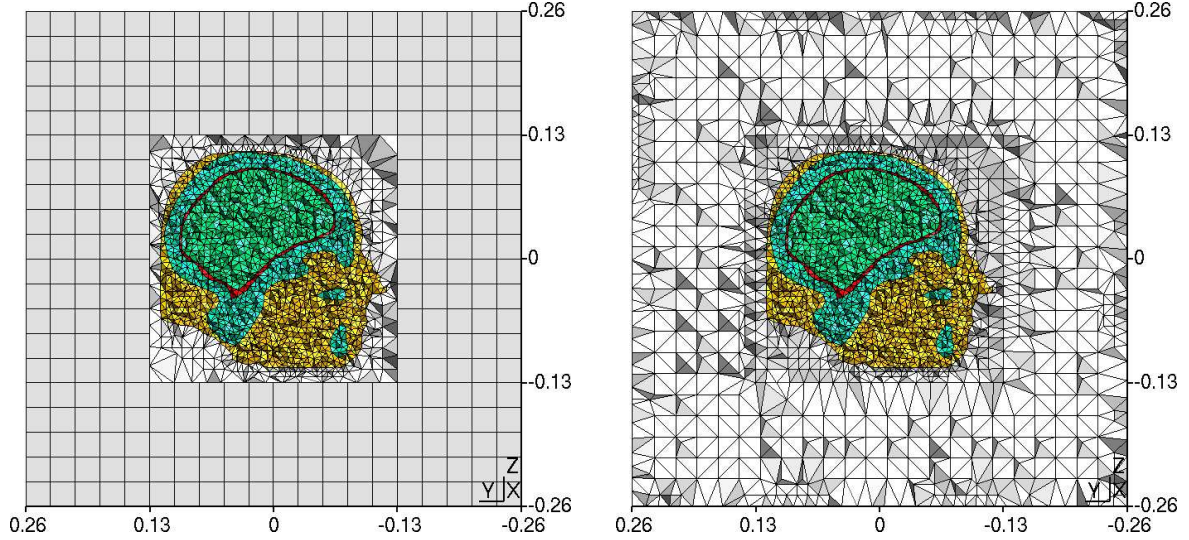


Figure 17: Planar views in a selected plane of the non-conforming hybrid tetrahedral-hexahedral mesh (left) and the conforming fully tetrahedral mesh (right) used for modelling the head tissues.

6. Concluding remarks

We have reported on our efforts to design a high-order accurate DG method for solving the time-domain Maxwell equations on non-conforming hybrid meshes. The proposed method combines a central numerical flux with a fourth-order leap-frog time integration. Within each mesh element, the EM field components are approximated using arbitrary high-order nodal polynomials. The geometrical flexibility is achieved by using unstructured triangular or tetrahedral meshes around complex geometric details and computational efficiency is then improved by using quadrilaterals or hexahedra to fill the remainder of the domain. The resulting DGTD method conserves a discrete form of the EM energy, and is stable under some CFL-like conditions. The key issues for the practical implementation of the proposed method are the generation of the hybrid meshes and the computation of flux matrices at the interfaces between hexahedral and tetrahedral faces. First, the mesh strategy adopted in this paper relies on the decomposition of the computational domain into non-overlapping subdomains, each one is meshed independently using the appropriate element type. Then, the hybrid interfaces are found using the method of separating axis and Boolean operations over polygons. This strategy demonstrates the benefits of relaxing the conformity requirement in the discretization process in view of facilitating the construction of meshes for complex propagation configurations, and improving the overall efficiency of the simulations. It is noteworthy that, most, if not all, of commercial or non-commercial mesh-generating softwares currently produce only conforming hybrid meshes by adding transition layers of pyramids or prisms. Second, the flux matrices are calculated using cubature rules over polygons. For a given accuracy, these rules are able to save more than 65% of the cubature points compared to other approaches. The proposed DGTD method is implemented in two and three space dimensions and tests on different wave propagation problems involving complex realistic geometries and heterogeneous media are presented. The spectral convergence for p -refinement method and the optimal convergence rate for h -refinement method have been verified. Results also show excellent agreement with exact solutions and no numerical artifacts caused by the non-conformity of the mesh are observed. Compared to DG

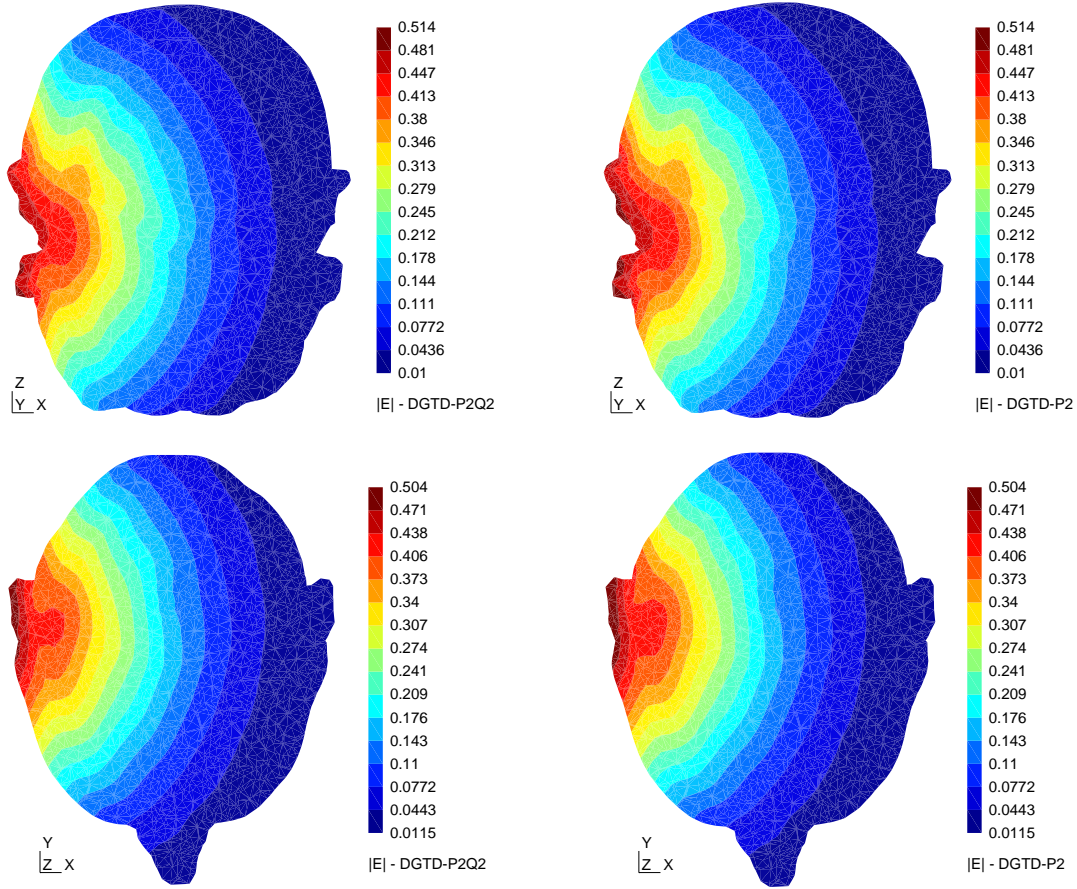


Figure 18: Contour lines of the modulus of the electric field, $|E|$, in selected cut planes of the head, using the hybrid DGTD- $\mathbb{P}_2\mathbb{Q}_2$ method (left) and the tetrahedral DGTD- \mathbb{P}_2 method (right).

methods on single element-type meshes, the DG method on hybrid meshes leads to clear reduction in computing time which is highly dependent on the interpolation order. In 2D, it is found that the hybrid mesh is able to produce at least a similar accuracy and allows the DG method to save up to 55% of the computing time and the number of degrees of freedom. In 3D, the gain becomes more significant, as the computational cost is reduced by up to 85%. A comparison between a fully hexahedral mesh and a fully tetrahedral mesh (obtained by converting each hexahedron to six tetrahedra) reveals that the tetrahedral method is always more accurate and more efficient than hexahedral method, excepting for linear interpolation orders. The parallelization of the proposed hybrid DG method and the comparison with DG methods on conforming hybrid meshes are left for future study.

Acknowledgements

The author would like to thank Dr. Seyed Mousavi of the University of California at Davis, USA, for providing the quadrature points over polygons to calculate flux matrices at hybrid interfaces. We also thank Dr. Nicolas Roquet for several discussions concerning the intersection tests for hybrid mesh generation. Finally, the author acknowledges partial support from the French National Agency for Research (ANR-10-JCJC-0905).

References

- [1] E. Abenius, U. Andersson, F. Edelvik, L. Eriksson, G. Ledfelt, Hybrid time domain solvers for the maxwell equations in 2d, *Int. J. Numer. Meth. Eng.* 53 (9) (2002) 2185–2199.

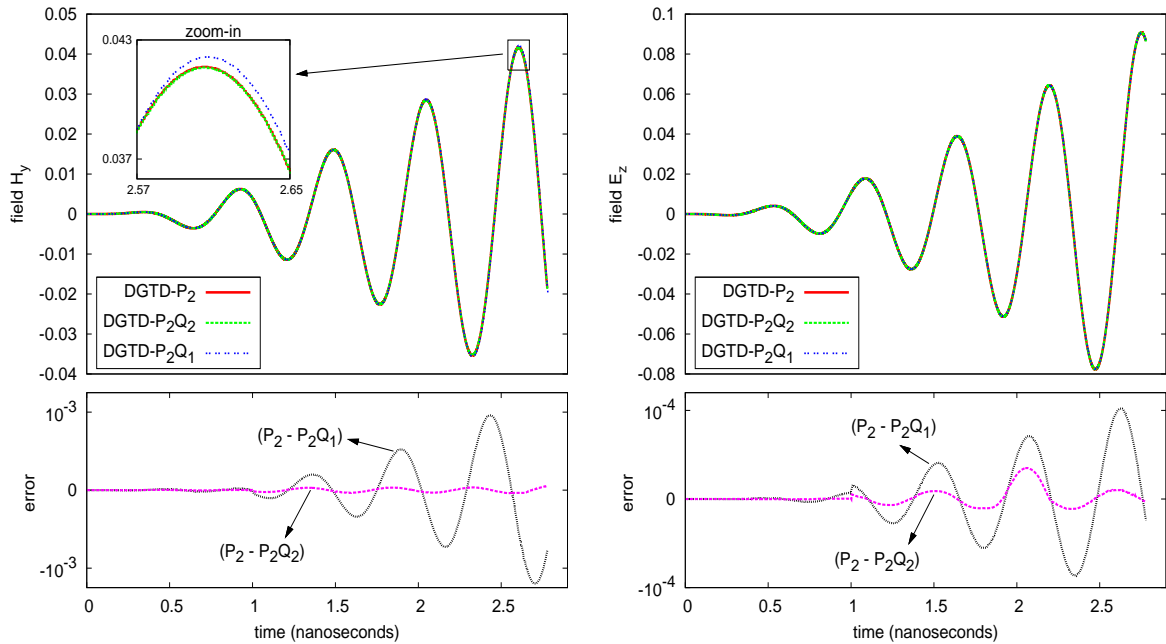


Figure 19: The top panels show the time evolution of H_y (left) and E_z (right) components at a selected point in the head. These solutions are obtained using the fully tetrahedral DGTD- \mathbb{P}_2 method (solid line) and the hybrid DGTD- $\mathbb{P}_2\mathbb{Q}_2$ (dashed line) and DGTD- $\mathbb{P}_2\mathbb{Q}_1$ (dotted line) methods. In the bottom panels we plot the differences between the tetrahedral and hybrid DG solutions denoted as $\mathbb{P}_2 - \mathbb{P}_2\mathbb{Q}_2$ and $\mathbb{P}_2 - \mathbb{P}_2\mathbb{Q}_1$.

- [2] T. Apel, N. Düvelmeyer, Transformation of hexahedral finite element meshes into tetrahedral meshes according to quality criteria, *Computing* 71 (4) (2003) 293–304.
- [3] C. A. Balanis, *Advanced Engineering Electromagnetics*, John Wiley & Sons, New York, 1989.
- [4] K. Barrett, Multilinear jacobians for isoparametric planar elements, *Finite Elem. Anal. Des.* 40 (8) (2004) 821–853.
- [5] M. Bergot, M. Duruflé, Higher-order discontinuous Galerkin method for pyramidal elements using orthogonal bases, *Numer. Meth. Part. Diff. Eq.* 29 (1) (2013) 144–169.
- [6] L. Branets, G. Carey, Extension of a mesh quality metric for elements with a curved boundary edge or surface, *J. Comput. Inf. Sci. Eng.* 5 (4) (2005) 302–308.
- [7] K. Busch, M. König, J. Niegemann, Discontinuous Galerkin methods in nanophotonics, *Laser & Photon. Rev.* 5 (6) (2011) 773–809.
- [8] G. Carey, Hexing the Tet, *Communications in Numerical Methods in Engineering* 18 (3) (2002) 223–227.
- [9] L. Chew, Guaranteed-quality mesh generation for curved surfaces, in: *9th Annual ACM Symposium Computational Geometry*, ACM Press, 1993, pp. 274–280.
- [10] B. Cockburn, G. Karniadakis, C. Shu (eds.), *Discontinuous Galerkin Methods. Theory, Computation and Applications*, vol. 11 of *Lecture Notes in Computational Science and Engineering*, Springer-Verlag, Berlin, 2000.
- [11] G. Cohen, X. Ferrieres, S. Pernet, A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell’s equations in time domain, *J. Comput. Phys.* 217 (2) (2006) 340–363.
- [12] R. Cools, Monomial cubature rules since “Stroud”: a compilation—part 2, *J. Comput. Appl. Math.* 112 (1-2) (1999) 21–27.
- [13] R. Cools, An encyclopaedia of cubature formulas, *J. Complex.* 19 (3) (2003) 445–453, online database (<http://nines.cs.kuleuven.be/ecf/>).
- [14] R. Cools, P. Rabinowitz, Monomial cubature rules since “Stroud”: a compilation, *J. Comput. Appl. Math.* 48 (3) (1993) 309–326.
- [15] S. Dossopoulos, B. Zhao, J.-F. Lee, Non-conformal and parallel discontinuous galerkin time domain method for maxwell’s equations: EM analysis of IC packages, *J. Comput. Phys.* 238 (0) (2013) 48–70.
- [16] D. Dunavant, Economical symmetrical quadrature rules for complete polynomials over a square domain, *Int. J. Numer. Meth. Eng.* 21 (10) (1985) 1777–1784.
- [17] D. Dunavant, High degree efficient symmetrical Gaussian quadrature rules for the triangle, *Int. J. Numer. Meth. Eng.* 21 (6) (1985) 1129–1148.
- [18] D. Dunavant, Efficient symmetrical cubature rules for complete polynomials of high degree over the unit cube, *Int. J. Numer. Meth. Eng.* 23 (3) (1986) 397–407.
- [19] C. Durochat, S. Lanteri, DGTD method on hybrid meshes for time domain electromagnetics, in: *URSI International Symposium on Electromagnetic Theory (EMTS)*, 2010, pp. 992–995.
- [20] C. Durochat, S. Lanteri, C. Scheid, High order non-conforming multi-element discontinuous Galerkin method for time-domain electromagnetics, in: *International Conference on Electromagnetics in Advanced Applications (ICEAA)*, Cape Town, South Africa, 2012, pp. 379–382.
- [21] D. Eberly, *3D Game Engine Design*, 2nd ed., Morgan Kaufmann Publishers, 2007.
- [22] D. Eberly, *Game Physics*, 2nd ed., Morgan Kaufmann Publishers, 2010.
- [23] F. Edelvik, G. Ledfelt, Explicit hybrid time domain solver for the Maxwell equations in 3D, *J. Sci. Comput.* 15 (1) (2000) 61–78.
- [24] H. Fahs, Development of a *hp*-like discontinuous Galerkin time-domain method on non-conforming simplicial meshes for electromagnetic

- wave propagation, *Int. J. Numer. Anal. Model.* 6 (2) (2009) 193–216.
- [25] H. Fahs, High-order leap-frog based discontinuous Galerkin method for the time-domain Maxwell equations on non-conforming simplicial meshes, *Numer. Math. Theor. Meth. Appl.* 2 (3) (2009) 275–300.
- [26] H. Fahs, Improving accuracy of high-order discontinuous Galerkin method for time-domain electromagnetics on curvilinear domains, *Int. J. Comput. Math.* 88 (10) (2011) 2124–2153.
- [27] H. Fahs, Investigation on polynomial integrators for time-domain electromagnetics using a high-order discontinuous Galerkin method, *Appl. Math. Model.* 36 (11) (2012) 5466–5481.
- [28] H. Fahs, L. Fezoui, S. Lanteri, F. Rapetti, Preliminary investigation of a non-conforming discontinuous Galerkin method for solving the time-domain Maxwell equations, *IEEE Trans. Magn.* 44 (6) (2008) 1254–1257.
- [29] H. Fahs, A. Hadjem, S. Lanteri, J. Wiart, M.-F. Wong, Calculation of the SAR induced in head tissues using a high order DGTD method and triangulated geometrical models, *IEEE Trans. Antennas Propag.* 59 (12) (2011) 4669–4678.
- [30] H. Fahs, S. Lanteri, A high-order non-conforming discontinuous Galerkin method for time-domain electromagnetics, *J. Comput. Appl. Math.* 234 (4) (2010) 1088–1096.
- [31] X. Ferrieres, J.-P. Parmentier, S. Bertuol, A. R. Ruddle, Application of a hybrid finite difference/finite volume method to solve an automotive EMC problem, *IEEE Trans. Electromagn. Compat.* 46 (4) (2004) 624–634.
- [32] G. Gassner, F. Lörcher, C.-D. Munz, J. Hesthaven, Polymorphic nodal elements and their application in discontinuous Galerkin methods, *J. Comput. Phys.* 228 (5) (2009) 1573–1590.
- [33] P.-L. George, F. Hecht, E. Saltel, Automatic mesh generator with specified boundary, *Comput. Methods Appl. Mech. Engrg.* 92 (3) (1991) 269–288.
- [34] S. Gottschalk, M. Lin, D. Manocha, OBB-tree: a hierarchical structure for rapid interference detection, in: *Proceedings of ACM Siggraph’96*, ACM, New York, USA, 1996, pp. 171–180.
- [35] J. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, *SIAM J. Numer. Anal.* 35 (2) (1998) 655–676.
- [36] J. Hesthaven, C. Teng, Stable spectral methods on tetrahedral elements, *SIAM J. Sci. Comput.* 21 (6) (1999) 2352–2380.
- [37] J. Hesthaven, T. Warburton, Nodal high-order methods on unstructured grids. I. Time-domain solution of Maxwell’s equations, *J. Comput. Phys.* 181 (1) (2002) 186–221.
- [38] J. Hesthaven, T. Warburton, *Nodal discontinuous Galerkin methods: Algorithms, Analysis and Applications*, Springer Texts Appl. Math., Springer-Verlag, Berlin, 2008.
- [39] P. Hubbard, Constructive solid geometry for triangulated polyhedra, Tech. Rep. CS-90-07, Brown University (1990).
- [40] P. Knabner, S. Korotov, G. Summ, Conditions for the invertibility of the isoparametric mapping for hexahedral finite elements, *Finite Elem. Anal. Des.* 40 (2) (2003) 159–172.
- [41] P. M. Knupp, On the invertibility of the isoparametric map, *Comput. Methods Appl. Mech. Engrg.* 78 (3) (1990) 313–329.
- [42] S. Mousavi, H. H. Xiao, N. Sukumar, Generalized gaussian quadrature rules on arbitrary polygons, *Int. J. Numer. Meth. Eng.* 82 (1) (2010) 99–113.
- [43] S. Pernet, X. Ferrieres, G. Cohen, High spatial order finite element method to solve Maxwell’s equations in time domain, *IEEE Trans. Antennas Propag.* 53 (9) (2005) 2889–2899.
- [44] S. Piperno, M. Remaki, L. Fezoui, A nondiffusive finite volume scheme for the three-dimensional Maxwell’s equations on unstructured meshes, *SIAM J. Numer. Anal.* 39 (6) (2002) 2089–2108.
- [45] P. Ratiu, B. Hillen, J. Glaser, D. Jenkins, *Medicine Meets Virtual Reality 11 - NextMed: Health Horizon*, vol. 11, chap. Visible Human 2.0 - the next generation, IOS Press, 2003, pp. 275–281.
- [46] M. Rivero, F. Feito, Boolean operations on general planar polygons, *Computers & Graphics* 24 (6) (2000) 881–896.
- [47] T. Rylander, A. Bondeson, Stability of explicit-implicit hybrid time-stepping schemes for Maxwell’s equations, *J. Comput. Phys.* 179 (2) (2002) 426–438.
- [48] P. Schneider, D. Eberly, *Geometric Tools for Computer Graphics*, 1st ed., Morgan Kaufmann Publishers, 2003.
- [49] S. Schnepf, T. Weiland, Efficient large scale electromagnetic simulations using dynamically adapted meshes with the discontinuous Galerkin method, *J. Comput. Appl. Math.* 236 (18) (2012) 4909 – 4924.
- [50] R. Sevilla, O. Hassan, K. Morgan, The use of hybrid meshes to improve the efficiency of a discontinuous Galerkin method for the solution of maxwell’s equations, *Comput. Struct.* 137 (2014) 2–13.
- [51] H. Songoro, M. Vogel, Z. Cendes, Keeping time with Maxwell’s equations, *IEEE Microw. Mag.* 11 (2) (2010) 42–49.
- [52] N. Sukumar, A. Tabarraei, Conforming polygonal finite elements, *Int. J. Numer. Meth. Eng.* 61 (12) (2004) 2045–2066.
- [53] A. Taflove, S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd ed., Artech House, Norwood, MA, 2005.
- [54] S. Wandzura, H. Xiao, Symmetric quadrature rules on a triangle, *Comput. Math. Appl.* 45 (12) (2003) 1829–1840.
- [55] R. B. Wu, T. Itoh, Hybrid finite-difference time-domain modeling of curved surfaces using tetrahedral edge elements, *IEEE Trans. Antennas Propag.* 45 (8) (1997) 1302–1309.
- [56] L. Zhang, T. Cui, H. Liu, A set of symmetric quadrature rules on triangles and tetrahedra, *J. Comp. Math.* 27 (1) (2009) 89–96.