

# High-order discontinuous Galerkin method for time-domain electromagnetics on geometry-independent Cartesian meshes

Héctor Navarro-García<sup>1</sup>  | Rubén Sevilla<sup>2</sup>  | Enrique Nadal<sup>1</sup>  | Juan José Ródenas<sup>1</sup> 

<sup>1</sup>Institute of Mechanical and Biomechanical Engineering, Universitat Politècnica de València, Valencia, Spain

<sup>2</sup>Zienkiewicz Centre for Computational Engineering, Faculty of Science and Engineering, Swansea University, Swansea, Wales, UK

## Correspondence

Héctor Navarro-García, Institute of Mechanical and Biomechanical Engineering, Universitat Politècnica de València, Cami de Vera, s/n, Valencia, 46022, Spain.

Email: hecnaga1@upv.es

## Funding information

Engineering and Physical Sciences Research Council, Grant/Award Number: EP/T009071/1; Ministerio de Ciencia, Innovación y Universidades, Grant/Award Number: FPU17/03993; Ministerio de Economía y Competitividad, Grant/Award Number: DPI2017-89816-R

## Abstract

In this work we present the Cartesian grid discontinuous Galerkin (cgDG) finite element method, a novel numerical technique that combines the high accuracy and efficiency of a high-order discontinuous Galerkin discretization with the simplicity and hierarchical structure of a geometry-independent Cartesian mesh. The elements that intersect the boundary of the physical domain require special treatment in order to minimize their effect on the performance of the algorithm. We considered the exact representation of the geometry for the boundary of the domain avoiding any nonphysical artifacts. We also define a stabilization procedure that eliminates the step size restriction of the time marching scheme due to extreme cut patterns. The unstable degrees of freedom are eliminated and the supporting regions of their shape functions are reassigned to neighboring elements. A subdomain matching algorithm and an a posteriori enrichment strategy are presented. Combining these techniques we obtain a final spatial discretization that preserves stability and accuracy of the standard body-fitted discretization. The method is validated through a series of numerical tests and it is successfully applied to the solution of problems of interest in the context of electromagnetic scattering with increasing complexity.

## KEYWORDS

Cartesian grid finite element method, discontinuous Galerkin, fictitious domain method, high-order discretization, Maxwell's equations

## 1 | INTRODUCTION

The finite-difference (FD) time-domain method (FDTD) presented by Yee<sup>1</sup> and developed by Taflove<sup>2</sup> has remained the predominant technique within the field of computational electromagnetics for the solution of Maxwell's equations in the time domain. However, in its simplest implementation, the FDTD method needs a structured mesh which is not capable of accurately representing complex geometries. A variety of methods that propose the combination of the FDTD with finite element (FE) or finite volume (FV) methods has been studied.<sup>3,4</sup> However, none of them has overcome the main limitations of the FDTD scheme. These combined approaches also rely on a piecewise linear approximation of the boundary which originates numerical diffraction phenomena. Furthermore, the low-order interpolation of the solution

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. *International Journal for Numerical Methods in Engineering* published by John Wiley & Sons Ltd.

introduces excessive numerical dispersion and/or dissipation, which limits its scope of application to simulations where the electromagnetic wave propagates along just a few wavelengths.

High-order methods are a powerful alternative to Yee's algorithm. The polynomial approximation of the geometry allows the boundary to be captured with greater precision which reduces the effect of the nonphysical diffraction. Likewise, by increasing the degree of the approximation, the dispersion and dissipation errors inherent to the numerical method are reduced.

Among all the available high-order methods, FE techniques based on the discontinuous Galerkin (DG) formulation have gained increasing interest since their appearance in the 1970s.<sup>5,6</sup> These strategies exhibit some desirable features that alternative approaches do not. Unlike for high-order FV or FD methods, the order of accuracy of the approximation depends on trial functions that are locally defined on each element. In comparison with standard FE methods, the order of the approximation can be easily adapted at the element level regardless of the discretization used in its surroundings. When combined with explicit time marching schemes, the mass matrix arising from the weak formulation of the problem is block diagonal, which means that the global system can be split into a set of element-by-element problems that can be computed in parallel.<sup>7,8</sup> Taking advantage of the multicore architectures of modern CPUs, this feature significantly speeds up the computations. Finally, the use of numerical fluxes provides an easy way to stabilize the numerical scheme for convection dominated problems.

The combination of DG and explicit Runge–Kutta methods leads to algorithms with a high convergence rate in both space and time.<sup>6</sup> However, explicit methods obey the Courant–Friedrichs–Levy (CFL)<sup>9</sup> condition that limits the maximum size of the time step that can be used. Furthermore, the overall CFL condition of a numerical approach depends on local properties of the discretization. In meshes with very different element sizes, due to local  $h$ -refinement, the smallest element defines the maximum size of the time step. In the literature, a variety of methods that seeks to minimize the effect of this dependency has been proposed. Strong-stability preserving Runge–Kutta methods<sup>10,11</sup> or the multistep Runge–Kutta methods<sup>12–14</sup> are some examples. As an alternative we may consider implicit methods.<sup>15,16</sup> These time integrators do not suffer from instability issues even for large step sizes but the cost to pay is the solution of the global system of equations. The matrix of the system to be solved each time step is no longer block-diagonal. Hybrid methods called locally implicit Runge–Kutta methods that seek to combine the best of both worlds have been also proposed.<sup>17–20</sup>

The accurate representation of the geometry is critical in the framework of high-order methods since it can become the limiting factor in terms of accuracy. Previous research shows that the approximate representation of the geometry must be at least of the same order as the solution in order to maintain the convergence rate of the method.<sup>21</sup> In this regard, the use of isoparametric approaches with curved elements is the most extended. However, it requires the generation of curvilinear high-order meshes, which is still challenging for complex geometries in three dimensions.<sup>22</sup> The use of exact boundary representation has shown advantages in this context,<sup>23,24</sup> but the automatic generation of meshes for NEFEM is even more challenging and, nowadays, it is only available for 2D geometries.<sup>25</sup>

The mesh generation problem can be overcome by using nonconforming methods, in which the analysis mesh is computed starting with a simple and easy-to-mesh shape that contains the problem domain. The result is a geometry independent discretization of the domain of interest. Among all the available nonconforming methods, we find the fictitious domain methods (FDM)<sup>26,27</sup> that capture the physical domain of the problem by assigning different properties to each point based on which domain it lies on. The Cartesian grid finite element method (cgFEM)<sup>28</sup> is an example of FDM whose main feature is the use of a mesh made up of regular quadrilateral (2D) or regular hexahedral (3D) elements. This spatial discretization presents very convenient characteristics from the computational point of view. The hierarchy of the mesh can be used to project information among meshes through  $h$ -refinement processes preventing from redundant calculations. Moreover, the structure of the mesh simplifies tasks related to locating points in elements or finding connectivities between entities of the mesh. The similarity of the elements leads to a major reduction in computational time and memory requirements as the integrals of all elements with the same integrand can be computed by scaling those obtained in a reference element. The cgFEM has been successfully applied for the solution of both image and geometry based structural analysis in a range of problem such as linear elasticity,<sup>29</sup> plasticity,<sup>30</sup> contact,<sup>31–33</sup> optimization,<sup>28,34,35</sup> image-based analysis,<sup>36,37</sup> and bone regeneration controlled by mechanoregulatory models.<sup>38</sup>

In this work, the use of the DG formulation with high-order approximations in the context of cgFEM for the solution of Maxwell's equations in the time domain is proposed for the first time. The resulting method brings together the advantages of both methods allowing the numerical simulation of the propagation of electromagnetic waves in complex domains with a high level of accuracy. The high fidelity of the high-order method is ensured by using integration strategies based on NEFEM that allow the exact representation of the geometry.<sup>21,39</sup> For elements with a small fraction of the area inside the computational domain a novel stabilization strategy is proposed to ensure that the maximum allowed time step leads to an

stable time marching scheme. The strategy consists of extending the solution of an element over the domain assigned to an adjacent one while reducing to zero the support of the shape functions defined on this second element. Therefore, the nodal values linked to this interpolation functions can be removed from the system of equations. Following this approach, we get rid of the unstable degrees of freedom. Independently, other authors have proposed similar strategies to improve the conditioning in elliptic problems<sup>40</sup> and to solve the Navier–Stokes equations with nonconforming meshes.<sup>41–43</sup> The strategy described in this contribution is more sophisticated than previous ones, which translates in greater robustness and versatility. Moreover, we performed a set of numerical test that let better understand the effects of this kind of stabilization techniques on the overall performance of the algorithm.

The remainder of the article is organized as follows. In Section 2 the DG formulation applied to the electromagnetic problem is briefly presented. Next, Section 3 summarizes the key aspects of cgFEM and points out some important features regarding its implementation. The combination of the two techniques that leads to the cgDG method is covered in Section 4. The maximum efficiency of the algorithm is sought by identifying casuistry and deriving calculation procedures that minimize the computational cost and the memory requirements related to the evaluation of the discrete operators. Section 5 is intended to present the stabilization strategy. Once the core idea of this approach is presented, a detailed description of the domain matching algorithm is described. The numerical tests and the results obtained for problems of interest in the context of computational electromagnetics are collected in Section 6. The performance of the proposed method is studied and applied to a range of problems with increasing difficulty, either as a consequence of an increase in frequency, an increase in the complexity of the geometry or both. Some conclusions and a series of future lines of research are presented in Section 7 to close this contribution.

## 2 | DISCONTINUOUS GALERKIN FORMULATION

In this section, we give a brief description of the DG formulation. The reader is urged to consult<sup>6,44–46</sup> for further information.

Let us consider the transient Maxwell's equations written in conservation dimensionless form and using Einstein's notation:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_k(\mathbf{U})}{\partial x_k} = \mathbf{S}(\mathbf{U}), \quad (1)$$

where  $\mathbf{U}$  is the vector of conservative variables,  $\mathbf{F}_k(\mathbf{U})$  is the hyperbolic flux in the  $k$ th direction, and  $\mathbf{S}(\mathbf{U})$  is the vector gathering all source terms. In the context of 2D scattering problems, those vectors are:

$$\mathbf{U} = \begin{Bmatrix} \varepsilon E_1 \\ \varepsilon E_2 \\ \mu H_3 \end{Bmatrix}, \quad \mathbf{F}_1 = \begin{Bmatrix} 0 \\ H_3 \\ E_2 \end{Bmatrix}, \quad \mathbf{F}_2 = \begin{Bmatrix} -H_3 \\ 0 \\ -E_1 \end{Bmatrix}, \quad \mathbf{S} = \begin{Bmatrix} (1 - \varepsilon) \frac{\partial E_1^i}{\partial t} \\ (1 - \varepsilon) \frac{\partial E_2^i}{\partial t} \\ (1 - \mu) \frac{\partial H_3^i}{\partial t} \end{Bmatrix} \quad (2)$$

for the *transverse electric* mode (TE<sup>z</sup>) and:

$$\mathbf{U} = \begin{Bmatrix} \mu H_1 \\ \mu H_2 \\ \varepsilon E_3 \end{Bmatrix}, \quad \mathbf{F}_1 = \begin{Bmatrix} 0 \\ -E_3 \\ -H_2 \end{Bmatrix}, \quad \mathbf{F}_2 = \begin{Bmatrix} E_3 \\ 0 \\ H_1 \end{Bmatrix}, \quad \mathbf{S} = \begin{Bmatrix} (1 - \mu) \frac{\partial H_1^i}{\partial t} \\ (1 - \mu) \frac{\partial H_2^i}{\partial t} \\ (1 - \varepsilon) \frac{\partial E_3^i}{\partial t} \end{Bmatrix} \quad (3)$$

for the *transverse magnetic* mode (TM<sup>z</sup>). In (2) and (3),  $E_j$  represents the  $j$  component of the scattered electric field vector,  $H_j$  represents the  $j$  component of the scattered magnetic field vector, and  $\varepsilon$  and  $\mu$  are the relative permittivity and permeability of the medium (we consider a linear isotropic material without current sources), respectively. Superscript  $i$  identifies the incident field. Note that we adopt the scattered field formulation to avoid the propagation of the incident field from the outer boundary of the model to the scatterer which reduces the computational cost of the simulation and has a positive impact on the accuracy of the solution. See Reference 2 for further details.

Let us consider a bounded domain  $\Omega$  and its decomposition into a set of nonoverlapping subdomains  $\Omega_e$  in such a way that  $\Omega = \bigcup_e \Omega_e$ . We locally approximate the solution via a polynomial interpolation defined by a set of nodal values on  $\Omega_e$ . The functions used to interpolate constitute a basis of the space of polynomials up to degree  $p$  in  $\Omega_e$ , denoted by

$P_p(\Omega_e)$ . The test functions considered to derive the weak form are polynomials also contained in the space  $P_p(\Omega_e)$  and they are gathered in a vector of the same size as  $\mathbf{U}$ , namely,  $\mathbf{W}$ . Multiplying (1) by  $\mathbf{W}$ , integrating over  $\Omega_e$  and integrating by parts, we obtain the following weak form in an element:

$$\int_{\Omega_e} \mathbf{W} \cdot \frac{\partial \mathbf{U}_e}{\partial t} d\Omega - \int_{\Omega_e} \frac{\partial \mathbf{W}}{\partial x_k} \cdot \mathbf{F}_k(\mathbf{U}_e) d\Omega + \int_{\partial\Omega_e} \mathbf{W} \cdot \mathbf{F}_n(\mathbf{U}_e) d\Gamma = \int_{\Omega_e} \mathbf{W} \cdot \mathbf{S}(\mathbf{U}_e) d\Omega, \quad (4)$$

where  $\mathbf{U}_e$  represents the restriction of the vector of conservative variables  $\mathbf{U}$  to the element domain and  $\mathbf{F}_n$  represents the flux in the normal direction  $\mathbf{n}$ , being  $\mathbf{n}$  the outward unit normal vector to the boundary  $\partial\Omega_e$ :

$$\mathbf{F}_n(\mathbf{U}) = n_k \mathbf{F}_k(\mathbf{U}). \quad (5)$$

To account for the discontinuous nature of the approximation, the normal flux is replaced by a numerical normal flux  $\mathcal{F}_n(\mathbf{U}_e, \mathbf{U}_{\text{out}})$  that depends not only on the solution within the element  $\mathbf{U}_e$  but also depends on the solution on the neighboring element  $\mathbf{U}_{\text{out}}(\mathbf{x})$ , defined as:

$$\mathbf{U}_{\text{out}}(\mathbf{x}) = \lim_{\delta \rightarrow 0^+} \mathbf{U}(\mathbf{x} + \delta \mathbf{n}) \quad \forall \mathbf{x} \in \partial\Omega_e. \quad (6)$$

Choosing the numerical flux is a crucial step as it dictates the stability and the accuracy of the numerical scheme. Some of the most widespread approaches consist of using a centered flux, an upwind flux or a linear combinations of the two previous schemes, see References 47-52. For linear hyperbolic problem, such as the one considered here, a typical choice consists of using a flux splitting approach. This formulation is based on dividing the flux across the boundary of the element into outgoing (superscript +) and incoming (superscript -) fluxes regarding the positive and negative eigenvalues of the Jacobian matrix  $\mathbf{A}_n = \partial \mathbf{F}_n / \partial \mathbf{U}$ , respectively:

$$\mathbf{F}_n(\mathbf{U}) = \mathbf{F}_n^+(\mathbf{U}) + \mathbf{F}_n^-(\mathbf{U}). \quad (7)$$

The upwind numerical flux is computed by considering that the incoming component of the flux is due to the distribution of electromagnetic field on the outside, while the outgoing flux depends on the distribution inside the element:

$$\mathcal{F}_n(\mathbf{U}_e, \mathbf{U}_{\text{out}}) = \mathbf{F}_n^+(\mathbf{U}_e) + \mathbf{F}_n^-(\mathbf{U}_{\text{out}}). \quad (8)$$

Substituting (8) into (4) and integrating by parts again, we obtain the following weak form in an element:

$$\int_{\Omega_e} \mathbf{W} \cdot \frac{\partial \mathbf{U}_e}{\partial t} d\Omega + \int_{\Omega_e} \mathbf{W} \cdot \frac{\partial \mathbf{F}_k(\mathbf{U}_e)}{\partial x_k} d\Omega + \int_{\partial\Omega_e} \mathbf{W} \cdot [\mathcal{F}_n(\mathbf{U}_e, \mathbf{U}_{\text{out}}) - \mathbf{F}_n(\mathbf{U}_e)] d\Gamma = \int_{\Omega_e} \mathbf{W} \cdot \mathbf{S}(\mathbf{U}_e) d\Omega. \quad (9)$$

Considering the jump operator  $[[\mathbf{G}(\mathbf{U}_e)]] = \mathbf{G}(\mathbf{U}_{\text{out}}) - \mathbf{G}(\mathbf{U}_e)$ , in which  $\mathbf{G}$  represents any operator, the difference between fluxes can be simplified leading to:

$$\int_{\Omega_e} \mathbf{W} \cdot \frac{\partial \mathbf{U}_e}{\partial t} d\Omega + \int_{\Omega_e} \mathbf{W} \cdot \frac{\partial \mathbf{F}_k(\mathbf{U}_e)}{\partial x_k} d\Omega + \int_{\partial\Omega_e} \mathbf{W} \cdot [[\mathbf{F}_n^-(\mathbf{U}_e)]] d\Gamma = \int_{\Omega_e} \mathbf{W} \cdot \mathbf{S}(\mathbf{U}_e) d\Omega. \quad (10)$$

Note that the integral of the flux over the boundary penalizes the difference between the incoming flux due to the internal and external fields, namely, the incoming flux jump. For an internal edge, both fields are elemental solutions and the weak continuity of the global solution is imposed as a result. For an external edge, that is, an edge over  $\partial\Omega$ , the outer state is not defined. As usual in the framework of hyperbolic equations, the Rankine-Hugoniot jump conditions are employed to compute the incoming flux jump that, when integrated over the boundary, translates into a weak enforcement of the boundary conditions.

The examples considered in this article focus on the electromagnetic scattering by perfect electric conductors (PEC). The tangential component of the total electric field vanish on the boundary of a PEC, namely,

$$\mathbf{n} \times \mathbf{E} + \mathbf{n} \times \mathbf{E}^i = \mathbf{0}. \quad (11)$$

The expression of the boundary term results

$$\llbracket \mathbf{F}_{\mathbf{n}}^{-}(\mathbf{U}_e) \rrbracket = \frac{1}{2} \left[ \llbracket H_3 \rrbracket - \sqrt{\frac{\varepsilon}{\mu}} (n_1 \llbracket E_2 \rrbracket - n_2 \llbracket E_1 \rrbracket) \right] \begin{Bmatrix} -n_2 \\ n_1 \\ -\sqrt{\frac{\mu}{\varepsilon}} \end{Bmatrix} \quad (12)$$

for the TE<sup>z</sup> mode and

$$\llbracket \mathbf{F}_{\mathbf{n}}^{-}(\mathbf{U}_e) \rrbracket = \frac{1}{2} \left[ \llbracket E_3 \rrbracket + \sqrt{\frac{\mu}{\varepsilon}} (n_1 \llbracket H_2 \rrbracket - n_2 \llbracket H_1 \rrbracket) \right] \begin{Bmatrix} n_2 \\ -n_1 \\ -\sqrt{\frac{\varepsilon}{\mu}} \end{Bmatrix} \quad (13)$$

for the TM<sup>z</sup> mode. The physical domain  $\Omega_{ph}$  may be both a closed domain or an infinite domain. To reproduce the infinite domain behavior on the boundary of  $\Omega$  an appropriate boundary condition that approximates the Silver–Muller radiation conditions is imposed. We consider a perfectly matched layer (PML) in combination with an absorbing boundary condition (ABC) to absorb outgoing waves. We use a first-order ABC that impose null incoming flux at the outer boundary, namely,

$$\llbracket \mathbf{F}_{\mathbf{n}}^{-}(\mathbf{U}_e) \rrbracket = -\mathbf{F}_{\mathbf{n}}^{-}(\mathbf{U}). \quad (14)$$

We recommend the reader to consult References 53–57 for further details on artificial boundary conditions for truncated infinite domains.

After the spatial discretization is performed, the resulting system of ordinary differential equations can be written as:

$$\mathbf{M} \frac{d\mathbf{U}^h}{dt} + \mathbf{R}(\mathbf{U}^h) = \mathbf{0}, \quad (15)$$

where  $\mathbf{M}$  is the mass matrix,  $\mathbf{U}^h$  is the vector which contains the nodal values of the conservative variables, and  $\mathbf{R}(\mathbf{U}^h)$  represents the vector of residuals.

Equation (15) is integrated in time using an explicit Runge–Kutta-type method. The use of high-order explicit integrators of this type decouples the system of Equation (15). The splitting of the global problem into a set of smaller local ones has important advantages from the computational point of view, such as a great reduction in the size of the systems of equations to be solved, leading to faster calculations and memory saving, and the possibility of applying parallel processing. However, for the algorithm to behave in a stable way, it is necessary to set a time step for the integration that complies with the CFL stability condition, which, in the particular case of the classic Runge–Kutta, is formulated:

$$\Delta t \leq C \frac{h}{c p^2} \quad (16)$$

being  $C$  a positive constant,  $h$  a length that quantifies the minimum size of the spatial discretization,  $c$  the speed of light in the medium, and  $p$  the degree of the polynomial approximation.

### 3 | CARTESIAN GRID FINITE ELEMENT METHOD

As mentioned in Section 1, cgFEM is a FDM characterized by the use of meshes exclusively made of regular quadrilateral elements in 2D or regular hexahedral elements in 3D. In the following, we present the key ideas of this discretization strategy. Even though they are also valid for 3D problems, in this exposition we focus on the 2D case, for the sake of simplicity.

The process to generate the analysis mesh starts defining the fictitious domain. In cgFEM, the fictitious domain is a squared domain that contains the physical domain, namely, the bounding box  $\mathcal{B}$ . Next, a set of grids containing all possible elements that may appear in the analysis mesh is computed. The grid pile is constructed following an iterative

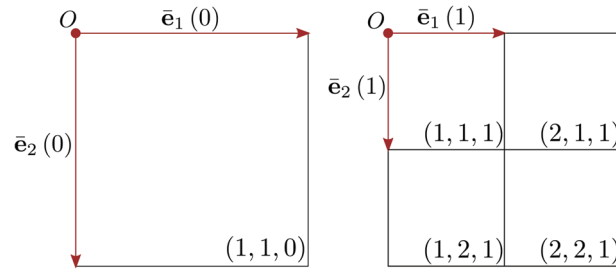


FIGURE 1 Virtual coordinates associated with the level 0 (left) and level 1 (right) elements

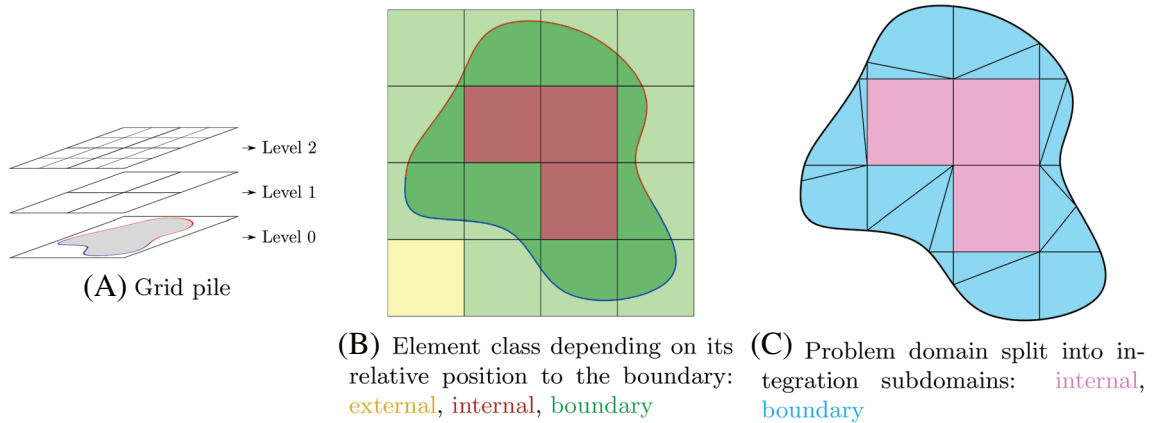


FIGURE 2 Stages in analysis mesh generation

procedure in which the current grid is computed by halving the elements of the previous grid in all space dimensions. The starting point is a grid that contains just an element of the same size as  $\mathcal{B}$ , which we refer to as the reference element. The procedure last for a number of iterations specified by the user which defines the minimum element size. Note that, if  $d$  is the number of spatial dimensions of the problem, the grid obtained after iteration  $l$ , namely, the level- $l$  grid, contains  $2^{dl}$  identical elements which are similar to the reference element.

Within this stage, each element may be identified by the virtual coordinates of its bottom-right vertex. The virtual coordinates system is a  $\mathbb{R}^{d+1}$  coordinate system with its origin on the upper-left corner of the bounding box and whose basis, in relation to the Cartesian basis  $\{\mathbf{e}_1, \mathbf{e}_2\}$ , are  $\{\bar{\mathbf{e}}_1(l), \bar{\mathbf{e}}_2(l), l\} = \{h(l)^{-1}\mathbf{e}_1, -h(l)^{-1}\mathbf{e}_2, l\}$ , where  $h(l)$  represents the size of the element for a given level  $l$ . In Figure 1, we present level-0 and level-1 grids and the virtual coordinates associated to its elements.

The next step is to find the intersections of the geometry with the grid pile. Depending on the location of the elements with respect to the boundary of the physical domain, these are classified into external, internal and boundary elements. Considering the computational domain  $\Omega$ , the element  $e$  with domain  $K_e$  is classified as:

- external element if  $K_e \cap \Omega = \emptyset$ .
- internal element if  $K_e \cap \Omega = K_e$
- boundary element if  $\{K_e \cap \Omega\} \supset K_e$

In Figure 2A,B, we show the grid pile up to level 2 and the classification of the elements in the top-level grid computed for an arbitrary domain.

The analysis grid is built up by selecting internal and boundary elements from the grid pile in such a way that there is no overlapping among elements and the physical domain is completely covered. This can be easily verified by using the virtual coordinates to explore the relations among elements in the grid pile.

To obtain the high-order nodal distribution of an element, the nodal distribution of the reference element is mapped to the physical elements. Notice that, by construction, this mapping is affine. Therefore, the Jacobian matrix of the transformation is constant and there is no loss of consistency in the interpolation, that is, a polynomial interpolation of

degree  $p$  in local coordinates of the reference element preserves its degree when transferred to Cartesian coordinates. As exposed later, these features provide great advantages for numerical integration.

Taking as starting point the analysis mesh, we need to define a set of subdomains that fits the geometry, see Figure 2C. These subdomains are employed to design the numerical quadratures used to evaluate the terms of the weak form of the problem. This way, the actual computational domain  $\Omega$  is consider, instead of  $\mathcal{B}$ . The procedure used to obtain the integration subdomains elementwise depends on the element type. Internal elements are treated as standard elements and, therefore, the integration domain matches with the element domain. In the case of boundary elements, the portion of the physical domain within the element splits into simpler subdomains (triangles or quadrilaterals) and special integration quadratures are defined over each of them. The integration over boundary elements must provide a sufficient level of accuracy in order to keep integration error below the discretization error. In the context of cgFEM, the two different strategies illustrated in Figure 3 are considered:

1. Polynomial approach: A polynomial curve is fitted to a set of sampled points (red squares in Figure 3) over each portion of the boundary delimited by intersection points with the element edges or vertices of the geometry (green squares in Figure 3). The integration subdomains are then computed considering this polynomial approximation of the exact geometry.
2. NEFEM approach: The parametric definition of the boundary provided by the CAD is taken into account. This approach allows to remove any geometric error induced by a polynomial approximation of the boundary.

In this contribution we only consider the NEFEM approach to ensure that nonphysical artifacts due to a poor representation of the domain do not appear. We follow this strategy to evaluate integrals both over the computational domain and over its boundary.

Concerning the imposition of boundary conditions in cgFEM, the fact that the boundary is not conforming with the mesh means that there are some differences in comparison with the standard finite element method. Since the mesh nodes are not placed on the boundary, generally speaking, the essential boundary conditions cannot be strongly imposed. Alternatively, they are imposed in weak form. In the case of Neumann boundary conditions, the integral along internal boundaries gives rise to terms in the RHS of the system of equations at all degrees of freedom (DOFs) of the element. Unlike in the standard finite element methods, where these terms only appear on the DOFs tied to nodes located on the contour.

Processing Dirichlet boundary conditions is significantly more complex. There is a range of techniques that allow the imposition of Dirichlet conditions in an weak form, among which we can mention the penalty methods, the vital vertex method, the mortar method or the Nitsche’s method. In the cgFEM we use a strategy developed in our research group and which has been validated for the elastic problem. The condition on the primal variable is imposed through a Lagrange multiplier field evaluated at the integration points of the curve defining the Dirichlet boundary of the physical domain. This field leads to an overconstrains system of equations, making it necessary to introduce a stabilization term in the formulation that ensures compliance with the inf-sup condition. To restrict the range of values that these multipliers can take and thus ensure optimal convergence, a stabilization term is added to the functional that penalizes the difference between the Lagrange multiplier field and a smoothed field on the dual variable obtained by postprocessing of the solution

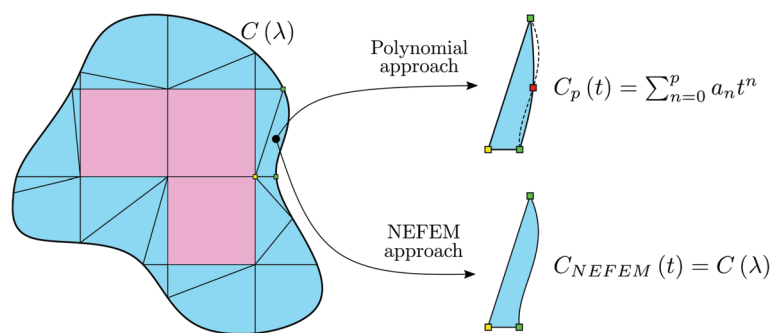


FIGURE 3 Example of different resulting integration subdomain depending on the boundary approximation: polynomial (top) and NEFEM (bottom)

of the problem. This stabilization strategy results in an iterative resolution process that ensures optimal convergence to the exact solution to the problem through refinement. The interested reader is addressed to Reference 29.

#### 4 | DISCONTINUOUS GALERKIN FORMULATION IN THE FRAMEWORK OF CGFEM

In the framework of the FDM, the calculated solution only makes physical sense when evaluated on the problem domain. Therefore, the approximate field  $\mathbf{U}^h(\mathbf{x}, t)$  is mathematically represented by piecewise defined function:

$$\mathbf{U}^h(\mathbf{x}, t) = \begin{cases} N_i(\mathbf{x}) \mathbf{U}_i^h(t) & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \in \Omega_e \\ \mathbf{0} & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \notin \Omega_e \end{cases}, \quad (17)$$

where  $\mathbf{U}_i^h(t)$  is the vector that contains the values of the conservative variables evaluated at the node  $i$  of the element  $e$  at the instant  $t$  and  $N_i(\mathbf{x})$  represents the shape functions of degree  $p$  associated with the node  $i$  evaluated at the global coordinate  $\mathbf{x}$ . This approximate solution is contained in a polynomial space up to degree  $p$  with support in  $\Omega_e, P_p(\Omega_e)$ , as stated in Section 2. A Fekete-nodal distribution<sup>58,59</sup> is considered to maximize the accuracy of the resulting approximation.

Regarding the construction of the weighting functions, a similar formulation is used:

$$W^h(\mathbf{x}) = \begin{cases} N_i(\mathbf{x}) W_i^h & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \in \Omega_e \\ 0 & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \notin \Omega_e \end{cases}. \quad (18)$$

In the following, we consider that flux operators  $\mathbf{F}_k$  and  $\mathbf{F}_n^-$  and the source operator  $\mathbf{S}$  are linear, as usual for the electromagnetic problem. Likewise, the analytical expression of the source term will be substituted by a polynomial approximation obtained by interpolating the values of the source evaluated at the nodes. Plugging (17) and (18) into (10) and taking into account the aforementioned assumptions we obtain:

$$W_i^h M_{ij} \frac{\partial \mathbf{U}_j^h}{\partial t} + W_i^h C_{ijk} \mathbf{F}_k(\mathbf{U}_j^h) + W_i^h \Phi_i = W_i^h M_{ij} \mathbf{S}(\mathbf{U}_j^h), \quad (19)$$

where  $\mathbf{M}$  is the mass matrix,  $\mathbf{C}_k$  is the convection matrix in the  $k$ th direction, and  $\Phi$  represents the electromagnetic flux through the boundary of the elements. These terms are computed as:

$$M_{ij} = \int_{\Omega_e} N_i N_j d\Omega, \quad (20)$$

$$C_{ijk} = \int_{\Omega_e} \partial_k N_i N_j d\Omega, \quad (21)$$

$$\Phi_i = \int_{\partial\Omega_e} N_i \mathbf{F}_n^-(\mathbf{U}_{out}) d\Gamma - \int_{\partial\Omega_e} N_i N_j \mathbf{F}_n^-(\mathbf{U}_j^h) d\Gamma. \quad (22)$$

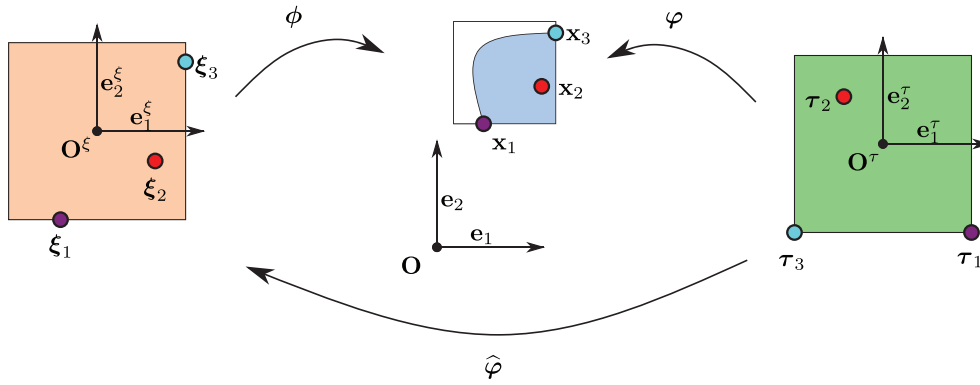
In general, coordinate transformations are used to map the domain of the integral into a space in which that domain correspond to a reference one for which an integration quadrature is available. We will be dealing with three different spaces, namely: the physical space  $\Omega_x$ , the local space  $\Omega_\xi$ , and the integration space  $\Omega_\tau$ . The coordinates of a point in each one of these will be denoted by  $\mathbf{x}$ ,  $\xi$ , and  $\tau$ , respectively. The mapping function between those spaces is:

$$\mathbf{x} = \boldsymbol{\phi}(\xi), \quad \mathbf{x} = \boldsymbol{\varphi}(\tau). \quad (23)$$

Note that  $\boldsymbol{\phi}(\xi) = N_i(\xi) \mathbf{x}_i$  which, because of the Cartesian structure of the mesh, is an affine transformation that formulates

$$\boldsymbol{\phi}(\xi) = \frac{1}{2} h_e \xi + \mathbf{x}_e, \quad (24)$$





**FIGURE 4** Diagram showing the local, physical, and integration spaces (from left to right) and their mappings. Circles of the same color represent the same point in each space

where  $h_e$  and  $\mathbf{x}_e$  represent the size and the center of the element  $e$ , respectively. That property let us obtain the projection of a point of the integration space on the local space as:

$$\xi = (\phi^{-1} \circ \varphi)(\tau) = \hat{\varphi}(\tau). \quad (25)$$

A graphical representation of these spaces and their relations is shown in Figure 4.

*Remark 1.* For the sake of simplicity, we expressed  $N_i(\mathbf{x})$  in terms of the global coordinate  $\mathbf{x}$ . However, these functions are defined using local coordinates  $\xi$  to improve accuracy, namely,  $N_i(\xi)$ . Thus,  $N_i(\mathbf{x})$  computes in two stages: mapping from global to local coordinates using  $\phi(\xi)$  and evaluating the function  $N_i(\xi)$ . This license does not affect the validity of the exposition.

Let us begin by studying the integrals over the interior of the domain. We focus on the case of the mass matrix. Applying the coordinate transformations defined above:

$$M_{ij} = \sum_{n=1}^{n_{is}} \int_{\Pi} N_i(\hat{\varphi}^n(\tau)) N_j(\hat{\varphi}^n(\tau)) |\mathbf{J}^{\varphi^n}(\tau)| d\Pi \quad (26)$$

where  $n_{is}$  is the number of integration subdomains within the element and  $\mathbf{J}^{\varphi^n}$  represents the Jacobian of the transformation  $\varphi^n$  associated with the integration subdomain  $n$ .

Internal elements are treated as standard finite elements and, therefore, they contain a single integration subdomain that matches the element domain. In the following, the superscripts  $n$  are removed for simplicity. Besides  $\phi$ , the mapping  $\varphi$  for internal elements is also affine since both the element domain and the reference domain for integration are hypercubes. As a consequence, the Jacobian determinant of the transformation is constant and can be taken out of the integrand. In the same way, given the similarity between the elements of the mesh, it is possible to relate their Jacobian taking into account the mesh level to which they belong according to:

$$\mathbf{J}^{\varphi_l} = 2^{L-l} \mathbf{J}^{\varphi_0}. \quad (27)$$

For convenience, we use the 0-level element to compute the reference Jacobian and we obtain:

$$M_{ij} = 2^{-dl} |\mathbf{J}^{\varphi_0}| \int_{\Pi} N_i(\hat{\varphi}(\tau)) N_j(\hat{\varphi}(\tau)) d\Pi. \quad (28)$$

If we study the integrand of the expression, we observe that it only depends on the interpolation functions and the mapping from the integration space to the local space. On the one hand, the interpolation functions are completely defined from a nodal distribution which is specific of the degree of approximation  $p$  used. On the other hand, as the mapping  $\hat{\varphi}$  is a

combination of two affine transformations affected by the Jacobian in the same way,  $\hat{\boldsymbol{\varphi}}$  is an affine transformation common to all the internal elements. Therefore, we can rewrite the expression as:

$$M_{ij} = 2^{-dl} M_{ij}^{0,p}, \quad (29)$$

where

$$M_{ij}^{0,p} = |\mathbf{J}^{\boldsymbol{\varphi}_0}| \int_{\Pi} N_i^p(\hat{\boldsymbol{\varphi}}(\boldsymbol{\tau})) N_j^p(\hat{\boldsymbol{\varphi}}(\boldsymbol{\tau})) d\Pi \quad (30)$$

represents the mass matrix of the 0-level element of degree  $p$  and  $d$  is the number of the spatial dimensions.

Proceeding in a similar way with the convection matrix we arrive at the expression:

$$C_{ijk} = 2^{1-dl} C_{ijk}^{0,p}, \quad (31)$$

where

$$C_{ijk}^{0,p} = |\mathbf{J}^{\boldsymbol{\varphi}_0}| (\mathbf{J}^{\boldsymbol{\varphi}_0})_{kl}^{-1} \int_{\Pi} \partial_l N_i^p(\hat{\boldsymbol{\varphi}}(\boldsymbol{\tau})) N_j^p(\hat{\boldsymbol{\varphi}}(\boldsymbol{\tau})) d\Pi \quad (32)$$

represents the convection matrix of the 0-level element of degree  $p$  and  $\partial_l$  symbolizes the derivative with respect to local coordinates.

With this formulation we compute the mass and convection matrices of any of the  $n_{ie}$  internal elements of the mesh by scaling the reference matrices. This allows us to save a large amount of memory space, since we move from having to save  $2n_{ie}$  matrices to  $2n_{ip}$  matrices, where  $n_{ip}$  stands from the number of different degrees of approximations that have been used for the spatial discretization.

In the case of boundary elements, the domain  $\Omega_e$  does not match the element domain  $K_e$  and the integral is computed as a sum of integrals over simpler subdomains which are obtained by splitting  $\Omega_e$ . Subsequently, one of the techniques mentioned in Section 3 is applied to obtain the transformation functions  $\boldsymbol{\varphi}$  and  $\hat{\boldsymbol{\varphi}}$ . It is not possible to do any simplification since the number of terms of the summation in (26) will depend on the geometry and approximation of  $\Omega_e$  and the integrand of each of these terms will depend on the transformations  $\boldsymbol{\varphi}$  and  $\hat{\boldsymbol{\varphi}}$ , whose definitions comes from the geometry and position in the element of each integration subdomain. Therefore, the domain integrals of each of the intersected elements are treated independently according to the general formulation. It is worth to remark that, even though the computational burden associated to these integrals is higher than that associated to internal elements, the number of intersected elements is small with respect to the total amount of elements in the mesh.

Regarding the term  $\boldsymbol{\Phi}_i$ , we differentiate two groups depending on the characteristics of the integrals:

- *Integrals of interfaces between elements:* the external and internal fields are defined by the solution of the finite elements analysis in the neighbor element and in the current element, respectively. Likewise, the values are computed by interpolating the solutions of those nodes located on the boundary under study, since the influence of the rest of the nodes is null. Due to the Cartesian structure of the mesh and its independence from geometry, all element faces possess a constant normal, which is aligned with one of the Cartesian directions. As a consequence, the incoming flow operator is identical for all point on the face and, since it is linear with respect to the conservative field, it can be taken from the integrand in (22). Furthermore, this operator can only respond to  $2d$  expressions, which correspond to the Cartesian directions. For the 2D case:  $\left\{ \mathbf{F}_{\bar{\mathbf{n}}}^- \text{ with } \bar{\mathbf{n}} \in \{(1, 0), (0, 1), (-1, 0), (0, -1)\} \right\}$ . With this in mind, we can express (22) as:

$$\boldsymbol{\Phi}_i = m_{ik} \mathbf{F}_{\bar{\mathbf{n}}}^- (\mathbf{U}_k^h) - m_{ij} \mathbf{F}_{\bar{\mathbf{n}}}^- (\mathbf{U}_j^h), \quad (33)$$

where:

$$m_{ij} = \int_{\partial\Omega_e} N_i N_j d\Gamma \quad (34)$$

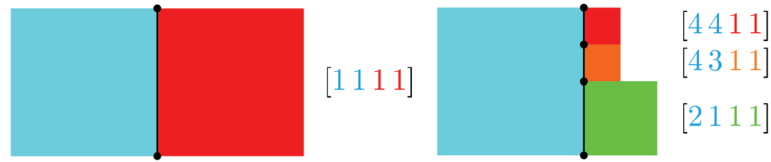


FIGURE 5 Topology of two different internal faces and code for processing

is a mass matrix of dimension  $d - 1$  and the subindices  $j$  and  $k$  identify the rows in  $\mathbf{U}^h$  that correspond to the nodes of the element and its neighbor, respectively. In turn, this set of integrals can be subdivided into:

- *Integrals over complete edges:*  $m_{ij}$  corresponds to the mass matrix of a Cartesian element of one lower dimension than the dimension of the problem. Following a similar development to that presented for the volume integrals, for a given element of degree  $p$ , we can express  $m_{ij}$  as a rescaling of a mass matrix of degree  $d - 1$  calculated for the 0-level element considering weighting functions of degree  $p$  and test functions of the degree  $n$ , where  $n$  is the degree of the polynomial field being evaluated (internal or external):

$$m_{ij} = 2^{(1-d)l} m_{ij}^{0,pn}, \quad (35)$$

$$m_{ij}^{0,pn} = |\mathbf{J}^{\mathcal{P}_0}| \int_{\partial\Omega} N_i^p(\hat{\boldsymbol{\phi}}(\boldsymbol{\tau})) N_j^n(\hat{\boldsymbol{\phi}}(\boldsymbol{\tau})) d\pi. \quad (36)$$

This way of obtaining boundary integrals on complete edges reduces the amount of information that needs to be stored in memory. However, it must be considered that, for  $h$ -refinement nonuniform meshes, an element may be connected through its boundary with more than one neighboring elements or that the integral on the face of the neighboring element does not cover its entire length but only part of it. This situation can be seen in Figure 5, where two different examples of interfaces between elements and their coding are represented. The encoding we use is composed of four numbers grouped in two pairs of two numbers. The first number of each pair represent is the ratio between the element side dimension and the size of the interface being integrated, while the second number represents the position of the interface on the element edge. It is possible to identify all the typologies of integrals over nonintersected element faces and calculate their respective reference matrices for the 0-level element. Something similar happens in the case of using  $p$ -refining strategies, where a mass matrix  $m_{ij}^{0,pn}$  will be obtained for each combination of degrees in the mesh.

- *Integrals over intersected edges:* When the element edge intersects the boundary of the physical domain, the integrand is null over the outer section and the integral restricts to  $\partial K_e \cap \Omega$ . This domain is different for each of the flux integrals over intersected faces and, therefore, their mass matrices are calculated individually according to:

$$\int_{\partial K_e \cap \Omega} N_i N_j d\Gamma. \quad (37)$$

- *Integrals over the boundary:* The calculation of the incoming numerical flux will be subjected to the boundary conditions imposed on the contour being integrated while the conservative variable distribution due to the numerical solution in the element will be a function of the calculated values for all its nodes, not only those on the element's boundary. This last characteristic is a feature of FDM methods and is a consequence of the physical domain boundary being internal to the mesh. Integrals over boundaries will be calculated following the general expression (22).

As a previous step to integral definition, we need to identify what elements meet at each of the interfaces in a process known as neighbor search. In the context of cgFEM, we can use the virtual numbering presented in Section 3 to find out the connectivity between elements. Using the virtual coordinates this process becomes trivial. Furthermore, we can also identify the set of different combination of degrees  $p$ , level jumps  $\Delta l$  and relative positions of neighbor elements in the mesh, avoiding unnecessary computations.

Intersected elements may contain disjointed regions where the solution is physically independent. If we use the standard discretization of cgFEM, the polynomial approximation would be shared by all the regions contained within the element. Trying to represent with a single smooth function a set of independent fields leads to a significant increase in the local error associated to these elements and the pollution of the global solution. This problem becomes more relevant

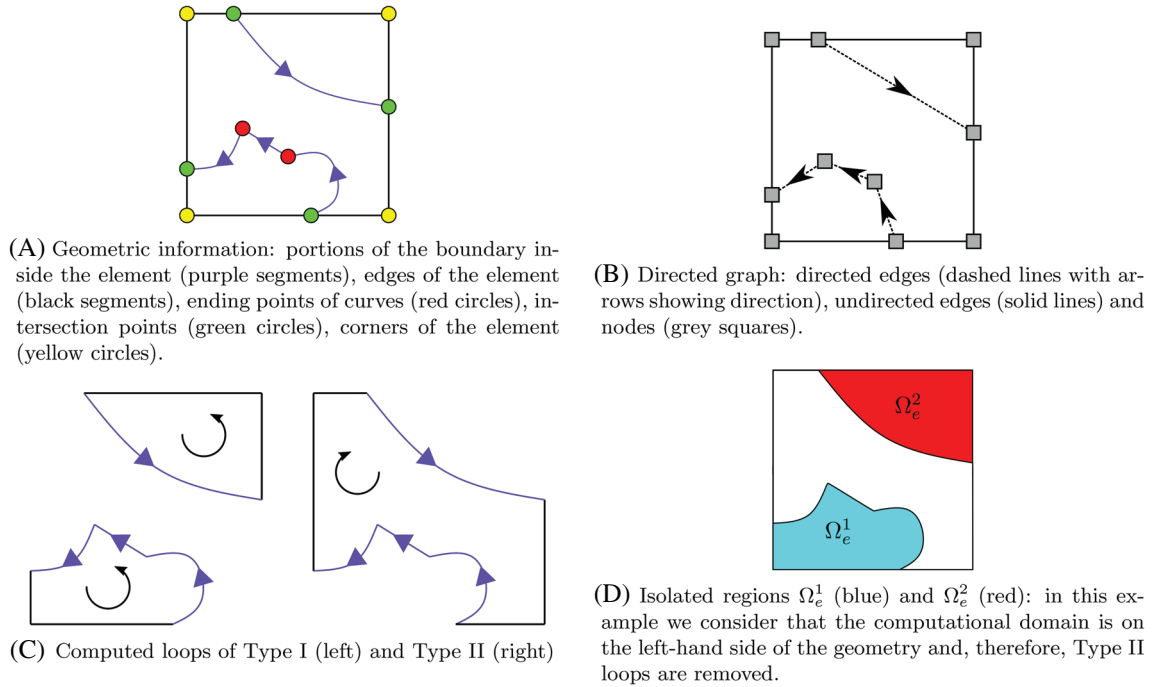


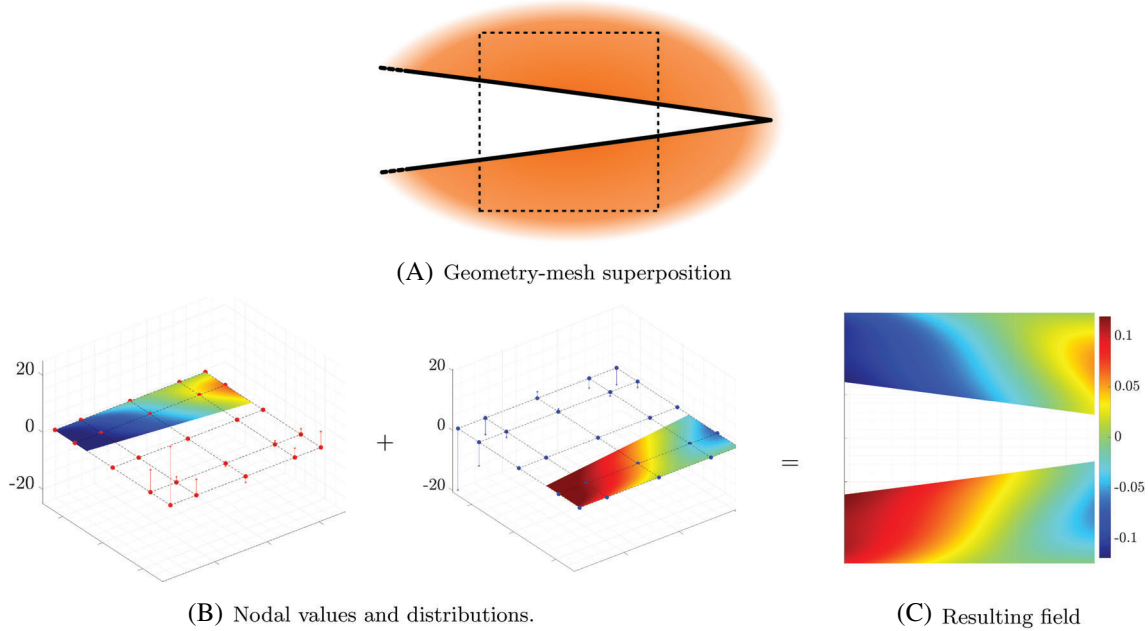
FIGURE 6 Algorithm employed to detect isolated regions within an element

as the size of the elements increases and the details of the geometry become smaller than the characteristic length of the mesh. This situation is quite usual in high-order methods.

In the following, we describe the algorithm that we employ to detect isolated regions. The reader can find an example of application in Figure 6. If we focus on a single element on the boundary, the output of the intersection routine is the set of curves internal to the element and the segments of the curves limited by intersection points and endings of the curves such that any point on the segment is internal to the element. Next, we split the contour of the element into a set of segments taking into account the intersection points and the corners (see Figure 6A). Then we obtain the connectivity of these segments and curves (hereafter the edges) through the intersection points and the corners (hereafter the vertices). The edges on the geometry are directed paths according to the definition of the geometry (see Figure 6B). To find an independent region in the element, we identify the tail of a directed edge, move to the successor, and go through the graph always taking the first edge from left to right at the vertices until we arrive to the starting vertex. This procedure is executed taking different starting points until all directed edges have been traversed once. The loops computed this way are classified as Type I. Then, it repeats changing the turning criterion to always take the first path from right to left. The new loops are classified as Type II (see Figure 6C). The region at both sides of the geometry is completely defined and, therefore, the turning criterion at the vertices characterizes the physical properties of the domain inside each loop. By the end of this algorithm, we have obtained all independent regions within the element and their physical properties. Once those regions outside the physical domain are removed, the remaining regions are necessarily isolated regions (see Figure 6D).

In this work, the spatial discretization is modified in those elements that contain two or more regions of the physical domain disjointed to each other (from now on referred to as isolated regions). Let us consider the element  $e$  located on the boundary of the physical domain  $\Omega$  in such a way that their intersection leads to a set of  $n_{\text{ir}}$  isolated regions in it, namely,  $\Omega_e = \cup_{r=1}^{n_{\text{ir}}} \Omega_e^r$ . See Figure 7B. The set of nodes  $\mathfrak{N}_e$  associate with the element  $e$  is split into  $n_{\text{ir}}$  subsets  $\{\mathfrak{N}_e^1, \mathfrak{N}_e^2, \dots, \mathfrak{N}_e^{n_{\text{ir}}}\}$  and the shape functions  $N_i$  used to interpolate the nodal values are redefined considering each subset  $\mathfrak{N}_e^r$  separately. The approximate field  $\mathbf{U}^h(\mathbf{x}, t)$  and the weighting functions  $W^h(\mathbf{x})$  in a boundary element with multiple isolated regions using standard summation notation formulate as:

$$\mathbf{U}^h(\mathbf{x}, t) = \begin{cases} \sum_{i \in \mathfrak{N}_e^r} N_i(\mathbf{x}) \mathbf{U}_i^h(t) & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \in \Omega_e^r \\ \mathbf{0} & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \notin \Omega_e \end{cases} \quad (38)$$



**FIGURE 7** Example of two independent fields within a single element. The approximate solution on the top region  $\Omega_e^1$  correspond to the interpolation of the nodal values at nodes drawn in red  $\mathfrak{N}_e^1$ , while the nodal values at blue nodes  $\mathfrak{N}_e^2$  define the approximate solution on the bottom region  $\Omega_e^2$ . The sets of red and blue nodes are linked to the same element, namely,  $\mathfrak{N}_e = \{\mathfrak{N}_e^1, \mathfrak{N}_e^2\}$ , but the support of the polynomial interpolations differ from one to the other. Thus, the resulting approximate solution on each isolated region is independent

and

$$W^h(\mathbf{x}) = \begin{cases} \sum_{i \in \mathfrak{N}_e^r} N_i(\mathbf{x}) W^h & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \in \Omega_e^r \\ \mathbf{0} & \text{if } \mathbf{x} \in K_e \text{ and } \mathbf{x} \notin \Omega_e^r \end{cases}, \quad (39)$$

respectively. The final discretization is completely independent from one region to another so this strategy allows the use of different polynomial degrees for each of them. Indeed, we employ an independent Fekete nodal distribution mapped on  $K_e$  for each subset  $\mathfrak{N}_e^r$ . An example of how the dissociation strategy works is shown in Figure 7.

In Reference 60 a similar dissociation approach was introduced to handle discontinuities internal to the elements in the context of 2D elasticity problems. That same approach was later on applied to other problems, see Reference 61 and the references therein. Our implementation surpasses existing approaches in the literature in the sense that the number of isolated region within an element is not restricted, the regions are bounded by the exact representation of the geometry provided by the CAD software, the FE approximation is defined at region-level and high-order polynomial basis are considered. Moreover, the DG solver adopted in this contribution let us weakly imposed the continuity of the solution across those interfaces that lie on the computational domain without the need of adding any additional term to the original formulation.

## 5 | STABILIZED SCHEME

As previously mentioned in Section 1, the computational advantage exhibited by the DG formulation is closely related to the use of explicit integrators to advance in time. The stabilization strategy proposed in this section seeks to minimize the negative effect that the presence of cut elements in the mesh has on the maximum stable time step of the explicit time integrator.

We want to identify and address the stabilization of unstable DOFs in the very first stages of the workflow in order to avoid redundant calculations. To do this, we need to select a local parameter that let us predict which intersected

elements will induce a time step substantially smaller than the one associated to noncut elements. The global stability of an explicit integrator is quantified by its CFL condition which depends on the polynomial degree and support of the most unfavorable element. Since the degree of approximation is not affected by the mesh–geometry interaction, a parameter that is related to the size of the support domain will be sought. In particular, we use the relative area  $\alpha_e^r$  as indicator. Following the notation presented in the previous section, this parameter represents the area ratio of the region  $r$  with respect to the extension of the whole element  $e$ , namely,

$$\alpha_e^i = \frac{|\Omega_e^i|}{|K_e|}, \quad (40)$$

where  $|\cdot|$  is the operator that provides the area of a given domain.

In Section 6.1, we show the convenience of this coefficient to predict unstable behaviors. The study of how its reduction affects the maximum step size shows a monotonic relationship between these magnitudes which justifies its use. Furthermore, it is a purely geometric parameter that is easy to evaluate in the initial stages of the algorithm. Finally, a set of DOFs will be considered unstable if the relative area of its support region is less than a certain value specified by the user  $\alpha_{th}$ .

*Remark 2.* Instability evaluation is performed at the level of isolated regions. For the sake of facilitating the comprehension of the algorithm, in the following we assume that each element contains one isolated region at most,  $n_{ir} = 1$ . Therefore, the DOFs linked to a region always correspond to those associated with the element that encompasses it, namely,  $\mathfrak{N}_e = \mathfrak{N}_e^1$ . With that in mind, we simplify the notation using  $\alpha_e$  and  $\Omega_e$  instead of  $\alpha_e^1$  and  $\Omega_e^1$ , respectively

The stabilization strategy we propose is based on modifying the support domains of the interpolation functions. Once the unstable region has been identified, its domain will be reassigned to the polynomial approximation defined on a neighboring element considered as stable. The resulting element is called extended element as, when evaluating the operators in weak form, this means that the integrals extend beyond the boundary of the element. The shape, nodal distribution, and shape functions of the extended elements remain the same before and after the reassignment procedure. Therefore, as far as implementation is concerned, they can be treated as standard intersected elements and do not entail any additional computational burden. Since the unstable regions are small with respect to the size of the extended element, problems related to polynomial instability due to extrapolation have not been detected. Once the reassignment has been completed, the DOFs initially associated with the unstable region have zero support and can be eliminated from the analysis mesh.

*Remark 3.* An alternative strategy to define the approximate solution on extended elements was also investigated. In this alternative approach, the reference element was mapped to a rectangle that contained all the regions associated with each element at hand. As the support of the shape functions is completely within the boundary of the nodal distribution, no extrapolation is applied. In comparison with the approach adopted in this contribution, this alternative did not provide any major advantage in terms of the overall stability of the method while the departure from the cgFEM data structure of the mesh reduced the efficiency of the resulting scheme.

The value of the maximum stable time step that may be expected from the resulting discretization will be greater than or equal to that associated with an element with a relative area of  $\alpha_{th}$ . In this way, we limit the maximum stable time step and eliminate its dependence on the cut elements.

Prior to go through the steps of the reassignment algorithm we need to define some further notation. Let us consider two adjacent elements  $i$  and  $j$  that intersect the physical domain  $\Omega$  leading to  $\Omega_i = K_i \cap \Omega$  and  $\Omega_j = K_j \cap \Omega$  where  $K_i$  and  $K_j$  are the domains of the whole elements. We represent as  $\alpha_{ij}$  the relative area of the combined domain  $\Omega_i \cup \Omega_j$  with respect to  $K_j$ . This corresponds to the extension of the element  $j$  over  $\Omega_i$ . Note that the subscripts identify the physical subdomains to combine and the element we compare to is that of the last index. Besides the combined relative area, we need to compute the centroids of the domains. For the domains  $\Omega_i$  and  $\Omega_j$ , these positions are represented as  $\bar{\mathbf{x}}_i$  and  $\bar{\mathbf{x}}_j$ , respectively. Finally, we identify as  $d_{ij}$  the distance between the centroids of the domains  $\Omega_i$  and  $\Omega_j$ , which formulates as  $d_{ij} = \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|$ .

The following briefly describes the algorithm used to reassign unstable domains. First, we identify the regions that limit the stability of the marching algorithm by computing the relative areas of all isolated regions within the boundary elements of the mesh and comparing them to  $\alpha_{th}$ . Next, the mesh is checked to verify that all selected regions belong to elements that are not greater than their neighbors. The elements that do not meet this condition are refined and the

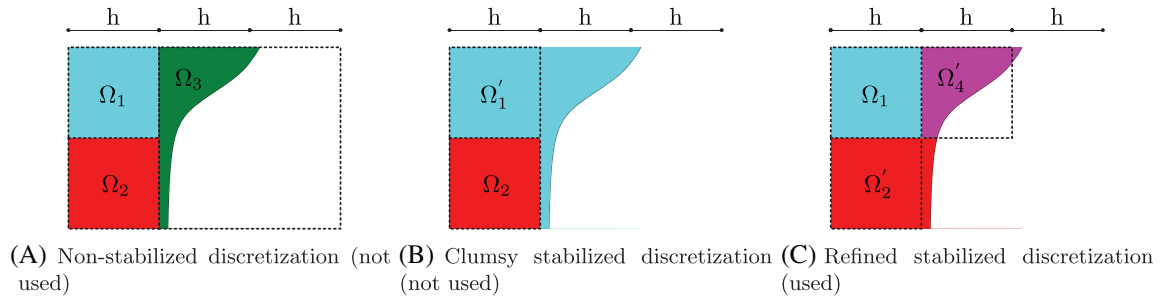


FIGURE 8 Comparison of different spatial discretizations

resulting elements are studied to identify new sources of instability. By the end of this procedure, we identify a total of  $n_{ue}$  unstable subdomains.

*Remark 4.* The refinement criterion presented in this section prevents from extending over regions even greater than the size of the element, which may lead to polynomial instabilities associated to extrapolation. Alternatively, mesh refinement may be understood as a way of splitting the original unstable regions into smaller ones. With this in mind, the strategy let us perform a better reassignment of the unstable region, linking each subregion to the most suitable element independently. We can even find children elements that are inherently stable without modifying the time integration scheme. In Figure 8, we consider an initial unstable discretization (Figure 8A) and compare the outputs of the reassignment algorithm when it is directly applied (Figure 8B) and when the mesh is refined in advance (Figure 8C). The figures show how this last approach, the one adopted in this contribution, leads to a more effective rearrangement of integration subdomains and to minor extrapolations. It is worth noting that the refinement criterion does not affect the stability of the base mesh since the stable time step is governed by the finest discretization.

Once the final mesh is obtained, we loop through the unstable regions in increasing order of relative area, namely,  $\alpha_e \leq \alpha_{e+1} \forall e \in \{1, 2, \dots, n_{ue}\}$ . Let us consider the unstable region  $\Omega_i$ . The first step is to identify the neighboring elements, which will be those that share an edge with the element  $i$ . This operation is trivial given the Cartesian structure of the mesh in cgFEM. Next, we explore the regions within the neighboring elements and discard those that are not adjacent to  $\Omega_i$ . We obtain the preliminary set of candidates  $\mathcal{N}_p$  that contains the elements that may be extended. In the following stage, the combined relative area resulting from each of the possible reallocations is computed, namely,  $\{\alpha_{ij} \forall j \in \mathcal{N}_p\}$ . The final set of candidates  $\mathcal{N}$  is computed removing from  $\mathcal{N}_p$  the elements whose combined relative area do not meet the condition  $\alpha_{ij} \geq \alpha_{th}$ . If  $\mathcal{N}$  is an empty set, the iteration ends and we move on to the next subdomain without performing any change on the discretization. If some elements remain, the centroid of all regions is evaluated  $\{\bar{\mathbf{x}}_k \forall k \in \{i, \mathcal{N}\}\}$  and the distance with respect to the centroid of the unstable region is computed  $\{d_{ij} \forall j \in \mathcal{N}\}$ . Finally, the unstable region is reassigned to the element that present the minimum distance,  $\min(\{d_{ij} \forall j \in \mathcal{N}\})$ . This procedure is depicted in Figure 9. If the final combination involves two initially unstable domains, both are removed from the set of unstable domains. If the element chosen for extension was a stable element, the region is marked as resolved. The process continues until there are no domains classified as unstable.

Note that, unstable neighbor elements are not removed from the first selection, but may remain as candidates. If, at the time of evaluating stability, the support resulting from merging the domains causes the initial discretization to become stable, it will make it likely to be extended. In this way, a smaller number of DOFs is eliminated and the size of the extensions is contained. By traversing the target domains in ascending order of relative area, we favor these types of assignments to occur.

When selecting the element to be extended, we may be interest in considering other characteristics beyond the proximity between domains. As an example, level of stability, orientation, or complexity of the assembled domain may be important features we must control. These aspects can be implicitly taken into account for the selection as a series of factors  $f_c$  that multiply the distance  $d_{ij}$  obtaining:

$$\bar{d}_{ij} = \prod_{k=1}^{n_f} f_k d_{ij}, \quad (41)$$

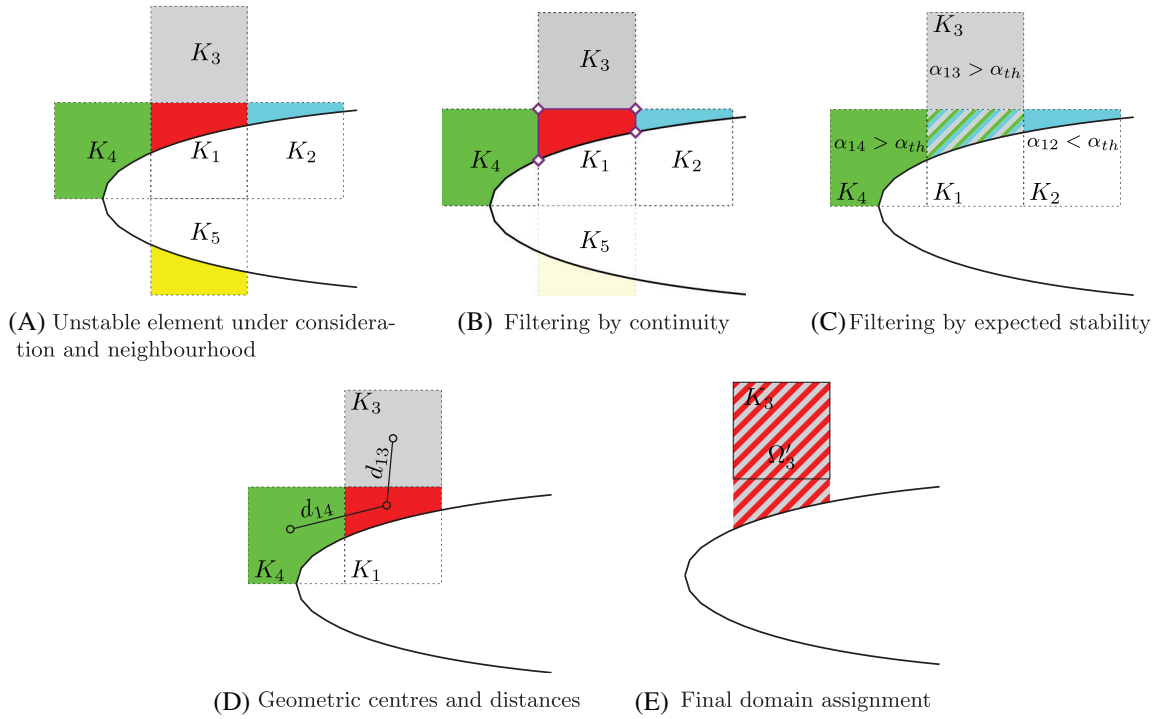


FIGURE 9 Procedure to reassign unstable regions

where  $n_f$  refers to the total number of modifiers taken into account.

In the current implementation we consider the following factors:

- $f_1$ : penalizes those elements with a relative area close to the threshold value. It gives an advantage to those elements with better characteristics from the stability point of view. This factor fits to the expression:

$$f_1 = 2(\tanh(A_1(\alpha_j - \alpha_{th})) + 1)^{-1} \quad (42)$$

being  $A_1$  the parameter that controls the range of values that are penalized. We recommend to set  $A_1 = 10$ .

- $f_2$ : penalizes those elements with an unfavorable distribution of the physical domain. It reduces the appearance of very slender integration domains. It is formulated as:

$$f_2 = \prod_{k=\{x,y\}} 2\left(\tanh\left(A_2\left(\frac{12}{h^4}I_{ij}^k - (\alpha_{th})^3\right)\right) + 1\right)^{-1} \quad (43)$$

where  $I_{ij}^k$  represents the moment of inertia of the merged region with respect to the  $k$ -axis that pass through its centroid and  $A_2$  is a number that limits the range of values that are penalized. We recommend to set  $A_2 = 10$ .

- $f_3$ : penalizes those elements that have already been extended. It indirectly reduces the possibility of integration domains with internal corners. It is computed as:

$$f_2 = \sum_{s=0}^{n_t} 2^{-s}, \quad (44)$$

where  $n_t$  represents number of times an element has been previously selected for extension.

Configurations may appear for which not all unstable regions are resolved by the end of the loop. This is the case shown in Figure 10A,B. When trying to reassign the domain encompassed in element  $K_1$ , the extension of the  $K_2$  and  $K_3$  elements is checked but in both cases, the resulting discretization is identified as unstable since  $\alpha_{1j} < \alpha_{th}$  with  $j = 2, 3$ . If we study elements  $K_2$  and  $K_3$  at the end of the loop, they will both have been removed from the mesh and their domains



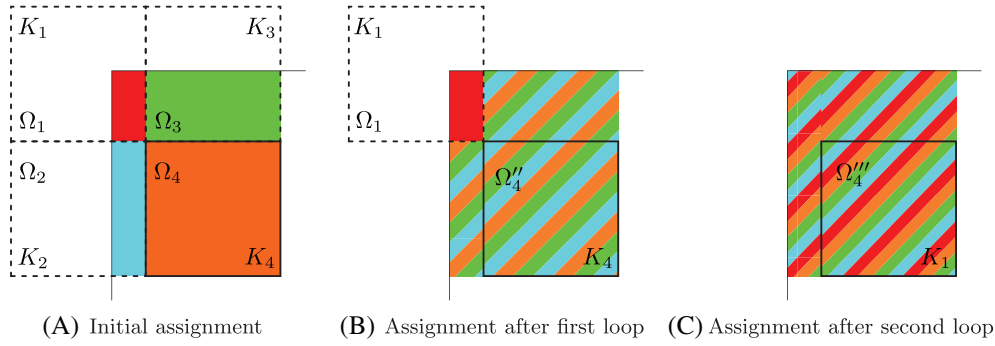


FIGURE 10 Example of two passes through loop to solve an unstable region

have been rearrange and now belongs to element  $K_4$ . If we start over the loop taking into account the reallocation, a new study for the physical domain in  $K_1$  leads to a valid reassignment. The domain could be reallocated to the element  $K_4$ , since it fulfills both the continuity conditions and the stability condition  $\alpha_{1234} \geq \alpha_{th}$ . Therefore, the domain reassignment process is performed iteratively until all target regions are solved.

The removal of nodes associated to unstable elements implies a reduction in the nodal density and, therefore, the richness of the spatial discretization decreases. To minimize the effect of stabilization on the discretization error, we propose to increase the degree of the polynomial approximation in the extension direction. We can easily do this thank to the special features of the cgDG. Inherited from cgFEM, the mesh is made up of quadrilaterals and we can easily modify the nodal distribution since it will be calculated as the tensor product of independent high-degree 1D nodal distributions. In addition, the element shape or size is independent of the interpolation, so changing the nodal distribution does not have any effect on the structure of the Cartesian mesh. Moreover, DG formulation let us locally enrich the spatial discretization regardless of the discretization used in the surrounding elements. The drawback of this technique is that increasing the degree reduces the stability of the discretization. Therefore, it will be necessary to achieve a balance between the quality of the spatial discretization and the size of the time step.

As previously mentioned, the stabilization strategy only affects the spatial discretization of the problem and is carried out in the preprocessing stage. Changes in the polynomial approximation have no effect on the DG formulation of the problem, so no additional terms need to be evaluated and the burden of computing the different operators is the same as for the nonstabilized scheme. Furthermore this strategy is transparent to the temporary integrator, so the methodology we propose can be used to solve any problem using DG formulation and with any explicit temporal integrator.

## 6 | NUMERICAL RESULTS

### 6.1 | Verification

In this section, a series of numerical tests that verify the proposed method and strategies are shown.

First, a convergence analysis will be performed. The optimal convergence rate of the original DG method for purely hyperbolic problems was proved by Peterson<sup>62</sup> to be equal to  $h^{p+1/2}$ , where  $h$  represents the characteristic length of the discretization and  $p$  is the degree of the polynomial approximation. Under some assumptions, the optimal convergence rate rises up to  $h^{p+1}$  as confirmed by LeSaint and Raviart<sup>63</sup> for Cartesian grids and by Richter<sup>64</sup> for some structured non-Cartesian grids. Considering the Cartesian structure of the mesh in cgDG a  $h^{p+1}$  convergence rate is expected.

The setup of the numerical test corresponds to a square domain  $[-\lambda, +\lambda] \times [-\lambda, +\lambda]$  with its sides oriented in the same direction as the Cartesian axes. We impose boundary conditions, initial field and source term in such a way that the analytical solution of the problem is given by:

$$\mathbf{U} = \begin{Bmatrix} -\varepsilon \sin(\pi t) \cos(\pi x) \sin(\pi y) \\ \varepsilon \sin(\pi t) \sin(\pi x) \cos(\pi y) \\ -\mu \cos(\pi t) \cos(\pi x) \cos(\pi y) \end{Bmatrix}. \quad (45)$$

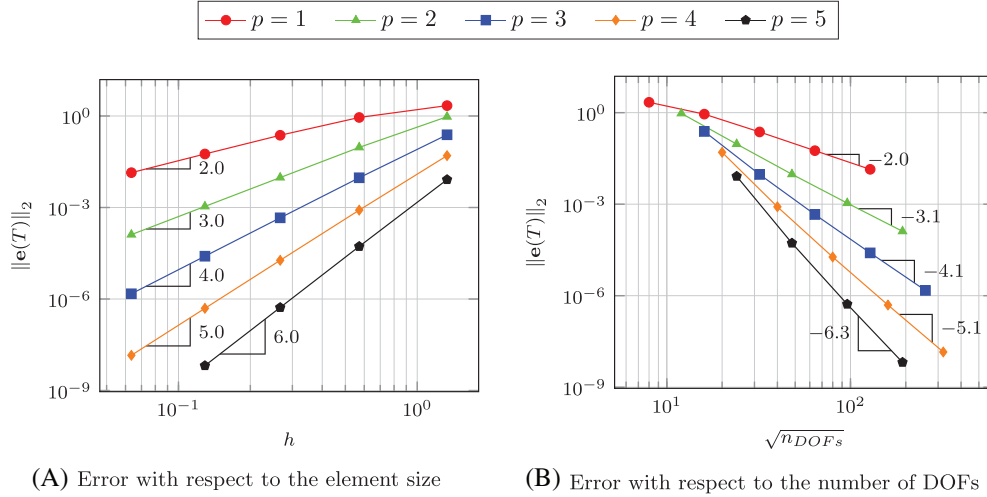


FIGURE 11 Convergence of the method

The set of increasingly refined meshes is defined so that the intersection points of the boundary elements with the geometry are at the midpoint of the edge. Starting with the coarsest mesh, each finer mesh possesses twice as many elements per direction as the immediate coarser one. Regarding the temporal integration, we use a small enough time step so that the spatial discretization error dominates over any other.

Figure 11A shows the errors of the cgDG solution field in the  $L_2$ -norm computed at time  $t = T$ ,  $\|\mathbf{e}(T)\|_2$ , with respect to the element size,  $h$ . The slope of the curves in the asymptotic region equals to  $p + 1$  which corresponds to the optimal convergence rate of DG when using Cartesian grids. This result puts the proposed method in advantage with respect to standard FE methods, that exhibits lower convergence rates. Figure 11B shows the same results versus the number of DOFs  $n_{DOFs}$ . It is worth noting that, to obtain a certain level of error, interpolations of a higher degree require a lower number of DOFs and, for this problem, a lower computational cost, due to the smoothness of the analytical solution. Furthermore, even for the same number of DOFs, using high-degree polynomials is preferable as it leads to a reduction in the number of elements and, therefore, the amount of numerical fluxes that needs to be evaluated. These results support the use of high-degree polynomial approximations to analyze problems involving wave propagation.

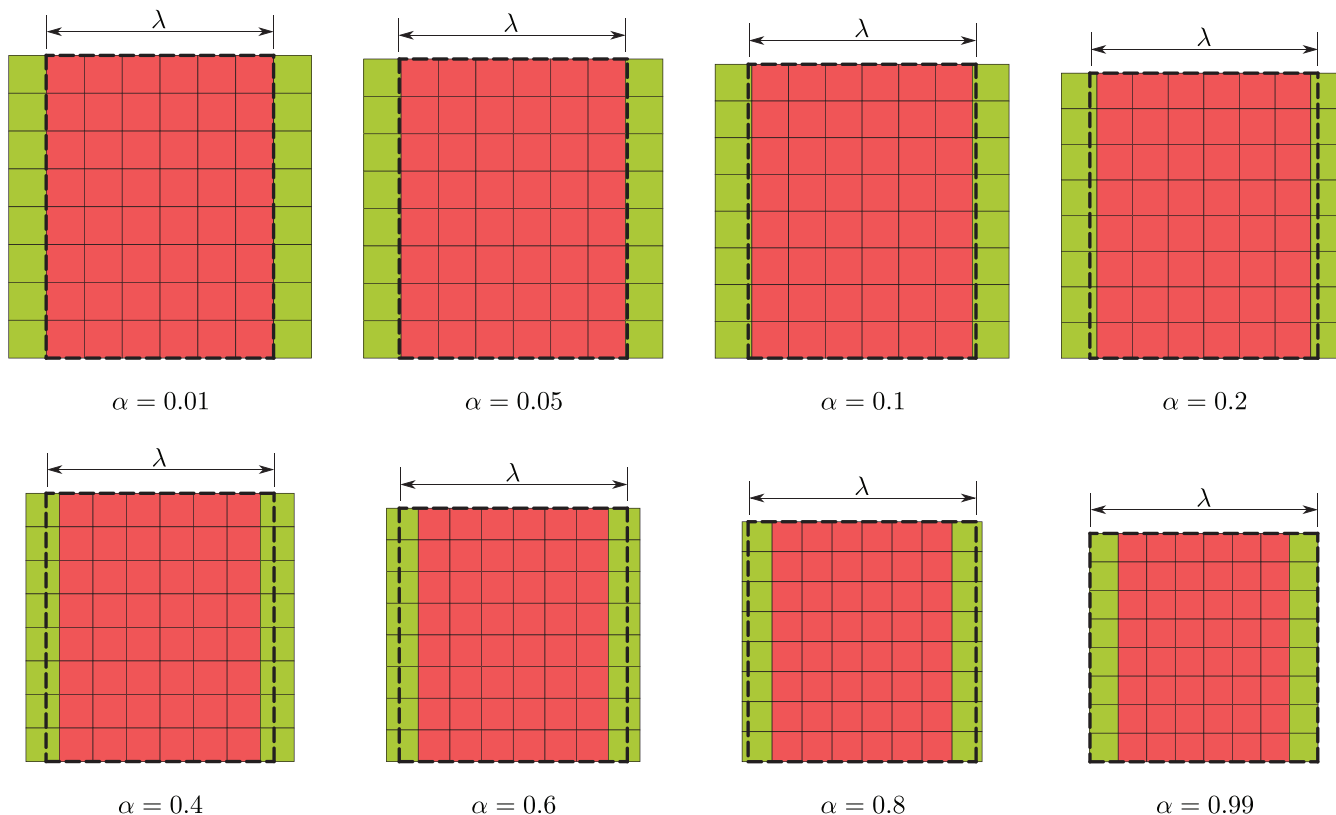
Next analysis pursues two goals: quantify the effect of the intersected elements on the stability of the method and evaluate the effectiveness of the domain extension strategy as a stabilization procedure. For this test, we consider a TE<sup>z</sup> electromagnetic plane wave that propagates in the positive direction of the  $x$ -axis through an infinite domain. The electric components of the wave are computed as:

$$\mathbf{E} = \mathbf{p} \cos(\omega t - \boldsymbol{\kappa} \mathbf{x}), \quad \boldsymbol{\kappa} = \omega \begin{Bmatrix} \cos \phi \\ \sin \phi \\ 0 \end{Bmatrix}, \quad \mathbf{p} = E_\phi \begin{Bmatrix} -\sin \phi \\ \cos \phi \\ 0 \end{Bmatrix} \quad (46)$$

with:

$$E_\phi = 1 \quad \phi = 0.$$

The infinite domain is modeled as a rectangular domain with periodic boundary conditions on the boundary. Given an initial distribution that satisfies (46), the system is advanced in time. With this configuration, we want to evaluate the stability of the method regardless of the influence of the boundary conditions. In the test that we propose, the fluxes through the boundary only depend on the numerical solution computed in the previous step. We run the same problem iteratively following a bisection method until we get the maximum time step for which the  $L_2$ -norm of the solution does not grow in time, hereinafter  $\Delta t_{st}$ . This result is an approximation of the time step that makes the spectral radius of the method equals to one.



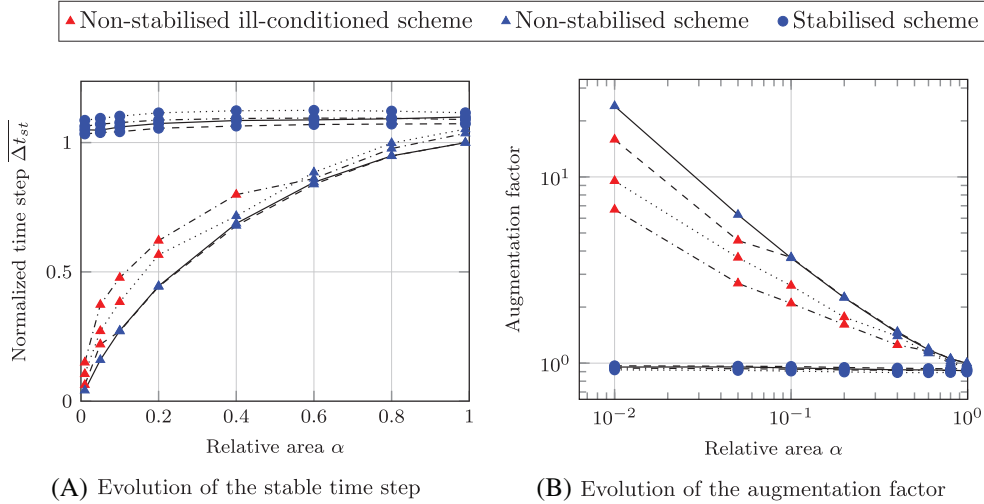
**FIGURE 12** Set of grids and computational domains used to test the stability of the method. The dashed lines represent the boundary of the computational domains. Elements in green are intersected elements, while elements in red are internal elements

Figure 12 gathers the set of base grids and computational domains considered in this test. All meshes are Cartesian grids made up of  $8 \times 8$  elements and the  $x_1$ -dimension of all computational domains equals the wavelength. However, the element sizes and the  $x_2$ -dimension of the computational domains vary so that  $\alpha = \{0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 0.99\}$  for the outer layer of elements on the left and on the right.

For each discretization, we considered two different scenarios. Firstly, all intersected and internal elements are active in the analysis mesh. Secondly, we apply the stabilization strategy before starting the integration loop. This second scenario uses a value of  $\alpha_{th}$  big enough so that all intersected elements will be classified as unstable and the stabilization procedure will be applied to them. In the final discretization all intersected elements of the initial mesh disappear since their domains are reassigned to the internal neighbors and their DOFs are also removed from the system of equations. We consider the interpolation degrees  $p = \{2, 4, 6, 8\}$ . In order to facilitate the understanding of the results,  $\Delta t_{st}$  is normalized with respect to the maximum stable time step calculated for an equivalent discretization without cut elements, namely,  $\overline{\Delta t_{st}}$ .

Figure 19 shows the evolution of  $\overline{\Delta t_{st}}$  with respect to  $\alpha$ . For the nonstabilized scheme we observe that  $\overline{\Delta t_{st}}$  drops with  $\alpha$ , tending to 0 as  $\alpha$  approaches this value. The slope of the curves is more pronounced as we move to smaller values of  $\alpha$ . If we focus on the stabilized discretization, the stable time step remains almost constant around one. The results show that the stability of the stabilized method is governed by the base mesh and is independent of the cut patterns that appear in the intersected elements.

The results also show a common problem of the FDM: when the analysis mesh contains any element that poses very little support, the system becomes ill-conditioned. As the degree of interpolation increases, the conditioning worsens and the domain necessary to ensure a certain precision of the solution increases. In Figure 13A, the results obtained for discretizations that exhibit this behavior have been represented in red. In these cases, the vector of unknowns has been calculated using the Moore–Penrose pseudoinverse.<sup>65</sup> It is important to point out that these situations only appear when nonstabilized discretization are used. The stabilization strategy eliminates the polynomial approximations with little support whose DOFs suffer from dependency problems. Therefore, the proposed stabilization strategy not only improves the



**FIGURE 13** Dependence of stable step size for the marching algorithm with respect to the relative ratio for both the standard and that stabilized discretizations

behavior of the method with regard to time integration, but also let us compute FE models that, even using very small time steps, could not be calculated accurately due to ill-conditioning.

In Figure 13B, the inverse of Figure 13A is drawn. On this occasion, the  $y$ -axis represents the factor by which the number of necessary iterations of the time loop is multiplied in order to reach the end of the simulation. Given that in most practical examples the calculation time corresponding to the time loop represents mostly the entire analysis time and that the computational cost of each iteration is independent of the step, this factor also applies to the overall analysis length. We see that the negative effects of  $\alpha$  from the point of view of time integration are significant for  $\alpha \leq 0.4$ . As an example, for a polynomial approximation of order  $p = 3$ , if the mesh contains an intersected element with  $\alpha = 0.2$ , this supposes that the simulation prolongs 2.26 times. Contrary, the stabilized scheme is not affected by the parameter  $\alpha$  and, therefore, the simulation times will be the same regardless of how the geometry intersects the mesh. In this latter case, the simulation time will only depends on the target level of accuracy, which will define the size of the elements and the approximation degree.

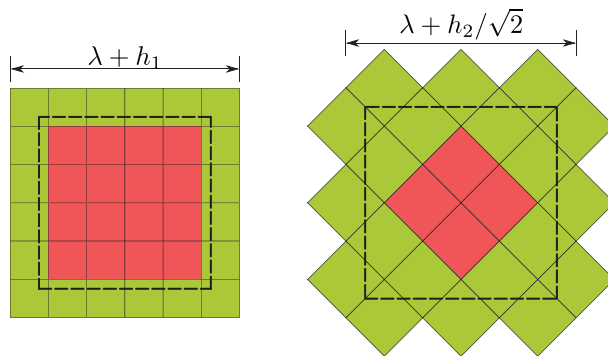
The same test has been carried out for a range of different element sizes. In all cases, the method behaves as in case described above.

Extending the support of the interpolation polynomials implies the reduction in the richness of the solution since the number of DOFs per wavelength reduces. To counteract the expected loss of accuracy, in Section 5 we proposed to increase the degree of the polynomial approximation used on the extended elements. The  $p$ -refinement must be done in such a way that the stabilized scheme keeps the magnitude of the discretization error. To calculate the expected errors, a characteristic length of the interpolation support domain is considered. We define the approximation degree on element  $i$  based on the characteristic length  $l_i^d$ , which is the size of the projection on the Cartesian axes of the integration domain associated to element  $i$ . The degree is assigned according to the expression:

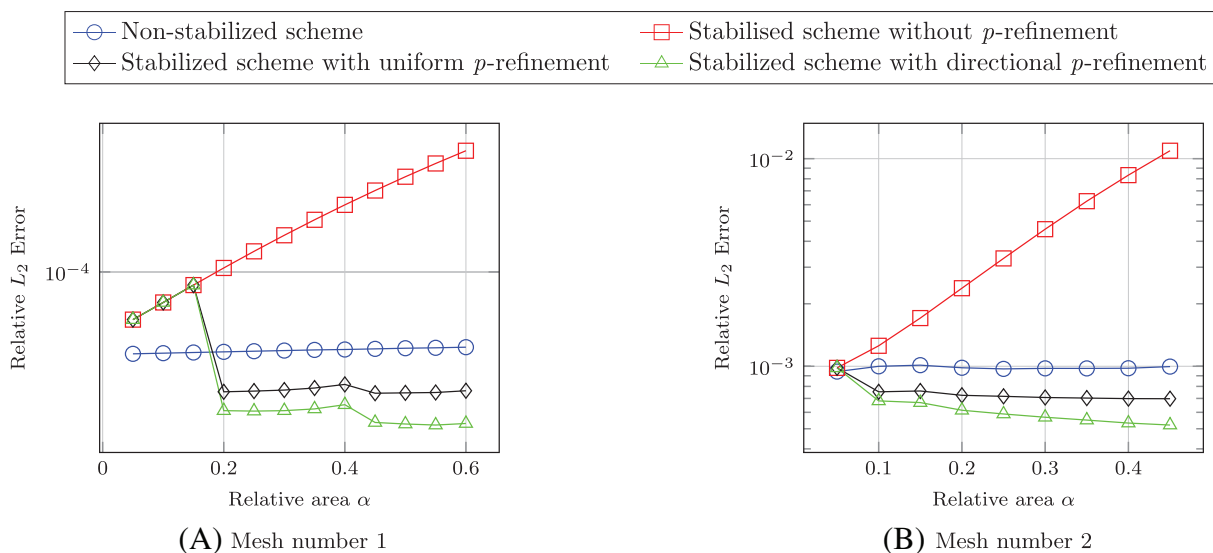
$$p_i = \left\lceil (p_r + 1) \frac{\ln(h_r)}{\ln(l_i)} - 1 \right\rceil, \quad (47)$$

where  $p_r$  and  $h_r$  are two parameter defined by the user that correspond to the degree and element size of a reference discretization, respectively. The previous expression is evaluated independently in each of the directions. Once the appropriate polynomial degrees for each direction are computed, we may use all this information to defined the final discretization, leading to a directional  $p$ -refinement; or we may only consider the worst case and use the same degree of approximation in both directions.

The following test seeks to quantify the effect of the stabilization strategy on the global error and the performance of the proposed  $p$ -refinement strategies. We use a numerical test that consists of a squared computational domain through which a wave is propagated. The distribution of such wave correspond to Equation (45). The dimensions of the domain are modified to sample the space of  $\alpha$  parameter. Two different meshes that differ in orientation are studied. The border



**FIGURE 14** Analysis meshes used to quantify the effect of the stabilization strategy on the discretization error. Mesh number 1 and 2 on the left and on the right, respectively



**FIGURE 15** Effect of the extension strategy on error

of the bounding box used to obtain mesh number 1 is aligned with the Cartesian axes, while the bounding box of mesh number 2 is rotated 45°. The base meshes remain unchanged throughout the test and we interpolate nodal values using fourth-degree polynomials. Figure 14 shows mesh number 1 and mesh number 2 for  $\alpha = 0.25$ .

In Figure 15, the red line represents the error in  $L_2$ -norm of the solution for time  $t = T$  for the stabilized scheme while the blue line shows evolution of the same magnitude for the nonstabilized one. First of all, it should be noted the higher error obtained for mesh number 2 is due to its elements being of a larger size. For both examples, we observe that the error for the stabilized scenario is always greater than that of the nonstabilized one. The difference increases as the relative area grows. The gap reaches a size of one order of magnitude. Furthermore, we see that the slope of the curve for the stabilized results is steeper for mesh 2 than for mesh 1. The explanation of this behavior may be related to the different size of the extension from the point of view of linear distance. Although the area of the unstable domain is the same, if we focus on a single direction of space, the extensions for mesh number 2 are greater than for mesh number 1. This yields a reduction of the nodal density in this direction which leads to an increase in the global error.

In contrast, we realize that using any  $p$ -refinement strategy driven by (47) we can keep the global error in the same order of magnitude as for the nonstabilized case, which shows that the refinement criterion is appropriate. As the extent of the extrapolation increases, the refinement criterion detects that the error distances from the target value and locally enriches the polynomial approximation to compensate this deviation. In Figure 15, each degree increment translates into a drop in the curves going from left to right. Looking closely, we realize that mesh number 1 experienced two increases in

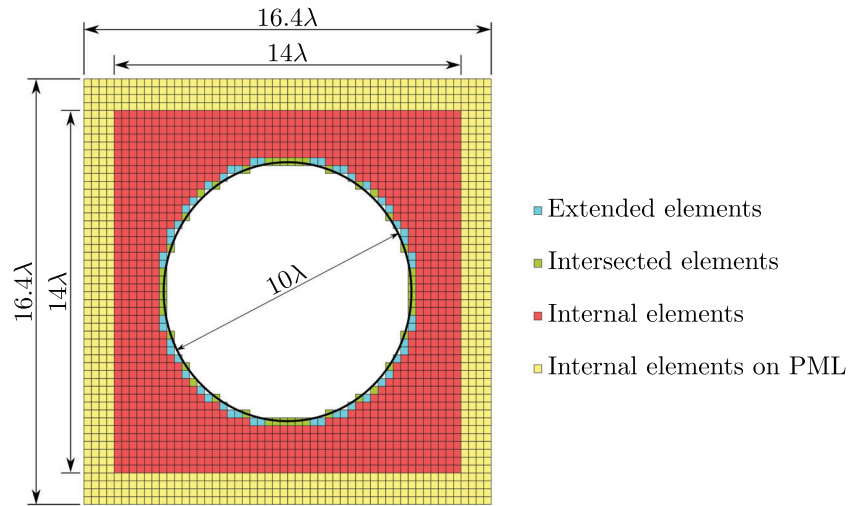


FIGURE 16 Mesh and element locations

the degree of interpolation during the test while this number goes up to five for mesh number 2. The higher interpolation degrees used in the computations are 6 and 9, respectively.

Comparing the results for both  $p$ -refinement strategies, the drops in the error are more pronounced when the  $p$ -refined discretization is computed regardless of the extension direction. Thus, this strategy leads to a lower global error. However, we may not forget that the error target we pursue is that provided by the nonstabilized scheme. From this perspective, the directional  $p$ -refinement strategy reaches the desired level of accuracy. In addition, this second alternative brings two main advantages. First, it lets us save some computational effort since the number of DOFs of the final discretization is smaller. Second, the stability of the model is hardly affected since the polynomial order only raises in the direction that can be balanced with a larger support. For these reasons, we adopt this last strategy for the remainder of this contribution.

As previously mentioned, the stability depends on the degree of the polynomial approximation as well as the extent of its support. Increasing the degree of the extended elements to preserve a certain level of accuracy negatively affects the maximum stable time step. A balance has to be found between the target error and the number of steps of the time integrator. In practice, this dependence does not entail an important limitation since the worst cases from the point of view of the time integration do have little influence on the error and vice versa. Values of  $\alpha_{th}$  around 0.3 have proven to be a good compromise solution for the set of problems studied so far.

## 6.2 | Scattering by a PEC circular cylinder

The second problem to be studied is an infinite cylinder with the properties of a PEC which reflects a plane wave that propagates perpendicularly to the height of the cylinder and whose  $\mathbf{H}$  or  $\mathbf{E}$  vector fields are aligned to the height at any point. This is a reference problem in the field of scattering as an analytical expression is available for the exact radar cross section (RCS) or, more precisely, the scattered width (SW), the counterpart of the RCS in 2D. See Reference 66 for further details. The 2D model considers a slice of the problem perpendicular to the height of the cylinder.

In the first analysis, a cylinder with a diameter equal to  $10\lambda$  is considered and solved both, with the  $TE^z$  and  $TM^z$  formulations, for an incident wave that propagates in the positive direction of the  $x$ -axis. The infinite domain is truncated to an extension of  $2\lambda$  and the infinite boundary condition is imposed by using a Berenguer PML with an ABC on the boundary.<sup>55</sup> It is worth mentioning that the PML strategies fit perfectly within the cgFEM framework. The Cartesian structure of the mesh offers an ideal setting to define the subdomains of the PML. Furthermore, by matching the boundary of these subdomains with the element edges, the increase in computational cost associated with the arising of a greater number of intersected elements is avoided.

The discretization used consists of a uniform mesh of elements of fourth degree and a nodal density of 13 nodes per wavelength. Figure 16 depicts the analysis mesh once the stabilization strategy has been applied.

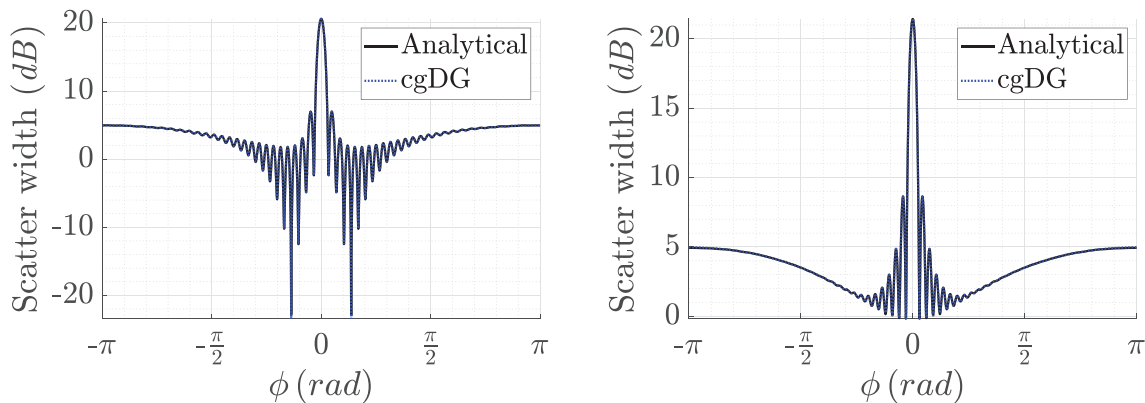


FIGURE 17 Bistatic scattered width for a  $10\lambda$  PEC cylinder

TABLE 1 Relative error ( $\times 10^{-4}$ ) in  $L_2$ -norm for the scattered width ( $\Delta\varphi = 9^\circ$ )

$\varphi$	0	$\Delta\varphi$	$2\Delta\varphi$	$3\Delta\varphi$	$4\Delta\varphi$	$5\Delta\varphi$
TE <sup>z</sup> Error	77.0	62.8	51.9	39.1	35.1	35.1
TM <sup>z</sup> Error	45.3	45.1	38.9	32.8	27.7	25.4

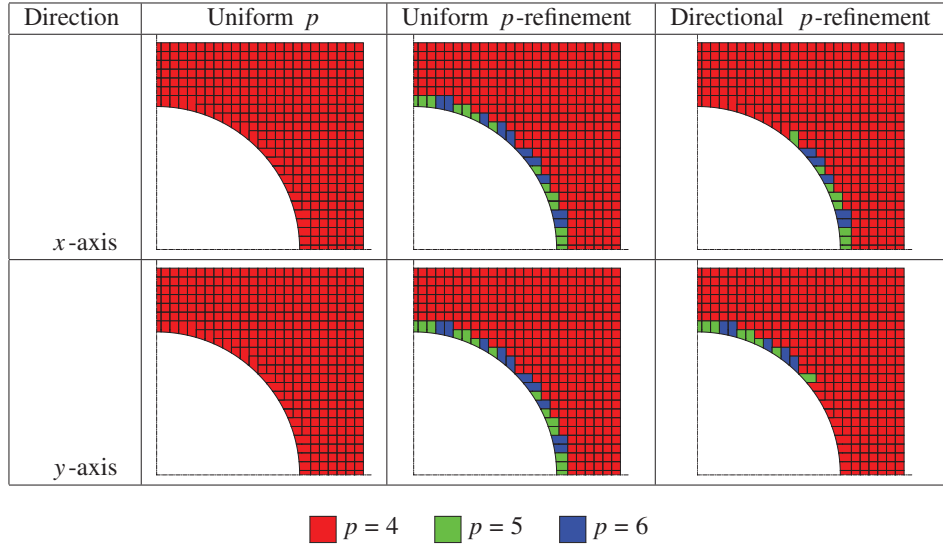
Figure 17 gathers the results obtained for the bistatic scattered width computed in both the TE<sup>z</sup> and the TM<sup>z</sup> modes. In these graphs  $\phi$  represents the bistatic angle. In both cases, we consider that the time harmonic steady state is reached when the  $L_2$ -norm of the SW varies in less than  $10^{-4}$  per unit in two consecutive cycles. Together with the numerical solution, the analytical solution is plotted. We see that the curves overlap for any  $\phi$ . Numerically, the relative error in  $L_2$ -norm of the SW is equal  $7.7 \times 10^{-3}$  for the TE<sup>z</sup> case and equal  $4.53 \times 10^{-3}$  for the TM<sup>z</sup> case. These results confirm the accuracy of the proposed method.

In the following test the influence of the direction  $\varphi$  in which the incident wave propagates on the overall solution is studied. The objective is to evaluate the influence of the apparent length of the element with respect to the wavelength on the accuracy of the simulation.

Changing the incidence direction leads to an equivalent change in the scattered field. We use this to evaluate if there is any propagation direction in the Cartesian mesh that drives to higher or lower accuracy. We start from the configuration of the previous case and carry out five analyses in which the angle of incidence is increased at a rate of  $\Delta\varphi = 9^\circ$ . Table 1 gathers the relative errors of the SW calculated according to both the TE<sup>z</sup> and the TM<sup>z</sup> mode for each of the angles in the  $L_2$ -norm. The values show a certain dependence to the orientation of the incident wave. However, the error of the SW always remains in the same order of magnitude, so we deduce that the method is not sensitive to the direction of propagation. In other words, the information equally propagates in all directions independently of the Cartesian structure of the mesh. The small variations that we observe are probably caused by the influence of the terms of higher degree presented in the nodal interpolation.

The results shown so far in this subsection rely on a uniform discretization both in terms of element size and approximation degree. The final test of this set focuses on the effects of the stabilization strategy in conjunction with the  $p$ -refinement strategy. Unlike the squared geometry studied in previous subsection, the circular boundary generates a great variety of cut patterns with different shapes and relative areas. In this example we reduce the domain of propagation up to  $1.5\lambda$  and simulate considering: uniform  $p$ , uniform  $p$ -refined and directional  $p$ -refined meshes; and the stability thresholds  $\alpha_{th} = \{0.2, 0.4, 0.6, 0.8\}$ . In Table 2 we put together six figures that show how the elemental interpolation degrees in each direction vary depending on the  $p$ -refinement strategy. These distributions arise for  $\alpha_{th} = 0.8$  and preserve symmetry with respect to  $x$  and  $y$  axes. The relative errors of the SW in the  $L_2$ -norm are gathered in Table 3. The results confirm that increasing the stability threshold has a negative effect on the accuracy as far as no compensation is considered. Regarding the performance of the  $p$ -refinement strategies, we do not find any difference in terms of error reduction while the directional  $p$ -refinement strategy lets save some computational cost. Therefore, we recommend the use of this approach.

**TABLE 2** Comparison of the spatial discretizations for the PEC cylinder problem depending on the  $p$ -refinement strategy used



**TABLE 3** Relative error ( $\times 10^{-4}$ ) in  $L_2$ -norm for the SW

$\alpha_{th}$	0.2	0.4	0.6	0.8
Uniform $p$	13.2	86.6	114.5	241.3
Uniform $p$ -refinement	7.6	8.9	11.5	13.6
Directional $p$ -refinement	7.6	9.0	11.0	13.8

### 6.3 | Scattering by a PEC NACA0012 aerofoil

In this section the scattering of a plane electromagnetic wave that falls out a NACA0012 profile modeled as a PEC is studied. The NACA0012 profile is a symmetrical shape that presents a singular point at the trailing edge. Special attention needs to be paid to the spatial discretization around this kind of points in order to accurately capture the distribution of the electromagnetic field.

In the first place, the effect of the discretization around the singular point on the calculation of the SW will be studied. We compute the bistatic SW produced by a profile with a chord length equal to  $10\lambda$ . The incident field correspond to a plane  $TE^z$  wave whose trajectory forms an angle of  $10^\circ$  with respect to the chord line. The outer boundary of the computational domain is a square whose side length is equal to  $13\lambda$ . The bounding box of the profile is centered inside this domain. We run a collection of seven simulations which differ each other by the geometry orientation. The chord line with respect to the  $x$ -axis  $\varphi_i$  varies from  $\varphi_0 = 0^\circ$  to  $\varphi_6 = -45^\circ$ , with increments of  $\Delta\varphi = -9^\circ$ . For each configuration, two analyses are performed: the first one employs a uniform mesh while the second one makes use of a  $hp$ -refined mesh. This last discretization uses top-level elements (smaller  $h$ ) in the surroundings of the leading and trailing edges. In all cases, the interpolation degree is assigned to each element according to Equation (47). Such approach means that refined elements are of lower order. Figure 18 shows examples of both meshes for  $\varphi_6$  as well as a detail of the elements around the singularity. This last representation makes it clear that multilevel jumps between adjacent elements are allowed. The setup completes deactivating the  $p$ -refinement strategy design to counteract of the extension and setting the stability limit to  $\alpha_{th} = 0.4$ .

Figure 19 shows the SW obtained with the uniform mesh (A) and with the  $hp$ -refined mesh (B). Both cases are compared with a reference solution computed on a verified FE software using a very fine high-order spatial discretization. We observe that the SW computed with the uniform mesh is much more sensitive to the position of the geometry than the one derived from the  $hp$ -adapted mesh. The curves overlap independently of the bistatic angle. We only observe small discrepancies at those angles for which the reflected intensity is lower. Moreover, a very good agreement with the reference values is observed which highlights the higher accuracy of the  $hp$ -adapted scheme. The results show that the method is



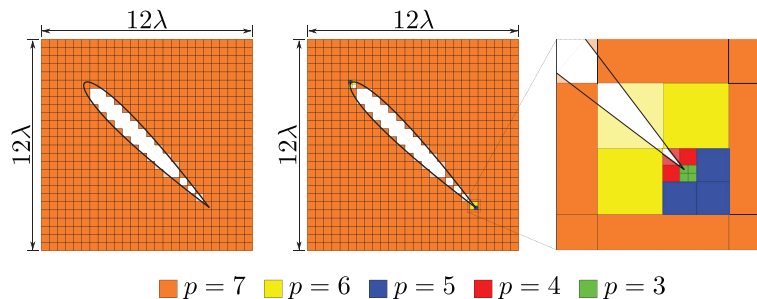


FIGURE 18 Comparison of the meshes used to discretize the PEC NACA0012 profile

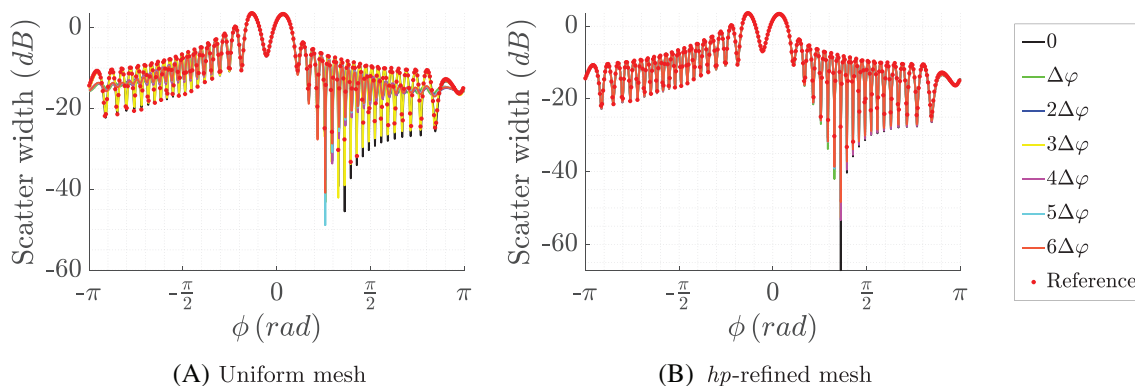


FIGURE 19 Influence of the *hp*-refinement procedure based on geometry on the scattered width

sensitive to the complexity of the geometry and its position with respect to the mesh, what translate into a higher complexity of the electromagnetic field within the element. This effect can be counteracted by using geometry-based *h*-refining strategies and subsequently using an appropriate degree assignment strategy to result in an *hp*-adapted discretization. In addition, these last results confirm the conclusion reached in the previous example: the precision of the method is independent of how information propagates.

The following example analyses the same geometry but subjected to a high frequency incident wave. This simulation supposes a great challenge as most numerical approaches suffer from numerical dispersion and/or dissipation which limits their scope of application. The wavelength is shortened so that the chord line becomes  $100\lambda$ . The direction of incidence is also modified, and set to coincide with the direction of the chord. The geometry is embedded in a domain that extends  $1\lambda$  in each direction beyond the geometry bounding box and is surrounded by a  $1\lambda$  thick PML. Regarding discretization, we consider a nodal density of 17 nodes per wavelength and a interpolation degree equal to 7 for the elements of the uniform mesh. We apply a *h*-refinement strategy that split the elements of the uniform mesh up to two times. Regarding criterion (47), the resulting mesh contains elements of degree  $p = \{7, 6, 5\}$ . This brings the total of 363,602 nodes and 5687 elements. These numbers are quite contained for a problem of these characteristics.

Figure 20 shows the spatial distribution of the nonzero components of the scattered electromagnetic field once it is fully developed. Figure 21 compares the SW computed with the cgDG implementation and a reference solution. The great match between these curves demonstrates the great performance of the proposed method when applied to high-frequency simulations.

### 6.4 | Scattering by complex geometries

The performance of the method when applied to two complex geometries is studied next. The first problem concerns the scattering caused by the middle section of a PEC satellite. The maximum dimension of the satellite bounding box

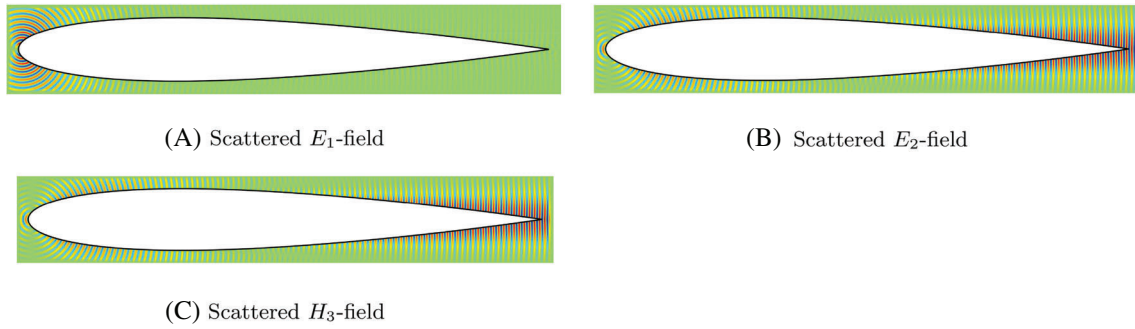


FIGURE 20 Scattered electromagnetic field for the  $100\lambda$  NACA profile

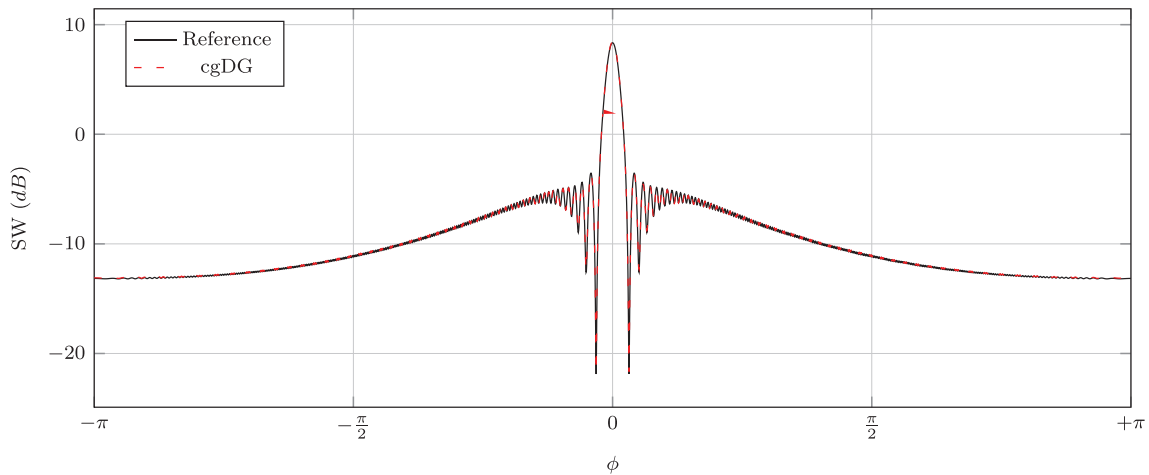


FIGURE 21 Scattered width for the  $100\lambda$  NACA profile

is  $40\lambda/3$  and the region of interest extends  $1.5\lambda$  beyond this bounding box. The computational domain is completed by a  $1.5\lambda$ -thick PML that surrounds that region. In this case, the problem is formulated in  $TM^z$  mode and the plane incident wave travels in the direction of the  $x$ -axis. The spatial discretization used for this analysis correspond to a uniform mesh formed by 1582 elements of degree 7, which makes a total of 303,744 DOFs. It should be noted that, despite the fact that the satellite geometry presents details that are small compared with the element size, the cgFEM technology allows this boundary to be taken into account without the need to refine the elements over the boundary until their sizes match those of the geometric features. The spatial discretization is governed by the complexity of the solution but, unlike most of the fitted methods, it is independent of how the geometry is defined. In Figure 22, the third component of the electric field is presented at the moment the steady state is reached. The figure shows how the method is able to capture the complexity of the field in the antenna cavity. In Figure 23 the SW obtained with cgDG is compared with that computed by a contrasted FE software using an overrefined discretization. We see that, although we are using quite a coarse spatial discretization, the results fit the reference solution<sup>25</sup> quite well. Furthermore, it is worth mentioning that the SW obtained with the proposed method is perfectly symmetric. This results from the fact that the spatial discretization presents symmetry about the  $x$ -axis, just like the physical problem (domain and boundary conditions). Obtaining this sort of symmetries for the spatial discretization in cgDG is straightforward since the mesh only depends on the definition of the bounding box and the splitting process, both of which present symmetry about the Cartesian axes. In contrast, most mesh generation techniques used in fitted methods evolves to nonsymmetric discretization regardless of the starting point.

Next, the scattering problem for the middle section of an airplane is considered. This boundary is particularly challenging from the point of view of the geometry. Small features compared with the global size the profile, such as the front antenna, originate resonance regions that have a great influence on the overall distribution of the electromagnetic field. In order to capture these local complexities of the solution we use an  $hp$ -adapted mesh with smaller elements in the

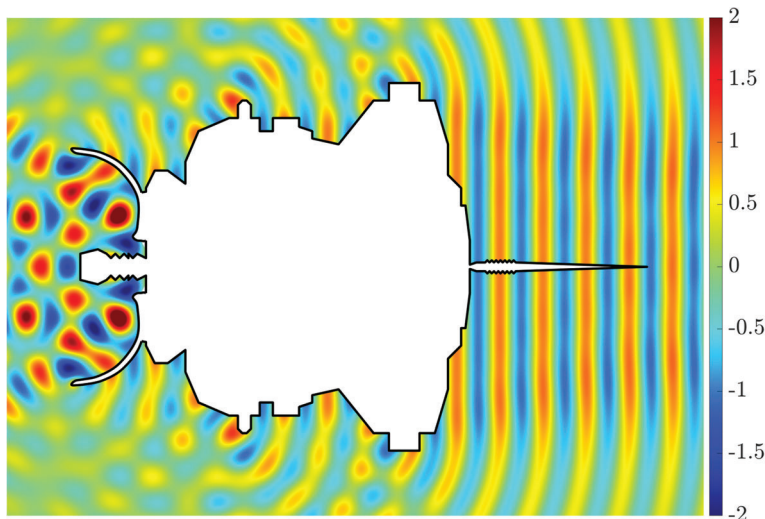


FIGURE 22  $H_3$  scattered field computed for the satellite profile

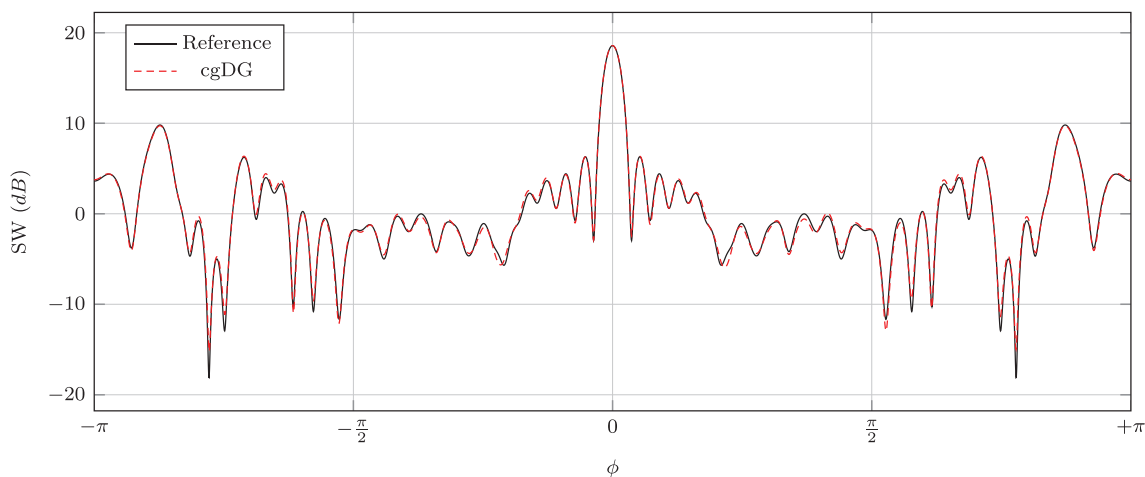


FIGURE 23 Bistatic scattered width obtained for the satellite profile

environment of this geometric feature. Additionally, as these structures are slender, they give rise to elements with several isolated regions within the physical domain. The dissociation strategy is used on these elements.

In the following, the setup of the airplane problem is depicted. The body is modeled as a PEC whose length is  $50\lambda$ . The physical domain extends  $2\lambda$  beyond the geometry bounding box in both directions with a  $2\lambda$ -thick PML surrounding it. We consider the  $TM^z$  version of the problem and study the scattering a plane wave that propagates from left to right. As aforementioned, this analysis uses an  $hp$ -adapted mesh with smaller elements around the most relevant geometric details and a base discretization of 20 nodes per wavelength. Figure 24 shows the complete mesh as well as a zoom of the antenna surroundings. Note the presence of elements with more than one disconnected regions. We studied how these elements evolve throughout the flow of the program. For the element  $K^i$ , at the beginning of the program two independent regions  $K_1^i$  and  $K_2^i$  are detected, each of which is assigned a different set of nodes  $K_1^i$  and  $K_2^i$ , respectively. When evaluating the stability, it is detected that the discretization of the  $K_2^i$  region is unstable and the support domain is reassigned to the element  $K_1^{i+1}$  giving rise to  $K_2^{i+1}$ . Finally, the DOFs linked to  $K_2^i$  are removed from the set of DOFs and the discretization  $K_1^{i+1}$  is enriched through  $p$ -refinement to obtain  $k_2^{i+1}$ .

The third component of the electric field are shown in Figure 25. Figure 26 shows some details of the solution in the regions where it exhibits a higher complexity. At the overall level, we see how the method is able to capture the influence of geometric details. We observed an increase in the amplitude of the scattered field, especially in the lower part of the

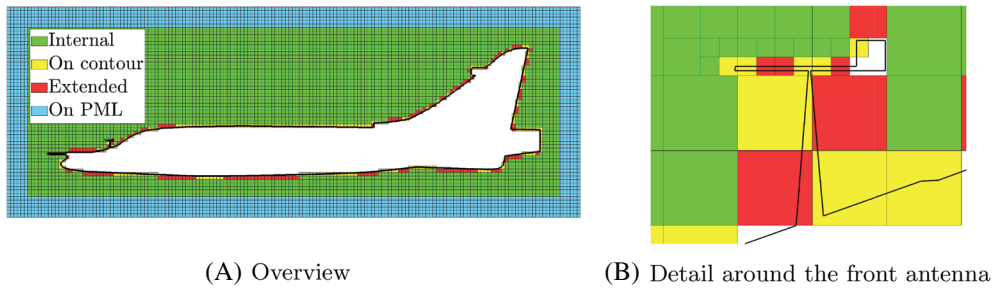


FIGURE 24 Analysis mesh for the aircraft section

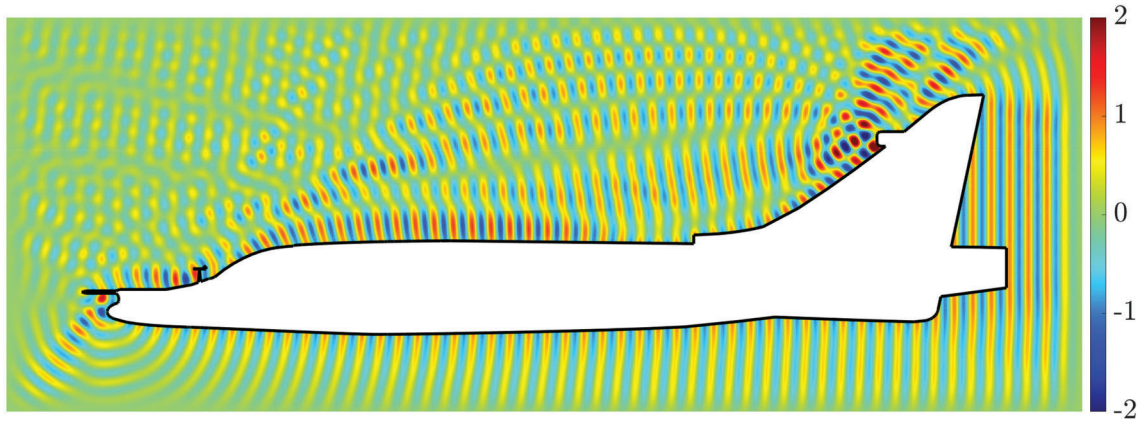


FIGURE 25  $H_3$  scattered field computed for the aircraft section

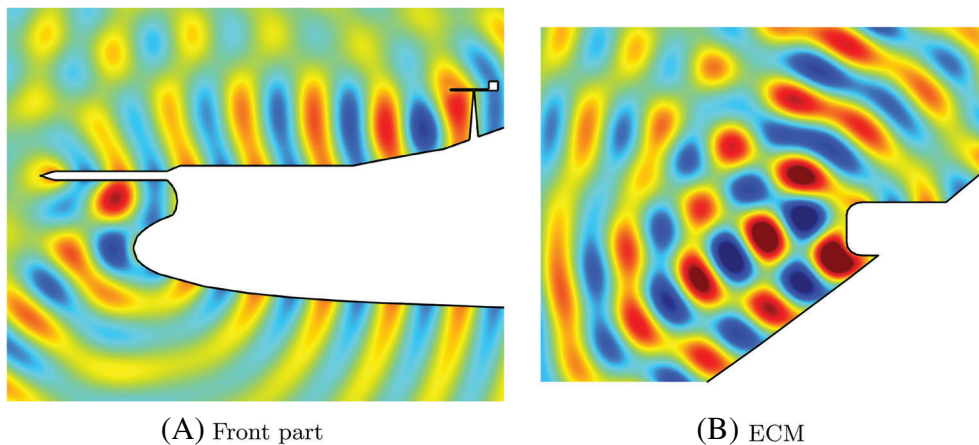


FIGURE 26 Zoom of the  $H_3$  field around geometrical details

refueling needle, the front part of the antenna placed in front of the cockpit and around in the electronic countermeasure (ECM) placed on the vertical stabilizer. However, when we go into detail, the polynomial approximation used is not able to correctly represent the very local complexity of the solution caused by the complexity of the boundary. This is clearly seen in the elements that contain the end of thin details of the geometry. This limitation does not affect the general performance of the method, as these are errors that affect locally and, thanks to the mesh refinement, do not pollute the global solution. Figure 27 confirms this assertion. The SW obtained with the proposed method fits the reference solution, computed with a much finer fitted NEFEM mesh<sup>25</sup> and an in-house high-order DG solver, really well, taken into account the total amount of 523,828 nodes considered in this simulation.

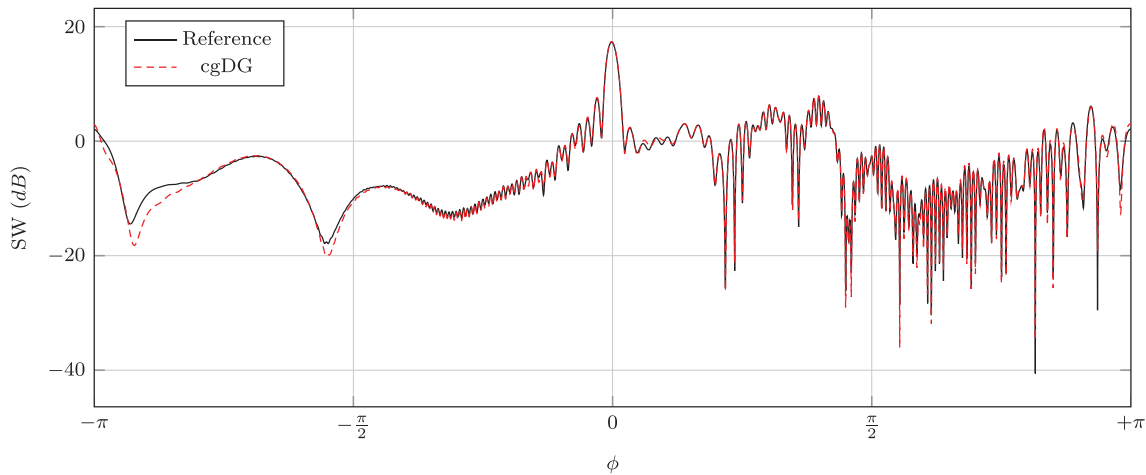


FIGURE 27 Bistatic scattered width obtained for the aircraft profile

## 7 | CONCLUSIONS

In this article, the cgDG has been presented. This high-order fictitious domain method possesses very convenient properties to address the simulation of wave propagation problem through intricate domains. Beyond developing the mathematical background of the conjunction, a collection of complementary strategies that ensure the highest performance of the resulting scheme are described. The stabilization strategy we implemented to deal with instabilities caused by intersected elements shows a very robust behavior and its use can be easily extrapolated to other numerical schemes or physical problems as it only affects the discretization of the space of interest. We have shown that, to maintain the accuracy of the approximation, any countermeasure must be considered and we propose the use of a priori directional  $p$ -refinement that cancels out the error increase due to the reduction in the number of degrees of freedom. The dissociation strategy we adopted to manage independent solution fields within the same elements also has a great influence on the overall performance of the methods. It allows the use of higher-order elements as the presence of this kind of elements no longer supposes a problem.

Next steps in the development of this method are its extension to 3D problems and its application to new physical phenomena such as fluid dynamics. The implementation of an error-based refinement strategies within the time loop would be of great interest as would let us more efficiently capture the solution on those regions such as cavities for electromagnetic simulations or turbulences for fluid simulations where the complexity of the field being analyzed increases. Regarding the stabilization strategy further research may be devoted to explore alternative stability criteria as well as new factors to tune up the weighted distance we used to select the extended elements. Even though this approach performs well for the collection of problems studied so far, new schemes may need slight variations. The dissociation strategy has shown to be a very powerful tool and further applications should be explored.

## ACKNOWLEDGMENTS

The authors thank the Ministry of Economy and Competitiveness (DPI2017-89816-R) and the Ministry of Science, Innovation and Universities (FPU17/03993) of the Government of Spain for the financial support. Ruben Sevilla also acknowledges the support of the Engineering and Physical Sciences Research Council (Grant number: EP/T009071/1).

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding authors upon reasonable request.

## ORCID

Héctor Navarro-García  <https://orcid.org/0000-0003-3160-7609>

Rubén Sevilla  <https://orcid.org/0000-0002-0061-6214>

Enrique Nadal  <https://orcid.org/0000-0002-2808-298X>

Juan José Ródenas  <https://orcid.org/0000-0003-2195-7920>

## REFERENCES

1. Yee KS. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans Antennas Propag.* 1966;14(3):302-307.
2. Taflove A, Hagness SC. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Vol 80. 3rd ed. Artech House; 2005.
3. Yee KS, Chen JS. The finite-difference time-domain (FDTD) and the finite-volume time-domain (FVTD) methods in solving Maxwell's equations. *IEEE Trans Antennas Propag.* 1997;45(3):354-363.
4. El Hachemi M, Hassan O, Morgan K, Rowse DP, Weatherill NP. Hybrid methods for electromagnetic scattering simulations on overlapping grids. *Commun Numer Methods Eng.* 2003;19(9):749-760.
5. Reed WH, Hill TR. *Triangular Mesh Methods for the Neutron Transport Equation (No. LA-UR-73-479; CONF-730414-2)*. Los Alamos Scientific Lab; 1973.
6. Cockburn B, Karniadakis GE, Shu C-W. The development of discontinuous Galerkin methods. *Lecture Notes in Computational Science and Engineering*. Vol 11. Springer; 2000:3-50.
7. Hesthaven JS, Warburton T. Nodal high-order methods on unstructured grids: I. time-domain solution of Maxwell's equations. *J Comput Phys.* 2002;181(1):186-221.
8. Sevilla R, Hassan O, Morgan K. The use of hybrid meshes to improve the efficiency of a discontinuous Galerkin method for the solution of Maxwell's equations. *Comput Struct.* 2014;137:2-13.
9. Courant R, Friedrichs K, Lewy H. On the partial difference equations of mathematical physics. *IBM J Res Dev.* 1967;11(2):215-234.
10. Gottlieb S, Shu CW, Tadmor E. Strong stability-preserving high-order time discretization methods. *SIAM Rev.* 2001;43(1):89-112.
11. Gottlieb S. On high order strong stability preserving Runge-Kutta and multi step time discretizations. *J Sci Comput.* 2005;25(1-2):105-128.
12. Grote MJ, Mitkova T. Explicit local time-stepping methods for Maxwell's equations. *J Comput Appl Math.* 2010;234(12):3283-3302.
13. Krivodonova L. An efficient local time-stepping scheme for solution of nonlinear conservation laws. *J Comput Phys.* 2010;229(22):8537-8551.
14. Piperno S. Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems. *ESAIM Math Modell Numer Anal.* 2006;40(5):815-841.
15. Hochbruck M, Pažur T. Implicit Runge-Kutta methods and discontinuous Galerkin discretizations for linear Maxwell's equations. *SIAM J Numer Anal.* 2015;53(1):485-507.
16. Kennedy CA, Carpenter MH. *Diagonally implicit Runge-Kutta Methods for Ordinary Differential Equations. A Review*. NASA Technical Memorandum ; 2016-219173. National Aeronautics and Space Administration, Langley Research Center; 2016.
17. Ascher UM, Ruuth SJ, Spiteri RJ. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl Numer Math.* 1997;25(2-3):151-167.
18. Kanevsky A, Carpenter MH, Gottlieb D, Hesthaven JS. Application of implicit-explicit high order Runge-Kutta methods to discontinuous-Galerkin schemes. *J Comput Phys.* 2007;225(2):1753-1781.
19. Dolean V, Fahs H, Fezoui L, Lanteri S. Locally implicit discontinuous Galerkin method for time domain electromagnetics. *J Comput Phys.* 2010;229(2):512-526.
20. Descombes S, Lanteri S, Moya L. Locally implicit time integration strategies in a discontinuous Galerkin method for Maxwell's equations. *J Sci Comput.* 2013;56(1):190-218.
21. Sevilla R, Fernández-Méndez S, Huerta A. NURBS-enhanced finite element method (NEFEM). *Int J Numer Methods Eng.* 2008;76(1):56-83.
22. Xie ZQ, Sevilla R, Hassan O, Morgan K. The generation of arbitrary order curved meshes for 3D finite element analysis. *Comput Mech.* 2013;51(3):361-374.
23. Sevilla R, Fernández-Méndez S, Huerta A. Comparison of high-order curved finite elements. *Int J Numer Methods Eng.* 2011;87(8):719-734.
24. Dawson M, Sevilla R, Morgan K. The application of a high-order discontinuous Galerkin time-domain method for the computation of electromagnetic resonant modes. *Appl Math Model.* 2018;55:94-108.
25. Sevilla R, Rees L, Hassan O. The generation of triangular meshes for NURBS-enhanced FEM. *Int J Numer Methods Eng.* 2016;108(8):941-968.
26. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: I. a stabilized Lagrange multiplier method. *Comput Methods Appl Mech Eng.* 2010;199(41-44):2680-2686.
27. Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: II. a stabilized Nitsche method. *Appl Numer Math.* 2012;62(4):328-341.
28. Nadal E, Ródenas JJ, Albelda J, Tur M, Tarancón JE, Fuenmayor FJ. Efficient finite element methodology based on cartesian grids: application to structural shape optimization. *Abstr Appl Anal.* 2013;1-19:2013.
29. Tur M, Albelda J, Nadal E, Ródenas JJ. Imposing Dirichlet boundary conditions in hierarchical Cartesian meshes by means of stabilized Lagrange multipliers. *Int J Numer Methods Eng.* 2014;98(6):399-417.
30. Tur M, Albelda J, Marco O, Ródenas JJ. Stabilized method of imposing Dirichlet boundary conditions using a recovered stress field. *Comput Methods Appl Mech Eng.* 2015;296:352-375.

31. Tur M, Albelda J, Navarro-Jimenez JM, Rodenas JJ. A modified perturbed Lagrangian formulation for contact problems. *Comput Mech.* 2015;55(4):737-754.
32. Navarro-Jiménez JM, Tur M, Fuenmayor FJ, Ródenas JJ. On the effect of the contact surface definition in the Cartesian grid finite element method. *Adv Model Simul Eng Sci.* 2018;5(1):12.
33. Navarro-Jiménez JM, Tur M, Albelda J, Ródenas JJ. Large deformation frictional contact analysis with immersed boundary method. *Comput Mech.* 2018;62(4):853-870.
34. Muñoz D, Ródenas JJ, Nadal E, Albelda J. 3D topology optimization with h-adaptive refinement using Cartesian grids finite element method (cgFEM). Proceedings of the 6th International Conference on Engineering Optimization EngOpt 2018; 2019: 778-788; Springer International Publishing, Cham.
35. Muñoz D, Albelda J, Ródenas JJ, Nadal E. Improvement in 3D topology optimization with h-adaptive refinement using the Cartesian grid finite element method. *Int J Numer Methods Eng.* 2021;1-28. <https://doi.org/10.1002/nme.6652>.
36. Giovannelli L, Ródenas JJ, Navarro-Jimenez JM, Tur M. Element stiffness matrix integration in image-based Cartesian grid finite element method. In: Zhang YJ, Tavares JMRS, eds. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol 8641. Springer Verlag; 2014:304-315.
37. Giovannelli L, Ródenas JJ, Navarro-Jiménez JM, Tur M. Direct medical image-based finite element modelling for patient-specific simulation of future implants. *Finite Elem Anal Des.* 2017;136:37-57.
38. Gutiérrez-Gil J, Nadal E, León OM, González-Estrada OA, Albelda J, Rodenas JJ. Application of the Cartesian grid finite element method to image-based bone remodelling process. Proceedings of the 5th ECCOMAS Young Investigators Conference; 2019:1-2; September.
39. Marco O, Sevilla R, Zhang Y, Ródenas JJ, Tur M. Exact 3D boundary representation in finite element analysis based on Cartesian grids independent of the geometry. *Int J Numer Methods Eng.* 2015;103(6):445-468.
40. Johansson A, Larson MG. A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary. *Numer Math.* 2013;123(4):607-628.
41. Sollie WEH, Bokhove O, van der Vegt JJW. Space-time discontinuous Galerkin finite element method for two-fluid flows. *J Comput Phys.* 2011;230(3):789-817.
42. Heimann F, Engwer C, Ippisch O, Bastian P. An unfitted interior penalty discontinuous Galerkin method for incompressible Navier-Stokes two-phase flow. *Int J Numer Methods Fluids.* 2013;71(3):269-293.
43. Xiao H, Febrianto E, Zhang Q, Cirak F. An immersed discontinuous Galerkin method for compressible Navier-Stokes equations on unstructured meshes. *Int J Numer Methods Fluids.* 2019;91(10):487-508.
44. Donea J, Huerta A. *Finite Element Methods for Flow Problems*. Wiley; 2003.
45. Hesthaven JS, Warburton T. *Nodal Discontinuous Galerkin Methods*. Texts in Applied Mathematics. Vol 54. Springer; 2008.
46. Di Pietro DA, Ern A. *Mathematical Aspects of Discontinuous Galerkin Methods, volume 69 of Mathématiques et Applications*. Springer; 2012.
47. Godunov S. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik.* 1959;47(3):271-306.
48. Lax PD. Hyperbolic systems of conservation laws II. *Commun Pure Appl Math.* 1957;10(4):537-566.
49. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *J Comput Phys.* 1981;43(2):357-372.
50. Harten A, Lax PD, van Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Rev.* 1983;25(1):35-61.
51. Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press; 2002.
52. Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer-Verlag Berlin Heidelberg; 2009.
53. Dan Givoli. *Numerical Methods for Problems in Infinite Domains*. Elsevier Science Ltd; 1956.
54. Berenger JP. A perfectly matched layer for the absorption of electromagnetic waves. *J Comput Phys.* 1994;114(2):185-200.
55. Jin J-M, Chew WC. Combining PML and ABC for the finite-element analysis of scattering problems. *Microw Opt Technol Lett.* 1996;12(4):192-197.
56. Bonnet F, Poupaud F. Berenger absorbing boundary condition with time finite-volume scheme for triangular meshes. *Appl Numer Math.* 1997;25(4):333-354.
57. Kabakian AV, Shankar V, Hall WF. Unstructured grid-based discontinuous Galerkin method for broadband electromagnetic simulations. *J Sci Comput.* 2004;20(3):405-431.
58. Sommariva A, Vianello M. Computing approximate Fekete points by QR factorizations of Vandermonde matrices. *Comput Math Appl.* 2009;57(8):1324-1336.
59. Bos L, De Marchi S, Sommariva A, Vianello M. Computing multivariate Fekete and Leja points by numerical linear algebra. *SIAM J Numer Anal.* 2010;48(5):1984-1999.
60. Hansbo A, Hansbo P. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Comput Methods Appl Mech Eng.* 2004;193(33-35):3523-3540.
61. Erik Burman, Susanne Claus, Peter Hansbo, Mats G. Larson, and André Massing. CutFEM: Discretizing geometry and partial differential equations. *Int J Numer Methods Eng.* 104(7):472-501, 2015.
62. Peterson TE. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J Numer Anal.* 1991;28(1):133-140.
63. LeSaint P, Raviart PA. On a finite element method for solving the neutron transport equation. In: de Boor C, ed. *Mathematical Aspects of Finite Elements in Partial Differential Equations*. Elsevier; 1974:89-123.

64. Richter GR. An optimal-order error estimate for the discontinuous Galerkin method. *Math Comput.* 1988;50(181):75.
65. Penrose R. A generalized inverse for matrices. *Math Proc Camb Philos Soc.* 1955;51(3):406-413.
66. Balanis CA. *Advanced Engineering Electromagnetics.* Wiley; 1989.

**How to cite this article:** Navarro-García H, Sevilla R, Nadal E, Ródenas JJ. High-order discontinuous Galerkin method for time-domain electromagnetics on geometry-independent Cartesian meshes. *Int J Numer Methods Eng.* 2021;1-32. doi: 10.1002/nme.6846