An automatic PML for acoustic finite element simulations in convex domains of general shape

Hadrien Beriot, Axel Modave

An automatic PML for acoustic finite element simulations with generally-shaped convex domains

H. Bériot[1] and A. Modave[2]

[1]Siemens Industry Software N.V., Interleuvenlaan 68, Researchpark Z1, Leuven (Belgium), hadrien.beriot@siemens.com
[2]POEMS, CNRS, Inria, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau (France), axel.modave@ensta-paris.fr

Abstract

We address the efficient finite element solution of exterior acoustic problems with truncated computational domains surrounded by perfectly matched layers (PMLs). The PML is a popular non-reflecting technique, which combines accuracy, computational efficiency and geometric flexibility. Unfortunately, the effective implementation of the PML for generally-shaped convex domains is tricky because of the geometric parameters that are required to define the PML medium. In this work, a comprehensive implementation strategy is proposed. This approach, which we called the automatically matched layer (AML) implementation, is versatile and fully automatic for the end-user. With the AML approach, the mesh of the layer is extruded, the required geometric parameters are automatically obtained during the extrusion step, and the effective implementation relies on a simple modification of the Jacobian matrix in the element-wise integrals. The AML implementation is validated and compared to other implementation strategies using numerical benchmarks in two and three dimensions, considering computational domains with regular and non-regular boundaries. A three-dimensional application with a generally-shaped domain generated using a convex hull is proposed to illustrate the interest of the AML approach for realistic industrial cases.

1 Introduction

Finite element methods (FEMs) are widely used in academia and industry to simulate acoustic wave propagation phenomena. With these methods, high-frequency oscillatory fields can be represented accurately in realistic settings with complicated geometries thanks to unstructured meshes, curved elements, high-order basis functions and $hp-$adaptivity [48]. Nevertheless, for many applications occuring in the free space, the computational domain must be truncated, and a specific non-reflecting technique must be used at the artifical boundary of the finite element mesh to represent the outward propagation of waves. Many approaches have been proposed and studied over the time. The basic impedance boundary condition is the simplest and cheapest approach, but it provides a rather poor accuracy. High-order absorbing boundary conditions (*e.g.* [1, 3, 19, 27, 47, 34, 36]), infinite elements (*e.g.* [17, 21, 25]) and perfectly matched layers (*e.g.* [5, 38, 9, 26, 49]) constitute families of more accurate techniques. In particular, the perfectly matched layers (PMLs) are easy to use and accurate for a limited computational cost, which makes them attractive for large-scale industrial simulations. They are frequently used in the aeroacoustic community [4, 32].

With the PML technique, the computational domain is surrounded with an artificial layer where the outgoing waves are damped. The governing equations are modified in such a way that any outgoing wave is perfectly transmitted from the domain to the layer, whatever the angle of incidence, which is the key property of this approach. The PML has been introduced by Bérenger [5] for time-dependent simulations of electromagnetic waves with rectangular truncated domains. It has been rapidly applied to other physical waves and to time-harmonic problems. In the frequency domain, the PML equations are simply obtained by performing a complex coordinate stretch in the governing equations [12, 40], which has been reinterpreted as a modification of the metric tensor [46, 31] and as a modification of the material properties [20, 41]. Mathematical analyses of the Helmholtz problem with a PML have been proposed in

[8, 10, 14, 28, 30, 31]. The selection of the absorbing function of the PML, which is critical for practical applications, has been studied in [9, 13, 15, 35]. In particular, it has been shown in [9, 13, 35] that specific unbounded absorbing functions provide high-fidelity solutions without requiring parameters' tuning, in contrast with commonly-used polynomial functions, for which some free parameters must be adjusted.

Most of the PML formulations have been derived for truncated domains with simple shapes, such as cuboids, spheres and cylinders. With the standard approach, the complex coordinate stretch is performed in a specific coordinate system, *i.e.* the Cartesian, spherical or cylindrical coordinate system, respectively (see *e.g.* [14, 44, 45]). Nevertheless, in many applications, an unnecessarily large free-space region must be discretized between the PML and the object under investigation should the truncated domain have one of these shapes. In order to improve the geometric flexibility of the PML, Teixeira and Chew [44, 45] proposed a *conformal* PML that is suited to convex truncated domains with regular boundary. They used a specific local curvilinear coordinate system based on a local parametrization of the boundary, with normal and tangent coordinates. The conformal PML was studied for the Helmholtz equation in [31]. The approach has been applied to time-domain simulations with a discontinuous Galerkin scheme [37] and to time-harmonic simulations with Bernstein-Bézier finite elements [18].

Unfortunately, the conformal PML requires the exact knowledge of some geometric information that may not be available in practical cases, which limits the applicability of the approach. In three dimensions, the main curvatures, the principal directions and the normal of the surface between the domain and the layer must be known, as well as the distance between any point of the layer and that surface. These data are easily derived for simple geometries, such as spherical and ellipsoidal surfaces, but it is not the case for more complicated surfaces. The data could be estimated *a priori* by computational procedures, but it would be at the price of additional processing steps. The conformal PML is *a priori* not suited to polyhedric domains with non-regular boundary, which is another drawback of the approach. Indeed, in some applicative cases, only polyhedric meshed domains are available.

To address generally-shaped truncated domains, few alternative PML realizations have been investigated. Zschiedrich *et al.* [50] proposed a two-dimensional PML implementation for polygonal domains where the complex coordinate stretch is performed in a local prismatoidal coordinate system. In [38, 39], Ozgun and Kuzuoglu proposed a *locally-conformal* PML finite-element implementation suited to convex domains of any shape. With their approach, the distance to the interface is estimated by solving a minimization problem, and the coordinate stretch is performed directly on the coordinates of the mesh nodes. That approach is very flexible and easy to implement, since it does not require further geometric data or any modification of the equations. However, the practical realization must be carefully handled, because it implies involved numerical approximations that can generate local instabilities [7]. To alleviate that issue, an alternative estimate procedure for the distance function has been proposed in [7].

In this work, we propose a novel finite element PML approach — which we called the *automatically matched layer* (AML) implementation — suited to generally-shaped convex domains with regular and non-regular boundaries. For convex domains with regular boundaries, this approach can be considered as a specific finite element implementation of the conformal PML that requires less geometric data than the original formulation. With the AML approach, the layer is generated at solver level by extruding the surface mesh of the domain boundary. The extrusion provides all the geometric data that are required, avoiding the need for supplementary pre-processing steps or costly optimization procedures. Then, the coordinate stretch is embedded into the finite element scheme through a rather simple modification of the Jacobian transformation on the reference element in the element-wise integrals. The accuracy of the obtained layer is improved in comparison with the approach based on complexified mesh nodes. In particular, unbounded absorbing functions, which are proved to be very efficient [9, 13, 35], can be used with the AML approach, in contrast to the approach of Ozgun and Kuzuoglu. Therefore, the whole strategy avoids some issues of the existing approaches while retaining the ease of implementation in existing codes and the accuracy by the use of unbounded absorbing functions.

This article is structured as follows. In section 2, the Helmholtz problem and the classical conformal PML are introduced, together with a description of the coordinate system associated to the interface between the domain and the layer. The AML approach, with the mesh extrusion and the PML implementation, is explained in section 3. In particuler, the links and the differences with the existing approaches are discussed. In sections 4 and 5, numerical results are proposed to validate the approach and to illustrate the use on a realistic 3D application. Finally, a conclusion is proposed in section 6.

Geometric data for reference benchmarks are provided in appendix A.

Notation. Scalars are denoted with italic lower-case letters (*e.g.* k and r). Vectors are denoted with italic bold lower-case letters (*e.g.* $\boldsymbol{x}$ and $\boldsymbol{n}$). Column matrices with the components of vectors in given coordinate systems are denoted with bold lowercase letters (*e.g.* $\mathbf{x} = [x_1\ x_2\ x_3]^\top$ and $\mathbf{n} = [n_1\ n_2\ n_3]^\top$). Matrices are denotes with bold uppercase letters (*e.g.* $\mathbf{J}$).

2 Conformal PML for the Helmholtz equation

We consider a general time-harmonic wave propagation problem defined on the unbounded domain $\mathbb{R}^d$, with the dimension $d = 2, 3$. The field $u(\boldsymbol{x})$ is the solution of the Helmholtz equation

$$-\Delta u - k^2 u = f(\boldsymbol{x}), \quad \text{in } \mathbb{R}^d, \tag{1}$$

with the Sommerfeld boundary condition at infinity,

$$\lim_{\|\boldsymbol{x}\|\to\infty} \|\boldsymbol{x}\|^{(d-1)/2} \left(\frac{\partial u}{\partial \|\boldsymbol{x}\|} - \imath k u \right) = 0, \tag{2}$$

where k is the (constant) wavenumber an $f(\boldsymbol{x})$ is a source term with compact support on the region $\Omega_{\text{dom}} \subset \mathbb{R}^d$. For finite element simulations, the domain Ω_{dom} is meshed and extended with a PML Ω_{pml} to represent the propagation of waves towards the infinity. The interface between the domain and the layer is noted Γ_{int}. The exterior border of the layer is noted Γ_{ext}. We take the convention that the time-dependence of the fields is $e^{-\imath\omega t}$, where ω is the angular frequency and t is the time.

The PML equations are obtained from the original equations by stretching a spatial coordinate in the complex plan, which introduces a directional damping of waves in the layer. For rectangular and cuboid domains, the stretching is performed in the Cartesian coordinate system. For circular and spherical domains, the radial coordinates are stretched in the corresponding coordinate systems [14]. This approach has also been used with elliptical and ellipsoidal domains [11]. It is however rather complicated to apply it for domains with more general shapes, since it require the use of a global coordinate system associated to the shape of the interface Γ_{int}. With the conformal PML [29, 31, 44, 45], the coordinate stretch is performed in a curvilinear coordinate system. This system is based on surfaces parallel to the interface Γ_{int}. The layer then has a constant thickness δ, and the exterior border and the interface are parallel. This approach is more flexible and exhibit some features which simplify the practical finite element implementation.

The coordinate stretch used to derive the conformal PML is described in section 2.1, together with the curvilinear coordinate system. The PML equation for the Helmholtz problem is derived in section 2.2. The variational formulation is given in section 2.3. Only the three-dimensional case is described, but it can be straightforwardly adapted to the two-dimensional case.

2.1 Coordinate stretch and curvilinear coordinates associated to Γ_{int}

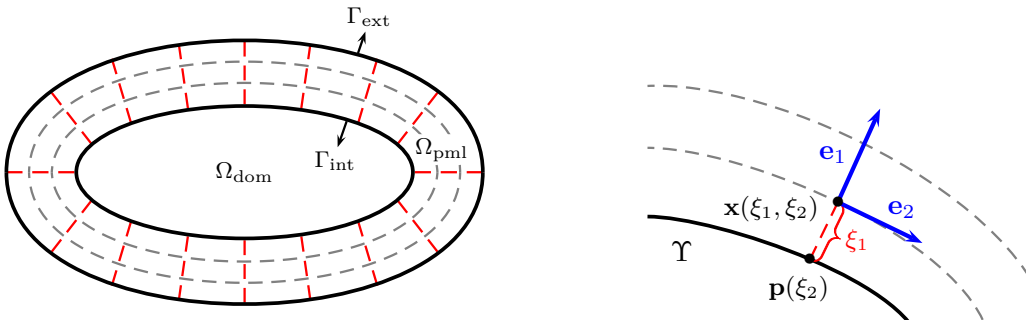
The PML equation will be derived by performing a complex coordinate stretch along the normal to the interface Γ_{int} between the domain Ω_{dom} and the layer Ω_{pml}. The domain is assumed to be convex and the interface is assumed to be sufficiently regular.

For each point $\boldsymbol{x}$ of the layer Ω_{pml}, we consider the closest point $\boldsymbol{p}$ belonging to the interface Γ_{int}. The distance between $\boldsymbol{x}$ and the interface Γ_{int} is given by $r := \text{dist}(\boldsymbol{x}, \Gamma_{\text{int}}) = \|\boldsymbol{x} - \boldsymbol{p}\| \in]0, \delta[$. The point $\boldsymbol{x}$ can then be represented in a unique way as

$$\boldsymbol{x}(r, \boldsymbol{p}) = \boldsymbol{p} + r\boldsymbol{n}(\boldsymbol{p}), \tag{3}$$

where $\boldsymbol{n}(\boldsymbol{p})$ is the unit normal of Γ_{int} at $\boldsymbol{p}$ pointing towards the exterior of Ω_{dom}. The *complex coordinate stretch* then consists in replacing the real coordinate r in equation (3) with the complex function $\tilde{r}(r)$ defined as

$$\tilde{r}(r) := r - \frac{1}{\imath k} f(r), \quad \text{with } f(r) := \int_0^r \sigma(r')\, dr', \quad \text{for } r \in]0, \delta[, \tag{4}$$



(a) Iso-value coordinate curves

(b) Orthogonal frame

Figure 1: Curvilinear coordinates and local frame associated with the boundary Γ_{int} in two dimensions. The curves of iso-value coordinates are represented in figure (a). Gray curves are parallel. Red lines are straight and perpendicular to Γ_{int}. Figure (b) shows the local frame and the radial coordinate ξ_1.

where $\sigma(r) \geq 0$ is the so-called *absorption function*. Because of the coordinate stretch, any outgoing wave is damped in the normal direction $\boldsymbol{n}(\boldsymbol{p})$, *i.e.*

$$\exp\left(\imath \boldsymbol{k} \cdot \boldsymbol{x}\right) \quad \overset{\text{stretch}}{\longrightarrow} \quad \exp\left(\imath \boldsymbol{k} \cdot \tilde{\boldsymbol{x}}\right) = \exp\left(\imath \boldsymbol{k} \cdot \boldsymbol{x}\right) \exp\left[-\cos(\theta) f(r)\right], \tag{5}$$

with $\tilde{\boldsymbol{x}}(r, \boldsymbol{p}) := \boldsymbol{p} + \tilde{r}(r)\boldsymbol{n}(\boldsymbol{p})$ and $\cos\theta = \boldsymbol{n} \cdot \boldsymbol{k}/k$.

The PML equation will be derived by performing the coordinate stretch in a specific curvilinear coordinate system (ξ_1, ξ_2, ξ_3) associated to the interface Γ_{int}. For each point $\boldsymbol{x}(\xi_1, \xi_2, \xi_3)$ of the layer, the coordinate $\xi_1 = r$ is the distance to the closest point belonging to Γ_{int}, and the coordinates ξ_2 and ξ_3 are provided by a local parametrization of Γ_{int}. We can then rewrite equation (3) as

$$\boldsymbol{x}(\xi_1, \xi_2, \xi_3) = \boldsymbol{p}(\xi_2, \xi_3) + \xi_1 \boldsymbol{n}(\xi_2, \xi_3) \tag{6}$$

and equation (4) as

$$\tilde{\xi}_1(\xi_1) := \xi_1 - \frac{1}{\imath k} f(\xi_1). \tag{7}$$

We introduce two unit tangent vectors $\boldsymbol{t}_2(\xi_2, \xi_3)$ and $\boldsymbol{t}_3(\xi_2, \xi_3)$ that satisfy

$$\frac{\partial \boldsymbol{n}}{\partial \xi_i} = \kappa_i \boldsymbol{t}_i, \quad \text{with } \boldsymbol{t}_i = \frac{\partial \boldsymbol{p}}{\partial \xi_i}, \quad \text{for } i = 2, 3, \tag{8}$$

where $\kappa_2(\xi_2, \xi_3)$ and $\kappa_3(\xi_2, \xi_3)$ are the principal curvatures of the surface Γ_{int} at $\boldsymbol{p}(\xi_2, \xi_3)$. The set of coordinates (ξ_1, ξ_2, ξ_3) forms an orthogonal curvilinear coordinate system, and the set of vectors $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3) := (\boldsymbol{n}, \boldsymbol{t}_2, \boldsymbol{t}_3)$ constitutes an orthonormal frame. The two-dimensional version of the coordinate system is illustrated in figure 1.

The Jacobian of the transformation from the curvilinear coordinates (ξ_1, ξ_2, ξ_3) to the Cartesian coordinates (x_1, x_2, x_3) is obtained by deriving equation (6) with respect to ξ_i, which gives

$$\frac{\partial \boldsymbol{x}}{\partial \xi_i} = h_i \boldsymbol{e}_i, \quad \text{for } i = 1 \ldots 3, \tag{9}$$

with the *scale factors* defined as $h_i(\xi_1, \xi_2, \xi_3) := \|\partial \boldsymbol{x}/\partial \xi_i\|$. For the system considered here, the scale factors are (see *e.g.* [44])

$$h_1 = 1, \tag{10}$$

$$h_i = 1 + \kappa_i(\xi_2, \xi_3)\, \xi_1, \quad \text{for } i = 2, 3. \tag{11}$$

Writing equation (9) in Cartesian coordinates gives the Jacobian matrix

$$\mathbf{J}_{x/\xi} := \frac{\partial(x_1, x_2, x_3)}{\partial(\xi_1, \xi_2, \xi_3)} = \begin{bmatrix} h_1 \mathbf{e}_1 & h_2 \mathbf{e}_2 & h_3 \mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & (1 + \kappa_2 \xi_1)\mathbf{t}_2 & (1 + \kappa_3 \xi_1)\mathbf{t}_3 \end{bmatrix}, \tag{12}$$

where $\mathbf{J}_{x/\xi}$ is a 3×3 matrix and $\mathbf{e}_i$, $\mathbf{n}$ and $\mathbf{t}_i$ are column matrices with the Cartesian components of $\boldsymbol{e}_i$, $\boldsymbol{n}$ and $\boldsymbol{t}_i$, respectively.

2.2 Derivation of the PML equation

The PML equation is straightforwardly obtained in the curvilinear coordinate system by writing explicitly the Helmholtz equation in curvilinear coordinates and by replacing ξ_1 with $\tilde{\xi}_1(\xi_1)$ in the obtained equation. Nevertheless, for the practical implementation in standard finite element codes, it is convenient to have the PML equation in Cartesian coordinates.

In [33], Matuszyk and Demkowicz described an elegant method to derive the PML equation and the variational formulation of the problem in Cartesian coordinates, when the complex stretching is performed in another coordinate system. Following their approach, the PML equation reads

$$-\nabla_x \cdot (\mathbf{\Lambda}_{\text{pml}} \nabla_x u) - \alpha_{\text{pml}}\, k^2 u = 0, \tag{13}$$

with the operator ∇_x, the matrix $\mathbf{\Lambda}_{\text{pml}}$ and the scalar α_{pml} defined as

$$\nabla_x := \begin{bmatrix} \partial_{x_1} & \partial_{x_2} & \partial_{x_3} \end{bmatrix}^\top, \qquad \mathbf{\Lambda}_{\text{pml}} := (\det \mathbf{J}_{\text{pml}})\, \mathbf{J}_{\text{pml}}^{-1} \mathbf{J}_{\text{pml}}^{-\top}, \qquad \alpha_{\text{pml}} := \det \mathbf{J}_{\text{pml}}, \tag{14}$$

where the matrix $\mathbf{J}_{\text{pml}}$ is the Jacobian of the transformation from the Cartesian coordinates (x_1, x_2, x_3) to the complexified Cartesian coordinates $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ (*i.e.* the Cartesian components of $\tilde{\boldsymbol{x}}$) defined as

$$\mathbf{J}_{\text{pml}} := \frac{\partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)}{\partial(x_1, x_2, x_3)}. \tag{15}$$

It is well known that the PML equation (13) can be interpreted as an anisotropic absorber with specific complex material properties (see *e.g.* some references in the electromagnetic community [20, 41, 44]). The PML can also be interpreted as a space with a complex metric tensor. This point of view is discussed by Teixeira and Chew in [45].

In the remainder of this section, we derive an explicit form for the Jacobian matrix (15). Using the chain rule, this matrix is factorized as

$$\mathbf{J}_{\text{pml}} = \frac{\partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)}{\partial(\tilde{\xi}_1, \tilde{\xi}_2, \tilde{\xi}_3)} \frac{\partial(\tilde{\xi}_1, \tilde{\xi}_2, \tilde{\xi}_3)}{\partial(\xi_1, \xi_2, \xi_3)} \frac{\partial(\xi_1, \xi_2, \xi_3)}{\partial(x_1, x_2, x_3)}. \tag{16}$$

This factorization highlights successive changes of variables: the transformation from Cartesian coordinates to curvilinear coordinates in the complex space, the use of the complex stretch to come back in the real space, and the transformation from curvilinear coordinates to Cartesian coordinates in the real space.

- The first matrix corresponds to $\mathbf{J}_{x/\xi}$, defined in equation (12), where the coordinate ξ_1 is replaced with $\tilde{\xi}_1(\xi_1)$. In $\mathbf{J}_{x/\xi}$, only the scale factors h_2 and h_3 depend on ξ_1. Using equations (7) and (11), we can write

$$\begin{aligned} h_i(\tilde{\xi}_1, \xi_2, \xi_3) &= 1 + \kappa_i(\xi_2, \xi_3)\, \tilde{\xi}_1(\xi_1) && (17)\\ &= 1 + \kappa_i(\xi_2, \xi_3) \left(\xi_1 - \frac{1}{\imath k} f(\xi_1) \right) && (18)\\ &= h_i(\xi_1, \xi_2, \xi_3)\, s_i(\xi_1, \xi_2, \xi_3), \quad \text{for } i = 2, 3, && (19) \end{aligned}$$

  where we have introduced the so-called *stretching functions*

$$s_i(\xi_1, \xi_2, \xi_3) := 1 - \frac{\kappa_i(\xi_2, \xi_3)}{\imath k\, h_i(\xi_1, \xi_2, \xi_3)} f(\xi_1), \quad \text{for } i = 2, 3. \tag{20}$$

  The first matrix can then be written as

$$\frac{\partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)}{\partial(\tilde{\xi}_1, \tilde{\xi}_2, \tilde{\xi}_3)} = \begin{bmatrix} h_1 \mathbf{e}_1 & s_2 h_2 \mathbf{e}_2 & s_3 h_3 \mathbf{e}_3 \end{bmatrix}. \tag{21}$$

- The second matrix is the Jacobian of the transformation from the real curvilinear coordinates to the stretched curvilinear coordinates. Because $\tilde{\xi}_1 = \tilde{\xi}_1(\xi_1)$ (equation (7)), $\tilde{\xi}_2 = \xi_2$ and $\tilde{\xi}_3 = \xi_3$, we have

$$\frac{\partial(\tilde{\xi}_1, \tilde{\xi}_2, \tilde{\xi}_3)}{\partial(\xi_1, \xi_2, \xi_3)} = \mathrm{diag}(s_1, 1, 1), \tag{22}$$

  where we have introduced the supplementary *stretching function*

$$s_1(\xi_1) := \frac{d\tilde{\xi}_1}{d\xi_1} = 1 - \frac{1}{\imath k}\sigma(\xi_1). \tag{23}$$

- The last matrix is the inverse of the Jacobian $\mathbf{J}_{x/\xi}$, which reads

$$\frac{\partial(\xi_1, \xi_2, \xi_3)}{\partial(x_1, x_2, x_3)} = \left[\, h_1^{-1}\mathbf{e}_1 \;\; h_2^{-1}\mathbf{e}_2 \;\; h_3^{-1}\mathbf{e}_3 \,\right]^\top. \tag{24}$$

Finally, using equation (21), (22) and (24) in equation (16) gives

$$\mathbf{J}_{\mathrm{pml}} = \mathbf{I} - \frac{1}{\imath k}\left(\sigma(\xi_1)\,\mathbf{n}\mathbf{n}^\top + \frac{\kappa_2}{h_2} f(\xi_1)\,\mathbf{t}_2\mathbf{t}_2^\top + \frac{\kappa_3}{h_3} f(\xi_1)\,\mathbf{t}_3\mathbf{t}_3^\top\right). \tag{25}$$

This matrix verifies

$$\mathbf{J}_{\mathrm{pml}} = \sum_{i=1}^{3} s_i \mathbf{e}_i\mathbf{e}_i^\top, \qquad \mathbf{J}_{\mathrm{pml}}^{-1} = \sum_{i=1}^{3} \frac{1}{s_i}\mathbf{e}_i\mathbf{e}_i^\top, \qquad \det \mathbf{J}_{\mathrm{pml}} = s_1 s_2 s_3. \tag{26}$$

It contains all the information relative to the geometry and the complex stretch.

2.3 Variational formulation

The truncated problem consists in solving the Helmholtz equation (1) in the domain Ω_{dom}, the PML equation (13) in the layer Ω_{pml}, continuity conditions at the interface Γ_{int} and a boundary condition at the exterior border Γ_{ext}. In this work, we use a homogeneous Neumann condition on Γ_{ext}. The region containing both the domain and the layer is denoted Ω. It is defined as an open set that verifies $\overline{\Omega} = \overline{\Omega}_{\mathrm{dom}} \cup \overline{\Omega}_{\mathrm{pml}}$.

Since the PML equation is a Helmholtz-type equation, the variational formulation of the problem is obtained rather immediately with the standard approach. It reads

$$\begin{aligned} &\text{Find } u \in H^1(\Omega) \text{ such that, } \forall v \in H^1(\Omega), \\ &\quad \int_{\Omega_{\mathrm{dom}}} \left[\nabla_x u \cdot \nabla_x v - k^2 uv\right] d\Omega \\ &\qquad + \int_{\Omega_{\mathrm{pml}}} \left[(\boldsymbol{\Lambda}_{\mathrm{pml}} \nabla_x u) \cdot \nabla_x v - \alpha_{\mathrm{pml}}\, k^2 uv\right] d\Omega = \int_{\Omega_{\mathrm{dom}}} fv\, d\Omega. \end{aligned} \tag{27}$$

Using the definitions of $\boldsymbol{\Lambda}_{\mathrm{pml}}$ and α_{pml}, the formulation can be rewritten as

$$\begin{aligned} &\text{Find } u \in H^1(\Omega) \text{ such that, } \forall v \in H^1(\Omega), \\ &\quad \int_{\Omega_{\mathrm{dom}}} \left[\nabla_x u \cdot \nabla_x v - k^2 uv\right] d\Omega \\ &\qquad + \int_{\Omega_{\mathrm{pml}}} \left[(\mathbf{J}_{\mathrm{pml}}^{-T} \nabla_x u) \cdot (\mathbf{J}_{\mathrm{pml}}^{-T} \nabla_x v) - k^2 uv\right] (\det \mathbf{J}_{\mathrm{pml}})\, d\Omega = \int_{\Omega_{\mathrm{dom}}} fv\, d\Omega. \end{aligned} \tag{28}$$

This second formulation could also be obtained by performing the complex coordinate stretch and the change of variables directly on the classical bilinear form of the Helmholtz equation [33]. It highlights the important role of the Jacobian matrix $\mathbf{J}_{\mathrm{pml}}$.

3 An automatic finite element implementation of the PML

The variational formulations (equations (27) and (28)) can be straightforwardly discretized with H^1-conforming finite elements by using the standard approach. Nevertheless, the direct implementation of the obtained finite element schemes for generally-shaped computational domains is more difficult because of the geometric data that are required.

For a direct implementation, the material properties ($\boldsymbol{\Lambda}_{\text{pml}}$ and α_{pml}) or, equivalently, the Jacobian matrix $\mathbf{J}_{\text{pml}}$ must be known at every point of the layer. They depend on the distance r, the normal direction $\mathbf{n}$, the tangential directions $\mathbf{t}_2$ and $\mathbf{t}_3$, and the curvatures κ_2 and κ_3. Unfortunately, these geometric data are available only for simple geometries, such as circular, elliptical or spherical domains. For more general domains, they could be provided by CAD tools or by numerical procedures (see *e.g.* [36]) at the price of additional preprocessing steps. Another issue is related to the approximation of the geometry by the mesh. Even with curvilinear elements, it could be impossible to align the finite element mesh with the interface Γ_{int} and the exterior border Γ_{ext}. A typical example is a circular domain meshed with straight-sided elements, which leads to a polygonal meshed interface. The definition of the geometric parameters then is ambiguous: should the distance r start at the meshed interface or at the circular interface? Finally, when only a mesh of the domain is available, without a reference geometry, we possibly have to address polyhedral boundaries, which falls *a priori* outside the scope of application of the conformal PML presented in the previous section.

With the proposed AML implementation, only the meshed interface is required. As in the previous section, the computational domain is assumed to be convex, but the meshed interface could be irregular. Our approach is based on a mesh extrusion generating the layer, and a local curvilinear coordinate system (η_1, η_2, η_3) associated to the meshed interface $\Gamma_{\text{int},h}$. By connecting directly the local coordinates (η_1, η_2, η_3) to the reference coordinates (u_1, u_2, u_3) used to compute the element-wise integrals, the issue of geometric data requirements may be fully alleviated.

The mesh extrusion, the reference coordinate system and the standard finite element scheme are described in section 3.1. The coordinate stretch and the curvilinear coordinates associated to the meshed interface $\Gamma_{\text{int},h}$ are discussed in section 3.2. The effective computation of the element-wise integrals for elements in the layer is discussed in section 3.3.

3.1 Mesh extrusion and geometric data

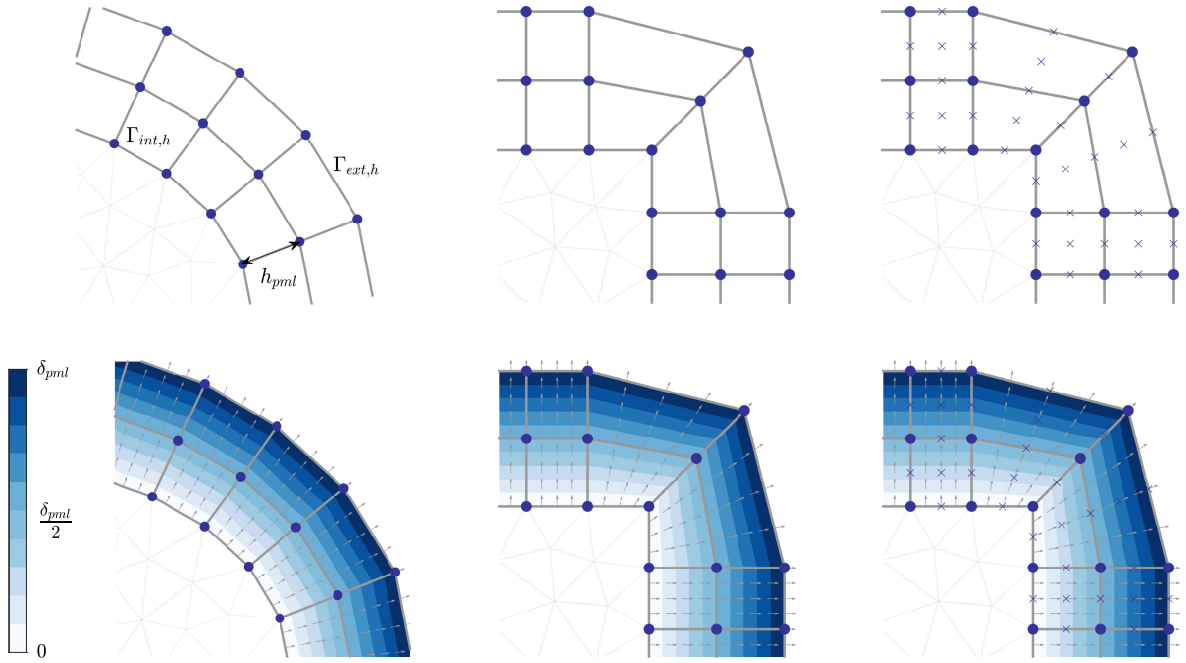
We consider a general mesh for the convex domain $\Omega_{\text{dom},h}$. The nature of the elements inside the domain $\Omega_{\text{dom},h}$ does not matter to present our approach, but we assume that the restriction of these elements on its exterior boundary, denoted $\Gamma_{\text{int},h}$, gives a conformal surface mesh made of linear straight elements ($N_{\text{geo}} = 1$) and quadratic curvilinear elements ($N_{\text{geo}} = 2$). Our approach could be extended to curvilinear elements of higher order or NURBS-enhanced methods, but this goes beyond the scope of this paper.

To generate the mesh of a layer, $\Omega_{\text{pml},h}$, the vertices and the second-order nodes belonging to the surface mesh $\Gamma_{\text{int},h}$ are extruded along a direction $\boldsymbol{n}_h$ which should correspond to the exterior normal. Nevertheless, the definition of the exterior normal of $\Gamma_{\text{int},h}$ is ambiguous at the points where the surface is not regular, which is the case, for instance, at the vertices and the edges of polyhedral surfaces. The extrusion direction $\boldsymbol{n}_h$ is then chosen following the empirical rules:

- for a given vertex of the surface, $\boldsymbol{n}_h$ is the average of the normal of the elements touching the vertex;
- for a second-order node defined in the middle of an edge or a face, $\boldsymbol{n}_h$ is the average of the extrusion directions defined on the vertices touching the edge or the face.

The two-dimensional version of this strategy is illustrated in figure 2. It is performed N_{pml} times with a constant extrusion distance h_{pml}. The obtained layer is structured and made of quadrangular elements (in two dimensions) or prismatic and hexahedral elements (in three dimensions) and has a thickness of $\delta_{\text{pml}} = N_{\text{pml}} h_{\text{pml}}$.

For each point $\boldsymbol{x}_h \subset \Omega_{\text{pml},h}$, we shall need values for the extrusion direction $\boldsymbol{n}_h$, for the closest point belonging to the interface $\boldsymbol{p}_h \subset \Gamma_{\text{int},h}$ and for the distance to the interface r_h. During the mesh extrusion,



(a) Case 1 (b) Case 2 (c) Case 3

Figure 2: Mesh extrusion for three cases in two dimensions with linear elements (cases (a) and (b)) and quadratic curvilinear elements (case (c)). The mesh vertices and the second-order nodes are represented. The mesh extrusion is performed twice here ($N_{\text{pml}} = 2$). The interpolated distance field $r_h \in [0, \delta_{\text{pml}}]$ and extrusion directions $\boldsymbol{n}_h$ are also shown.

these geometric data can be recorded at the nodes. Indeed, for each extruded node, $\boldsymbol{n}_h$ and $\boldsymbol{p}_h$ are the same as for the origin node belonging to the interface. For a node generated at the n^{th} extrusion, the distance r_h is equal to nh_{pml}. At every node, we have $\boldsymbol{x}_h = \boldsymbol{p}_h + r_h \boldsymbol{n}_h$.

To get values for the geometric data inside the elements we use a polynomial extrapolation. Let us consider a hexahedral element D^e of the layer. The reference hexahedron is defined as

$$\mathsf{D}_{\text{ref}} := \{(u_1, u_2, u_3) : u_i \in [-1, 1], i = 1 \dots 3\}, \tag{29}$$

where (u_1, u_2, u_3) are the reference coordinates. In the reference coordinate system, the polynomial representation of the position vector reads

$$\boldsymbol{x}^e(u_1, u_2, u_3) = \sum_{n_1=1}^{N_{\text{geo}}+1} \sum_{n_2=1}^{N_{\text{geo}}+1} \sum_{n_3=1}^{N_{\text{geo}}+1} \ell_{n_1}(u_1)\, \ell_{n_2}(u_2)\, \ell_{n_3}(u_3)\, \boldsymbol{x}^e_{n_1,n_2,n_3}, \tag{30}$$

where $\{\ell_n\}_{n=1}^{N_{\text{geo}}+1}$ are the $N^{\text{th}}_{\text{geo}}$-order Lagrange functions and $\boldsymbol{x}^e_{n_1,n_2,n_3}$ is the mesh node of the physical hexahedron D^e corresponding to the set of indices (n_1, n_2, n_3). Similarly, we can obtain the extrusion direction $\boldsymbol{n}^e$, the closest point $\boldsymbol{p}^e$ and the numerical distance r^e at every point of the element D^e by extrapolating the data recorded at the nodes. The extrusion direction and the closest points do not vary in the stretching direction, while the numerical distance varies only in that direction. In addition, the numerical distance varies linearly. If the reference coordinate u_1 is aligned with the stretching direction, we can then write

$$\boldsymbol{x}^e(u_1, u_2, u_3) = \boldsymbol{p}^e(u_2, u_3) + r^e(u_1)\, \boldsymbol{n}^e(u_2, u_3) \tag{31}$$

with

$$r^e(u_1) = (n-1)h_{\text{pml}} + \frac{u_1 + 1}{2} h_{\text{pml}}, \tag{32}$$

if the element D^e has been generated at the n^{th} extrusion, with $n = 1 \dots N_{\text{pml}}$. Figure 2 presents the numerical distance field obtained using this strategy in two dimensions.

3.2 Coordinate stretch and curvilinear coordinates associated to $\Gamma_{\text{int},h}$

Similarly to the approach of section 2.1, a complex coordinate stretch is performed along the extrusion direction $\boldsymbol{n}_h$. The main differences here are that the interface $\Gamma_{\text{int},h}$ can be irregular and the extrusion direction $\boldsymbol{n}_h$ is an interpolated field.

To perform the complex coordinate stretch in the extrusion direction, we introduce a local curvilinear coordinate system $(\eta_1^e, \eta_2^e, \eta_3^e)$ associated to the interface $\Gamma_{\text{int},h}$ and based on the reference coordinate system. The first coordinate is the numerical distance. The second and third curvilinear coordinates are the second and third reference coordinates, which constitute a local parametrization of the interface. We then have $\eta_1^e(u_1) = r^e(u_1)$, $\eta_2^e(u_2) = u_2$ and $\eta_3^e(u_3) = u_3$. The position vector can be rewritten with a dependence on the curvilinear coordinates,

$$\boldsymbol{x}^e(\eta_1^e, \eta_2^e, \eta_3^e) = \boldsymbol{p}^e(\eta_2^e, \eta_3^e) + \eta_1^e \boldsymbol{n}^e(\eta_2^e, \eta_3^e). \tag{33}$$

Replacing the coordinate η_1^e with the stretched coordinate $\tilde{\eta}_1^e(\eta_1^e) = \eta_1^e - f(\eta_1^e)/\imath k$ leads to the complex vector

$$\tilde{\boldsymbol{x}}^e(\eta_1^e, \eta_2^e, \eta_3^e) := \boldsymbol{x}^e(\tilde{\eta}_1^e, \eta_2^e, \eta_3^e) = \boldsymbol{x}^e(\eta_1^e, \eta_2^e, \eta_3^e) - \frac{1}{\imath k} f(\eta_1^e) \boldsymbol{n}^e(\eta_2^e, \eta_3^e). \tag{34}$$

Thanks to these formula, we can obtain Jacobian matrices to switch between the Cartesian coordinate system and the curviliear coordinate system in the real and complex spaces.

3.3 Finite element scheme and automatic PML implementation

The variational formulation (28) is solved by using an H^1-conforming finite element method. The numerical solution u_h is constructed on the mesh with hierarchical Lobatto shape functions (see *e.g.* [6, 43]). The space of discrete solutions is denoted $V_h \subset H^1(\Omega_h)$, with $\overline{\Omega}_h = \overline{\Omega}_{\text{dom},h} \cup \overline{\Omega}_{\text{pml},h}$. Following the classical approach, the discrete variational formulation reads

$$\begin{aligned} &\text{Find } u_h \in V_h \text{ such that, } \forall v_h \in V_h, \\ &\int_{\Omega_h} \left[\nabla_x u_h \cdot \nabla_x v_h - k^2 u_h v_h \right] d\Omega_h \\ &\quad + \int_{\Omega_h} \left[(\mathbf{J}_{\text{pml},h}^{-\top} \nabla_x u_h) \cdot (\mathbf{J}_{\text{pml},h}^{-\top} \nabla_x v_h) - k^2 u_h v_h \right] (\det \mathbf{J}_{\text{pml},h})\, d\Omega_h = \int_{\Omega_{\text{dom},h}} f v_h \, d\Omega_h, \end{aligned} \tag{35}$$

where a Jacobian matrix $\mathbf{J}_{\text{pml},h} = \partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)/\partial(x_1, x_2, x_3)$ has to be defined.

We could derive explicitly $\mathbf{J}_{\text{pml},h}$ by performing successive changes of variables between the Cartesian coordinates and the curvilinear coordinates in the complex and real spaces, following the strategy used in section 2.1. Nevertheless, the computation is simplified by combining $\mathbf{J}_{\text{pml},h}$ with the Jacobian matrix of the mapping with the reference element, which is generally used to compute element-wise integrals.

Effective computation of the element-wise integrals

Let us consider an entry (A, B) of the matrix of the global system obtained by the discretization of the problem. The element-wise integrals corresponding to the contribution of an element $\mathsf{D}^e \subset \Omega_{\text{pml},h}$ in the variational formulation (35) reads

$$\int_{\mathsf{D}^e} \left[(\mathbf{J}_{\text{pml},h}^{-\top} \nabla_x \Psi_A) \cdot (\mathbf{J}_{\text{pml},h}^{-\top} \nabla_x \Psi_B) - k^2 \Psi_A \Psi_B \right] (\det \mathbf{J}_{\text{pml},h})\, d\mathsf{D}^e, \tag{36}$$

where Ψ_A and Ψ_B are global basis functions (defined on Ω_h) associated to the global unknowns u_A and u_B. We assume that A and B are such that D^e belongs to the support of Ψ_A and Ψ_B.

Using the reference mapping between the physical element D^e and the reference element D_{ref}, the integral can be rewritten as

$$\int_{\mathsf{D}_{\text{ref}}} \left[(\mathbf{J}_{\text{pml},h}^{-\top} \mathbf{J}_{\text{ref}}^{-\top} \nabla_u \psi_a) \cdot (\mathbf{J}_{\text{pml},h}^{-\top} \mathbf{J}_{\text{ref}}^{-\top} \nabla_u \psi_b) - k^2 \psi_a \psi_b \right] (\det \mathbf{J}_{\text{pml},h}) (\det \mathbf{J}_{\text{ref}})\, d\mathsf{D}_{\text{ref}}, \tag{37}$$

where $\mathbf{J}_{\text{ref}} := \partial(x_1, x_2, x_3)/\partial(u_1, u_2, u_3)$ is the reference mapping, $\nabla_u = [\partial_{u_1}\ \partial_{u_2}\ \partial_{u_3}]^\top$ is the reference gradient, ψ_a and ψ_b are local basis functions defined on D_{ref}, and a and b are the local indices associated to the global indices A and B. We have $\psi_a(\mathbf{u}) = \Psi_A(\mathbf{x}^e(\mathbf{u}))$ and $\psi_b(\mathbf{u}) = \Psi_B(\mathbf{x}^e(\mathbf{u}))$ with $\mathbf{u} \in \mathsf{D}_{\text{ref}}$. Combining the Jacobian matrices gives

$$\int_{\mathsf{D}_{\text{ref}}} \Big[(\mathbf{J}^{-\top} \nabla_u \psi_a) \cdot (\mathbf{J}^{-\top} \nabla_u \psi_b) - k^2 \psi_a \psi_b \Big] \, (\det \mathbf{J}) \, d\mathsf{D}_{\text{ref}} \tag{38}$$

where the combined Jacobian matrix is defined as $\mathbf{J} := \mathbf{J}_{\text{pml},h} \mathbf{J}_{\text{ref}}$.

The effective computation of $\mathbf{J}$ is obtained rather straightforwardly. First, we re-factorize the matrix $\mathbf{J}$ as

$$\mathbf{J} = \frac{\partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)}{\partial(x_1, x_2, x_3)} \frac{\partial(x_1, x_2, x_3)}{\partial(u_1, u_2, u_3)} = \frac{\partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)}{\partial(\eta_1, \eta_2, \eta_3)} \frac{\partial(\eta_1, \eta_2, \eta_3)}{\partial(u_1, u_2, u_3)}, \tag{39}$$

with changes of variables between the complex Cartesian coordinates, the real curvilinear coordinates associated to $\Gamma_{\text{int},h}$ and the reference coordinates. Using equation (34), we easily derive the first Jacobian matrix,

$$\frac{\partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)}{\partial(\eta_1, \eta_2, \eta_3)} = \frac{\partial(x_1, x_2, x_3)}{\partial(\eta_1, \eta_2, \eta_3)} - \frac{1}{\imath k} \Big[\, \sigma(\eta_1)\, \mathbf{n}^e \ \ f(\eta_1)\, \partial_{\eta_2} \mathbf{n}^e \ \ f(\eta_1)\, \partial_{\eta_3} \mathbf{n}^e \, \Big]. \tag{40}$$

Seen the definition of the curvilinear coordinates, we have

$$\frac{\partial(\eta_1, \eta_2, \eta_3)}{\partial(u_1, u_2, u_3)} = \operatorname{diag}\Big(\partial_{u_1} \eta_1, 1, 1 \Big). \tag{41}$$

with $\partial_{u_1} \eta_1 = h_{\text{pml}}/2$. Finally, we get

$$\mathbf{J} = \mathbf{J}_{\text{ref}} - \frac{1}{\imath k} \Big[\, (\partial_{u_1} \eta_1)\, \sigma(\eta_1)\, \mathbf{n}^e \ \ f(\eta_1)\, \partial_{u_2} \mathbf{n}^e \ \ f(\eta_1)\, \partial_{u_3} \mathbf{n}^e \, \Big]. \tag{42}$$

Seen equation (42), the practical implementation of the AML simply consists in adding an imaginary part to the Jacobian matrix used in the element-wise integrals. This part only requires a spatial representation of the distance $r^e(u_1)$ and the extrusion direction $\mathbf{n}^e(u_2, u_3)$, which the nodal values are automatically obtained during the mesh extrusion and then extrapolated on the layer, as explained in section 3.2.

Discussion

Although our approach is based on a complex stretch in a coordinate system associated to the mesh, it can in fact be seen as a specific implementation of the conformal PML. Indeed, using equations (21) and (22), we have, for the conformal PML,

$$\frac{\partial(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)}{\partial(\xi_1, \xi_2, \xi_3)} = \frac{\partial(x_1, x_2, x_3)}{\partial(\xi_1, \xi_2, \xi_3)} - \frac{1}{\imath k} \Big[\, \sigma(\xi_1)\, \mathbf{n} \ \ \kappa_2 f(\xi_1)\, \mathbf{t}_2 \ \ \kappa_3 f(\xi_1)\, \mathbf{t}_3 \, \Big]. \tag{43}$$

We obtain the corresponding Jacobian for the AML implementation (equation (40)) by taking $(\xi_1, \xi_2, \xi_3) = (\eta_1, \eta_2, \eta_3)$ with the approximations $\mathbf{n} \approx \mathbf{n}^e$, $\kappa_2 \mathbf{t}_2 \approx \partial_{\eta_2} \mathbf{n}^e$ and $\kappa_3 \mathbf{t}_3 \approx \partial_{\eta_3} \mathbf{n}^e$ over each element D^e of the layer. In the case where the interface is smooth, the AML then is a conformal PML. For non-smooth interfaces, the AML can be seen as an approximate PML with effective geometric parameters at the corners. This approach is similar to the one proposed in [36] for high-order absorbing boundary conditions, where effective geometric parameters were obtained with empirical procedures to deal with corners.

The approach proposed by Ozgun and Kuzuoglu in [38, 39], which consists in replacing the coordinates of the mesh nodes with the stretched coordinates, is *a priori* simpler that the approach described here. With that approach, the Jacobian matrix used in the element-wise integrals would be

$$\mathbf{J} = \Big[\, \partial_{u_1} \tilde{\mathbf{x}}^e \ \ \partial_{u_2} \tilde{\mathbf{x}}^e \ \ \partial_{u_3} \tilde{\mathbf{x}}^e \, \Big], \tag{44}$$

where $\tilde{\mathbf{x}}^e$ corresponds to a polynomial representation of the stretched coordinates. Seen the definition of the stretched position (equation (34)), the real part of this matrix is identical to the one of the matrix obtained with our approach (equation (42)), but the imaginary part is different. With the approach of Ozgun and Kuzuoglu, the stretched coordinates are interpolated with polynomial shape functions, and then derivated numerically in equation (44). Nevertheless, both steps introduce approximation errors when the integrated absorbing function $f(\eta_1)$ that is in the imaginary part of equation (34) is not represented accurately with the polynomial shape functions. For instance, the widely used quadratic absorbing function $\sigma(\eta_1) = \alpha\eta_1^2$ requires third-degree polynomial shape functions for an exact representation, and a supplementary degree for an accurate numerical derivation. The very efficient unbounded absorbing functions introduced by Bermúdez *et al.* [9] cannot be represented accurately with polynomials. By contrast, with our approach, the absorbing function $\sigma(\eta_1)$ and the integrated function $f(\eta_1)$ are evaluated exactly in the Jacobian matrix (equation 42) without any polynomial interpolation and numerical derivation. The unbounded absorbing functions can be used efficiently.

4 Numerical validation and comparison

In this section, the efficiency of the PML implementations is studied for truncated computational domains with regular and non-regular borders in two and three dimensions. The AML approach is compared to the direct implementation of the PML (when the exact geometric data are available) and to the implementation based on complex mesh nodes proposed by Ozgun and Kuzuoglu [38, 39]. The reference benchmarks and the PML parameters used for the numerical simulations are detailed in section 4.1. The validation and comparison results for truncated domains with regular and non-regular borders are presented in sections 4.2 and 4.3, respectively.

4.1 Reference benchmarks and PML parameters

The numerical simulations represent the scattering of the plane wave $u_{\text{inc}}(\boldsymbol{x}) = e^{\imath kx}$ with the propagation direction $\hat{\boldsymbol{k}} = \boldsymbol{e}_x$ by a sound-hard object centered at the origin. In two dimensions, the scattering object is the disk of radius $a_{\text{sca}} = 1$ and the resulting scattered field is

$$u_{\text{ref}}(\boldsymbol{x}) = -\sum_{m=0}^{\infty} \epsilon_m \imath^m \frac{{J_m}'(ka_{\text{sca}})}{{H_m^{(1)}}'(ka_{\text{sca}})} H_m^{(1)}(kr)\cos(m\theta), \quad r \geq a_{\text{sca}}, \tag{45}$$

where (r, θ) are the polar coordinates, J_m is the m^{th}-order Bessel function, $H_m^{(1)}$ is the m^{th}-order first-kind Hankel function, and ϵ_m is the Neumann function which is equal to 1 for $m = 0$ and 2 otherwise. In three dimensions, the scattering object is the sphere of radius $a_{\text{sca}} = 1$ and the resulting scattered field is

$$u_{\text{ref}}(\boldsymbol{x}) = -\sum_{m=0}^{\infty} \imath^m (2m+1) \frac{{j_m}'(ka_{\text{sca}})}{{h_m^{(1)}}'(ka_{\text{sca}})} h_m^{(1)}(kr)\, P_m(x/r), \quad r \geq a_{\text{sca}}, \tag{46}$$

where $r = \|\boldsymbol{x}\|$ is the radial coordinate, j_m is the m^{th}-order spherical Bessel function, $h_m^{(1)}$ is the m^{th}-order first-kind spherical Hankel function, and P_m is the m^{th}-order Legendre polynomial.

Truncated computational domains with different shapes are considered. For each case, the domain is meshed with the mesh generator Gmsh [22]. A layer surrounding the domain is generated by extrusion at run time by the solver, which is a Matlab code. The mesh of the domain is made of triangular or tetrahedral elements, and the extruded layer is composed of quadrangular or prismatic elements in two and three dimensions, respectively. Quadratic curvilinear elements are used for the accurate geometric representation of the problem ($N_{\text{geo}} = 2$). The finite element solution is computed with hierarchical polynomial shape functions of maximal degree $p = 1$, 2 or 3 (see *e.g.* [6, 43]). The Neumann boundary condition $\partial_n u = -\partial_n u_{\text{inc}}$ is prescribed on the boundary of the scattering object, and a homogeneous Neumann condition is used on the exterior boundary of the extruded layer.

The accuracy of the PML critically depends on the absorbing function $\sigma(\xi_1)$, the thickness δ of the layer and the spatial discretization (see e.g. [9, 15, 35]). These parameters must then be chosen with great

care. For a given discretization, increasing the thickness of the layer generally improves the accuracy, but it also increases the number of degrees of freedom and the size of the algebraic system. Therefore, in order to limit the computational cost, the layer should be as thin as possible with an optimized absorbing function.

Polynomial absorbing functions are widely used. These functions ensure a progressive damping of the outgoing waves in the layer. In particular, the cubic function

$$\sigma_{\text{cub}}(\xi_1) = \bar{\sigma}(\xi_1/\delta)^3 \tag{47}$$

is a very frequent choice, even though a supplementary free parameter $\bar{\sigma}$ is introduced. An approach to select the parameter $\bar{\sigma}$ consists in choosing *a priori* the reflection coefficient for outgoing waves with normal incidence. The reflection coefficient of the planar PML reads

$$R_{\theta_{\text{inc}}} = \exp\left[-2\cos(\theta_{\text{inc}})\int_0^\delta \sigma(\xi_1')\,d\xi_1'\right], \tag{48}$$

where θ_{inc} is the angle of incidence. For the cubic function, the constant amplitude $\bar{\sigma}$ should then be $\bar{\sigma} = (2/\delta)(\ln R_0^{-1})$, where δ and R_0 are given. Nevertheless, $\bar{\sigma}$ must not be too large to avoid any significant dispersion error due to the spatial discretization [15, 35]. For practical applications, the parameter selection is a critical issue.

In an alternative approach, Bermúdez *et al.* [9] studied unbounded absorbing functions which cancel the reflection coefficient (48). In particular, the hyperbolic function

$$\sigma_{\text{hyp}}(\xi_1) = \frac{1}{\delta - \xi_1} \tag{49}$$

provides good accuracy without requiring the tuning of supplementary free parameters for finite element simulations [9, 35]. Nevertheless, because this function is singular on the exterior border of the layer, the numerical approximation of the element-wise integrals requires some care. For the AML implementation and the direct implementation of the conformal PML, we have used the hyperbolic function and Gauss-Legendre quadratures without any difficulty. Nevertheless, the implementation with complex mesh nodes has not provided satisfactory results with this absorbing function. That implementation is not suited to this kind of absorbing function (see discussion at the end of section 3.3).

In all the cases, the thickness of the layer is equal to $N_{\text{pml}}h_{\text{pml}}$, where N_{pml} is the number of extrusions and h_{pml} is the extrusion distance. The distance h_{pml} is taken equal to h, the characteristic size of the mesh cells in the computational domain. The influence of N_{pml} on the accuracy is studied in the following sections.

4.2 Numerical results for computational domains with regular boundary

The AML implementation is compared to the other approaches for two computational domains with regular boundary: the disk of radius $a = 1.1$ and the ellipse of semi-axes $a_x = 1.6$ and $a_y = 1.1$. The scattering disk of unit radius is placed in the middle of these domains. The simulations are performed with the wavenumber $k = 25$, the polynomial degrees $p = 1$, 2 and 3, the characteristic size of the mesh cells $h = p\lambda/d_\lambda$, the wavelength $\lambda = 2\pi/k$ and the resolution rate $d_\lambda = 20$. The implementations are tested with different layer thicknesses (or, equivalently, different numbers of mesh extrusions N_{pml}) and both hyperbolic and cubic absorbing functions.

For a quantitative comparison of the implementations, we consider the relative L^2-error between the numerical solution u_{num} and the reference solution (45) on the computational domain,

$$\frac{\|u_{\text{num}} - u_{\text{ref}}\|_{L^2(\Omega)}}{\|u_{\text{ref}}\|_{L^2(\Omega)}}.$$

We also consider the relative L^2-error between the reference solution and its L^2-projection onto the finite element space,

$$\frac{\|\mathcal{P}u_{\text{ref}} - u_{\text{ref}}\|_{L^2(\Omega)}}{\|u_{\text{ref}}\|_{L^2(\Omega)}}.$$

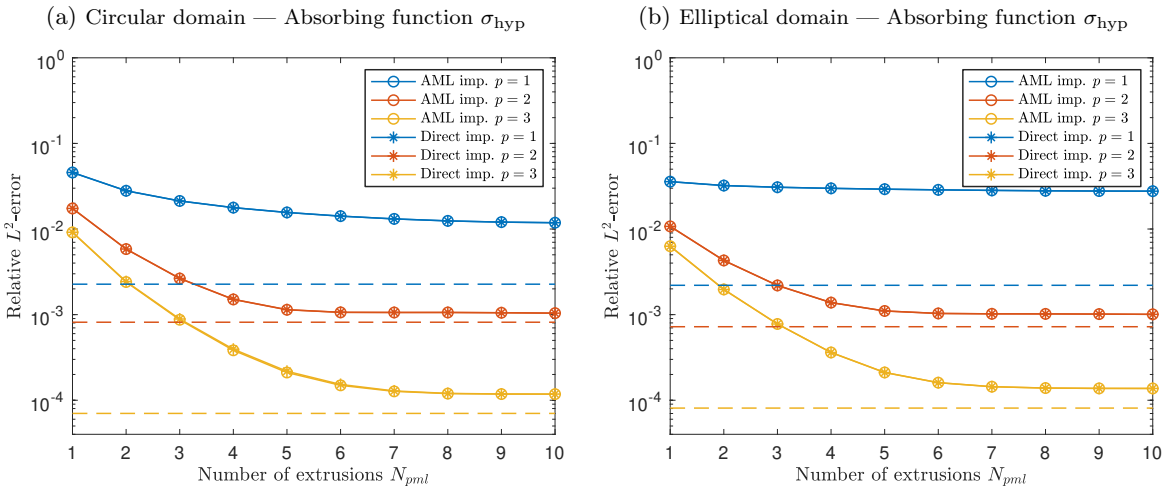


Figure 3: Comparison of the implementations with the hyperbolic absorbing function σ_{hyp} for the benchmarks with the circular domain *(left)* and the elliptical domain *(right)*. The relative L^2-error is plotted as a function of the layer thickness for finite element schemes with polynomial degrees $p = 1$, 2 and 3. Dashed line: best interpolation error in the truncated domains.

By Céa's lemma, this error corresponds to the best numerical solution that can be reached on each mesh, whatever the boundary treatment.

The AML implementation and the direct implementation of the conformal PML are firstly tested with the hyperbolic absorbing function σ_{hyp} (49). The exact geometric parameters used for the direct implementation of the conformal PML are provided in appendix A. The complex mesh nodes approach is not considered because it does not provide satisfactory results with σ_{hyp}. On Figure 3, the relative L^2-error is plotted as a function of the layer thickness for both implementations, with different polynomial degrees and both circular and elliptical domains. In all the cases, the AML implementation and the direct implementation give the same errors. The error is obviously larger with thin layers and small polynomial degrees. For these cases, the numerical dispersion is significant. Increasing the layer thickness decreases the error until a plateau that is close to the best interpolation error for $p = 2$ and 3. With $p = 1$, the plateau is substantially higher than the best interpolation error. The level of this plateau is higher in the elliptical case.

The PML implementations are now tested with the cubic absorbing function σ_{cub} (47). When using this function, the parameter $\bar{\sigma}$ must be chosen. Here, it is taken equal to $(2/\delta)(\ln R_0^{-1})$, where the tuning parameter R_0 has to be adjusted. We firstly study the parameter selection with the direct implementation of the PML in the circular case. Figure 4 shows the error curves obtained with the cubic function and different values of R_0, as well as the curves corresponding to the hyperbolic function, for both $p = 2$ and 3. For thick layers and large values of R_0, which correspond to small values of $\bar{\sigma}$, the relative errors are rather high and do not vary with the polynomial degree p. The errors are dominated by modeling errors: the outgoing waves are not sufficiently damped, and they are reflected at the exterior border of the layer. For thick layers and small values of R_0 (*i.e.* $R_0 \leq 10^{-5}$ for $p = 2$ and $R_0 \leq 10^{-6}$ for $p = 3$), they are effectively damped, and the errors are close to the best interpolation error. For thin layers, the errors increase because of the numerical dispersion. The errors are similar with the hyperbolic and cubic functions for layers with several cells. The cubic function is outperformed by the hyperbolic function for layers with one or two mesh cells. In the remaining of the manuscript, the cubic function is always used with $R_0 = 10^{-6}$.

Figure 5 shows the error curves obtained with the AML implementation and the complex mesh nodes approach equipped with the cubic function σ_{cub}. The results of the direct implementation, not shown, are nearly identical to those with the AML. For thin layers, the errors are always smaller with the AML than with the complex mesh nodes approach. As discussed at the end of section 3.3, approximation errors are introduced with the second approach because of the interpolation of the complex coordinates on the finite

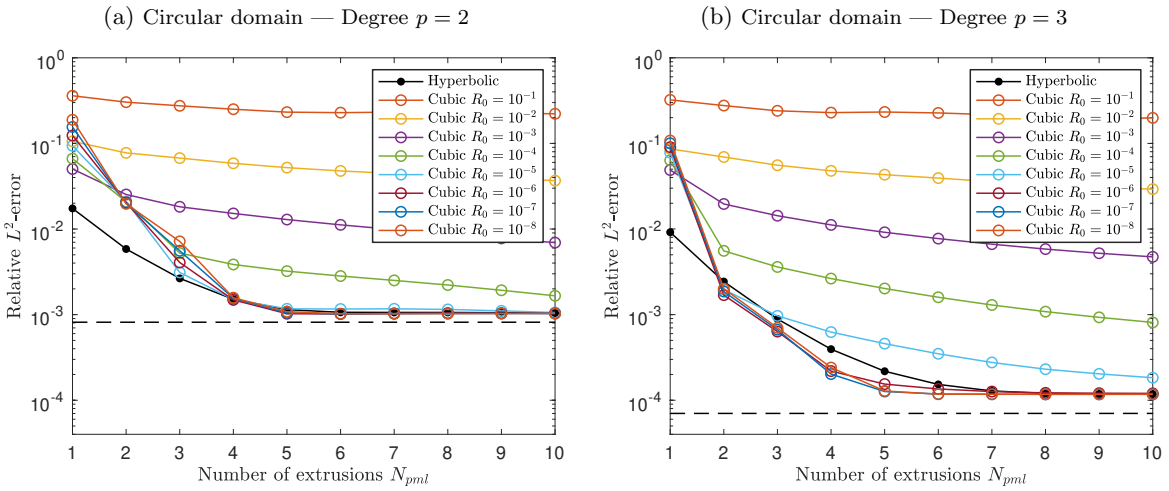


Figure 4: Comparison of the absorbing functions for the benchmark with the circular domain and $p = 2$ *(left)* or 3 *(right)*. The relative L^2-error is plotted as a function of the layer thickness for σ_{hyp} and σ_{cub} with different values of R_0. Dashed line: best interpolation error in the truncated domain.

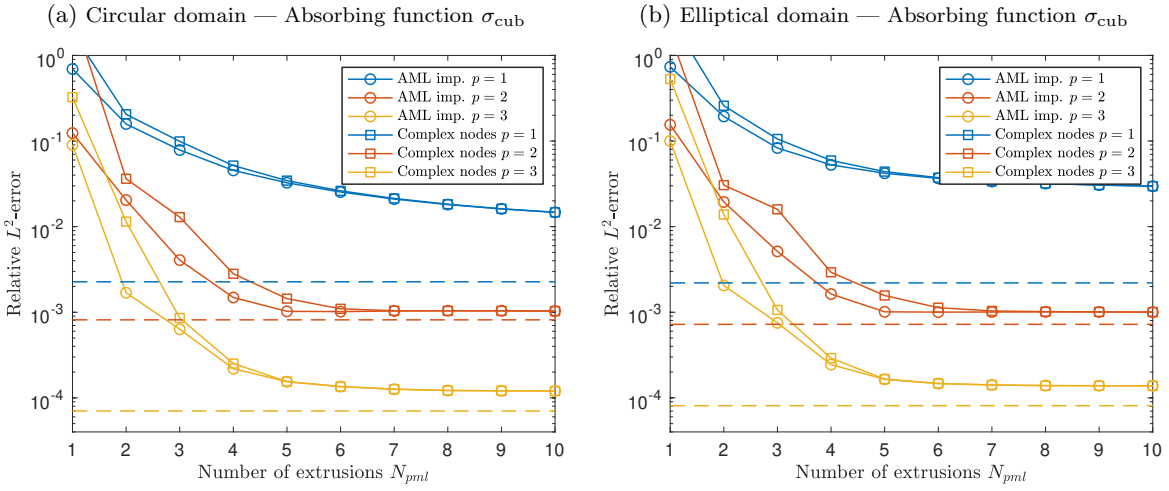


Figure 5: Comparison of the implementations with the cubic absorbing function σ_{cub} (with $R_0 = 10^{-6}$) for the benchmarks with the circular domain *(left)* and the elliptical domain *(right)*. The relative L^2-error is plotted as a function of the layer thickness for finite element schemes with polynomial degrees $p = 1$, 2 and 3. The direct implementation of the PML *(not shown)* gives the same errors as the AML implementation. Dashed line: best interpolation error in the truncated domain.

element mesh. This interpolation, which is the main difference between both implementations, explains the higher errors observed with the second one. In all the cases, increasing the layer thickness decreases the errors until plateaus, which are the same levels for all the implementations and both absorbing functions. These plateaus vary only with the polynomial degree p.

In a nutshell, the direct implementation of the conformal PML and the AML implementation give the same results, which validates our approach. The implementation based on complex mesh nodes provides similar results, but only for large layers with a polynomial absorbing function that must tuned.

4.3 Numerical results for computational domains with corners

The PML implementations are studied and compared for computational domains having corners with right and non-right angles. To analyse the influence of the angle on the error, polygonal and polyhedral domains are considered with different numbers of edges and faces, respectively. The circular and spherical domains, which are limit cases, are considered as well. In two dimensions, the scattering disk of unit radius is placed in the middle of polyhedral domains of midradius 1.65. In three dimensions, the scattering sphere of unit radius and polyhedral domains of midradius 2 are considered. In both cases, the polynomial degree is $p = 2$ and the resolution rate is $d_\lambda = 20$ The wavenumber is $k = 25$ in two dimensions and $k = 10$ in three dimensions.

Only the AML implementation and the complex mesh nodes approach, which can address any kind of convex truncated domains, are systematically compared. The standard PML is suited only to domains with smooth borders and right angles. In the latter case, the coordinate stretch is performed in two or three directions, which must be orthogonal. Strategies to address non-right angles have been proposed for absorbing layers that are defined differently [24, 23, 16]. Nevertheless, these strategies do not apply to standard PML formulations, and they involve specific modifications of the computational scheme at the corners. By contrast, the implementations considered here correspond to the standard PML on the regular parts of the domain boundary, with a corner treatment implicitly embedded in the coordinate stretch based on the mesh extrusion.

Figure 6 shows snapshots of the numerical solutions and the error distributions, together with the meshes, for numerical simulations performed with polygonal and circular domains. The AML implementation was used with $N_{\text{pml}} = 4$ mesh extrusions and the absorbing function σ_{hyp}. For the circular case, the error distribution is clearly dominated by the dispersion error of the finite element scheme. For the squared and pentagonal domains, the levels of error remain low, but structures in the error distributions indicate that spurious errors are generated at the boundary of the domain, and likely at the corners. Finally, the largest levels of error are observed with the triangular domain. For that case, the extruded mesh cells close to the corners have the worst aspect ratio. In addition, the exterior border is rather close to the scatterer. This is the most challenging case for the PML implementations.

For a quantitative comparison of the implementations, the L^2-error is computed for different configurations with the polygonal and circular domains. We consider the AML implementation with σ_{hyp}, and the complex mesh nodes approach with σ_{cub}. As a reference, the standard Cartesian PML is tested with both absorbing functions for the configurations with the square domain. The relative error is plotted as a function of the layer thickness for the different configurations on figure 7. Let us note that the relative projection error has nearly the same value for all the computational domains. Different observations can be made:

- When the hyperbolic function is used (Figure 7a), the errors obtained with the Cartesian PML are very close to the projection error, and do not vary with the layer thickness. With the AML implementation, the errors are slightly larger for the circular and polyhedral domains with obtuse angles. The errors significantly larger, by at least one order of magnitude, in the triangular case. This corroborates the preliminary qualitative study.
- When the cubic function is used (Figure 7b), the errors obtained with the Cartesian PML remain close to the projection error, but only for thick layers (*i.e.* with $N_{\text{pml}} \geq 5$). Either the parameter R_0 is not optimal, or the cubic function cannot provide results as good as the hyperbolic function for thin layers. With the complex mesh nodes approach, the errors are similar or slightly higher than the errors obtained with the Cartesian PML.
- In all the cases, increasing the layer thickness decreases the errors until they plateau, at the same levels for the AML implementation and the complex mesh nodes approach, despite the different absorbing functions. The plateau is much higher in the triangular case.

To summarize, both the AML implementation and the complex mesh nodes approach are equivalent for large layers. The results are satisfactory, except for the triangular case, where the border is close to the scattering object and the angles are acute. For thin layers, which allow for a reduction of the computational cost, combining the AML with the hyperbolic function is the most efficient approach.

The implementations are finally tested in three dimensions, with tetrahedral, cubic, octahedral, icosahedral and spherical domains. In figure 8, the error curves are shown for the AML implementation

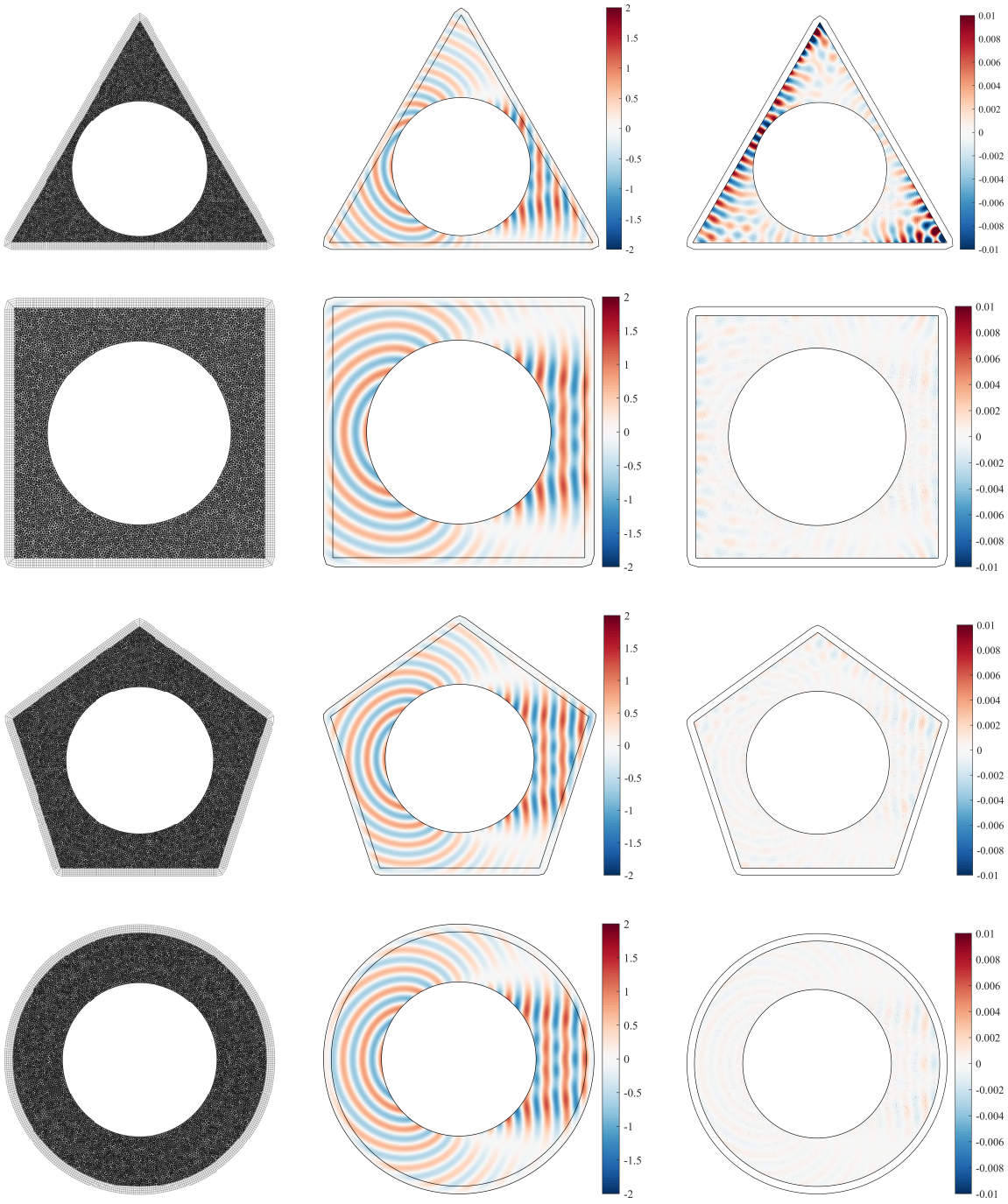


Figure 6: Polygonal computational domains: mesh *(left)*, real part of the numerical solution *(center)* and corresponding error distribution *(right)* obtained with the AML implementation. The layer is generated with $N_{\text{pml}} = 4$ mesh extrusions and the hyperbolic absorbing function is used.

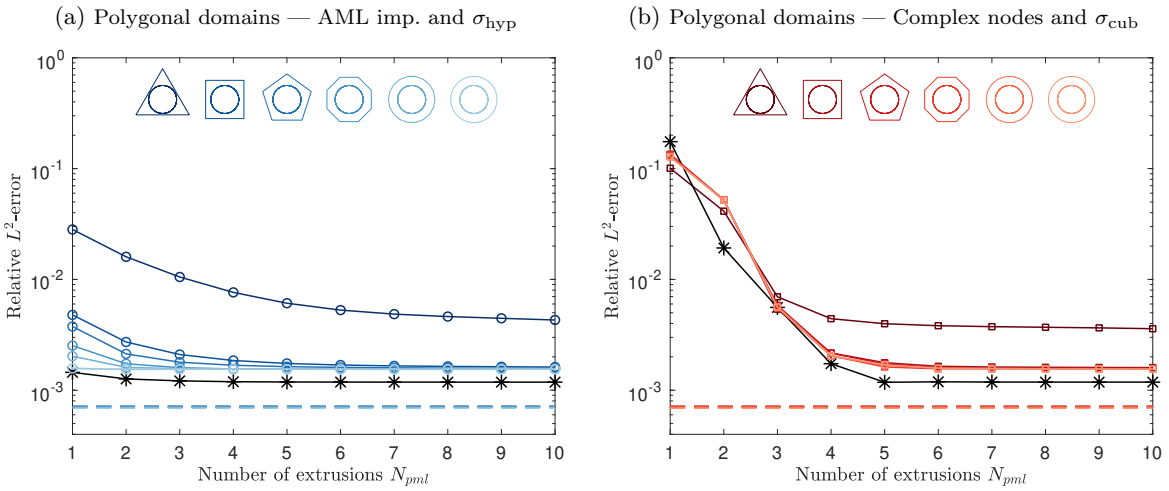


Figure 7: Polygonal computational domains: comparison of the AML equipped with σ_{hyp} *(left)* and the complex mesh nodes approach equipped with σ_{cub} *(right)*. Each colored line correspond to one polyhedral domain. Light and dark colors correspond to polyhedra with many and few edges, respectively. Black lines correspond to the results with the standard Cartesian PML for the squared domain. Dashed line: best interpolation error in the truncated domains.

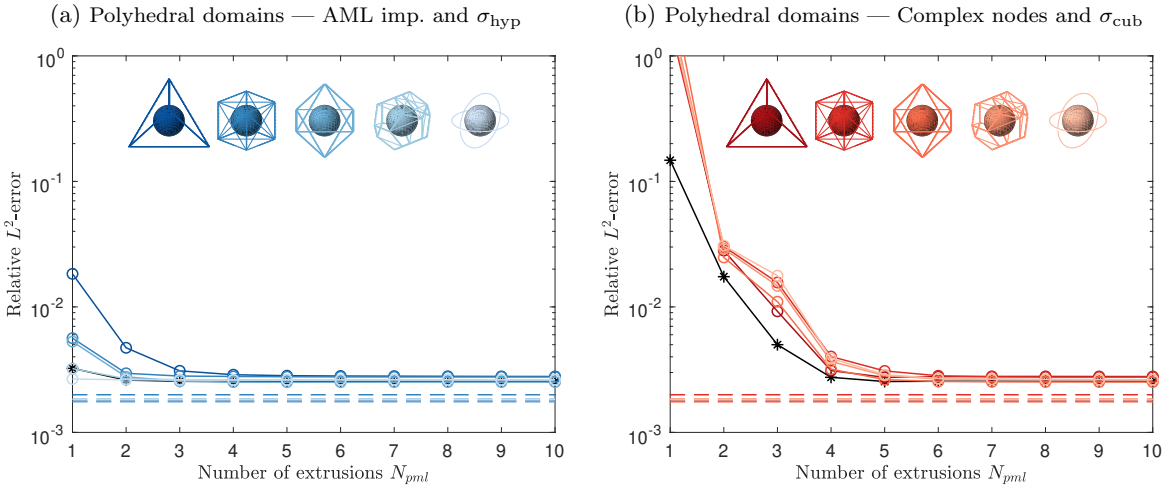


Figure 8: Polyhedral computational domains: comparison of the AML equipped with σ_{hyp} *(left)* and the complex mesh nodes approach equipped with σ_{cub} *(right)*. Each colored line correspond to one polyhedral domain. Light and dark colors correspond to polyhedra with many and few edges, respectively. Black lines correspond to the results with the standard Cartesian PML for the cubic domain. Dashed line: best interpolation error in the truncated domains.

with σ_{hyp} and the complex mesh nodes approach with σ_{cub}. The error curves corresponding to the Cartesian PML with the cubic domain are shown as well. These results confirm the observations made for the two-dimensional results. Both the AML implementation and the complex mesh nodes approach are efficient for sufficiently large layers. For thin layers, using the AML implementation with the hyperbolic absorbing function is the best strategy. For the icosahedral and the spherical cases, only one mesh extrusion is necessary.

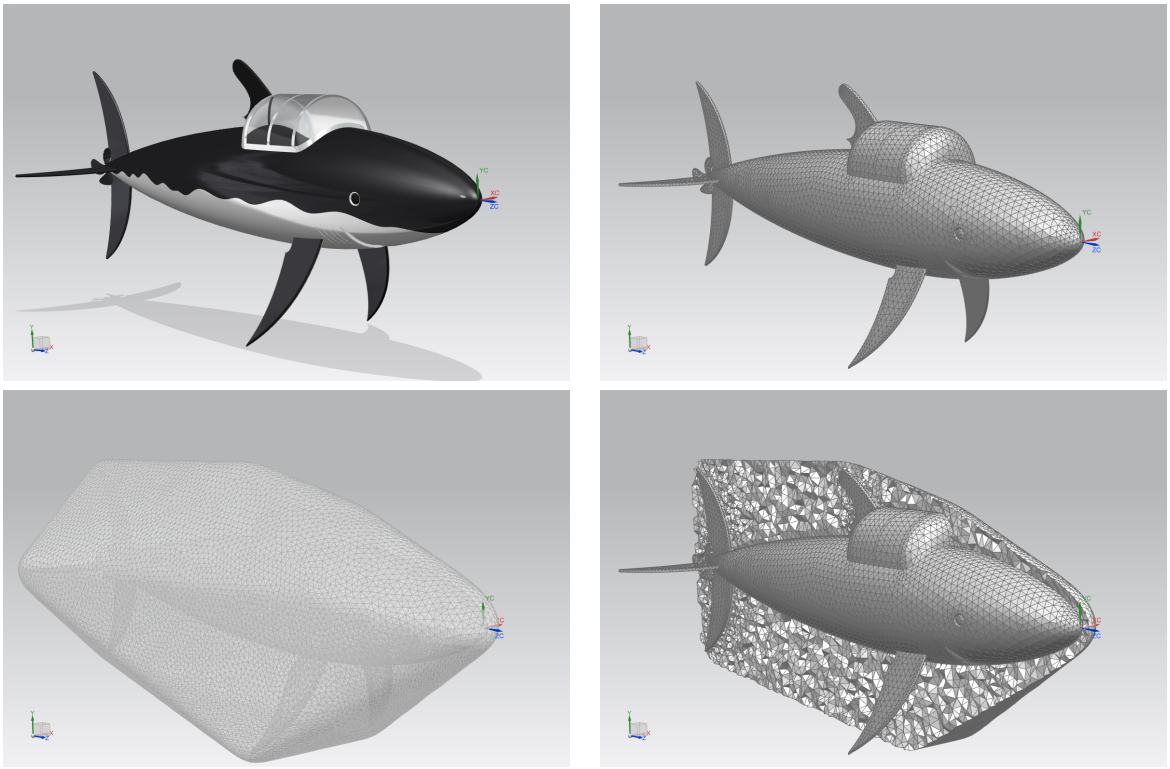


Figure 9: Applicative benchmark. Original CAD model of unit length *(top left)*, corresponding surface mesh with characteristic length $h = 15 \times 10^{-3}$ *(top right)*, convex exterior surface automatically generated at the minimal distance $2h$ from the submarine *(bottom left)* and cut view of the final tetrahedral mesh *(bottom right)*.

5 Application with an automatically generated convex domain

To illustrate the interest of the AML approach for realistic applications, we consider the scattering of a plane wave by a three-dimensional scattering object: the *shark* submarine. The original geometry[1] and the mesh of the submarine are shown in Figure 9. The submarine is placed in the middle of a computational domain where the pressure field is computed. In order to minimize the number of degrees of freedom, the exterior surface of this computational domain is generated automatically by using a complex hull algorithm. The exterior surface mesh is then extruded to generate the absorbing layer. The final model is set up and ran using the simulation package Simcenter 3D (version 2019.2) [42] developed by Siemens Industry Software, in which the AML implementation has been developed.

We describe step-by-step the process to get the mesh of the computational domain and the layer. First, a surface mesh of 6-noded triangular elements is generated on the original CAD geometry of the submarine, which is of unit total length. The characteristic length of the elements is $h \approx 15 \times 10^{-3}$. Then, an approximate convex hull of this surface mesh is computed by applying the quickhull algorithm [2]. To create the exterior surface of the computational domain, the convex hull is scaled in the exterior normal direction, and it is homogeneously triangulated with the element length h. The minimal distance between the submarine and the scaled surface is equal to $2h$. In a third step, a quadratic tetrahedral mesh of the volume between the submarine and the scaled surface is generated. The final mesh of the domain, shown in the bottom right part of Figure 9, is made of 302 803 10-noded tetrahedral elements and 439 884 nodes. As expected, close to the nose or at the fine tips, the exterior surface of the domain is only 2-elements away from the submarine surface. Finally, the mesh of the PML region, composed of 10-noded prismatic elements, is extruded by the solver at run time. The required geometric data are generated automatically (see discussion in Section 3.1). The extrusion length is $h_{\text{pml}} = h$, and the layer thickness is $\delta_{\text{pml}} = N_{\text{pml}} h_{\text{pml}}$. Only the AML implementation and the hyperbolic absorbing function are

[1] CAD model freely available on the Internet: https://grabcad.com/library/submarine-shark-1

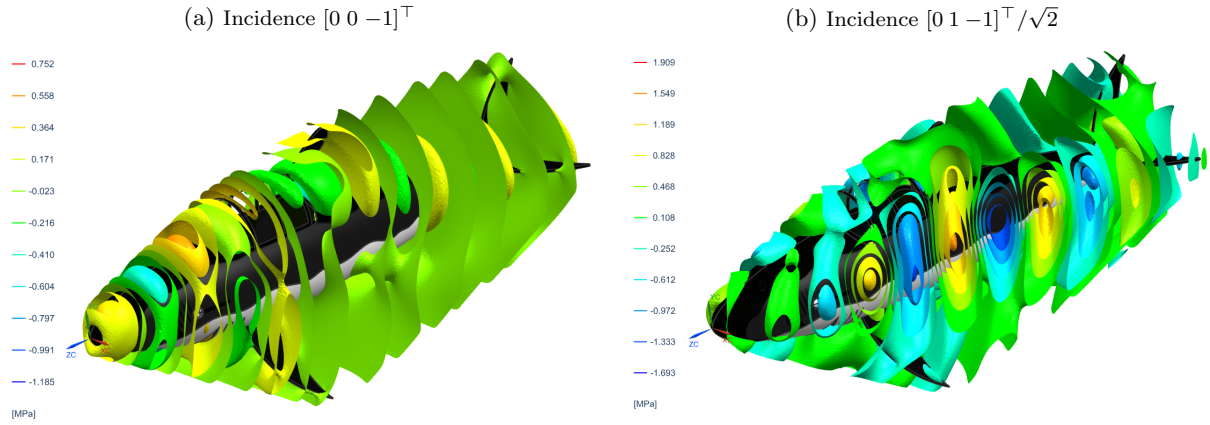


Figure 10: Applicative benchmark. Real part of the scattered fields obtained in the physical domain for both incident fields. The AML implementation is used with the absorbing function σ_{hyp} and $N_{\text{pml}} = 5$.

Table 1: Applicative benchmark. Model characteristics for truncated domains with different shapes. The distance between the interface and the scatterer is equal to $2h$ and an extruded layer with $N_{\text{pml}} = 5$ is generated. The volume of the domain corresponds to the sum of the volumes of the mesh cells.

Shape of domain	Volume of domain	# Elements		# DoFs			Memory for facto.
		Domain	Ext. layer	Domain	Ext. layer	Total	
Convex envelope	$6.8e^{-2}$	302 803	109 670	439 884	438 700	878 584	11.6 Gb
Cuboid	$19.7e^{-2}$	546 837	151 020	773 064	604 100	1 377 164	23.5 Gb
Cylinder	$21.1e^{-2}$	608 502	146 140	854 097	584 580	1 438 677	28.3 Gb

considered.

The simulations were performed for two incident plane waves hitting the submarine frontally ($\mathbf{k}_{\text{inc}} = [0\ 0\ -1]^\top$) and with an oblique incidence ($\mathbf{k}_{\text{inc}} = [0\ 1\ -1]^\top/\sqrt{2}$), respectively, with the wavenumber $k = 40$, the polynomial degree $p = 2$ and the resolution rate $d_\lambda = 20$. The real part of the scattered fields obtained in the computational domain is represented in Figure 10 for both incident fields in the case $N_{\text{pml}} = 5$. The orientation of the global axes is visible on the figures. The computations were performed on a DELL Precision 7520 laptop, with 3.10 GHz clock speed and 32 Gb of RAM. The most computationally intensive operations were the system assembly and the factorization, which required respectively 4.5 and 62 seconds on 4 threads. The factorization required 11.6 Gb of memory.

We first study the influence of the PML on the accuracy by comparing simulations performed with three layer thicknesses: $N_{\text{pml}} = 1$, 5 and 10. Figure 11 shows the directivity of the scattered field on a circle of radius 2 in the xz-plane, for both incidence directions and each layer thickness. For each incidence, the results obtained with the different thicknesses are very close. For layers with $N_{\text{pml}} = 5$ and 10, the curves are superimposed. Therefore, a thin absorbing layer, with a thickness of only few mesh cells, is sufficient to get accurate results, although the interface between the domain and the layer is very close to the scatterer.

The computational cost, in terms of runtime and memory storage, is directly related to the total number of degrees of freedom, and then to the volume of the computational domain. Using the automatically generated convex hull to define the domain, instead of a more standard geometry (*e.g.* cuboid, cylinder or spheroid), reduces the volume of the computational domain, yielding significant savings on the computational resources. To illustrate this, we have generated cuboidal and cylindrical computational domains aligned with the main axis of the submarine. For each geometry, the minimal distance between the exterior boundary and the submarine is equal to $2h$, and the exterior surface mesh is extruded to generate a layer with $N_{\text{pml}} = 5$. The characteristics of the resulting finite element models are compiled in Table 1.

The model with the convex envelope is significantly less computationally intensive than the cuboidal

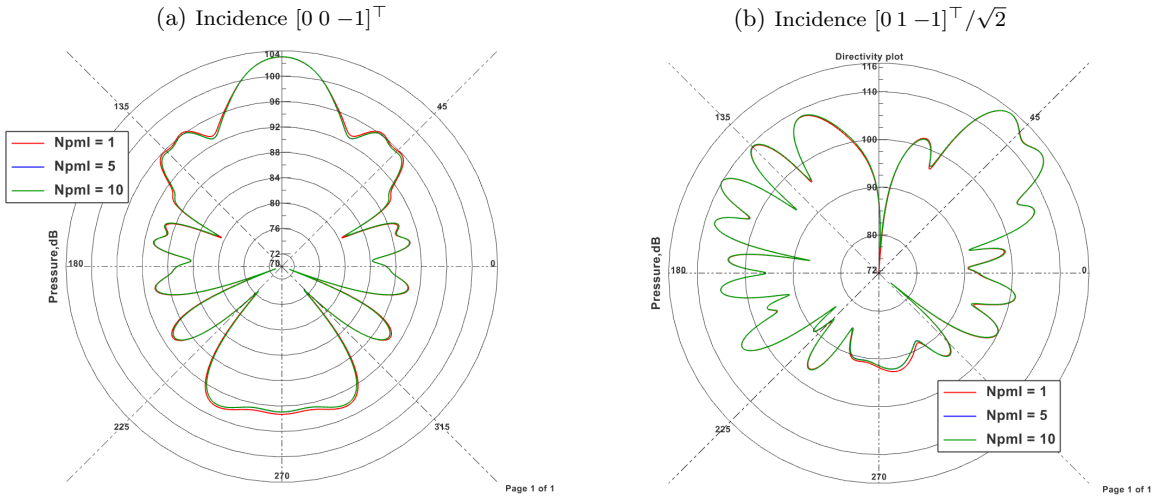


Figure 11: Applicative benchmark. Directivity plots of the scattered field obtained on a circle or radius $a = 2$, in the zx-plane, centered at $z = 0.5a$.

and cylindrical models. The volume and the number of degrees of freedom in the computational domain are reduced by approximately a factor of three and two, respectively. This difference may be explained by the presence of several mesh refinement regions close to the scatterer, which capture the small geometric details of the submarine. The PML unknowns represent a significant proportion of the total number of degrees of freedom in all the cases: 50%, 44% and 40% for the convex, cuboidal and cylindrical cases, respectively. The use of a convex envelope allows to reduce the size of the resulting global system of around 40% as compared to a canonical truncation. This results into a significant memory footprint reduction of 60% (resp. 50%) in comparison to the cylindrical (resp. cuboidal) case. Let us note that, the layer where generated by extrusion, even at the corners, using the procedure described in Section 3.1. With the standard Cartesian and cylindrical PML approaches, where the complex stretch is performed in orthogonal directions, the number of elements and the number of degrees of freedom would be larger at the corners, resulting in a larger computational cost.

In conclusion, it is important to limit the volume of the computational domain and the thickness of the layer to control the computational cost, especially in three dimensions. This advocates the use of the AML implementation of the PML, which brings geometric flexibility for the definition of the computational domain, and which allows to use efficient unbounded absorbing functions, such as the hyperbolic one.

6 Conclusion

In this article, a comprehensive strategy to implement the PML is proposed for the accurate finite element solution of scattering problems with generally-shaped convex computational domains. The approach, which we called the AML implementation, is fully automatic, in the sense that the end-user only needs to generate the mesh of the main domain, and to choose the layer thickness. The layer does not have to be modelled with a CAD tool, and no geometric information on the layer is required *a priori*. The approach is very versatile. It can be used with high-order finite element schemes, with unbounded absorbing functions, and for convex domains with regular and non-regular borders.

The AML implementation relies on a mesh extrusion generating the layer. Geometric data recorded during the extrusion step are interpolated on the finite element space, providing the required geometric parameters. No further geometric data is required. A rather simple modification of the Jacobian matrix in the element-wise finite element integrals is performed to progressively attenuate the solution inside the layer.

For computational domains with regular borders, the AML can be seen as a specific implementation of the conformal PML. With the conformal PML, a complex coordinate stretch is performed in a local coordinate system associated to the border of the domain. The direct implementation is rather easy thanks to complex material parameters. Unfortunately, the definition of these material parameters required the explicit knowledge of geometric parameters at all points of the layer (*i.e.* the distance to the border of the domain, the stretching direction, the principal curvatures and the principal directions), which can be difficult to obtain in practical cases (*e.g.* with complicated geometries or when only a mesh of the domain is available). Rather than using empirical approaches to estimate these parameters, the complex coordinate stretch is performed in the interpolated extrusion direction with the AML approach. This simple strategy allows to alleviate the issue of the accurate estimation of the geometric parameters. Numerical results have shown that the AML is equivalent to the direct implementation of the conformal PML in terms of accuracy for academic settings where the geometric parameters are known.

The proposed approach can deal with computational domains having corners, where the geometric parameters are normally not well-defined. The interpolated geometric parameters are used at the corners as an empirical strategy. The numerical results have shown that the method is very accurate for corners with obtuse angles, and that the accuracy decreases with more acute angles. For settings with right angles, the AML is not equivalent to the Cartesian PML, but the error remains at an acceptable level for the considered cases.

In term of easiness of implementation, the AML has been compared to the direct PML implementation and the implementation based on complex mesh nodes proposed by Ozgun and Kuzuoglu [38, 39]. Contrarily to the direct PML implementation, the AML approach does not require the explicit definition of the principal curvatures and the principal directions, though they could be estimated with the interpolated geometric parameters. The minimal geometric knowledge is also a feature of the complex mesh nodes implementation, but that approach cannot be applied with unbounded absorbing functions, which is a severe limitation.

The AML implementation of the PML, combined with the hyperbolic absorbing function, is an accurate, automatic and computationally efficient approach to solve large-scale scattering problems. The geometric flexibility allows for the use of a convex hull to minimize the volume of the computational domain, thereby reducing the size of the problem and the computational cost.

Acknowledgements

The first author gratefully acknowledges VLAIO *(Flanders Innovation & Entrepreneurship)* for their support of the EUREKA project HEAVENS *"Holistic environment for advanced vibro-acoustic engineering for non-specialists"* coordinated by Dr. Onur Atak from Siemens Industry Software.

A Geometric data and Jacobian matrices for 2D reference cases

In this appendix, geometric data and Jacobian matrices used for the direct implementation of the conformal PML are provided. In the general two-dimensional case, the Jacobian matrix of the PML transformation, its inverse and its determinant read, respectively,

$$\mathbf{J}_{\text{pml}} = \sum_{i=1}^{2} s_i \mathbf{e}_i \mathbf{e}_i^\top, \qquad \mathbf{J}_{\text{pml}}^{-1} = \sum_{i=1}^{2} \frac{1}{s_i} \mathbf{e}_i \mathbf{e}_i^\top, \qquad \det \mathbf{J}_{\text{pml}} = s_1 s_2, \tag{50}$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ constitute an orthonormal frame, and s_1 and s_2 are the stretching functions.

For a circular domain of radius R, the Cartesian components of the orthonormal directions are

$$\mathbf{e}_1 = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \qquad \mathbf{e}_2 = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}, \tag{51}$$

and the stretching functions are

$$s_1 = 1 - \frac{1}{\imath k}\sigma(\xi_1), \qquad s_2 = 1 - \frac{1}{\imath k}\frac{1}{r} f(\xi_1), \tag{52}$$

with the polar coordinates (r, θ) and the stretched coordinate $\xi_1 = r - R$.

For an elliptical domain of semi-axes a_x and a_y, the Cartesian components of the orthonormal directions are

$$\mathbf{e}_1(\mathbf{p}) = \frac{1}{\left[(p_x a_y/a_x)^2 + (p_y a_x/a_y)^2\right]^{1/2}} \begin{bmatrix} p_x a_y/a_x \\ p_y a_x/a_y \end{bmatrix}, \tag{53}$$

$$\mathbf{e}_2(\mathbf{p}) = \frac{1}{\left[(p_x a_y/a_x)^2 + (p_y a_x/a_y)^2\right]^{1/2}} \begin{bmatrix} -p_y a_x/a_y \\ p_x a_y/a_x \end{bmatrix}, \tag{54}$$

where $\mathbf{p} = [p_x, p_y]$ is the closest point belonging to the boundary of the domain. One has $p_x^2/a_x^2 + p_y^2/a_y^2 = 1$. The stretching functions are

$$s_1 = 1 - \frac{1}{\imath k}\sigma(\xi), \qquad s_2 = 1 - \frac{1}{\imath k}\frac{1}{R(\mathbf{p}) + \xi} f(\xi), \tag{55}$$

with the radius of curvature

$$R(\mathbf{p}) = \frac{\left[(p_x a_y/a_x)^2 + (p_y a_x/a_y)^2\right]^{3/2}}{a_x a_y} \tag{56}$$

and the stretched coordinate $\xi = \mathrm{dist}(\mathbf{x}, \mathbf{p})$.

References

[1] X. Antoine, H. Barucq, and A. Bendali. Bayliss-Turkel-like radiation conditions on surfaces of arbitrary shape. *Journal of Mathematical Analysis and Applications*, 229(1):184–211, 1999.

[2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

[3] A. Bayliss and E. Turkel. Radiation boundary conditions for wave-like equations. *Communications on Pure and applied Mathematics*, 33(6):707–725, 1980.

[4] E. Bécache, A.-S. Bonnet-Ben Dhia, and G. Legendre. Perfectly matched layers for time-harmonic acoustics in the presence of a uniform flow. *SIAM Journal on Numerical Analysis*, 44(3):1191–1217, 2006.

[5] J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114(2):185–200, 1994.

[6] H. Bériot, A. Prinn, and G. Gabard. Efficient implementation of high-order finite elements for Helmholtz problems. *International Journal for Numerical Methods in Engineering*, 106(3):213–240, 2016.

[7] H. Bériot and M. Tournour. On the locally-conformal perfectly matched layer implementation for Helmholtz equation. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 2009, pages 503–513. Institute of Noise Control Engineering, 2009.

[8] A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An exact bounded perfectly matched layer for time-harmonic scattering problems. *SIAM Journal on Scientific Computing*, 30(1):312–338, 2007.

[9] A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. *Journal of Computational Physics*, 223(2):469–488, 2007.

[10] J. H. Bramble and J. E. Pasciak. Analysis of a Cartesian PML approximation to acoustic scattering problems in R2 and R3. *Journal of Computational and Applied Mathematics*, 247:209–230, 2013.

[11] G. Bunting, A. Prakash, T. Walsh, and C. Dohrmann. Parallel ellipsoidal perfectly matched layers for acoustic Helmholtz problems on exterior domains. *Journal of theoretical and computational acoustics*, 26(02):1850015, 2018.

[12] W. C. Chew and W. H. Weedon. A 3d perfectly matched medium from modified maxwell's equations with stretched coordinates. *Microwave and optical technology letters*, 7(13):599–604, 1994.

[13] R. Cimpeanu, A. Martinsson, and M. Heil. A parameter-free perfectly matched layer formulation for the finite-element-based solution of the Helmholtz equation. *Journal of Computational Physics*, 296:329–347, 2015.

[14] F. Collino and P. Monk. The perfectly matched layer in curvilinear coordinates. *SIAM Journal on Scientific Computing*, 19(6):2061–2090, 1998.

[15] F. Collino and P. B. Monk. Optimizing the perfectly matched layer. *Computer methods in applied mechanics and engineering*, 164(1-2):157–171, 1998.

[16] E. Demaldent and S. Imperiale. Perfectly matched transmission problem with absorbing layers: Application to anisotropic acoustics in convex polygonal domains. *International Journal for Numerical Methods in Engineering*, 96(11):689–711, 2013.

[17] L. Demkowicz and F. Ihlenburg. Analysis of a coupled finite-infinite element method for exterior Helmholtz problems. *Numerische Mathematik*, 88(1):43–73, 2001.

[18] A. El Kacimi, O. Laghrouche, D. Ouazar, M. Mohamed, M. Seaid, and J. Trevelyan. Enhanced conformal perfectly matched layers for bernstein–bézier finite element modelling of short wave scattering. *Computer Methods in Applied Mechanics and Engineering*, 355:614–638, 2019.

[19] B. Engquist and A. Majda. Absorbing boundary conditions for numerical simulation of waves. *Proceedings of the National Academy of Sciences*, 74(5):1765–1766, 1977.

[20] S. D. Gedney. An anisotropic PML absorbing media for the FDTD simulation of fields in lossy and dispersive media. *Electromagnetics*, 16(4):399–415, 1996.

[21] K. Gerdes. A review of infinite element methods for exterior Helmholtz problems. *Journal of Computational Acoustics*, 8(01):43–62, 2000.

[22] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[23] M. N. Guddati and K.-W. Lim. Continued fraction absorbing boundary conditions for convex polygonal domains. *International Journal for Numerical Methods in Engineering*, 66(6):949–977, 2006.

[24] M. N. Guddati and J. L. Tassoulas. Continued-fraction absorbing boundary conditions for the wave equation. *Journal of Computational Acoustics*, 8(01):139–156, 2000.

[25] T. Hohage and L. Nannen. Hardy space infinite elements for scattering and resonance problems. *SIAM journal on numerical analysis*, 47(2):972–996, 2009.

[26] T. Huttunen, J. P. Kaipio, and P. Monk. The perfectly matched layer for the ultra-weak variational formulation of the 3D Helmholtz equation. *International Journal for Numerical Methods in Engineering*, 61(7):1072–1092, 2004.

[27] R. Kechroud, X. Antoine, and A. Soulaimani. Numerical accuracy of a Padé-type non-reflecting boundary condition for the finite element solution of acoustic scattering problems at high-frequency. *International Journal for Numerical Methods in Engineering*, 64(10):1275–1302, 2005.

[28] S. Kim and J. E. Pasciak. Analysis of a Cartesian PML approximation to acoustic scattering problems in R2. *Journal of Mathematical Analysis and Applications*, 370(1):168–186, 2010.

[29] M. Lassas, J. Liukkonen, and E. Somersalo. Complex Riemannian metric and absorbing boundary conditions. *Journal des Mathematiques Pures et Appliquees*, 80(7):739–768, 2001.

[30] M. Lassas and E. Somersalo. On the existence and convergence of the solution of pml equations. *Computing*, 60(3):229–241, 1998.

[31] M. Lassas and E. Somersalo. Analysis of the PML equations in general convex geometry. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 131(5):1183–1207, 2001.

[32] P. Marchner, H. Beriot, X. Antoine, and C. Geuzaine. Stable Perfectly Matched Layers with Lorentz transformation for the convected Helmholtz equation. working paper or preprint, Apr. 2020.

[33] P. J. Matuszyk and L. F. Demkowicz. Parametric finite elements, exact sequences and perfectly matched layers. *Computational Mechanics*, 51(1):35–45, 2013.

[34] M. Medvinsky, S. Tsynkov, and E. Turkel. Direct implementation of high order BGT artificial boundary conditions. *Journal of Computational Physics*, 376:98–128, 2019.

[35] A. Modave, E. Delhez, and C. Geuzaine. Optimizing perfectly matched layers in discrete contexts. *International Journal for Numerical Methods in Engineering*, 99(6):410–437, 2014.

[36] A. Modave, C. Geuzaine, and X. Antoine. Corner treatments for high-order local absorbing boundary conditions in high-frequency acoustic scattering. *Journal of Computational Physics*, 401:109029, 2020.

[37] A. Modave, J. Lambrechts, and C. Geuzaine. Perfectly matched layers for convex truncated domains with discontinuous Galerkin time domain simulations. *Computers & Mathematics with Applications*, 73(4):684–700, 2017.

[38] O. Ozgun and M. Kuzuoglu. Near-field performance analysis of locally-conformal perfectly matched absorbers via Monte Carlo simulations. *Journal of computational Physics*, 227(2):1225–1245, 2007.

[39] O. Ozgun and M. Kuzuoglu. Non-maxwellian locally-conformal PML absorbers for finite element mesh truncation. *IEEE transactions on antennas and propagation*, 55(3):931–937, 2007.

[40] C. M. Rappaport. Interpreting and improving the pml absorbing boundary condition using anisotropic lossy

mapping of space. *IEEE Transactions on Magnetics*, 32(3):968–974, 1996.

[41] Z. S. Sacks, D. M. Kingsland, R. Lee, and J.-F. Lee. A perfectly matched anisotropic absorber for use as an absorbing boundary condition. *IEEE transactions on Antennas and Propagation*, 43(12):1460–1463, 1995.

[42] Siemens Industry Software. Simcenter Nastran User's Guide (Version 2019.2), 2019. https://docs.plm.automation.siemens.com/tdoc/scnastran/2019_2/help/.

[43] P. Solin, K. Segeth, and I. Dolezel. *Higher-order finite element methods*. Chapman and Hall/CRC, 2003.

[44] F. Teixeira and W. Chew. Analytical derivation of a conformal perfectly matched absorber for electromagnetic waves. *Microwave and Optical technology letters*, 17(4):231–236, 1998.

[45] F. Teixeira and W. Chew. Complex space approach to perfectly matched layers: a review and some new developments. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 13(5):441–455, 2000.

[46] F. Teixeira and W. C. Chew. Differential forms, metrics, and the reflectionless absorption of electromagnetic waves. *Journal of Electromagnetic Waves and Applications*, 13(5):665–686, 1999.

[47] R. Tezaur, A. Macedo, C. Farhat, and R. Djellouli. Three-dimensional finite element calculations in acoustic scattering using arbitrarily shaped convex artificial boundaries. *International Journal for Numerical Methods in Engineering*, 53(6):1461–1476, 2002.

[48] L. L. Thompson. A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America*, 119(3):1315–1330, 2006.

[49] E. Turkel and A. Yefet. Absorbing PML boundary layers for wave-like equations. *Applied Numerical Mathematics*, 27(4):533–557, 1998.

[50] L. Zschiedrich, R. Klose, A. Schädle, and F. Schmidt. A new finite element realization of the perfectly matched layer method for Helmholtz scattering problems on polygonal domains in two dimensions. *Journal of Computational and applied mathematics*, 188(1):12–32, 2006.