



Metric-based anisotropic mesh adaptation for 3D acoustic boundary element methods

Stéphanie Chaillat, Samuel P Groth, Adrien Loseille

► To cite this version:

Stéphanie Chaillat, Samuel P Groth, Adrien Loseille. Metric-based anisotropic mesh adaptation for 3D acoustic boundary element methods. *Journal of Computational Physics*, 2018, 372, pp.473 - 499. 10.1016/j.jcp.2018.06.048 . hal-01895636

HAL Id: hal-01895636

<https://hal.science/hal-01895636>

Submitted on 15 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Metric-based anisotropic mesh adaptation for 3D acoustic boundary element methods

Stéphanie Chaillat^{a,*}, Samuel P. Groth^b, Adrien Loseille^c

^a*Laboratoire POEMS UMR CNRS-ENSTA-INRIA,
ENSTA-UMA, 828 Bd des Maréchaux, 91120 Palaiseau Cedex, FRANCE*

^b*Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

^c*Gamma 3 Team, INRIA Saclay-Ile de France, 1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau, FRANCE*

Abstract

This paper details the extension of a metric-based anisotropic mesh adaptation strategy to the boundary element method for problems of 3D acoustic wave propagation. Traditional mesh adaptation strategies for boundary element methods rely on Galerkin discretizations of the boundary integral equations, and the development of appropriate error indicators. They often require the solution of further integral equations. These methods utilise the error indicators to mark elements where the error is above a specified tolerance and then refine these elements. Such an approach cannot lead to anisotropic adaptation regardless of how these elements are refined, since the orientation and shape of current elements cannot be modified.

In contrast, the method proposed here is independent of the discretization technique (e.g., collocation, Galerkin). Furthermore, it completely remeshes at each refinement step, altering the shape, size, and orientation of each element according to an *optimal* metric based on a numerically recovered Hessian of the boundary solution. The resulting adaptation procedure is truly anisotropic and independent of the complexity of the geometry. We show via a variety of numerical examples that it recovers optimal convergence rates for domains with geometric singularities. In particular, a faster convergence rate is recovered for scattering problems with complex geometries.

Keywords: Acoustic wave scattering, Boundary Element Method, fast BEMs, Mesh adaptation, Anisotropic meshes

1. Introduction

We consider the scattering of time-harmonic acoustic waves by three-dimensional obstacles embedded within an unbounded, homogeneous domain. Various numerical methods can be used to solve such a problem, however a natural and popular approach is to reformulate the problem as a boundary integral equation. The main advantage of such a reformulation is to restrict the computational domain to the boundary of the obstacle and to exactly fulfill the outgoing radiation condition. The numerical solution of boundary integral equations is known as the Boundary Element Method (BEM) (often called the Method of Moments in the electromagnetic community). Despite the reduction in dimension of the computational domain, the main drawback of the BEM is the fully-populated nature of the system matrix. The cost of BEM simulations is thus prohibitively high when large-scale problems are concerned. Hence there has been a great deal of work since the inception of the BEM to reduce its computational cost.

If no acceleration technique is used, the storage of such a system is $\mathcal{O}(N^2)$, where N is the number of degrees of freedom on the scatterer boundary (e.g., the number of nodes in the mesh of the discretized boundary). The cost of solving the dense system using a direct method such as Gaussian elimination requires $\mathcal{O}(N^3)$ flops, whereas solution via an iterative method such as GMRES requires $\mathcal{O}(N_{\text{iter}}N^2)$, where N_{iter} is the number of iterations. In the last decades, different approaches have been proposed to speed up the solution of dense systems. The best-known method is probably the fast multipole method (FMM) originally proposed by Greengard and Rokhlin [35] which enables a fast evaluation of the matrix-vector product required by the iterative solver. Initially developed for N-body simulations, the FMM has since been extended to oscillatory kernels [25, 34]. Now it is widely used in many application fields and has shown its capabilities in the context of mechanical engineering problems solved with the BEM [16, 45]. An alternative approach designed for dense systems is based on the concept of hierarchical matrices (\mathcal{H} -matrices) [9]. The principle of \mathcal{H} -matrices is to partition the initial dense linear system, and then reduce it to a data-sparse one

*Corresponding author

Email addresses: stephanie.chaillat@ensta-paristech.fr (Stéphanie Chaillat), samuelpgroth@gmail.com (Samuel P. Groth), adrien.loseille@inria.fr (Adrien Loseille)

by finding sub-blocks in the matrix that can be accurately approximated by low-rank matrices. The efficiency of hierarchical matrices relies on the possibility to approximate, under certain conditions, the underlying kernel function by low-rank matrices. The approach has been shown to be very efficient for asymptotically smooth kernels (e.g., the Laplace kernel) and efficient in a pre-asymptotic regime for oscillatory kernels such as Helmholtz or elastodynamic kernels [19].

Mesh adaptation is an additional technique to reduce the computational cost of a numerical method. The principle is to optimize (or at least improve) the positioning of a given number of degrees of freedom on the geometry of the obstacle, in order to yield simulations with superior accuracy compared to those obtained via the use of uniform meshes. Adaptation is particularly important for scattering obstacles that contain geometric singularities, i.e., corners and ridges, which lead to a rapid variation of the surface solution near these singularities. For such problems, meshes graded toward these singularities must be employed in order to yield accurate approximations. In addition, for wave scattering problems, we may exploit the directionality of the waves in order to further reduce the number of degrees of freedom. The best strategy to achieve these goals is via so-called “anisotropic” mesh adaptation for which an extensive literature exists for volume-based methods such as the finite element method and the discontinuous Galerkin method [1]. However, there is relatively little research attention being paid to mesh adaptation in a boundary element context. One possible explanation is the large computational cost of standard BEMs. With the development of fast BEMs such as the Fast Multipole accelerated BEM (FM-BEM) [22] or \mathcal{H} -matrix accelerated BEM (\mathcal{H} -BEM) [12], the capabilities of the BEM are greatly improved such that efficient adaptive mesh strategies are needed not only to reduce further the computational cost, but also to certify the numerical results by assessing that the theoretical convergence order is observed during the computations.

In the BEM community, the majority of the research on mesh adaptation has been confined to isotropic techniques with a focus on the Laplace equation (see, e.g., the exhaustive review [27]) and extensions to the Helmholtz equation being made only fairly recently [5, 6, 7]. These isotropic techniques are usually based on *a posteriori* error analysis from which error indicators are derived. An indicator is then used to steer the mesh refinement by systematically marking and refining only elements where the error is above a specified threshold - a process known as Dörfler marking [15]. The derivation of appropriate local error estimators [26] is a significant challenge owing to the non-locality of boundary integral operators. This difficulty is the main reason why adaptivity for the BEM is a much less well-explored research topic in comparison to adaptivity for the FEM where the relevant operators are local differential operators. In many works convergence rates for error estimates are proven rigorously, e.g., [28, 14, 30]. However, it is seen that Dörfler refinement techniques do not usually recover the optimal convergence rates for 3D problems with anisotropic features [4]. Anisotropic variants of this strategy have been considered in [4, 28] however with rectangular elements for cube or cube-like shapes (where all the ridges are right angles). For these shapes they obtain the optimal convergence rate, however for general shapes (or complex geometries), their approach would not perform as well. The additional drawback of previously published works is the problem-dependent or integral equation-dependent nature of the error estimates. Also, the error analysis of these methods requires a Galerkin discretization and hence a higher computational cost than, say, a collocation discretization.

The first novelty of the present work is the extension of metric-based anisotropic mesh adaptation (AMA) to the BEM. The metric-based AMA proposed in [38, 39] does not employ a Dörfler marking strategy but rather generates a sequence of non-nested meshes with a specified complexity (proportional to the number of vertices or elements). The different meshes are defined according to a metric field derived from the evaluation of the linear interpolation error of the (unknown) exact solution on the current mesh. From a theoretical point of view, a continuous metric is derived from the Hessian of the exact solution. From a practical point of view, an approximate metric is derived from the numerical solution only (obtained via the BEM on a mesh). This approximate Hessian is based on the extension of typical (volumetric) derivative recovery operators [46] to the case of numerical boundary solutions. In AMA, the size, shape, and orientation of elements are adjusted simultaneously. The advantages of this approach are that it is ideally suited to solutions with anisotropic features, it is independent of the underlying PDE and discretization technique (collocation, Galerkin, etc.), and it is inexpensive. The metric-based AMA approach, as outlined above, has never been applied to the BEM. The purpose of this paper is to detail and report on the first application of metric-based AMA within a boundary element setting. Furthermore, we address some issues encountered when using an iterative solver for the FM-BEM on the resulting refined anisotropic meshes. In particular, we present two simple techniques to reduce the number of GMRES iterations required to achieve convergence when anisotropic elements are contained in the mesh.

The second novelty of this work is the combination of two acceleration techniques, namely metric-based anisotropic mesh adaptivity (AMA) and Fast Multipole acceleration. If no fast BEM is used, the capabilities of anisotropic mesh techniques cannot be fully demonstrated for realistic large scale scattering scenarios. This original combination permits us to show the performance of AMA strategy for complex real-world scattering problems such as acoustic scattering from an aircraft (see Section 5).

The outline of the paper is as follows. In Section 2, we recall the boundary integral equation formulation for wave scattering problems, the discretization of the resulting boundary integral equations (BIEs), and the acceleration of the solution of the arising linear system via the FM-BEM. In Section 3, we outline the AMA approach and extend the relevant results from [38, 39] for the case of a 3D volume solution to a 3D surface solution. In Section 4, we demonstrate the capabilities of the FM-BEM solver and present results illustrating the utility of a sequence of meshes to reduce the number of iterations required by the iterative BEM solver. Finally, in Section 5, an array of numerical examples is presented demonstrating the effectiveness of the AMA approach for the boundary element method. We show that optimal convergence rates are achieved with AMA for scatterers with geometric singularities, as opposed to sub-optimal rates when employing uniform meshes. Further we see how the anisotropic meshing exploits the wave directionality for higher frequency problems, thereby reducing the number of mesh elements required. For large-scale real-world scattering scenarios, we observe enormous gains over the employment of uniform meshes.

2. Standard and accelerated boundary element methods

2.1. Boundary integral equations for 3D Helmholtz equation

Consider a closed bounded domain $\Omega \subset \mathbb{R}^3$ with boundary Γ and let $\Omega^+ := \mathbb{R}^3 \setminus \overline{\Omega}$ denote the exterior scattering domain. Suppose that we have an incident plane wave u^i with wavenumber $k := 2\pi/\lambda$ (where λ is the wavelength) and direction \mathbf{d} ; this may be written as

$$u^i(\mathbf{x}) = e^{ik\mathbf{d} \cdot \mathbf{x}}, \quad \mathbf{x} \in \mathbb{R}^3.$$

u^i is scattered by Ω leading to a scattered field u^s which, when combined with u^i , gives the total field u in Ω^+ . This total field is such that

$$\nabla^2 u + k^2 u = 0 \quad \text{in } \Omega^+. \quad (1)$$

In this work, we consider only sound-soft (Dirichlet) boundary conditions (BCs)

$$u = 0 \quad \text{on } \Gamma. \quad (2)$$

Furthermore, we require that $u^s = u - u^i$ satisfies the Sommerfeld radiation condition

$$\lim_{r \rightarrow \infty} \left(\frac{\partial u^s}{\partial r} - ik u^s \right) = 0, \quad (3)$$

where $r = |\mathbf{x}|$. We note that the adaptation method presented later is general in that it can be applied to problems with sound-hard (Neumann), mixed, or transmission boundary conditions, in addition to the sound-soft case considered here. The sound-soft BCs, when compared to the other kinds of BC, give rise to the most severe singularities in the boundary solution when corners and ridges are present. Hence it is for this case that mesh adaptivity is most imperative and therefore why it is our focus in this paper.

The first main difficulty arising in the numerical solution to this exterior boundary value problem is related to the unboundedness of the computational domain Ω^+ . Integral equation based methods are one of the possible tools to overcome this issue. The approach is based on the potential theory [23]. For any positive real number k , let

$$\Phi_k(\mathbf{x}, \mathbf{y}) = \frac{e^{ik|\mathbf{x}-\mathbf{y}|}}{4\pi|\mathbf{x}-\mathbf{y}|}$$

be the fundamental solution of the 3D Helmholtz equation. The single-layer potential operator is defined by

$$\mathcal{S}\phi(\mathbf{x}) := \int_{\Gamma} \Phi_k(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) ds(\mathbf{y}), \quad \mathbf{x} \notin \Gamma.$$

The trace of the single-layer potential is given by applying the exterior Dirichlet trace to \mathcal{S} such that we have

$$(\mathcal{S}\phi)|_{\Gamma} = S\phi$$

where the boundary integral operator S is defined by

$$S\phi(\mathbf{x}) := \int_{\Gamma} \Phi_k(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}) ds(\mathbf{y}), \quad \mathbf{x} \in \Gamma.$$

There exist various possible integral equations to obtain the Cauchy data. For the numerical examples considered in Section 5, we use the simplest choice for the sound-soft problems. We employ the Dirichlet formulation using only the single-layer operator

$$Sq = -u^i, \quad (4)$$

which is usually called the electric field integral equation (*EFIE*) since, in the electromagnetic setting, the unknown represents the electric field. The solution of the integral equation (4) is the trace of $q = -\frac{\partial u}{\partial \mathbf{n}}$ on the boundary of the domain. The value in the domain is then obtained by using the following boundary integral representation

$$u^s(\mathbf{x}) = Sq(\mathbf{x}), \quad \mathbf{x} \in \Omega^+. \quad (5)$$

When the problem at hand does not have a thickness, we require a slight modification of the BIE for these problems since here the unknown is the jump in $\partial u / \partial \mathbf{n}$ across the surface, rather than $\partial u / \partial \mathbf{n}$ itself. We denote this jump by $[\frac{\partial u}{\partial \mathbf{n}}]$. So the BIE to solve is [44]:

$$-u^i(\mathbf{x}) = \int_{\Gamma} \Phi_k(\mathbf{x}, \mathbf{y}) \left[\frac{\partial u}{\partial \mathbf{n}} \right](\mathbf{y}) ds(\mathbf{y}), \quad \mathbf{x} \in \Gamma. \quad (6)$$

It is known [42] that the Integral Equation (4) is not uniquely solvable for all wavenumbers k . Hence, in practice, it is often preferable to employ so-called “combined” formulations, such as the *combined integral formulations* [20], which are uniquely solvable for all k but are more computationally expensive. However, for the purpose of the present paper, the choice of integral equation is not particularly important since our focus is to demonstrate the application of AMA to BEMs. Thus we choose to employ the simplest BIE formulation in order to keep the computational cost at a minimum.

2.2. Boundary element discretization and Fast Multipole acceleration

The main ingredients of the BEM are a transposition of the concepts developed for the Finite Element Method [11]. First, the numerical solution of the boundary integral equation (4) is based on a discretization of the surface Γ into N_E isoparametric boundary elements of order one, i.e., three-node triangular elements. Each physical element E_e on the approximate boundary is mapped onto a reference element Δ_e via an affine mapping

$$\boldsymbol{\xi} \in \Delta_e \rightarrow \mathbf{y}(\boldsymbol{\xi}) \in E_e, \quad 1 \leq e \leq N_e.$$

Δ_e is the reference triangle in the (ξ_1, ξ_2) -plane. The N interpolation points $\mathbf{y}_1, \dots, \mathbf{y}_N$ are chosen as the vertices of the mesh. Each unknown field q is approximated with globally continuous, piecewise-linear shape functions $(v_i(\mathbf{y}))_{1 \leq i \leq N}$: $v_i(\mathbf{y}_j) = \delta_{ij}$ for $1 \leq i, j \leq N$. A boundary element E_e contains exactly 3 interpolation nodes $(\mathbf{y}_k^e)_{1 \leq k \leq 3}$ associated with 3 basis functions $(v_k^e)_{1 \leq k \leq 3}$. These basis functions are related to the canonical basis $(\hat{v}_k)_{1 \leq k \leq 3}$ defined on the reference element Δ_e by $v_k^e(\mathbf{y}(\boldsymbol{\xi})) = \hat{v}_k(\boldsymbol{\xi})$. Each unknown field is approximated on the element E_e by

$$q(\mathbf{y}) \approx \sum_{k=1}^3 q^k v_k^e(\mathbf{y}),$$

where q^k denotes the approximation of the nodal value of $q(\mathbf{y}_k)$. To discretize the boundary integral equation (4), we employ the collocation approach. It consists in enforcing the equation at a finite number of collocation points \mathbf{x} . To have a solvable discrete problem, one has to choose N collocation points. The N approximation nodes thus defined also serve as collocation points, i.e., $(\mathbf{x}_i)_{1 \leq i \leq N} = (\mathbf{y}_j)_{1 \leq j \leq N}$. In addition, a standard Gaussian quadrature formula is used to evaluate the integral in (4). This discretization process transforms (4) into a square complex-valued linear system of size N of the form

$$\mathbb{A}\mathbf{q} = \mathbf{b}, \quad (7)$$

where the (N) -vector \mathbf{q} collects the degrees of freedom (DOFs) while the (N) -vector \mathbf{b} arises from the imposed incident wave field. Assembling the full dense matrix \mathbb{A} [11] requires the computation of all element integrals for each collocation point, thus requiring an $\mathcal{O}(N^2)$ computational time and memory. In addition, BEM matrix equations such as (7) are here solved iteratively using the GMRES algorithm. With reference to (7), each GMRES iteration requires one evaluation of $\mathbb{A}\mathbf{q}$ for a given \mathbf{q} , a task requiring a computing time of order $\mathcal{O}(N^2)$. To lower this $\mathcal{O}(N^2)$ complexity, which is unacceptable for large BEM models, fast BEM solution techniques such as the Fast Multipole Method (FMM) must be employed.

The goal of the FMM is to accelerate the evaluation of the matrix-vector product $\mathbb{A}\mathbf{q}$ for a given \mathbf{q} , required at each iteration of an iterative solver applied to the BEM-discretized equations. Moreover, the governing BEM matrix

is never explicitly formed, which leads to a storage requirement well below the $\mathcal{O}(N^2)$ memory required for holding it. Substantial savings in both computational time and memory requirements are thus achieved. In general terms, the FMM exploits a reformulation of the fundamental solution in terms of products of functions of \mathbf{x} and of \mathbf{y} , so that (unlike in the traditional BEM) integrations with respect to \mathbf{y} can be reused when the collocation point \mathbf{x} is changed. On decomposing the position vector $\mathbf{r} = \mathbf{y} - \mathbf{x}$ into $\mathbf{r} = (\mathbf{y} - \mathbf{y}_0) + \mathbf{r}_0 - (\mathbf{x} - \mathbf{x}_0)$, where \mathbf{x}_0 and \mathbf{y}_0 are two poles and $\mathbf{r}_0 = \mathbf{y}_0 - \mathbf{x}_0$ (Figure 1a) and invoking a well-known plane wave decomposition, the Helmholtz fundamental solution is written as [25]:

$$\Phi_k(\mathbf{x}, \mathbf{y}) = \lim_{L \rightarrow +\infty} \int_{\hat{\mathbf{s}} \in S} e^{ik\hat{\mathbf{s}} \cdot (\mathbf{y} - \mathbf{y}_0)} \mathcal{G}_L(\hat{\mathbf{s}}; \mathbf{r}_0; k) e^{-ik\hat{\mathbf{s}} \cdot (\mathbf{x} - \mathbf{x}_0)} d\hat{\mathbf{s}}, \quad (8)$$

where S is the unit sphere of \mathbb{R}^3 and the transfer function $\mathcal{G}_L(\hat{\mathbf{s}}; \mathbf{r}_0; k)$ is defined in terms of Legendre polynomials P_p and spherical Hankel functions of the first kind $h_p^{(1)}$ by:

$$\mathcal{G}_L(\hat{\mathbf{s}}; \mathbf{r}_0; k) = \frac{ik}{16\pi^2} \sum_{0 \leq p \leq L} (2p+1) i^p h_p^{(1)}(k|\mathbf{r}_0|) P_p(\cos(\hat{\mathbf{s}}, \mathbf{r}_0)). \quad (9)$$

It can be shown that Expression (8) is valid only for well-separated sets of collocation and integration points clustered

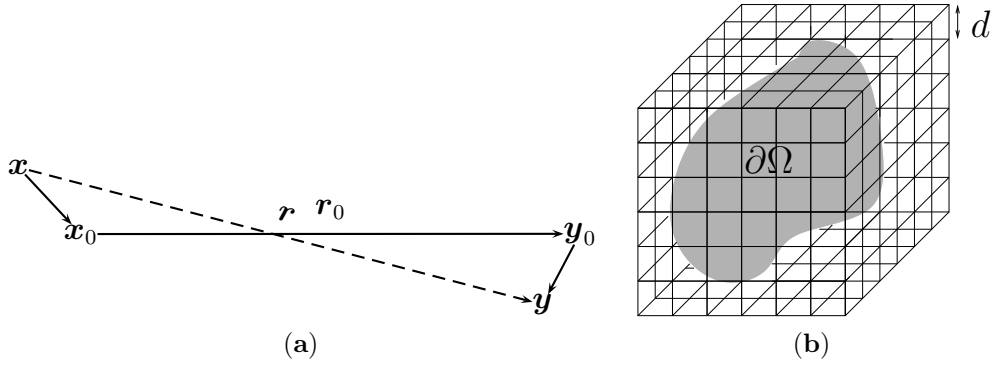


Figure 1: Fast Multipole Method: (a) Decomposition of the position vector and (b) 3D cubic grid embedding the boundary.

around poles \mathbf{x}_0 and \mathbf{y}_0 .

In the single-level FMM, a 3D cubic grid of linear spacing d embedding the whole boundary Γ is then introduced to drive the computation (see Figure 1b). The FMM basically consists in using decomposition (8), with the poles \mathbf{x}_0 and \mathbf{y}_0 being chosen as the cell centres of the cells \mathcal{C}_x and \mathcal{C}_y , whenever \mathbf{x} and \mathbf{y} belong to non-adjacent cubic cells (i.e., cells that do not share a corner, Figure 2). The treatment of such Fast Multipole (FM) contributions exploits the plane wave decomposition (8) of the fundamental solution, truncated at a finite L and in a manner suggested by its multiplicative form. One notes that $\mathbf{q} \rightarrow \mathbb{A}\mathbf{q}$ is a discretized version of the single-layer integral operator $q \rightarrow Sq$. The FM-BEM essentially consists in an accelerated method for the evaluation of discretized integral operators on given densities. The efficient evaluation of Sq needed in (4) is decomposed into three steps:

1. Evaluation of the multipole moments for each cell \mathcal{C}_y

$$\mathcal{R}(\hat{\mathbf{s}}; \mathcal{C}_y) = \int_{\Gamma} e^{ik\hat{\mathbf{s}} \cdot (\mathbf{y} - \mathbf{y}_0)} q(\mathbf{y}) dS(\mathbf{y}).$$

2. Application of the truncated transfer functions

$$\mathcal{L}(\hat{\mathbf{s}}; \mathcal{C}_x) = \sum_{\mathcal{C}_y \notin \mathcal{A}(\mathcal{C}_x)} \mathcal{G}_L(\hat{\mathbf{s}}; \mathbf{r}_0) \mathcal{R}(\hat{\mathbf{s}}; \mathcal{C}_y).$$

3. Numerical evaluation of the integration over the unit sphere with a quadrature rule

$$(Sq)^{FM}(\mathbf{x}) \approx \sum_q w_q \left[e^{-ik\hat{\mathbf{s}}_q \cdot (\mathbf{x} - \mathbf{x}_0)} \mathcal{L}(\hat{\mathbf{s}}_q; \mathcal{C}_x) \right].$$

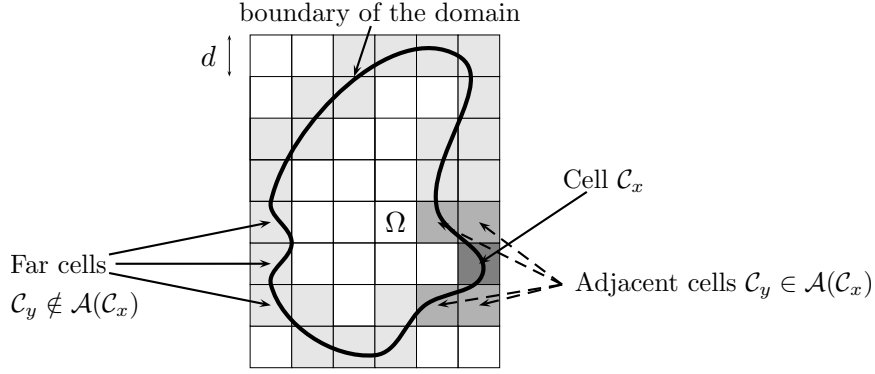


Figure 2: FM-BEM: Definition of the adjacent cells.

Conversely, when \mathbf{x} and \mathbf{y} belong to adjacent cells, traditional BEM evaluation methods are used instead.

To improve further the computational efficiency of the FM-BEM, standard (i.e., non-FMM) calculations must be confined to the smallest possible spatial regions while retaining the advantage of clustering the computation of influence terms into non-adjacent large groups whenever possible. This is achieved by recursively subdividing cubic cells into eight smaller cubic cells. New pairs of non-adjacent smaller cells, to which plane wave expansions are applicable, are thus obtained from the subdivision of pairs of adjacent cells. The cell-subdivision approach is systematized by means of an octree structure of cells. At each level ℓ , the linear cell size is denoted d^ℓ . The level $\ell = 0$, composed of only one cubic cell containing the whole surface Γ , is the tree root. The subdivision process is further repeated until the finest level $\ell = \bar{\ell}$, implicitly defined by a preset subdivision-stopping criterion ($d^{\bar{\ell}} \geq d^{\min}$), is reached. Level- $\bar{\ell}$ cells are usually termed leaf cells. This is the essence of the multi-level FMM, whose theoretical complexity is $\mathcal{O}(N \log N)$ per GMRES iteration both for computational time and memory requirements. More details on the implementation of the FM-BEM for waves can be found in [17].

2.3. Regularity of the boundary data

A standard rule of thumb for wave approximation problems is that between 6 and 10 degrees of freedom per wavelength are required to achieve “engineering accuracy” with numerical methods [41]. The most popular way to distribute these degrees of freedom is over a uniform mesh of the surface. However, for calculations involving scatterers with ridges, it is well known that this accuracy can be severely diminished if uniform meshes are employed [36]. Furthermore, the convergence of BEMs for these problems will be suboptimal. This is due to the low regularity of the boundary data at ridges. In particular, at a ridge, we expect the acoustic pressure u to behave as [36]

$$u \sim r^{\frac{\pi}{\alpha}} \quad \text{as } r \rightarrow 0, \quad (10)$$

where r is the distance from the ridge, and α is the exterior angle. Hence the flow velocity will behave as

$$\frac{\partial u}{\partial \mathbf{n}} \sim r^{\frac{\pi}{\alpha}-1} \quad \text{as } r \rightarrow 0. \quad (11)$$

At corners where two or more ridges meet, the singularities in u and $\partial u / \partial \mathbf{n}$ are even more severe and do not have simple expressions such as (10) and (11) (see, e.g., [8]).

In order to accurately approximate the solutions of such scattering problems and recover the optimal convergence order, the mesh must be appropriately refined towards the ridges. We note that the optimal order of convergence in the L^2 -norm for \mathbb{P}_1 discretizations is $\mathcal{O}(N^{-1})$ where N is the number of DOFs. Optimal *a priori* refinement strategies can be devised via a consideration of the polynomial interpolation of functions with the appropriate singularities [3, 33]. However, for complicated scattering geometries with vertices, designing an optimal mesh is challenging and often infeasible. Hence adaptive meshing strategies have proven popular. In the next section we describe the adaptive strategy employed here.

3. Metric-based anisotropic mesh adaptation

In this section, we derive new estimates for the metric-based anisotropic mesh adaptation procedure in the case of 3D surface solutions. Further details on the approach derivation for volume solutions and examples of application can be found in [38, 39], or in the comprehensive review [2].

The goal of mesh adaptation is to find the optimal mesh that permits the minimization of the approximation error due to the discretization scheme, i.e., the BEM in our context. To date, most mesh adaptation approaches for the BEM rely explicitly on the local error estimation of the boundary approximation to drive the adaptation procedure. The derivation of appropriate local error estimators is a significant challenge owing to the non-locality of boundary integral operators. This difficulty is the main reason why adaptivity for the BEM is a much less well-explored research topic in comparison to adaptivity for the FEM where the relevant operators are local differential operators.

The metric-based mesh adaptation method employed in this paper is also linked to the error in the surface approximation but in a much less explicit way. In addition, it is not linked to the underlying boundary integral operators. Rather, our method is connected to the linear interpolation error of the (unknown) exact solution on the mesh (interpolation with higher order functions has not yet been considered) within a specified norm. Then, given a number of degrees of freedom, meshes which minimize this interpolation error are designed in an anisotropic setting. Via arguments relating the BEM approximation error to this interpolation error, such meshes are seen to be close to optimal for BEM solutions. Many authors have considered such a technique, e.g., [21]. However this is the first time it is used in the boundary element context. In what follows, we focus on the description of metric-based mesh adaptation in the BEM setting, i.e., we assume that the solution is only provided on the boundary Γ .

3.1. Metric-based anisotropic mesh adaptation for a surface solution

A convenient framework to generate anisotropic meshes is that of Riemannian metric spaces. A Riemannian metric space is defined by a metric tensor $\mathcal{M}(\mathbf{x})$ (where \mathcal{M} is a symmetric positive definite matrix), i.e., is a smoothly varying function of the physical variable \mathbf{x} . In the context of anisotropic mesh generation, each mesh vertex \mathbf{x} has an assigned value $\mathcal{M}(\mathbf{x})$ which dictates the size and orientation of adjacent elements. By generating a unit mesh in the corresponding Riemannian metric space, we obtain an anisotropic mesh refined in Euclidean space according to the metric \mathcal{M} . This is the fundamental idea of metric-based mesh adaptation as introduced in [31].

A surface mesh, associated with a Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in \Gamma}$ is a triangulation of the surface Γ . A triangle K , which is defined by its edges $\{\mathbf{e}_i\}_{i=1}^3$, is *unit* with respect to \mathcal{M} if the length of each edge is unit in this metric, i.e., if

$$\|\mathbf{e}_i\|_{\mathcal{M}} := \sqrt{{}^t\mathbf{e}_i \mathcal{M} \mathbf{e}_i} = 1, \quad \text{for } i = 1, 2, 3. \quad (12)$$

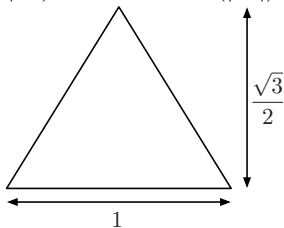
At this point it is important to note that in the BEM context, the mesh is 3D but the elements are 2D. We need to introduce the operator R_K that restricts a 3D quantity to the 2D local element frame. R_K can be deduced from the edges of K , such that

$$R_K(\mathcal{M}^{-1/2}) = \begin{pmatrix} {}^t\mathbf{t}_1 \mathcal{M}^{-1/2} \mathbf{t}_1 & {}^t\mathbf{t}_1 \mathcal{M}^{-1/2} \mathbf{t}_2 \\ {}^t\mathbf{t}_1 \mathcal{M}^{-1/2} \mathbf{t}_2 & {}^t\mathbf{t}_2 \mathcal{M}^{-1/2} \mathbf{t}_2 \end{pmatrix},$$

where

$$\mathbf{t}_1 = \frac{\mathbf{e}_1}{\|\mathbf{e}_1\|} \quad \text{and} \quad \mathbf{t}_2 = \frac{[{}^t\mathbf{e}_1 \mathbf{e}_1] \mathbf{e}_2 - [{}^t\mathbf{e}_1 \mathbf{e}_2] \mathbf{e}_1}{\|[{}^t\mathbf{e}_1 \mathbf{e}_1] \mathbf{e}_2 - [{}^t\mathbf{e}_1 \mathbf{e}_2] \mathbf{e}_1\|}.$$

If the three edges of K are of unit length with respect to the metric tensor \mathcal{M} , then the area of K computed in the Riemannian ($|K|_{\mathcal{M}}$) or Euclidean ($|K|$) space is, respectively,



$$|K|_{\mathcal{M}} = \frac{\sqrt{3}}{4}, \quad \text{and} \quad |K| = \frac{\sqrt{3}}{4} \det(R_K(\mathcal{M}^{-1/2})).$$

The existence of a conforming *unit mesh* in which each triangle is perfectly unit with respect to a given Riemannian metric space is not guaranteed in general. Hence the objective of seeking a unit mesh must be relaxed somewhat. We seek a *quasi-unit* mesh with respect to \mathcal{M} , which is a mesh composed of quasi-unit triangles. A triangle K is quasi-unit with respect to \mathcal{M} if

$$\frac{1}{\sqrt{2}} \leq \|\mathbf{e}_i\|_{\mathcal{M}} \leq \sqrt{2}, \quad \text{for } i = 1, 2, 3, \quad \text{and} \quad |K|_{\mathcal{M}} = \frac{\sqrt{3}}{4}. \quad (13)$$

These values come from practical considerations, see discussions in [37]. Using the above notion of a quasi-unit mesh, the generation of anisotropic meshes is simplified. The mesh generator is only required to generate a quasi-unit mesh in the prescribed Riemannian metric space, and this is shown to be always possible in [37]. The generated adapted mesh is anisotropic in Euclidean space.

The principle of AMA theory is then to minimize the error coming from the linear interpolation of the (unknown) exact solution. In the BEM context, we cannot directly apply the results proposed for 2D or 3D volume solutions as even though we work in 3D, we only have the solution on a surface. In particular, we cannot recover the Hessian of the numerical solution which is needed for the AMA procedure. One alternative would be to extend the theory for 3D volume solutions by considering the covariant derivatives. This approach would involve complex developments of differential geometry. In addition, we only focus on a linear approximation of the surface. These developments are mainly relevant for higher order approximations. A more natural and simple approach in the BEM context is to extend the surface solution in a tubular neighborhood, i.e., for a collection of points whose distance to the surface is smaller than ε . More precisely, to any solution u_s defined on the surface Γ , we associate the lifting u defined on the tubular neighborhood Γ_ε by

$$u(\mathbf{x}) = u_s(\mathcal{P}(\mathbf{x}))$$

where $\mathcal{P}(\mathbf{x})$ is the projection onto the surface Γ . The function u is now a 3D volume solution and we can define its Hessian.

3.2. Controlling the linear interpolation error

We suppose that the solution u is sufficiently smooth to possess a second order Taylor approximation. For 3D volume solutions, the quadratic approximation u^q of u is given by its truncated Taylor expansion centered at point \mathbf{a} :

$$u(\mathbf{x}) \approx u^q(\mathbf{x}) := u(\mathbf{a}) + {}^t\nabla u(\mathbf{a})(\mathbf{x} - \mathbf{a}) + \frac{1}{2} {}^t(\mathbf{x} - \mathbf{a}) H(\mathbf{a})(\mathbf{x} - \mathbf{a}),$$

where H is a symmetric matrix representing the Hessian of the extended solution u .

We now have all the tools to derive an estimation of the interpolation error. We follow the same ideas than in [38]. We note Π_h the discrete linear interpolation operator, i.e., $\Pi_h u$ is the linear interpolant of u on the mesh \mathcal{H}_N (with N nodes). The starting point is to show that the interpolation error in the L^1 norm of the solution u on a triangular element K is

$$\|u - \Pi_h u\|_{L^1(K)} = \frac{|K|}{24} \left| \sum_{i=1}^3 {}^t\mathbf{e}_i H \mathbf{e}_i \right| \leq \frac{|K|}{24} \sum_{i=1}^3 {}^t\mathbf{e}_i |H| \mathbf{e}_i, \quad (14)$$

where the \mathbf{e}_i are the edges of the triangle K provided in the local element frame and for every symmetric matrix H , $|H|$ denotes the positive symmetric matrix deduced from H by taking the absolute values of its eigenvalues. The proof is based on the evaluation of the pointwise interpolation error within the element, i.e., $(u - \Pi_h u)(\mathbf{x})$ for $\mathbf{x} \in K$. It only uses the mapping onto the reference element and the knowledge that the solution is a quadratic function. This error is then integrated over the element.

Proof: The reference element K_{ref} is defined by its three vertices:

$$\hat{\mathbf{x}}_1 = {}^t(0, 0), \quad \hat{\mathbf{x}}_2 = {}^t(1, 0), \quad \hat{\mathbf{x}}_3 = {}^t(0, 1).$$

The mapping onto the current element K is given by:

$$\mathbf{x} = \mathbf{x}_1 + B_K \hat{\mathbf{x}} \quad \text{with} \quad B_K = [\mathbf{x}_2 - \mathbf{x}_1, \mathbf{x}_3 - \mathbf{x}_1] = [\mathbf{e}_1, \mathbf{e}_2], \quad \mathbf{x} \in K, \quad \hat{\mathbf{x}} \in K_{ref},$$

where we have introduced the edges: $\mathbf{e}_1 = \mathbf{x}_2 - \mathbf{x}_1$, $\mathbf{e}_2 = \mathbf{x}_3 - \mathbf{x}_1$, and $\mathbf{e}_3 = \mathbf{x}_3 - \mathbf{x}_2$. The quadratic function u reads in the frame of K_{ref} :

$$u(\mathbf{x}(\hat{\mathbf{x}})) = \frac{1}{2} {}^t\mathbf{x}_1 H \mathbf{x}_1 + \frac{1}{2} {}^t\mathbf{x}_1 H B_K \hat{\mathbf{x}} + \frac{1}{2} {}^t\hat{\mathbf{x}} B_K H \mathbf{x}_1 + \frac{1}{2} {}^t\hat{\mathbf{x}} B_K H B_K \hat{\mathbf{x}}.$$

As we consider linear interpolation, the linear and constant terms of $u(\mathbf{x}(\hat{\mathbf{x}}))$ are exactly interpolated such that only quadratic terms are kept. We introduce $\tilde{u}(\mathbf{x}) = \frac{1}{2} {}^t\hat{\mathbf{x}} B_K H B_K \hat{\mathbf{x}}$, then it becomes

$$(u - \Pi_h u)(\mathbf{x}) = (\tilde{u} - \Pi_h \tilde{u})(\mathbf{x}).$$

We rewrite \tilde{u} in matrix form as

$$\tilde{u}(\mathbf{x}(\hat{\mathbf{x}})) = \frac{1}{2} {}^t \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \begin{bmatrix} {}^t\mathbf{e}_1 H \mathbf{e}_1 & {}^t\mathbf{e}_1 H \mathbf{e}_2 \\ {}^t\mathbf{e}_2 H \mathbf{e}_1 & {}^t\mathbf{e}_2 H \mathbf{e}_2 \end{bmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix},$$

and then \tilde{u} in K_{ref} reads:

$$\tilde{u}(\mathbf{x}(\hat{\mathbf{x}})) = \frac{1}{2} \left(({}^t\mathbf{e}_1 H \mathbf{e}_1) \hat{x}^2 + ({}^t\mathbf{e}_2 H \mathbf{e}_2) \hat{y}^2 + 2({}^t\mathbf{e}_1 H \mathbf{e}_2) \hat{x} \hat{y} \right).$$

\tilde{u} is now linearly interpolated on K_{ref} . Its linear interpolant $\Pi_h \tilde{u}(\hat{\mathbf{x}})$ writes $a\hat{x} + b\hat{y} + c$, where the coefficients $(a, b, c) \in \mathbb{R}^3$ satisfy the following linear system ensuring $\Pi_h u(\mathbf{v}_i) = u(\mathbf{v}_i)$ for all $i \in [1, 3]$, with \mathbf{x}_i the vertex i :

$$\begin{cases} \Pi_h \tilde{u}(\mathbf{x}_1) = \tilde{u}(\mathbf{x}(0, 0)) = 0 = c, \\ \Pi_h \tilde{u}(\mathbf{x}_2) = \tilde{u}(\mathbf{x}(1, 0)) = \frac{1}{2} ({}^t \mathbf{e}_1 H \mathbf{e}_1) = a, \\ \Pi_h \tilde{u}(\mathbf{x}_3) = \tilde{u}(\mathbf{x}(0, 1)) = \frac{1}{2} ({}^t \mathbf{e}_2 H \mathbf{e}_2) = b. \end{cases}$$

The solution of the previous linear system gives the final expression of $\Pi_h \tilde{u}$:

$$\Pi_h \tilde{u}(\mathbf{x}(\hat{\mathbf{x}})) = \frac{1}{2} [({}^t \mathbf{e}_1 H \mathbf{e}_1) \hat{x} + ({}^t \mathbf{e}_2 H \mathbf{e}_2) \hat{y}].$$

The exact pointwise interpolation error is then given by

$$(u - \Pi_h u)(\mathbf{x}(\hat{\mathbf{x}})) = \frac{1}{2} [({}^t \mathbf{e}_1 H \mathbf{e}_1) (\hat{x}^2 - \hat{x}) + ({}^t \mathbf{e}_2 H \mathbf{e}_2) (\hat{y}^2 - \hat{y}) + 2({}^t \mathbf{e}_1 H \mathbf{e}_2) \hat{x} \hat{y}].$$

Classically, for every function F , its integration over K can be computed through its expression in K_{ref} :

$$\int_K F(\mathbf{x}) dx dy = \int_{K_{ref}} F(\mathbf{x}(\hat{\mathbf{x}})) \|\mathbf{e}_1 \wedge \mathbf{e}_2\| d\hat{x} d\hat{y} = 2|K| \int_{K_{ref}} F(\mathbf{x}(\hat{\mathbf{x}})) d\hat{x} d\hat{y}.$$

Consequently, the interpolation error in the L^1 norm, evaluated by a direct integration of $|(u - \Pi_h u)(\mathbf{x})|$, is given by

$$\|u - \Pi_h u\|_{L^1(K)} = \frac{2|K|}{24} |{}^t \mathbf{e}_1 H \mathbf{e}_2 - {}^t \mathbf{e}_1 H \mathbf{e}_1 - {}^t \mathbf{e}_2 H \mathbf{e}_2|.$$

The cross term can be expressed only in terms of $\mathbf{e}_i H \mathbf{e}_i$ for $i = 1, \dots, 3$ by remarking that $\mathbf{e}_1 - \mathbf{e}_2 + \mathbf{e}_3 = 0$. It follows

$$2{}^t \mathbf{e}_1 H \mathbf{e}_2 = {}^t \mathbf{e}_1 H \mathbf{e}_1 + {}^t \mathbf{e}_2 H \mathbf{e}_2 - {}^t \mathbf{e}_3 H \mathbf{e}_3,$$

and we deduce:

$$2|{}^t \mathbf{e}_1 H \mathbf{e}_2 - ({}^t \mathbf{e}_1 H \mathbf{e}_1 + {}^t \mathbf{e}_2 H \mathbf{e}_2)| = \left| \sum_{i=1}^3 {}^t \mathbf{e}_i H \mathbf{e}_i \right|.$$

When all the eigenvalues of H are greater than or equal to zero, we have

$$\left| \sum_{i=1}^3 {}^t \mathbf{e}_i H \mathbf{e}_i \right| = \sum_{i=1}^3 {}^t \mathbf{e}_i H \mathbf{e}_i.$$

On the other hand, when H has positive and negative eigenvalues, we have only the following inequality

$$\left| \sum_{i=1}^3 {}^t \mathbf{e}_i H \mathbf{e}_i \right| \leq \sum_{i=1}^3 {}^t \mathbf{e}_i |H| \mathbf{e}_i.$$

This property is used to conclude the proof in the general case. ■

We consider now all elements K which are unit with respect to the metric \mathcal{M} . The next step consists in introducing the metric \mathcal{M} , i.e, in showing that

$$\sum_{i=1}^3 {}^t \mathbf{e}_i H \mathbf{e}_i = \frac{3}{2} \text{trace}(R_K(\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}})).$$

Proof: We consider first the simple case where $R_K(H) = I_2$ and $R_K(\mathcal{M}) = I_2$ (where I_2 is the 2×2 identity matrix). The regular triangle $K_0 = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ unit for I_2 is defined by the vertices:

$$\mathbf{x}_1 = (0, 0), \mathbf{x}_2 = (1, 0), \mathbf{x}_3 = \left(\frac{1}{2}, \frac{\sqrt{3}}{2} \right).$$

We first show the following preliminary result: *For every line (D) passing through one of the vertices of K_0 , the sum of the square lengths of the edges projected on (D) is invariant.*

Without loss of generality, we assume that (D) passes through the vertex \mathbf{x}_1 of K_0 . If (D) is defined by the vector

$$\mathbf{n} = (\cos(v), \sin(v)),$$

with $v \in \mathbb{R}$, then the length of the three edges of K_0 projected on (D) are given by:

$$a = \mathbf{e}_1 \cdot \mathbf{n} = \cos(v), \quad b = \mathbf{e}_2 \cdot \mathbf{n} = \frac{1}{2} \cos(v) + \frac{\sqrt{3}}{2} \sin(v), \quad c = \mathbf{e}_3 \cdot \mathbf{n} = \frac{-1}{2} \cos(v) + \frac{\sqrt{3}}{2} \sin(v).$$

Such that $a^2 + b^2 + c^2 = 3/2$. When $R_K(\mathcal{M})$ is different from I_2 , we use $R_K(\mathcal{M}^{-\frac{1}{2}})$ that maps the unit ball of I_2 onto the unit ball of $R_K(\mathcal{M})$. (D) is now directed by one of the eigenvectors of $R_K(\mathcal{M})$, e.g., \mathbf{v}_j , and which is passing through \mathbf{x}_1 . The lengths a , b and c are thus multiplied by h_j , i.e., the size prescribed by $R_K(\mathcal{M})$ in the direction \mathbf{v}_j . Consequently, the square length of the edges projected on (D) are multiplied by h_j^2 . It comes using the preliminary result:

$$\sum_{i=1}^3 ({}^t \mathbf{e}_i \mathbf{v}_j)^2 = \frac{3}{2} h_j^2.$$

Considering this relation for the two principal directions of $R_K(\mathcal{M})$, we obtain:

$$\sum_{i=1}^3 \|\mathbf{e}_i\|_2^2 = \sum_{j=1}^2 \sum_{i=1}^3 ({}^t \mathbf{e}_i \mathbf{v}_j)^2 = \frac{3}{2} (h_1^2 + h_2^2) = \frac{3}{2} \text{trace}(R_K(\mathcal{M})^{-1}). \quad (15)$$

We now consider the general case where a symmetric matrix $R_K(H)$ is involved in the estimation instead of $R_K(H) = I_2$. $R_K(H)$ being symmetric, it has two real eigenvalues $(\mu_i)_{i=1,2}$ along the principal directions $(\mathbf{u}_i)_{i=1,2}$. If K is a regular triangle with edges $(\mathbf{e}_i)_{i=1,3}$, according to the preliminary result, the sum of the projected square length of edges $(\mathbf{e}_i)_{i=1,3}$ on each principal direction \mathbf{u}_j is equal to $3/2$. We deduce:

$$\mu_j \sum_{i=1}^3 ({}^t \mathbf{e}_i \mathbf{u}_j)^2 = \frac{3}{2} \mu_j, \text{ for } j \in [1, 2] \text{ and } \sum_{j=1}^2 \sum_{i=1}^3 \mu_j ({}^t \mathbf{e}_i \mathbf{u}_j)^2 = \frac{3}{2} (\mu_1 + \mu_2).$$

The previous equality reads:

$$\sum_{i=1}^3 {}^t \mathbf{e}_i H \mathbf{e}_i = \frac{3}{2} \text{trace}(R_K(H)).$$

We deduce the general case by taking $R_K(\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}})$ as a symmetric matrix. In that case, each edge \mathbf{e}_i of the regular triangle is mapped on $\tilde{\mathbf{e}}_i = R_K(\mathcal{M}^{-\frac{1}{2}}) \mathbf{e}_i$. The new triangle defined by edges $(\tilde{\mathbf{e}}_i)_{i=1,3}$ is unit with respect to $R_K(\mathcal{M})$. This concludes the proof. \blacksquare

Using that $|K| = \frac{\sqrt{3}}{4} \det(R_K(\mathcal{M}^{-1/2}))$, the interpolation error (14) is finally expressed for a surface solution by:

$$\|u - \Pi_h u\|_{L^1(K)} = \frac{\sqrt{3}}{64} \det(R_K(\mathcal{M}^{-\frac{1}{2}})) |\text{trace}(R_K(\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}}))|. \quad (16)$$

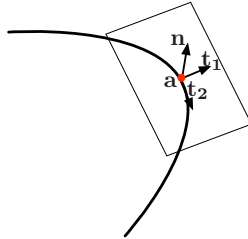


Figure 3: To extend the concept of metric-based error estimates, we introduce a surface restriction R_s on the local tangent plane.

Importantly, (16) implies that the interpolation error does not depend on the element shape, i.e., on the mesh generator. The metric alone contains enough information to describe completely the linear interpolation error in the L^1 norm. In addition, this interpolation error can be expressed with continuous quantities only. Indeed \mathcal{M} and H can

be defined continuously anywhere. The restriction R_K is replaced by its continuous counterpart R_s , i.e., the tangent plane restriction. For a smooth surface, the operator R_s is defined by using the local Frenet frame $[\mathbf{t}_1, \mathbf{t}_2, \mathbf{n}]$ (see Fig. 3). We thus define a continuous linear interpolation error, independent from the discrete mesh similarly to [38] by removing the surface contribution of (16)

$$2 \frac{\|u - \Pi_h u\|_{L^1(K)}}{|K|} = \frac{1}{8} |\text{trace}(R_s(\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}}))| \leq \frac{1}{8} \text{trace}(R_s(\mathcal{M}^{-\frac{1}{2}} |H| \mathcal{M}^{-\frac{1}{2}}))$$

Then, we define the upper bound of the continuous interpolation error by

$$e_{R_s(\mathcal{M})}(\mathbf{x}) = \frac{1}{8} \text{trace}(R_s(\mathcal{M}^{-\frac{1}{2}} |H| \mathcal{M}^{-\frac{1}{2}})). \quad (17)$$

Finally, the continuous interpolation error calculated on the continuous mesh \mathbf{M} is given by

$$E_p(\mathbf{M}) = \int_{\Gamma} \left(e_{R_s(\mathcal{M})}(\mathbf{x}) \right)^p d\mathbf{x}, \quad p \geq 1. \quad (18)$$

3.3. Numerical validation on an analytical example

In this section, we validate the previous continuous analysis and the assumption to work in the tangent plane. To this end, we compare the continuous evaluation of the interpolation with the discrete evaluation in a simple analytic case. For the surface, we consider a quarter of a cylinder (S) defined by the following parametric equations:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sigma(u, v) = \begin{pmatrix} \cos(u) \\ \sin(u) \\ v \end{pmatrix},$$

with $u \in [0, \pi/2]$ and $v \in [0, 1]$. The Frenet frame is then given by $\mathbf{n} = {}^t(x, y, 0)$, $\mathbf{t}_1 = {}^t(0, 0, 1)$ and $\mathbf{t}_2 = {}^t(-y, x, 0)$, so that the restriction R_s is defined everywhere. We also consider a continuous description of the mesh given by the following metric:

$$\mathcal{M}(x, y, z) = \alpha \begin{pmatrix} h_x^{-2} & & \\ & h_y^{-2} & \\ & & h_z^{-2} \end{pmatrix},$$

with $h_x(x, y, z) = 0.2$, $h_y(x, y, z) = 0.1(y+1) + 0.05(x-1)$ and $h_z(x, y, z) = 0.2(z+2)$. In the continuous mesh setting, we introduce the complexity \mathcal{N} of a mesh. It is the continuous counterpart of the number of vertices and it governs the size of the meshes, i.e.,

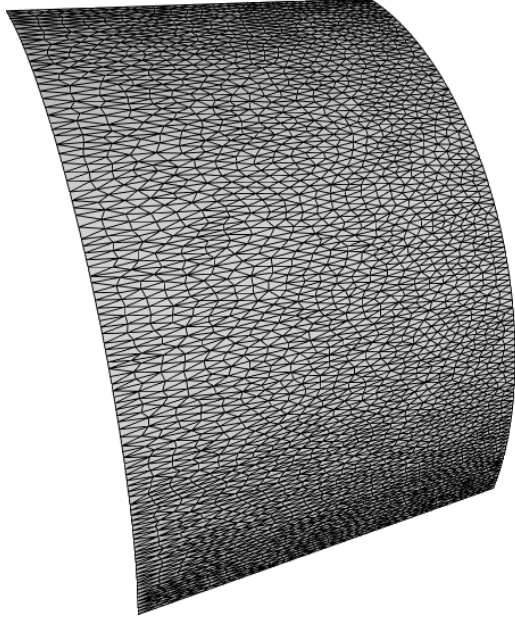
$$\mathcal{N} := \int_{\Gamma} d(\mathbf{x}) d\mathbf{x} = \int_{\Gamma} \sqrt{\det(R_s(\mathcal{M})(\mathbf{x}))} d\mathbf{x},$$

where $d = (h_1 h_2)^{-1}$ is the density and $(h_i)_{i=1,2}$ are the local sizes along the principal directions of $R_s(\mathcal{M})$. In this analytical example, α is a parameter that controls the complexity of the continuous mesh.

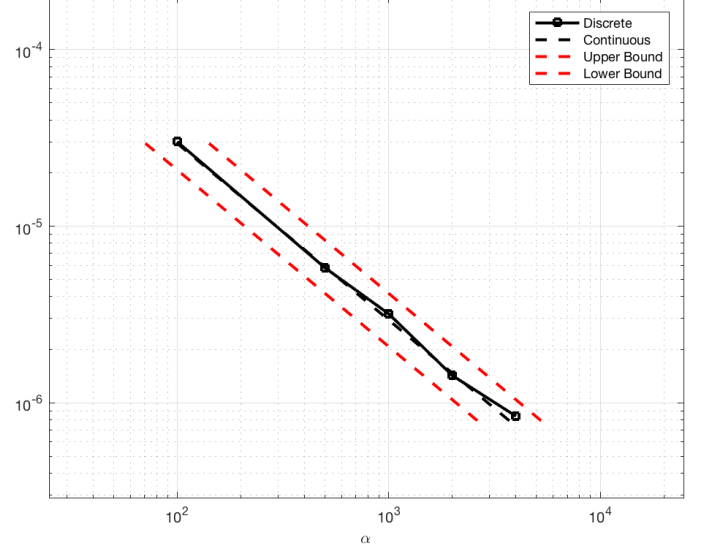
Given the quadratic function $u(x, y, z) = x^2 - y^2 + xz$, we can compute the right hand side of (18) using a formal integration. We have :

$$\int_S e_{R_s(\mathcal{M})}(\mathbf{a}) dS = \frac{0.01038536836\pi}{8\alpha}.$$

We then compare this continuous estimation with a discrete one based on a sequence of meshes generated with $\alpha = [100, 500, 1000, 2000, 4000]$. The interpolation error is computed with a numerical integration. Figure 4a represents the mesh for $\alpha = 100$. In Figure 4b, we plot the continuous and discrete errors. We also give an upper and lower bound as the mesh generator is not able to generate a perfectly unit-mesh but only quasi-unit meshes. An edge is considered to be quasi-unit when its length computed in the metric is in $[1/\sqrt{2}, \sqrt{2}]$. We observe that the discrete interpolation error fits the continuous one, and that the deviation is always contained within the lower and upper bounds (corresponding to the generation of a quasi-unit mesh instead of a perfectly unit one). This example confirms the relevance of the use of the tangent plane restriction. We now have all the tools to derive the mesh minimising the continuous interpolation error.



(a)



(b)

Figure 4: (a) Adapted mesh on the cylinder for $\alpha = 100$ and (b) comparison between a continuous and discrete evaluation of the boundary interpolation error.

3.4. Deriving the optimal continuous mesh

The introduction of the continuous interpolation is very important to find the optimal mesh, i.e., the mesh minimizing the error introduced in the BEM solution. Indeed, the initial discrete optimization problem reads:

Find the surface mesh \mathcal{H}_N^{opt} with N nodes, minimising the discrete interpolation error $\|u - \Pi_h u\|_{L^p(\Gamma_h)}$, (\mathcal{P}_d)

where Γ_h is the discrete support representing Γ . This problem is generally intractable practically and would require some simplifications. By replacing the discrete interpolation error with its continuous equivalent, we do not have anymore to deal with discrete quantities. Using the mathematical tools defined on the continuous mesh space instead of the discrete one, the optimization problem is simplified. It is the originality of this work (similarly to [38]), to solve this problem in a continuous setting. We consider in the following the continuous interpolation error controlled in L^p norm (with $p \geq 1$). The global optimization problem becomes to find the optimal continuous mesh \mathbf{M}^* minimizing the continuous interpolation error in the L^p norm:

$$\text{Given a complexity } \mathcal{N}, \text{ find } \mathbf{M}^* = \operatorname{argmin}_{\mathbf{M}} \left(\int_{\Gamma} \left(e_{R_s(\mathcal{M})}(\mathbf{x}) \right)^p d\mathbf{x} \right)^{1/p}. \quad (\mathcal{P}_c)$$

The optimization problem (\mathcal{P}_c) is well-posed and the interpolation error **is expressed continuously without the need to define a discrete support**. In the following section, (\mathcal{P}_c) is solved in two steps.

We need first to introduce some concepts of the continuous mesh framework (detailed in [38]). These meshes are represented by a Riemannian metric space $\mathbf{M} = (R_s(\mathcal{M})(\mathbf{x}))_{\mathbf{x} \in \Gamma}$. Locally, the spectral decomposition of $R_s(\mathcal{M})(\mathbf{x})$ is

$$R_s(\mathcal{M})(\mathbf{x}) = \mathcal{R}(\mathbf{x}) \Lambda(\mathbf{x})^t \mathcal{R}(\mathbf{x}),$$

where $\mathcal{R}(\mathbf{x})$ is an orthonormal matrix that provides the local orientation given by the eigenvectors $(\mathbf{v}_i(\mathbf{x}))_{i=1,2}$. The $(\lambda_i(\mathbf{x}))_{i=1,2}$ are the corresponding eigenvalues. As a result, the diagonal matrix Λ is either

$$\begin{bmatrix} \lambda_1(\mathbf{x}) & 0 \\ 0 & \lambda_2(\mathbf{x}) \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} h_1^{-2}(\mathbf{x}) & 0 \\ 0 & h_2^{-2}(\mathbf{x}) \end{bmatrix}.$$

Introducing the density $d = (h_1 h_2)^{-1}$ and the anisotropic quotients $r_i = h_i^2 (h_1 h_2)^{-1}$, we can write

$$R_s(\mathcal{M})(\mathbf{x}) = d(\mathbf{x}) \mathcal{R}(\mathbf{x}) \begin{bmatrix} r_1^{-1} & \\ & r_2^{-1} \end{bmatrix}^t \mathcal{R}(\mathbf{x}).$$

It follows from (17) that

$$e_{R_s(\mathcal{M})} = \frac{1}{8} \left(d^{-1}(\mathbf{a}) \sum_{i=1}^2 r_i(\mathbf{a}) {}^t \mathbf{v}_i(\mathbf{a}) |R_s(H)| \mathbf{v}_i(\mathbf{a}) \right),$$

where \mathbf{v}_i denote again the eigenvectors of the metric $R_s(\mathcal{M})$.

Optimal density and anisotropic ratios for a fixed orientation. It is now possible to solve the continuous optimization problem. We introduce $\gamma_i = {}^t \mathbf{v}_i(\mathbf{a}) |R_s(H)| \mathbf{v}_i(\mathbf{a})$, i.e., the directions of the continuous mesh. The first step in the solution of (\mathcal{P}_c) is to fix γ_1 and γ_2 . Since the density is constant, we can write $r_2 \sim r_1^{-1}$. The function $e_{R_s(\mathcal{M})}$ locally writes:

$$e_{R_s(\mathcal{M})} = \frac{d^{-1}}{8} (r_1 \gamma_1 + r_1^{-1} \gamma_2),$$

such that, we have to solve:

$$\min_{(r_1, d)} \int_{\Gamma} d^{-p} (r_1 \gamma_1 + r_1^{-1} \gamma_2)^p, \quad \text{under the linear constraint: } \int_{\Gamma} d = \mathcal{N},$$

where d is now an unknown of the optimization problem and the constraint is linear. In addition, the problem is uncoupled. It is thus natural to exhibit first the optimal anisotropic quotient r_1 and the optimal density in a second step.

To minimize $E_p(\mathbf{M})$ defined in (18), we use the calculus of variations. The classical Euler-Lagrange necessary condition states that for the variation $\delta \mathbf{M} = (\delta r_1, \delta d)$, we have:

$$\forall \delta r_1, \forall \delta d \text{ with } \int_{\Gamma} \delta d = 0, \quad \delta E_p(\mathbf{M}; \delta r_1) + \delta E_p(\mathbf{M}; \delta d) = 0. \quad (19)$$

If we note $\beta = (r_1 \gamma_1 + r_1^{-1} \gamma_2)$, (19) for $\delta d = 0$ leads to

$$\delta E_p(\mathbf{M}; \delta r_1) = \int_{\Gamma} p d^{-p} \beta^{p-1} (\gamma_1 - r_1^{-2} \gamma_2) \delta r_1 = 0. \quad (20)$$

It follows $r_1 = \sqrt{\gamma_2/\gamma_1}$, $r_2 = \sqrt{\gamma_1/\gamma_2}$ and $e_{R_s(\mathcal{M})}$ becomes

$$e_{R_s(\mathcal{M})} = \frac{d^{-1}}{8} (r_1 \gamma_1 + r_1^{-1} \gamma_1) = \frac{1}{4} d^{-1} (\gamma_1 \gamma_2)^{\frac{1}{2}}. \quad (21)$$

Then (19) for $\delta r_1 = 0$ leads to

$$\delta E_p(\mathbf{M}; \delta d) = \int_{\Gamma} p \frac{1}{4^p} d^{-p-1} (\gamma_1 \gamma_2)^{\frac{p}{2}} \delta d = 0 \quad \text{with the constraint } \int_{\Gamma} \delta d = 0. \quad (22)$$

A condition to ensure (22) is given by $d^{-p-1} (\gamma_1 \gamma_2)^{\frac{p}{2}} = C$ where C is a real constant. Using the constraint of the constant complexity \mathcal{N} , the final expression of d is

$$d = \mathcal{N} \left(\int_{\Gamma} (\gamma_1 \gamma_2)^{\frac{p}{2(p+1)}} \right)^{-1} (\gamma_1 \gamma_2)^{\frac{p}{2(p+1)}}. \quad (23)$$

Finally, the optimal continuous mesh \mathbf{M}^* obtained by solving (\mathcal{P}_c) is described by

$$\lambda_i^* = (h_i^*)^{-2} = \mathcal{N} \left(\int_{\Gamma} (\gamma_1 \gamma_2)^{\frac{p}{2(p+1)}} \right)^{-1} (\gamma_1 \gamma_2)^{-\frac{1}{2(p+1)}} \gamma_i, \quad \text{and } r_i^* = \frac{(\gamma_1 \gamma_2)^{\frac{1}{2}}}{\gamma_i} \quad \forall i = 1, 2. \quad (24)$$

Uniqueness. It is then possible to prove that the optimal continuous mesh \mathbf{M}^* given by (24) is the unique solution of (\mathcal{P}_c) verifying $E_p(\mathbf{M}^*)^p < E_p(\mathbf{M})^p$ for all \mathbf{M} having the same fixed $(\gamma_i)_{i=1,2}$.

Proof: Evaluating E_p to the power p at the optimal point \mathbf{M}^* , it becomes

$$E_p(\mathbf{M}^*)^p = \frac{1}{4^p} \mathcal{N}^{-p} \left(\int_{\Gamma} (\gamma_1 \gamma_2)^{\frac{p}{2(p+1)}} \right)^{p+1}. \quad (25)$$

A generic \mathbf{M} of complexity \mathcal{N} is given by its two anisotropic quotients $(r_i)_{i=1,2}$ and its density d . To take into account the constraint on the density, the density is rewritten as $d = \mathcal{N}(\int_{\Gamma} f)^{-1} f$, where f is a strictly positive function. The error committed with a generic \mathbf{M} is then

$$E_p(\mathbf{M})^p = \frac{\mathcal{N}^{-p}}{8^p} \left(\int_{\Gamma} f \right)^p \int_{\Gamma} f^{-p} (r_1 \gamma_1 + r_2 \gamma_2)^p.$$

To prove $E_p(\mathbf{M}^*)^p < E_p(\mathbf{M})^p$, we use the generalized arithmetic-geometric inequality which comes from the strict concavity of the logarithm function (since $r_1 \gamma_1 \neq r_2 \gamma_2$, otherwise it would be the optimal continuous mesh):

$$\ln \left(\frac{1}{2} (r_1 \gamma_1 + r_2 \gamma_2) \right) > \frac{1}{2} (\ln(r_1 \gamma_1) + \ln(r_2 \gamma_2)) = \ln \left(r_1^{\frac{1}{2}} \gamma_1^{\frac{1}{2}} r_2^{\frac{1}{2}} \gamma_2^{\frac{1}{2}} \right) = \ln \left(\gamma_1^{\frac{1}{2}} \gamma_2^{\frac{1}{2}} \right) \implies (r_1 \gamma_1 + r_2 \gamma_2)^p > 2^p \left(\gamma_1^{\frac{1}{2}} \gamma_2^{\frac{1}{2}} \right)^p,$$

where we have used that $r_1 r_2 = 1$. Finally, if we denote $g = (\gamma_1 \gamma_2)^{\frac{p}{2(p+1)}}$, we have

$$\begin{cases} E_p(\mathbf{M}^*)^{\frac{p}{p+1}} &= \frac{1}{4^{\frac{p}{p+1}}} \mathcal{N}^{-\frac{p}{p+1}} \int_{\Gamma} g, \\ E_p(\mathbf{M})^{\frac{p}{p+1}} &> \frac{1}{4^{\frac{p}{p+1}}} \mathcal{N}^{-\frac{p}{p+1}} \left(\int_{\Gamma} f \right)^{\frac{p}{p+1}} \left(\int_{\Gamma} f^{-p} g^{p+1} \right)^{\frac{1}{p+1}}. \end{cases}$$

Using the Hölder inequality, we have that

$$\begin{aligned} \left(\int_{\Gamma} f \right)^{\frac{p}{p+1}} \left(\int_{\Gamma} f^{-p} g^{p+1} \right)^{\frac{1}{p+1}} &= \left(\int_{\Gamma} f \right)^{\frac{p}{p+1}} \left(\int_{\Gamma} \left(\frac{g}{f^{\frac{p}{p+1}}} \right)^{p+1} \right)^{\frac{1}{p+1}} \\ &\geq \left(\int_{\Gamma} f^{\frac{p}{p+1}} \left(\frac{g}{f^{\frac{p}{p+1}}} \right) \right) = \int_{\Gamma} g, \end{aligned} \quad (26)$$

as $\frac{p+1}{p} \geq 1$, $p+1 \geq 1$ and $\frac{p}{p+1} + \frac{1}{p+1} = 1$. Relation (26) implies $E_p(\mathbf{M}^*) < E_p(\mathbf{M})$ for all \mathbf{M} having the same fixed γ_1, γ_2 . The optimal solution \mathbf{M}^* is thus unique. \blacksquare

Optimal orientations. The last step consists in deriving the optimal directions of the continuous mesh \mathbf{M}^* . We know from (21) and (23) that for a given set of directions $(\mathbf{v}_i)_{i=1,2}$, the optimal interpolation error reads:

$$E_p(\mathbf{M}^*) = \frac{\mathcal{N}^{-1}}{4} \left(\int_{\Gamma} (\det(\mathbb{G}))^{\frac{p}{2(p+1)}} \right)^{\frac{p+1}{p}},$$

where \mathbb{G} is the diagonal matrix composed of γ_1 and γ_2 . It is then possible to minimize the optimal interpolation error by seeking the directions \mathbf{v}_i minimizing $\det(\mathbb{G})^{\frac{p}{2(p+1)}}$, or equivalently minimizing $\det(\mathbb{G})$. Geometrically, $\det(\mathbb{G})$ is the area of the rectangle defined by $(\mathbf{v}_i)_{i=1,2}$ and of length $({}^t \mathbf{v}_i | R_s(H) | \mathbf{v}_i)_{i=1,2}$. We denote by $(\lambda_i)_{i=1,2}$ the eigenvalues of $|R_s(H)|$ and $(\mathbf{u}_i)_{i=1,2}$ its principal directions. The length of a side computed with respect to $|R_s(H)|$ is:

$${}^t \mathbf{v}_i | R_s(H) | \mathbf{v}_i = \lambda_1 ({}^t \mathbf{v}_i \mathbf{u}_1)^2 + \lambda_2 ({}^t \mathbf{v}_i \mathbf{u}_2)^2.$$

We denote by θ the angle between \mathbf{u}_1 and \mathbf{v}_1 (see Fig. 5). The problem is now to minimize

$$({}^t \mathbf{v}_1 | R_s(H) | \mathbf{v}_1) ({}^t \mathbf{v}_2 | R_s(H) | \mathbf{v}_2) = (\lambda_1 \cos^2 \theta + \lambda_2 \sin^2 \theta) (\lambda_1 \sin^2 \theta + \lambda_2 \cos^2 \theta) = (\lambda_1 - \lambda_2)^2 \sin^2 \theta \cos^2 \theta.$$

It results that $\det(\mathbb{G})$ is minimal when the vectors $(\mathbf{v}_i)_{i=1,2}$ are aligned with the principal directions of $|R_s(H)|$. Using (24), we conclude that the optimal metric both in sizes and directions is given by the following theorem.

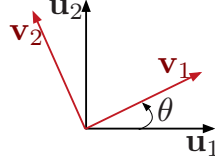


Figure 5: Determination of the optimal orientations of the continuous mesh.

Theorem 1. *Let u be a twice continuously differentiable function and H its Hessian, the optimal continuous mesh \mathbf{M}^* minimizing Problem (P_c) reads locally:*

$$R_s(\mathcal{M}^*) = D^* \det(|R_s(H)|)^{\frac{-1}{2(p+1)}} |R_s(H)|, \text{ with } D^* = \mathcal{N} \left(\int_{\Gamma} \det(|R_s(H)|)^{\frac{p}{2(p+1)}} \right)^{-1}. \quad (27)$$

\mathbf{M}^* is unique, locally aligned with the eigenvectors basis of $R_s(H)$ and has the same anisotropic quotients as $R_s(H)$. In addition, \mathbf{M}^* provides an optimal explicit bound of the interpolation error in the L^p norm (with $p \geq 1$):

$$E_p(\mathbf{M}^*) = \frac{\mathcal{N}^{-1}}{4} \left(\int_{\Gamma} \det(|R_s(H)|)^{\frac{p}{2(p+1)}} \right)^{\frac{p+1}{p}}. \quad (28)$$

We now have all the tools to generate in practice a discrete mesh approximating the continuous optimal solution. We can use any metric-based adaptive mesh generators as soon as the generated mesh is quasi-unit. In addition, we provide bounds on the discrete interpolation error when the continuous mesh is projected into the space of discrete meshes by means of an adaptive mesh generator. If the mesh generator achieves a quasi-unit mesh with respect to \mathbf{M}^* , the following bounds for the discrete interpolation error are obtained:

$$\frac{1}{2} E_p(\mathbf{M}^*) \leq \|u - \Pi_h u\|_{L^p(\Gamma_h)} \leq 2 E_p(\mathbf{M}^*),$$

or equivalently,

$$\frac{1}{8} \mathcal{N}^{-1} \left(\int_{\Gamma} \det(|R_s(H)|)^{\frac{p}{2(p+1)}} \right)^{\frac{p+1}{p}} \leq \|u - \Pi_h u\|_{L^p(\Gamma_h)} \leq \frac{1}{2} \mathcal{N}^{-1} \left(\int_{\Gamma} \det(|R_s(H)|)^{\frac{p}{2(p+1)}} \right)^{\frac{p+1}{p}}. \quad (29)$$

In other words, \mathbf{M}^* defined by (27) allows us to generate an optimal adapted mesh to control the interpolation error in the L^p norm. One main advantage of this approach is to be completely generic, independent from the PDE and the numerical scheme.

3.5. Application to solutions given by a BEM approximation

The above results are valid for the linear interpolant $\Pi_h u$ of the exact solution u . However, neither $\Pi_h u$ or u are known. Only the approximation of u obtained by the BEM is at hand. In order to relate the interpolation error to the approximation error, we consider a reconstruction operator \mathcal{R}_h which is applied to the numerical approximation u^h . This operator can be a recovery process, a hierarchical basis, or an operator connected to an a posteriori estimate. We make the assumption that the reconstruction $\mathcal{R}_h u^h$ is better than u^h for a given norm $\|\cdot\|$, i.e., that

$$\|u - \mathcal{R}_h u^h\| \leq \alpha \|u - u^h\|, \quad \text{where } 0 \leq \alpha < 1.$$

Then, using the triangle inequality, we have

$$\|u - u^h\| \leq \frac{1}{1-\alpha} \|u^h - \mathcal{R}_h u^h\|.$$

If $\Pi_h \mathcal{R}_h \phi^h = \phi^h$ for all $\phi^h \in \mathbb{P}^1$, i.e., the reconstruction operator is exact at the nodes, then

$$\|u - u^h\| \leq \frac{1}{1-\alpha} \|\mathcal{R}_h u^h - \Pi_h \mathcal{R}_h u^h\|.$$

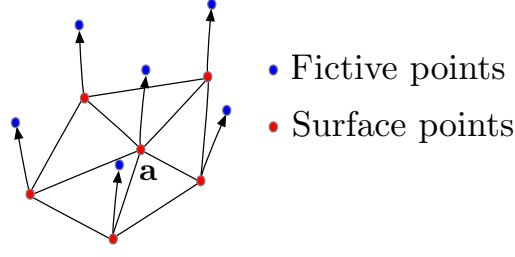


Figure 6: Fictive points created to ensure the tubular neighborhood of \mathbf{a} . The solution at blue node is set to the surface solution.

Then we have from (29) that

$$\|u - u^h\| \leq \frac{1}{2(1-\alpha)} \mathcal{N}^{-1} \left(\int_{\Gamma} \det(|R_s(H_{\mathcal{R}_h u^h})|) \right)^{\frac{p}{2(p+1)}} \frac{p+1}{p}.$$

Consequently we apply Estimate (28) to $\mathcal{R}_h u^h$.

There are numerous techniques in the literature to approximate or “recover” the Hessian (see, e.g., [43, 46]). From a practical point of view, we only need to recover a pointwise 3D Hessian at the nodes of the current surface mesh. The method we employ here is based on a local quadratic recovery and uses the boundary solution only. The lifting solution is deduced from u_s , following the assumption of Section 3.1. The starting point is the quadratic approximation of the lifting function u at a point $\mathbf{a} = (x_a, y_a, z_a)$ located on the surface of the tubular neighborhood (such that we have $\mathcal{P}(\mathbf{a}) = \mathbf{a}$):

$$u(\mathbf{x}) \approx u_s(\mathbf{a}) + {}^t \nabla u(\mathbf{a}) \mathbf{dx} + \frac{1}{2} {}^t \mathbf{dx} H(\mathbf{a}) \mathbf{dx},$$

where $\mathbf{dx} = \mathbf{x} - \mathbf{a}$. The gradient of u being a 3D vector, it reads

$${}^t \nabla u(\mathbf{a}) = (a \ b \ c).$$

Similarly, the Hessian is a 3×3 symmetric matrix:

$$H(\mathbf{a}) = \begin{bmatrix} d & e & f \\ e & g & h \\ f & h & i \end{bmatrix}.$$

At each surface mesh node, we thus have 9 unknowns to recover. The simple approach we use is a least square. We start with the patch composed of the elements adjacent to the node \mathbf{a} , see Figure 6. The set of points is then augmented by creating points along the normal direction to the surface. This extruded geometry is depicted in Figure 6 and uses standard boundary layer mesh generation procedures [40] (especially at complex corners or along ridges). The normal spacing is set to the minimal edge length of the patch of the current point. If the number of unknowns of this augmented system is not sufficient, an additional layer of neighbours is taken into account.

3.6. Refinement strategy

All the numerical examples shown in the following section use the same refinement strategy. Given a user-prescribed sequence of complexity $\mathbf{N} = [\mathcal{N}_1, \dots, \mathcal{N}_k]$, we seek for the sequence of corresponding optimal meshes $\mathbf{H} = [\mathcal{H}_1, \dots, \mathcal{H}_k]$. This process is non-linear, i.e., both the mesh and the solution have to be converged. The following iterative algorithm is used to generate \mathcal{H}_i :

Start from the mesh $\mathcal{H}_i^0 = \mathcal{H}_{i-1}$ (or from the initial uniform mesh \mathcal{H}_0 at iteration 0).

for all $k=0 : n_{iter}$ **do**

 Compute the approximation u^h on the mesh \mathcal{H}_i^k with the iterative FM-BEM solver (software `COFFEE`).

 Compute the recovered Hessian from u^h and deduce Metric (27) at the nodes of the mesh, with complexity \mathcal{N}_i .

 From (27), a new quasi-unit mesh \mathcal{H}_i^{k+1} of complexity \mathcal{N}_i is generated with `FEFLO.A` [37].

end for

For each case, the complexity sequence is always defined as $\mathcal{N}_{i+1} = 2\mathcal{N}_i$ with $n_{iter} = 5$.

4. Validation and enhancements of the iterative FM-BEM solver

In this section we first validate the efficiency of the FM-BEM solver at hand, **COFFEE**. We also describe several necessary enhancements to improve the efficiency of the FM-BEM solver when anisotropic elements are present in the mesh. This concerns the change of the initial guess for the GMRES solver along with the design of a simple but efficient preconditioner.

4.1. Validation of the fast BEM solver

The fast BEM solver for 3D Helmholtz problems, used in this article has been implemented in our in-house solver **COFFEE** [17]. Since so far the code has only been validated for 3D elastodynamic problems, this section is devoted to the verification of the accuracy and numerical efficiency of the solver for the scalar acoustic case of interest here. To this end, we consider the simple case of the diffraction of an incident plane wave by a unit sphere for which an analytical solution is known [13].

In this section, we consider a uniform triangulated mesh with a density of points per wavelength equal to 10. The tolerance of the GMRES iterative solver is set to 10^{-4} . In all the numerical examples presented, we use the CERFACS GMRES package [29]. We report in this section only results for the EFIE with Dirichlet boundary conditions and similar results are obtained for the MFIE with Neumann boundary conditions. In Table 1, we report the number of degrees of freedom N together with the number of iterations required to achieve convergence in GMRES for both the standard BEM and the FM-BEM solvers. The two solvers are seen to converge with the same rate. Importantly, we were not able to solve problems larger than about 50 000 DOFs with the standard BEM due to time and memory constraints. We also report the relative L^2 -norm of the discrepancy between the computed solution and the analytical one, computed in the domain (on a circle of radius 2 enclosing the obstacle). Both solvers give the same accuracy.

N k	2 562	10 242	40 962	328 606	626 333
# iter (BEM)	18	37	67	/	/
L^2 error (BEM)	$3.66 \cdot 10^{-3}$	$2.01 \cdot 10^{-3}$	$1.12 \cdot 10^{-3}$	/	/
# iter FM-BEM	18	38	67	366	387
L^2 error (FM-BEM)	$3.66 \cdot 10^{-3}$	$2.01 \cdot 10^{-3}$	$1.12 \cdot 10^{-3}$	$8.09 \cdot 10^{-4}$	$7.65 \cdot 10^{-4}$

Table 1: Validation of the FM-BEM solver for 3D Helmholtz equations: comparison of the accuracy and convergence of the iterative BEM and FM-BEM solvers.

In addition, to check the numerical efficiency of the solver, we report in Figure 7 the theoretical and numerical complexities in terms of computational time for one matrix-vector product. The black circles represent the computational time for the standard BEM and the red squares represent the time for the FM-BEM. As expected, the standard BEM is seen to be of the order of $\mathcal{O}(N^2)$ operations per iteration and the FM-BEM of the order of $\mathcal{O}(N \log N)$ operations per iteration.

4.2. Enhancements of the FM-BEM for anisotropic meshes and adaptivity

One important point in the adaptive process is the ability of the iterative solver to converge to the solution within the desired threshold. For small-scale examples, the FM-BEM is not necessary (only the standard BEM) and the number of GMRES iterations required to solve the matrix system is not a pressing issue. However, as soon as large-scale problems are considered, the use of the FM-BEM is mandatory to compute the solution on a given mesh. The number of GMRES iterations is known to increase with the size of the problem [18]. The presence of anisotropic elements in the mesh also increases the number of GMRES iterations. As a result, the number of iterations may be a limiting factor when employing the FM-BEM on anisotropic meshes. Here we discuss two simple methods to reduce the number of GMRES iterations in the context of a mesh adaptation strategy. The first one is a diagonal scaling preconditioner and the second one is based on the use of an initial guess for the GMRES obtained from the previous level of adaptation. To illustrate the efficiency of these preconditioners, we consider the spherical resonator test case: a screen curved into a sphere with a circular aperture to create a resonator. The sphere has unit radius and the circular opening of radius $\pi/10$ has its centre on the positive x-axis, see Figure 13a.

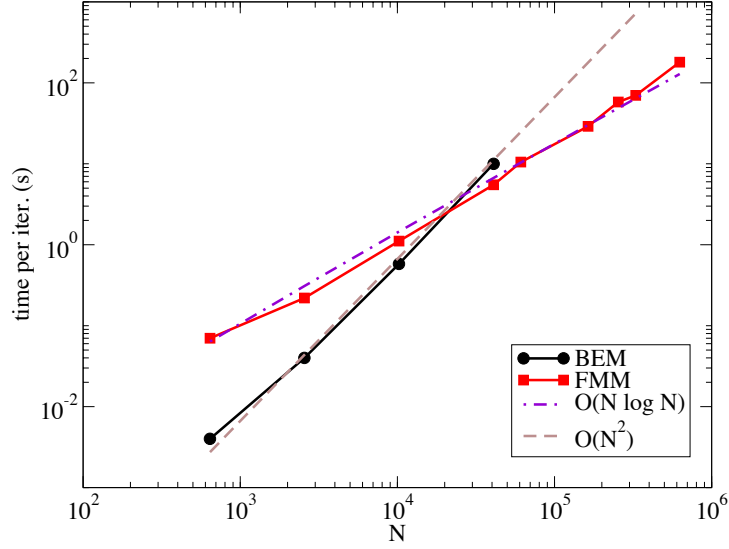


Figure 7: Comparison of the theoretical and numerical complexities in terms of computational time for one matrix-vector product. The black circles represent the computational time for the standard BEM and the red squares represent the time for the FM-BEM. In addition, we report the two theoretical complexities.

Preconditioning with a diagonal scaling. When performing BEM calculations on adapted meshes containing elements with drastically different sizes, the diagonal and off-diagonal entries will also vary drastically. A simple way to avoid this issue is to introduce a preconditioner which scales all the diagonal entries of the system matrix to 1. We first write the matrix system to be solved as

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

We propose the following right-preconditioning strategy with the preconditioner \mathbf{M} defined as

$$\mathbf{M}_{ij} = \begin{cases} \mathbf{A}_{ij}, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \quad (30)$$

Therefore the preconditioned system to be solved becomes:

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = \mathbf{M}^{-1}\mathbf{y}. \quad (31)$$

We report in Table 2 the number of GMRES iterations required to achieve convergence with a tolerance of 10^{-4} with and without this preconditioner. We see that the gain in terms of number of iterations and computational times increases with the sizes of the meshes, to reach a reduction by a factor 4.

#DOFs	# iter without Prec.	# iter with Prec.
152	43	30
323	57	37
898	77	50
2 051	124	61
4 031	180	74
8 163	257	90
16 932	412	104

Table 2: Spherical resonator: Number of GMRES iterations required to achieve convergence with a tolerance of 10^{-4} both without and with the diagonal scaling preconditioner (30).

Using an appropriate initial guess for GMRES. A further advantage of an adaptive mesh refinement is the possibility to use a computed approximation at a previous level of refinement as an initial guess for the GMRES solver. This

choice is expected to be more efficient than a guess chosen a priori, such as the zero vector or the incident wave evaluated on the surface. The simple modification consists in interpolating the previous solution onto the newly generated mesh (see the refinement strategy, Section 3.6). In our case, this is done as an additional post-processing step in **FEFLO.A**. In Table 3, we illustrate the improvement obtained with such an initial guess. More precisely, we consider the same resonator problem as above and solve the system at each step of refinement twice, once with the zero vector as the initial guess and once with the linear interpolation of the previous step numerical solution onto the current mesh as the initial guess. Again the tolerance for the GMRES solver is 10^{-4} . The gain is less impressive than the one obtained with the diagonal scaling preconditioner, but an additional gain of 10% is obtained.

#DOFs	# iter without Ini. Guess	# iter with Ini. Guess
152	30	30
323	37	38
898	50	45
2 051	61	52
4 031	74	63
8 163	90	77
16 932	104	90

Table 3: Spherical resonator: number of iterations (with the diagonal scaling preconditioner) with and without the use of an interpolated initial guess.

To sum up, the preconditioning by scaling the diagonal entries of the matrix yields a large reduction in the number of GMRES iterations. Furthermore, this reduction is greater for meshes with larger anisotropic ratios. Next we saw that the initialization of GMRES using the previous solution also serves to save a number of iterations, a saving which increases with the size and degree of anisotropy of the mesh.

5. Validation of the adaptive mesh strategy for the FM-BEM

We illustrate in this section the capabilities of the adaptive mesh strategies for different 3D configurations: scattering of plane waves by open surfaces (a planar screen and a sphere with aperture), by a closed simple shape (a cube with cavity [24]), and by a large complex structure (an aircraft).

5.1. Planar screen

We consider the scattering of plane waves (with unit amplitude and direction $\mathbf{d} = (-1, 0, 0)$) by a sound-soft unit screen centered at the origin, i.e., the screen occupies the region $[-\frac{1}{2}, \frac{1}{2}] \times [-\frac{1}{2}, \frac{1}{2}]$. Such configurations lead to surface solutions with the most severe singular behavior, and are a good illustration of the capabilities of an adaptive mesh strategy.

It is well known (see, e.g., [36]) that at the edge of a screen, we have

$$\left[\frac{\partial u}{\partial \mathbf{n}} \right] \sim (kr)^{-0.5} \quad \text{as } kr \rightarrow 0, \quad (32)$$

where r is the distance from the edge. At the corner of the screen, this singularity is more severe, and was shown in [8] to take the form

$$\left[\frac{\partial u}{\partial \mathbf{n}} \right] \sim (kr)^{-0.704} \quad \text{as } kr \rightarrow 0.$$

The optimal convergence rate here (for our \mathbb{P}_1 discretization) is $\mathcal{O}(N^{-1})$. Owing to the singular behavior, the approximated solution on a sequence of uniformly refined meshes should yield a sub-optimal convergence rate.

We employ the adaptive strategy proposed in Section 3.6 for three wavenumbers ($k = 5, 20, 80$) and compare the results to those obtained with a sequence of uniformly refined meshes. In figures 8b-d, we present the relative L^2 errors in the scattered field evaluated on a circle in the (x, y) -plane of radius 2 (Figure 8a).

We observe that the use of the anisotropic adaptive strategy improves the computational efficiency for this scattering problem by almost an order of magnitude. For example, for $k = 5$, to achieve an accuracy of $1.37 \cdot 10^{-2}$ with a uniform refinement 3 622 DOFs are required while the anisotropic refinement produced an approximation with an accuracy of $1.78 \cdot 10^{-2}$ with only $N = 271$ DOFs. For $k = 20$, to achieve an accuracy of $1.12 \cdot 10^{-2}$ with a uniform refinement 13 929 DOFs are required while the anisotropic refinement produced an approximation with an accuracy of $1.06 \cdot 10^{-2}$ with

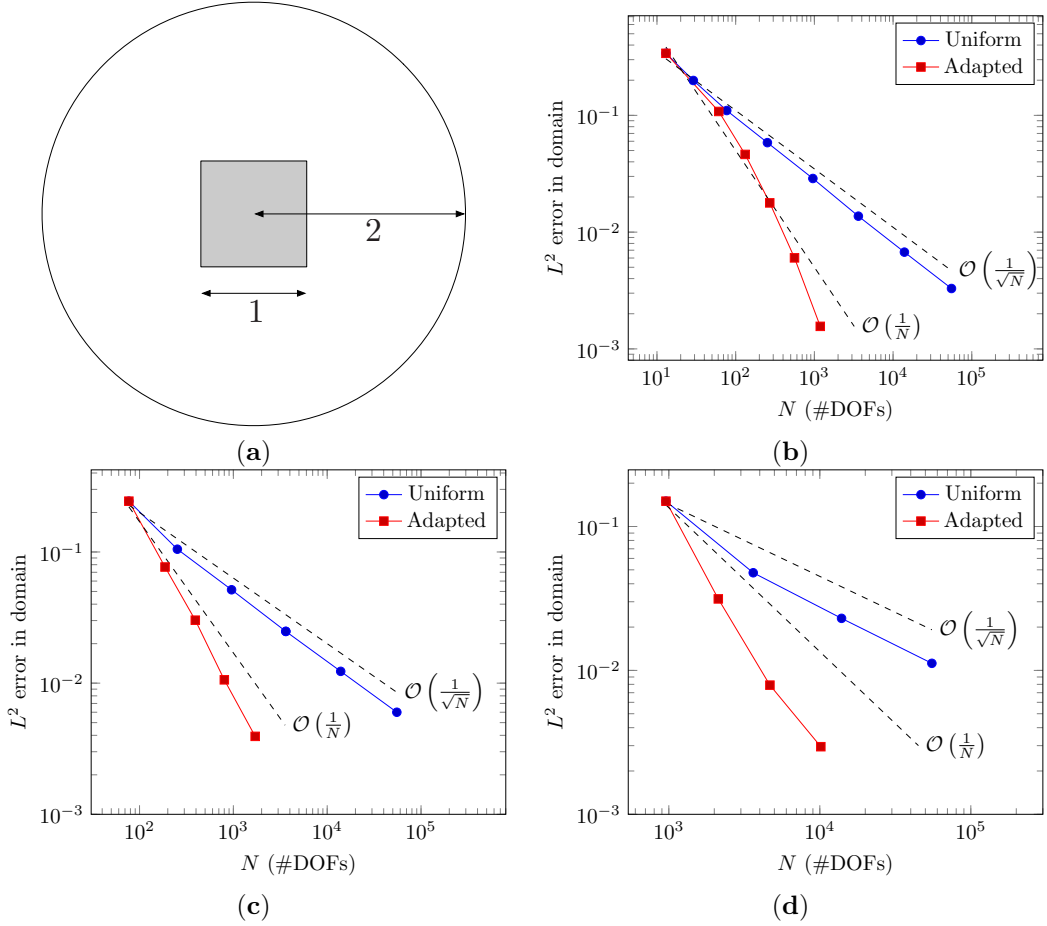


Figure 8: Scattering by a screen: Relative L^2 errors in the scattered field computed on the circle (a) for uniform and adapted meshes for three wavenumbers: (b) $k = 5$, (c) $k = 20$ and (d) $k = 80$.

only 801 DOFs. Finally, for $k = 80$, to achieve an accuracy of $1.12 \cdot 10^{-2}$ with a uniform refinement 55 224 DOFs are required while the anisotropic refinement produced an approximation with an accuracy of $7.9 \cdot 10^{-3}$ with only 4 672 DOFs. In addition, for all three wavenumbers the uniform refinement yields a convergence rate of approximately $\mathcal{O}(N^{-0.5})$ in the relative L^2 error. This sub-optimality is related to the prominent edge singularities of the form (32). While, with the adaptive mesh strategy, the optimal convergence rate of $\mathcal{O}(N^{-1})$ is recovered. In fact, it appears that we even improve slightly on this rate, especially at the higher wavenumbers. This improvement can be attributed to the exploitation of the directionality of the wave propagation.

In Figure 9, we depict the final surface mesh for $k = 5$ and $k = 20$. For $k = 5$ (resp. $k = 20$), the mean anisotropic ratio is around 30 (resp. 60). If an isotropic error estimate was used, the obtained mesh would contained 55 608 DOFs, i.e., 15 times more DOFs than the anisotropic mesh for $k = 20$ at 5th iteration. This also emphasizes the advantage of using anisotropic estimates over isotropic estimates.

Finally, the level of anisotropy on the sides of the screen increases with the complexity, see Figure 10, showing the strong directionality of the error.

5.2. Cube with cavity

We next consider the scattering by a cube with cavity [24] which is the 3D analogue of the 2D trapping domain [10]. We use the adaptive strategy for the sound-soft problem and the incident direction is $\mathbf{d}^i = (-2, -1, -1)$ throughout. The singularities in the boundary solution for this problem should be less severe than for the screen. At an edge, the unknown should behave as

$$\frac{\partial u}{\partial \mathbf{n}} \sim r^{-1/3}.$$

The general rule is that we expect the acoustic pressure u , to behave near an edge as

$$u \sim r^{\pi/\alpha} \quad \text{as } r \rightarrow 0,$$

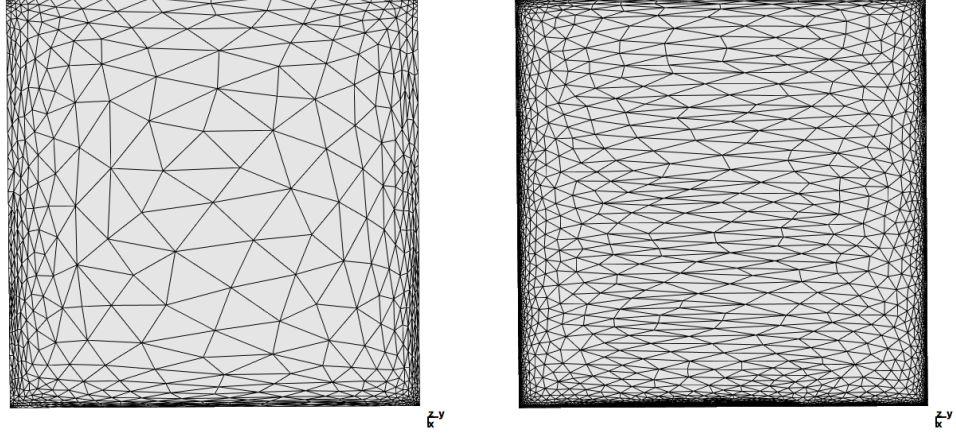


Figure 9: Screen case: final adapted meshes for $k = 5$ (left) and $k = 20$ (right).

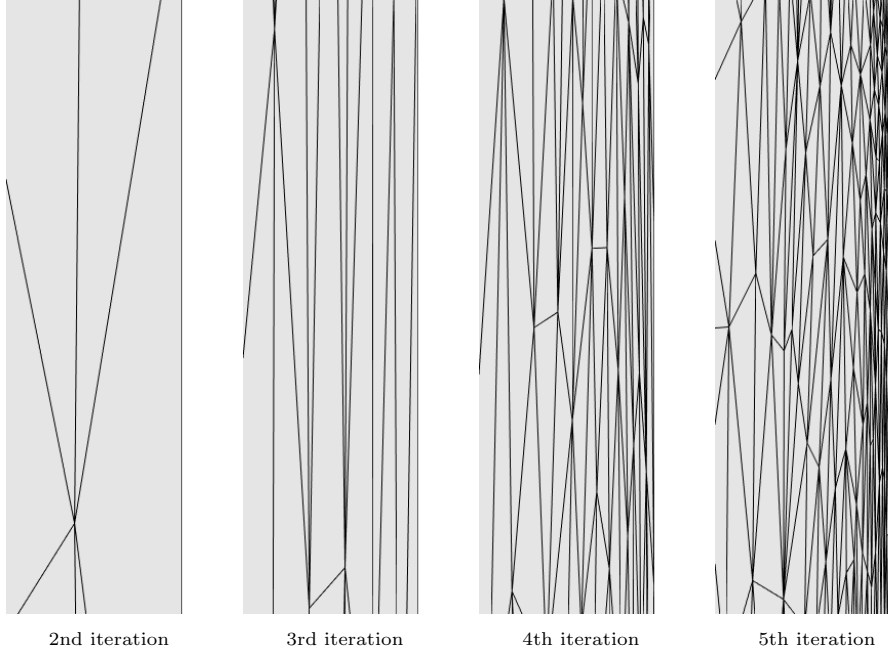


Figure 10: Screen case: evolution of the anisotropic ratio from 2nd to 5th iteration for $k = 20$ at side $x = 0.5$.

where r is the distance from the edge and α is the exterior angle subtended at the edge. Hence the flow velocity will behave as

$$\frac{\partial u}{\partial \mathbf{n}} \sim r^{\pi/\alpha-1} \quad \text{as } r \rightarrow 0.$$

Again, the errors for the adapted and uniform meshes are computed on a circle of radius 2 and are shown in Figure 11**b-d** for the wavenumbers $k = 5, 10, 20$. Similar to the screen problem, the uniform refinement yields a suboptimal convergence rate. However, this time the rate is closer to $\mathcal{O}(N^{-2/3})$, i.e., superior to that for the screen due to a less severe singularity.

In Figure 12 we display the mesh at the fourth step of the adaptive process and the real part of the corresponding BEM approximation of the boundary solution. We note, on the top of the cube, the refinement towards the ridges as well as the exploitation of the directionality. To estimate the gain due to the level of anisotropy, we have generated the equivalent isotropic mesh having the minimal sizes of the anisotropic mesh. The isotropic mesh is composed of 19 409 DOFs, i.e., 4 times the number of DOFs in the equivalent anisotropic mesh.

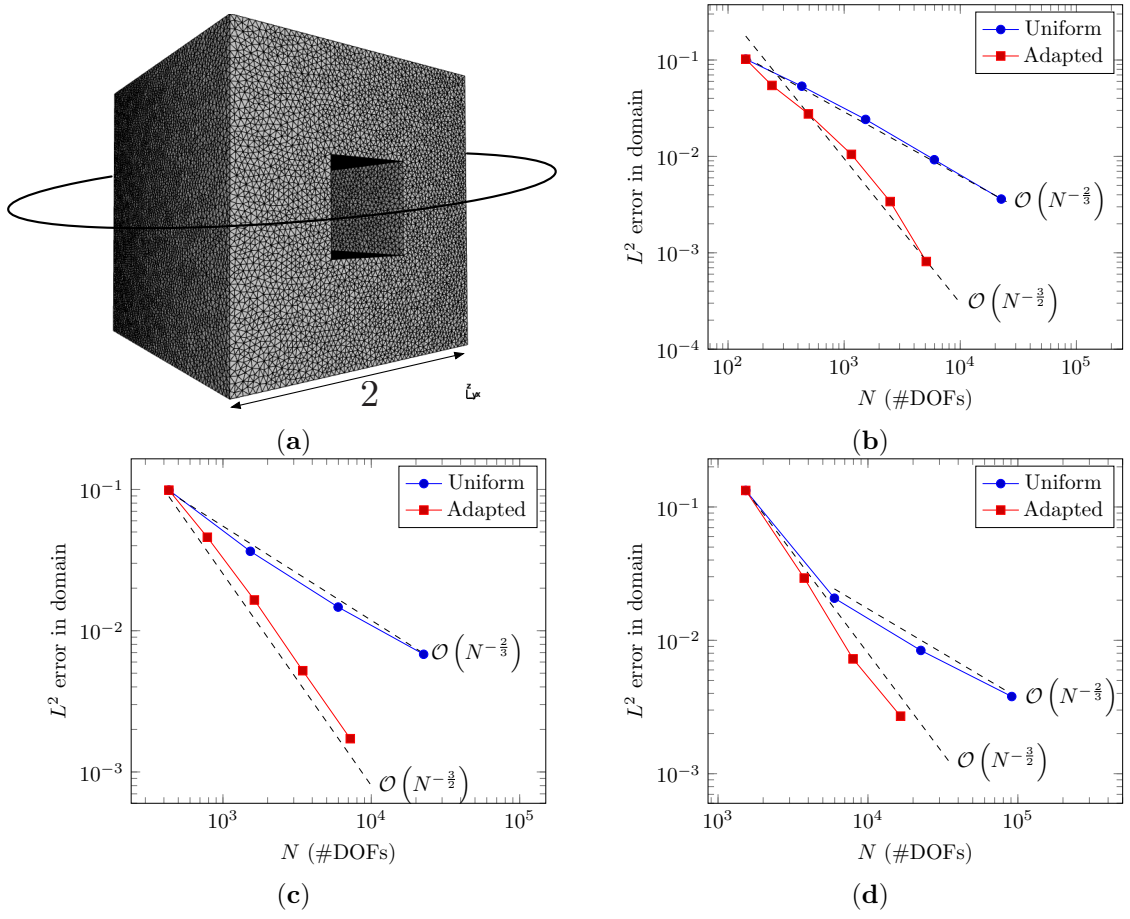


Figure 11: Scattering by a sound-soft cube with cavity: Relative L^2 errors for uniform and adapted meshes for (b) $k = 5$, (c) $k = 10$ and $k = 20$.

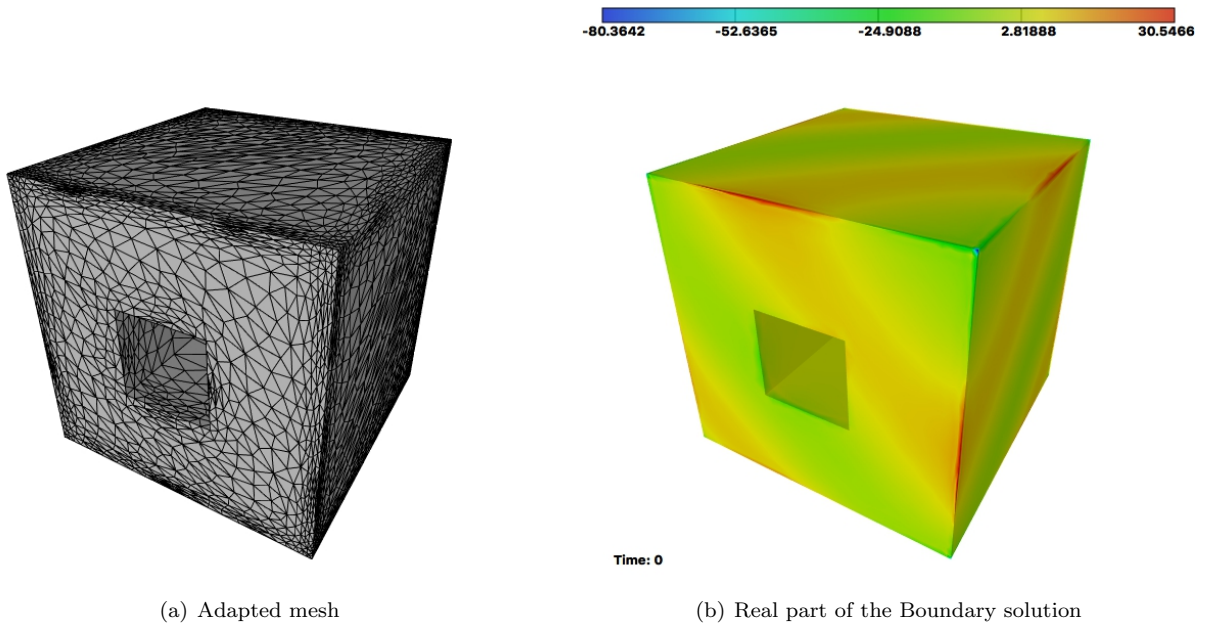


Figure 12: Cube with cavity: Adapted mesh and real part of the boundary solution at fourth step of the adaptation for the sound-soft cube with cavity. The incident wave has direction $\mathbf{d}^i = (-2, -1, -1)$ and $k = 5$.

5.3. Spherical resonator

We consider again the case of a spherical resonator (Figure 13a). Such objects are of interest in the context of anisotropic mesh adaptation because they require the retention of some level of isotropy in the refinement in order to maintain a good approximation of the surface by planar triangles. The incident wave is a plane wave travelling in the negative x -direction with a wavenumber of $k = 10$. In Figure 13b, we represent the modulus of the solution computed in a 2D planar slice of the domain (the screen in the (x, y) plane with $z = 0$) using the boundary integral representation (5).

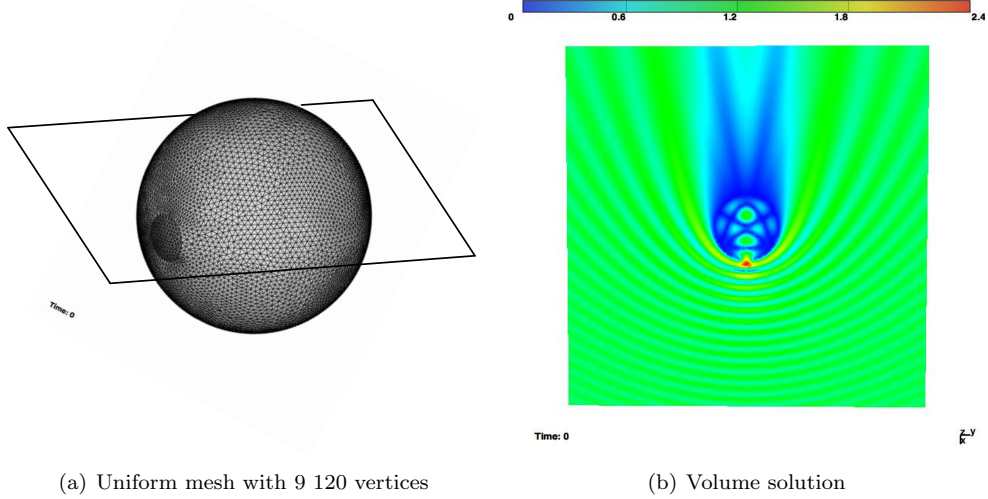


Figure 13: Spherical resonator: (a) Uniform mesh of the spherical resonator and screen where the volume solution is computed; (b) corresponding modulus of total field in the (x, y) -plane.

Since this example is an isotropic example, we perform an experiment demonstrating the capabilities of our mesh adaptation technique with three configurations: with full anisotropy (similarly to what is presented in previous examples), and two additional ones with different levels of isotropy enforced. The first level is called “semi-isotropy” and the second one is a fully isotropic case. For the semi-isotropy approach, the maximum dimension of the adapted triangles is limited accordingly to the complexity. Precisely, we enforce

$$\max(h) = \frac{C}{\sqrt{N}},$$

where h is the maximum dimension of each triangle. This restricts the elongation of the elements but still permits some elongation controlled by the parameter C . In this work, C is set to 3.5 after some numerical experimentation.

In Figure 14, we give some meshes and corresponding solutions obtained during the adaptation process for the three strategies.

In Figure 15, we report the relative L^2 errors for uniform and (anisotropic, isotropic and semi-isotropic) adapted meshes with $k = 5$. Similarly to the case of the planar screen, a uniform refinement yields a convergence rate of $\mathcal{O}(N^{-\frac{1}{2}})$. The anisotropic and semi-isotropic adaptive strategies initially achieve a rate of $\mathcal{O}(N^{-\frac{3}{2}})$ which is super optimal, however this rate begins to slow as higher accuracies are reached. The reason is that the shape to which the refinement is leading is not the correct one due to the elongated elements, see Figure 14. By enforcing some isotropy the convergence rate of $\mathcal{O}(N^{-\frac{3}{2}})$ is maintained up to a larger mesh. However, it still slows once the relative error falls below approximately 10^{-3} . On the other hand, for the fully isotropic refinement, which restricts the height to width ratio of the triangles, at first the convergence rate is similar to the one obtained by a uniform refinement then for accuracies smaller than 10^{-2} , the convergence rate is $\mathcal{O}(N^{-\frac{3}{2}})$. Furthermore, it appears that the isotropic refinement maintains this rate asymptotically since it is also refining towards the spherical shape accurately as well as resolving the edge singularity. Nevertheless, it appears that if one is interested in achieving accuracies of the order of 10^{-2} or 10^{-3} , the anisotropic adaptive strategy is superior.

This spherical example illustrates the limitations of the current fully anisotropic mesh strategy. The potential anisotropy, provided by the error estimate, has to comply with the surface principal curvatures. In this example, the full anisotropic error estimate implies a deformed sphere that the solver is not able to recover. The semi-anisotropic approach reduces the anisotropy to limit this artifact. For more complex geometries, it is then necessary to intersect

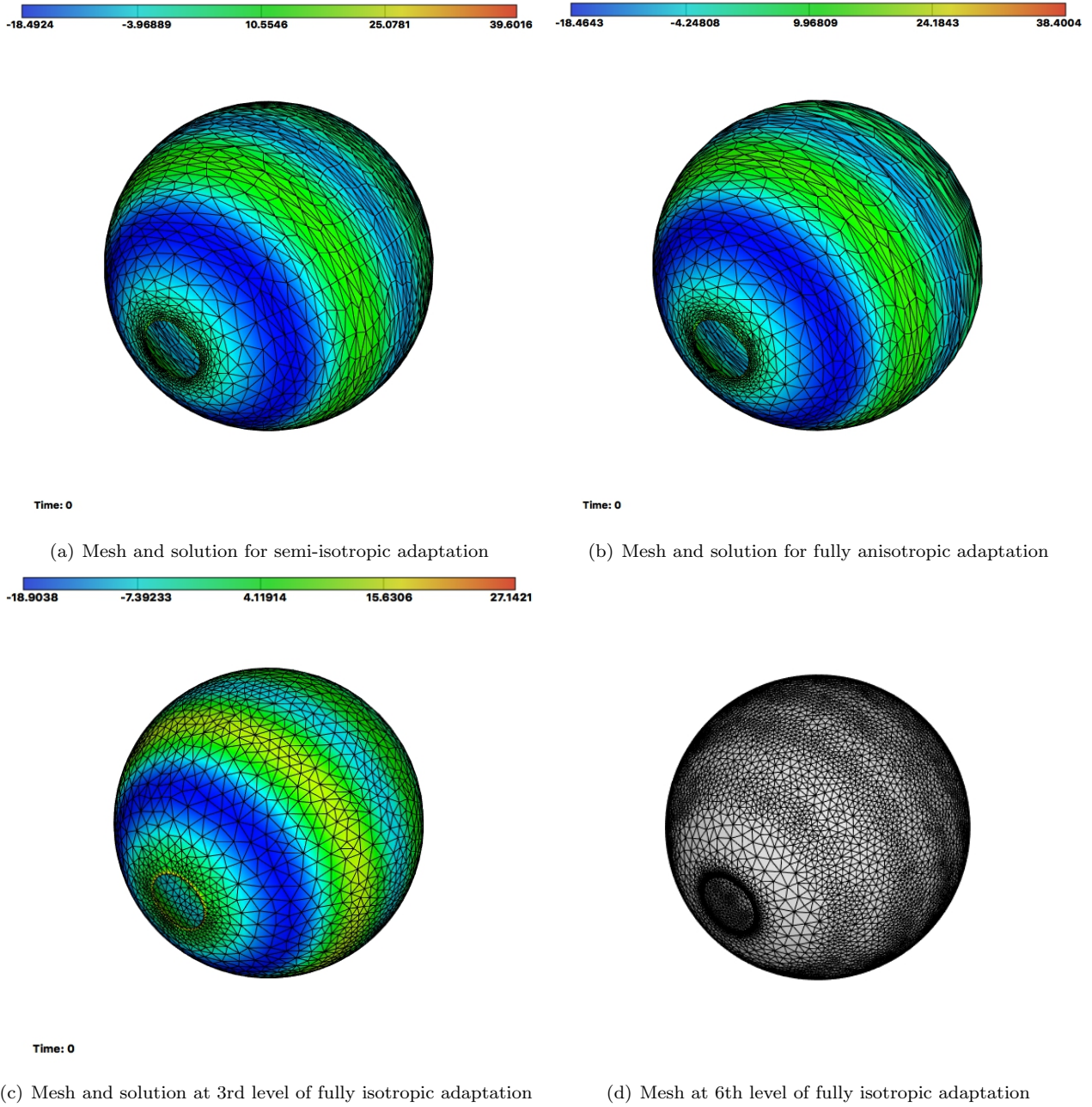


Figure 14: Spherical resonator: Adapted meshes with real part of the surface solution for (a) semi-isotropic, (b) fully anisotropic (c) fully isotropic adaptation strategies, and (d) mesh at 6th level of fully isotropic adaptation.

the metric provided by the error estimate with a geometric metric representing the surface curvatures, see [37]. We use such a metric for the following example. It is clear from this example that defining a high-order adaptive mesh strategy, for instance for \mathbb{P}_2 interpolation, is necessary to recover anisotropy while preserving the geometry.

5.4. F15 aircraft

In this test case, we illustrate the robustness of the technique with respect to the complexity of the input geometry. We consider the geometry of an unarmed F15 aircraft, depicted in Figure 16a. The computation is started with the uniform mesh of Figure 16b composed of almost 80 000 vertices and 163 000 triangles. The incident plane wave has direction $\mathbf{d}^i = (1/2, 1, 0)$ and wavenumber $k = 6$. In this numerical experiment, 15 adapted meshes are generated. The imposed complexity is kept constant for 3 iterations, and then it is multiplied by a factor 2. The prescribed complexities are

$$\{10\,000, 20\,000, 40\,000, 80\,000, 160\,000\}.$$

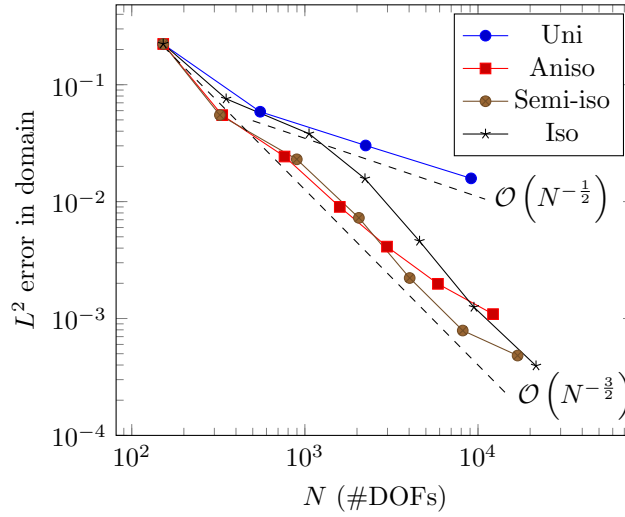


Figure 15: Scattering by a spherical resonator: Relative L^2 errors for uniform and (anisotropic, isotropic and semi-isotropic) adapted meshes with $k = 5$.

Iteration	# vertices	# triangles	# Solver iterations	Meshing CPU Time	Solver CPU time
3	10 562	21 120	392	42 s	19 mn 47 s
6	15 687	31 370	521	57 s	30 mn 36 s
9	25 428	50 852	707	1 mn 34 s	52 mn 25 s
12	43 217	86 430	593	2 mn 44 s	65 mn 38 s
15	80 611	161 218	364	5 mn 13 s	70 mn 53 s

Table 4: F15 aircraft statistics of the adapted meshes, solver iteration and CPU times.

The real part of the boundary solution is represented in Figure 17. We see the high directionality of the diffracted waves with different patterns for the front and back part of the aircraft. In Table 4, we report the sizes of the generated meshes. We observe that the size is almost doubled at each sequence of fixed complexity. Note that in this case, a surface metric is intersected with Metric (27) with $p = 2$ to guarantee a good surface resolution. The expression of the surface metric is described in [37]. The meshing CPU time is the combined time taken to evaluate the error estimate, generate the adapted mesh, and interpolate the solution from the previous mesh to the current mesh. This interpolated solution is used as a restart for the acoustic solver. The number of solver iterations ranges from 300 to more than 700. The maximal number of iterations is obtained at the 9th iteration, where new wave reflections are captured in the mesh. For the subsequent iterations, the number of solver iterations is almost constant, showing that no more new scales in the solution appear with smaller meshes. In term of sizes, we report in Table 5 the minimal size of the anisotropic mesh. To evaluate the gain of using an anisotropic approach, an isotropic and uniform mesh have been generated with the same level of discretization as its anisotropic counterpart. The factors provided in Table 5 are the multiplication factors needed to obtain the sizes of the isotropic and uniform meshes, respectively. We observe that if an isotropic approach was used, the size of the meshes would be approximately 3 times greater. And if uniform meshes were employed, the size of the meshes to achieve the same accuracy would be between 300 and 6 million times greater! For instance, the uniform mesh having the same accuracy than the 15th mesh is composed of more than 500 million vertices. This illustrates the optimality of the anisotropic approach in term of ratio accuracy/CPU time. Figure 18 shows closer views of the mesh (at the last iteration of the adaptive process) near the singularities of the geometry of the aircraft. Most of the anisotropy is reached in these areas when the diffracted waves have low regularity. Finally in Figure 19, we display the total field in the planes $y = 0$ and $z = 0$.

6. Conclusion

This paper details the first extension of a metric-based anisotropic mesh adaptation strategy to the boundary element method for 3D acoustic wave propagation problems. Following the works of [38, 39] for 3D volume solutions, we have derived the optimal continuous mesh minimizing the \mathbb{P}_1 interpolation error for 3D surface solutions. The method proposed is independent of the discretization technique (e.g., collocation, Galerkin) and independent of the

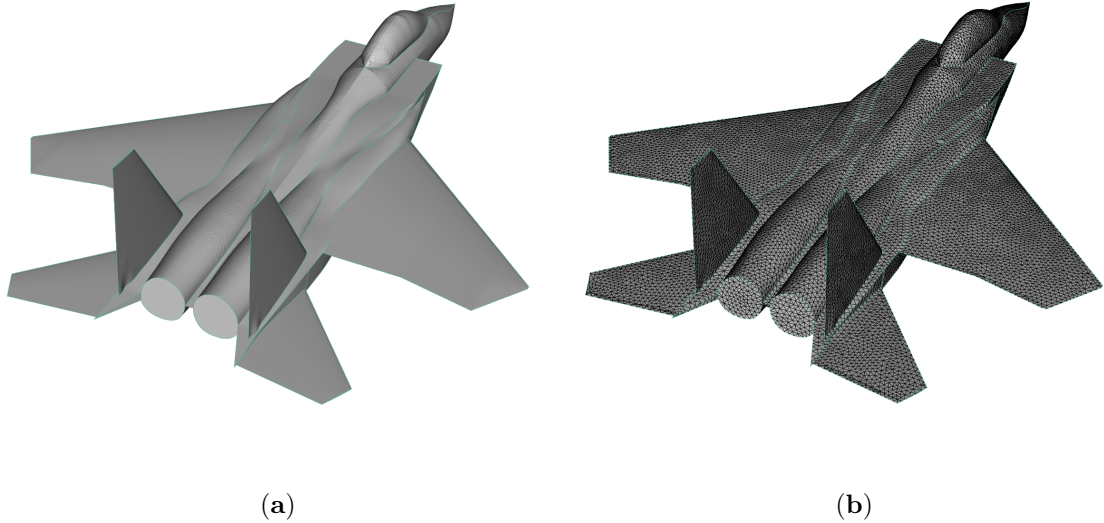


Figure 16: F15 aircraft: (a) geometry and (b) initial uniform mesh.

Iteration	Minimal Size	Gain over isotropic	Gain over uniform
3	$9 \cdot 10^{-3}$	3.9	378
6	$4 \cdot 10^{-3}$	4.9	2 535
9	$1.3 \cdot 10^{-3}$	3.0	47 572
12	$5.6 \cdot 10^{-4}$	3.2	377 246
15	$1.8 \cdot 10^{-4}$	3.5	6 195 211

Table 5: F15 aircraft: Expected gain in terms of DOFs of using an anisotropic approach compared to isotropic or uniform refinement. The size of the corresponding isotropic or uniform mesh has to be multiplied by the factor given in the 3rd and 4th column respectively.

complexity of the geometry. It completely remeshes at each refinement step, altering the shape, size, and orientation of elements according to the *optimal metric* based on a numerically recovered Hessian of the boundary solution. The resulting adaptation is truly anisotropic and we have shown via a variety of numerical examples that it recovers optimal convergence rates for domains with geometric singularities: a planar screen, a sphere with aperture, a cube with cavity, and an F15 aircraft. We note that this optimal rate of convergence cannot be achieved when using uniform refinement since the rapid variation of the boundary solution near geometric singularities requires special meshing treatment. We observed that the sizes of the generated anisotropic meshes are relatively small, with a complexity increase of between 3 and 4 times, and up to millions of times being required to achieve the same level of accuracy with isotropic and uniform refinement, respectively.

For the solver, high anisotropy tends to increase the number of iterations to solve the linear system. However, the mesh adaptation framework naturally provides a good initial guess for the iterative solver by simply interpolating the solution of the previous mesh onto the current mesh. This was shown to decrease the number of iterations at very little expense. We further showed that applying a simple diagonal scaling preconditioner decreases the number of iterations significantly for anisotropic meshes.

The first encouraging anisotropic BEM results presented in this work pave the way for further developments. The first important future direction is to improve the treatment of curved surfaces. It has been observed numerically that the level of anisotropy must be controlled to ensure a good approximation of the scatterer surface. For complex geometries, a more thorough method needs to be employed where two metrics are used, one based on the recovered Hessian and the other based on the surface curvature. Another approach is to propose an adaptation strategy for \mathbb{P}_k interpolation with $k \geq 2$ on curved meshes. Another important improvement concerns the FMM. The version employed in this work is not optimal with respect to computational times for non uniform meshes. It would be of interest to employ an FMM which is tailored to the adapted meshes such as the one proposed in [32].

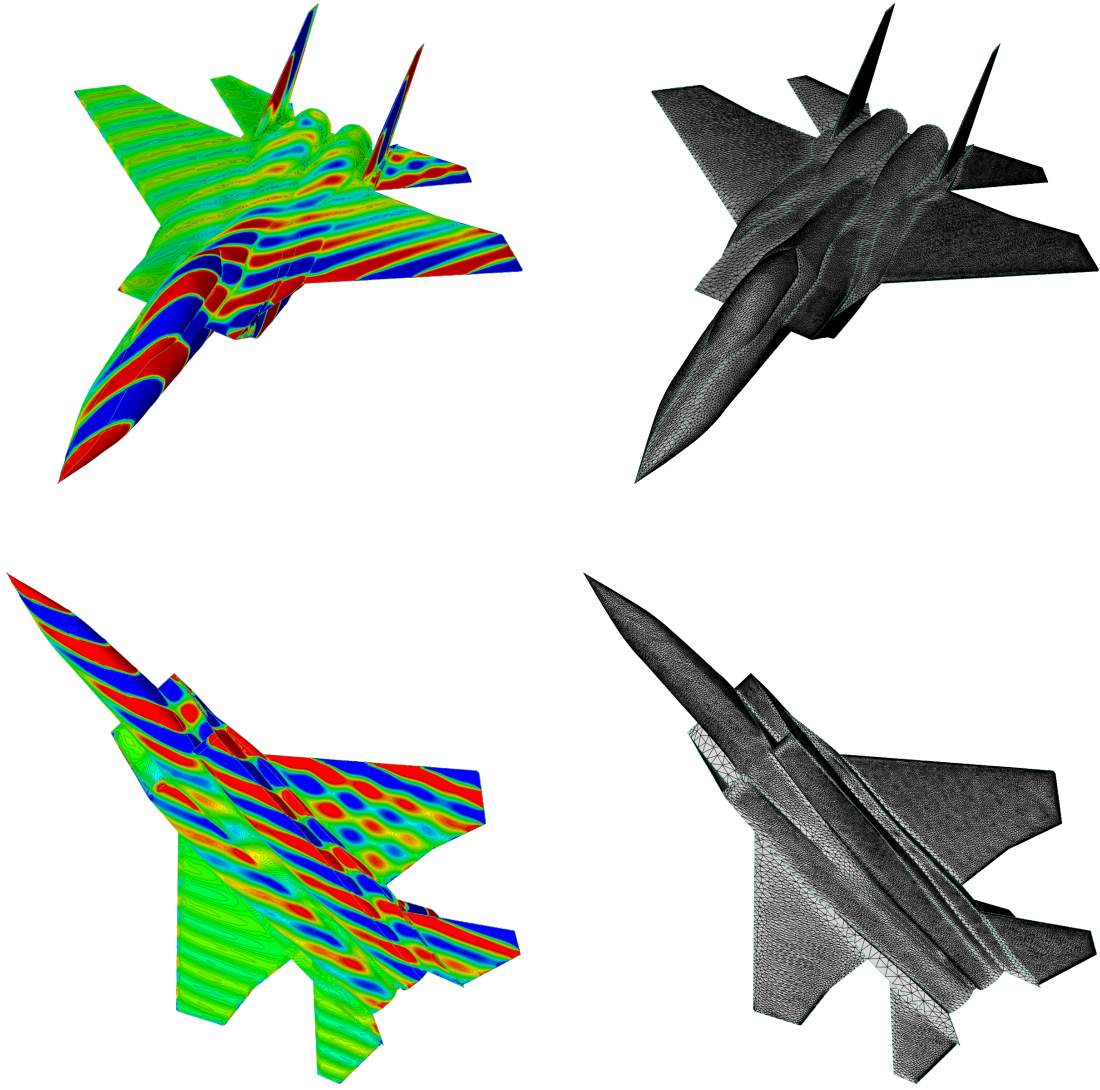


Figure 17: F15 aircraft geometry : Solution on top and back for the last iteration of the adaptive process.

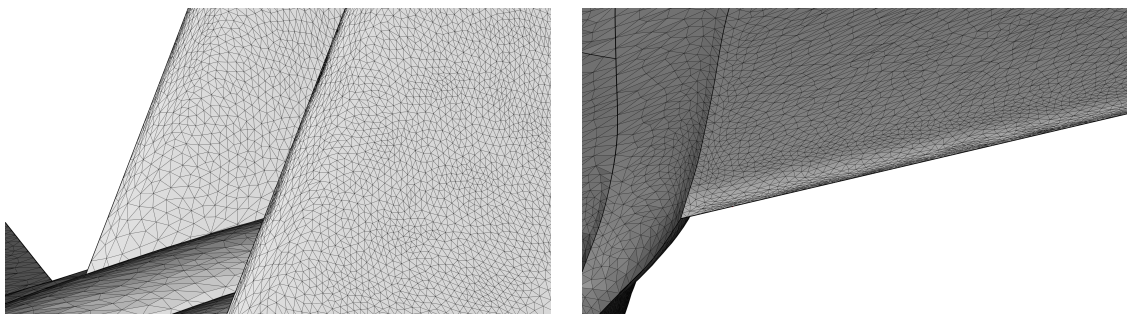


Figure 18: F15 aircraft geometry : Closer view of the mesh at the last iteration of the adaptive process.

Acknowledgements

This work has been funded by the Agence Nationale de la Recherche (RAFFINE project, ANR-12-MONU-0021).

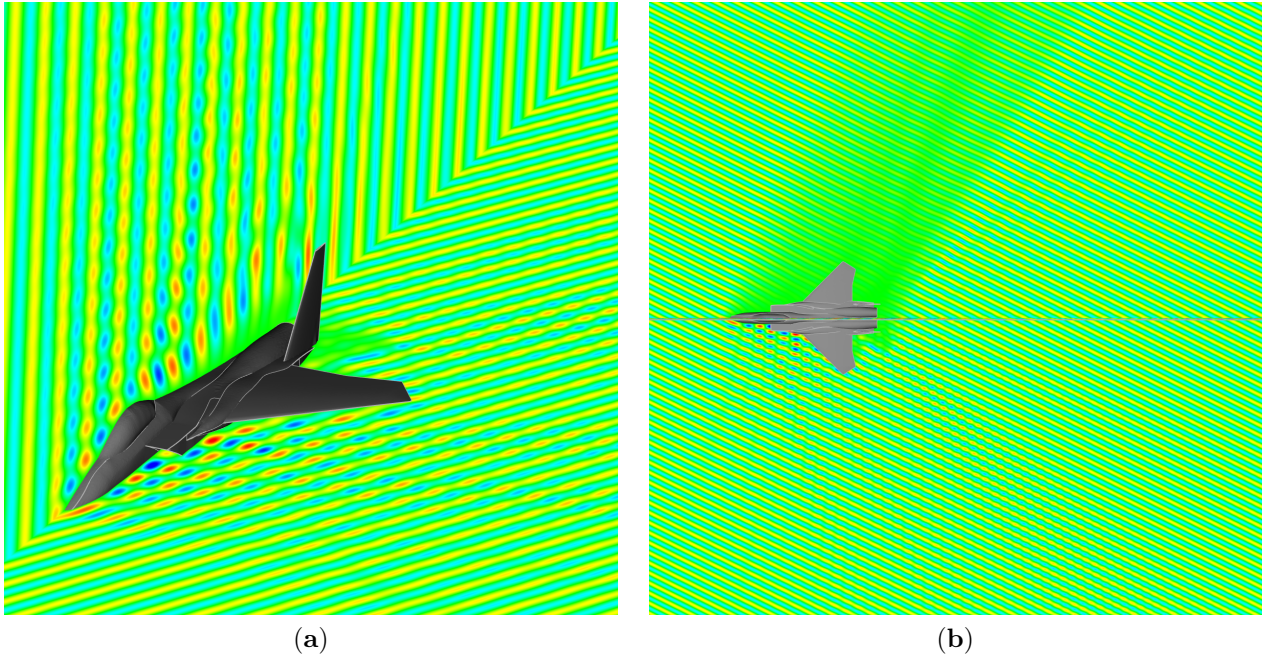


Figure 19: F15 aircraft: Real part of the total field in the planes $y = 0$ (a) and $z = 0$ (b).

References

- [1] M. AINSWORTH AND J. T. ODEN, *A posteriori error estimation in finite element analysis*, vol. 37, John Wiley & Sons, 2011.
- [2] F. ALAUZET AND A. LOSEILLE, *A decade of progress on anisotropic mesh adaptation for computational fluid dynamics*, *Computer-Aided Design*, 72 (2016), pp. 13–39.
- [3] T. APEL, A. SÄNDIG, AND J. WHITEMAN, *Graded mesh refinement and error estimates for finite element solutions of elliptic boundary value problems in non-smooth domains*, *Mathematical methods in the Applied Sciences*, 19 (1996), pp. 63–85.
- [4] M. AURADA, S. FERRAZ-LEITE, AND D. PRAETORIUS, *Estimator reduction and convergence of adaptive bem*, *Applied Numerical Mathematics*, 62 (2012), pp. 787–801.
- [5] M. BAKRY, *Fiabilité et optimisation des calculs obtenus par des formulations intégrales en propagation d'ondes*, PhD thesis, ENSTA ParisTech, 2016.
- [6] M. BAKRY, *A goal-oriented a posteriori error estimate for the oscillating single layer integral equation*, *Applied Mathematics Letters*, 69 (2017), pp. 133–137.
- [7] M. BAKRY, S. PERNET, AND F. COLLINO, *A new accurate residual-based a posteriori error indicator for the BEM in 2D-acoustics*, *Computers & Mathematics with Applications*, 73 (2017), pp. 2501–2514.
- [8] Z. P. BAZANT, *Three-dimensional harmonic functions near termination or intersection of gradient singularity lines: a general numerical method*, *International Journal of Engineering Science*, 12 (1974), pp. 221–243.
- [9] M. BEBENDORF, *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*, Springer, 2008.
- [10] T. BETCKE AND E. A. SPENCE, *Numerical estimation of coercivity constants for boundary integral operators in acoustic scattering*, *SIAM Journal on Numerical Analysis*, 49 (2011), pp. 1572–1601.
- [11] M. BONNET, *Boundary integral equation methods for solids and fluids*, John Wiley, 1995.
- [12] S. BÖRM, L. GRASEDYCK, AND W. HACKBUSCH, *Introduction to hierarchical matrices with applications*, *Engineering analysis with boundary elements*, 27 (2003), pp. 405–422.
- [13] J. BOWMAN, T. SENIOR, AND P. USLENGHI, *Electromagnetic and acoustic scattering by simple shapes*, Taylor & Francis Group, 1988.
- [14] C. CARSTENSEN, *An a posteriori error estimate for a first-kind integral equation*, *Mathematics of Computation of the American Mathematical Society*, 66 (1997), pp. 139–155.
- [15] C. CARSTENSEN, M. FEISCHL, M. PAGE, AND D. PRAETORIUS, *Axioms of adaptivity*, *Computers & Mathematics with Applications*, 67 (2014), pp. 1195–1253.

- [16] S. CHAILLAT AND M. BONNET, *Recent advances on the fast multipole accelerated boundary element method for 3D time-harmonic elastodynamics*, Wave Motion, 50 (2013), pp. 1090–1104.
- [17] S. CHAILLAT, M. BONNET, AND J. SEMBLAT, *A multi-level fast multipole BEM for 3-D elastodynamics in the frequency domain*, Computer Methods in Applied Mechanics and Engineering, 197 (2008), pp. 4233–4249.
- [18] S. CHAILLAT, M. DARBAS, AND F. LE LOUËR, *Fast iterative boundary element methods for high-frequency scattering problems in 3D elastodynamics*, Journal of Computational Physics, 341 (2017), pp. 429–446.
- [19] S. CHAILLAT, L. DESIDERIO, AND P. CIARLET, *Theory and implementation of \mathcal{H} -matrix based iterative and direct solvers for Helmholtz and elastodynamic oscillatory kernels*, Journal of Computational physics, 351 (2017), pp. 165–186.
- [20] S. N. CHANDLER-WILDE, I. G. GRAHAM, S. LANGDON, AND E. A. SPENCE, *Numerical-asymptotic boundary integral methods in high-frequency acoustic scattering*, Acta. Numer., (2012), pp. 89–305.
- [21] L. CHEN, P. SUN, AND J. XU, *Optimal anisotropic meshes for minimizing interpolation errors in L^p -norm*, Mathematics of Computation, 76 (2007), pp. 179–204.
- [22] R. COIFMAN, V. ROKHLIN, AND S. WANDZURA, *The fast multipole method for the wave equation: A pedestrian prescription*, IEEE Antennas and Propagation Magazine, 35 (1993), pp. 7–12.
- [23] D. COLTON AND R. KRESS, *Integral equation methods in scattering theory*, SIAM, 2013.
- [24] M. DARBAS, E. DARRIGRAND, AND Y. LAFRANCHE, *Combining analytic preconditioner and fast multipole method for the 3-D Helmholtz equation*, Journal of Computational Physics, 236 (2013), pp. 289–316.
- [25] E. DARVE, *The fast multipole method: numerical implementation*, Journal of Computational Physics, 160 (2000), pp. 195–240.
- [26] B. FAERMANN, *Local a-posteriori error indicators for the Galerkin discretization of boundary integral equations*, Numerische Mathematik, 79 (1998), pp. 43–76.
- [27] M. FEISCHL, T. FUHRER, N. HEUER, M. KARKULIK, AND D. PRAETORIUS, *Adaptive Boundary Element Methods: A Posteriori Error Estimators, Adaptivity, Convergence, and Implementation*, Arch. Computat. Methods Eng., 22 (2015), pp. 309–389.
- [28] S. FERRAZ-LEITE AND D. PRAETORIUS, *Simple a posteriori error estimators for the h-version of the boundary element method*, Computing, 83 (2008), pp. 135–162.
- [29] V. FRAYSSÉ, L. GIRAUD, S. GRATTON, AND J. LANGOU, *Algorithm 842: A set of gmres routines for real and complex arithmetics on high performance computers*, ACM Transactions on Mathematical Software (TOMS), 31 (2005), pp. 228–238.
- [30] T. GANTUMUR, *Adaptive boundary element methods with convergence rates*, Numerische Mathematik, 124 (2013), pp. 471–516.
- [31] P. GEORGE, F. HECHT, AND M. VALLET, *Creation of internal points in Voronoi’s type method. Control adaptation*, Advances in engineering software and workstations, 13 (1991), pp. 303–312.
- [32] I. G. GRAHAM, L. GRASEDYCK, W. HACKBUSCH, AND S. A. SAUTER, *Optimal panel-clustering in the presence of anisotropic mesh refinement*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 517–543.
- [33] I. G. GRAHAM, W. HACKBUSCH, AND S. SAUTER, *Hybrid Galerkin boundary elements on degenerate meshes*, Mathematical Aspects of Boundary Element Methods, 414 (1999), p. 140.
- [34] L. GREENGARD, J. HUANG, V. ROKHLIN, AND S. WANDZURA, *Accelerating fast multipole methods for the Helmholtz equation at low frequencies*, IEEE Computational Science and Engineering, 5 (1998), pp. 32–38.
- [35] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, Journal of Computational physics, 73 (1987), pp. 325–348.
- [36] P. JUHL, *A note on the convergence of the direct collocation boundary element method*, Journal of Sound and Vibration, 212 (1998), pp. 703–719.
- [37] A. LOSEILLE, *Chapter 10 - unstructured mesh generation and adaptation*, in Handbook of Numerical Methods for Hyperbolic Problems, R. Abgrall and C.-W. Shu, eds., vol. 18 of Handbook of Numerical Analysis, Elsevier, 2017, pp. 263 – 302.
- [38] A. LOSEILLE AND F. ALAUZET, *Continuous mesh framework part I: well-posed continuous interpolation error*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 38–60.
- [39] ———, *Continuous mesh framework part II: validations and applications*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 61–86.
- [40] A. LOSEILLE AND R. LÖHNER, *Robust Boundary Layer Mesh Generation*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 493–511.
- [41] S. MARBURG, *Discretization requirements: how many elements per wavelength are necessary?*, in Computational Acoustics of Noise Propagation in Fluids-Finite and Boundary Element Methods, Springer, 2008, pp. 309–332.
- [42] J. C. NÉDÉLEC, *Acoustic and Electromagnetic Equations: Integral Representations for Harmonic Problems*, Springer, New York, 2001.

- [43] M. PICASSO, F. ALAUZET, H. BOROUCAKI, AND P.-L. GEORGE, *A numerical study of some hessian recovery techniques on isotropic and anisotropic meshes*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1058–1076.
- [44] E. P. STEPHAN, *Boundary integral equations for screen problems in \mathbb{R}^3* , Integral Equations and Operator Theory, 10 (1987), pp. 236–257.
- [45] T. TAKAHASHI, *A wideband fast multipole accelerated boundary integral equation method for time-harmonic elastodynamics in two dimensions*, International Journal for Numerical Methods in Engineering, 91 (2012), pp. 531–551.
- [46] M.-G. VALLET, C.-M. MANOLE, J. DOMPIERRE, S. DUFOUR, AND F. GUIBAULT, *Numerical comparison of some hessian recovery techniques*, International Journal for Numerical Methods in Engineering, 72 (2007), pp. 987–1007.