Efficient high order and domain decomposition methods for the time-harmonic Maxwell's equations

Marcella Bonazzoli

École Doctorale Sciences Fondamentales et Appliquées
Laboratoire Jean Alexandre Dieudonné

Thèse de doctorat

Présentée en vue de l'obtention du
grade de docteur en Mathématiques
de
Université Côte d'Azur

par
Marcella Bonazzoli

MÉTHODES D'ORDRE ÉLEVÉ ET MÉTHODES DE DÉCOMPOSITION DE DOMAINE EFFICACES POUR LES ÉQUATIONS DE MAXWELL EN RÉGIME HARMONIQUE

EFFICIENT HIGH ORDER AND DOMAIN DECOMPOSITION METHODS FOR THE TIME-HARMONIC MAXWELL'S EQUATIONS

Dirigée par Francesca Rapetti, Maître de conférences, Université Côte d'Azur
et codirigée par Victorita Dolean, Maître de conférences, Université Côte d'Azur

Soutenue le 11 septembre 2017
Devant le jury composé de :

Daniele Boffi	Professeur, Università degli Studi di Pavia	Rapporteur
Victorita Dolean	Maître de conférences, Université Côte d'Azur	Directrice de thèse
Christophe Geuzaine	Professeur, Université de Liège	Rapporteur
Roland Masson	Professeur, Université Côte d'Azur	Examinateur
Frédéric Nataf	Directeur de recherche CNRS, UPMC, Paris 6	Examinateur
Francesca Rapetti	Maître de conférences, Université Côte d'Azur	Directrice de thèse

Remerciements

J'adresse mes premiers remerciements à mes directrices de thèse, Francesca et Victorita, sans qui je n'en serais pas là aujourd'hui. Vous m'avez beaucoup appris pendant ces trois années, et je n'aurais pas pu progresser autant dans mon travail de recherche sans votre soutien et confiance en moi. Je tiens également à remercier Daniele Boffi et Christophe Geuzaine d'avoir accepté d'être les rapporteurs de cet écrit, ainsi que les autres membres de mon jury de thèse, Roland Masson et Frédéric Nataf. Je suis très honorée que chacun de vous ait accepté de participer à l'évaluation de cette thèse.

Je souhaite remercier tous les membres du projet MEDIMAX pour la collaboration fructueuse et enrichissante : Iannis Aliferis, Marion Darbas, Maya de Buhan, Ibtissam El Kanfoud, Frédéric Hecht, Pierre Jolivet, Claire Migliaccio, Frédéric Nataf, Richard Pasquetti, Christian Pichot et Pierre-Henri Tournier. En particulier, je tiens à remercier Frédéric Nataf pour m'avoir invitée à plusieurs reprises à Paris et pour avoir suivi mon travail de thèse régulièrement avec gentillesse et disponibilité. Merci à Frédéric Hecht pour m'avoir dévoilé certains mystères au coeur de FreeFem++. Un grand merci à Pierre-Henri pour sa patience exceptionnelle dans le partage de ses connaissances et pour le travail côte à côte en dépit de la distance. Parmi les personnes avec qui j'ai eu la chance de travailler figurent Ivan Graham et Euan Spence : j'ai beaucoup appris et c'était un plaisir de travailler avec eux. Je garde un très bon souvenir de la collaboration intense avec Xavier Claeys et Pierre Marchand pendant les six semaines de l'école d'été du Cemracs.

Je tiens à exprimer ma gratitude à tous ceux qui m'ont accueillie au Laboratoire J.A. Dieudonné, depuis mon arrivée lors du stage de Master, notamment à Julia, Jean-Marc, Roland, Angélique et Victoria. Un grand merci à Manuelle et Chiara qui ont suivi avec efficacité la gestion de mes nombreux déplacements vers tous les coins du monde.

Je n'oublierai jamais tous les bons moments passés avec les doctorants, stagiaires et post-doctorants du laboratoire et d'ailleurs : Alexis et Alexis, Ali, Amine, Anthony, Armand, Arthur, Bastien, Björn, Bienvenu, Brice, Byron, Charles, Chiara, Cristina, David et David, Eduard, Eléonore, Eliot, Emiliano, Emmanuel, Farah, Fernanda, Giulia et Giulia, Guillaume, Huda, Huong, Jean-Baptiste, Jie, Jonathan, Julian, Julie, Julien, Katia, Laurence, Liana, Ludovick, Luis, Maksym, Massimo, Mayya, Mélisande, Nabil, Nancy, Nathalie, Nicolas, Olivier, Reine, Rinel, Robert, Rodrigo, Samira, Simon, Stefan, Thi, Van Bien, Victor, Vincent, Vladimir, Yash, Willy, Zeinab, Zhiyan. J'ai une pensée spéciale pour les co-bureaux, les cruciverbistes, les compagnons de RU, les GDRs goûter et cinéma, les joueurs de Cranium au Colloque des doctorants et les participants au Cemracs.

Je voudrais remercier aussi mes anciens professeurs qui m'ont donné le goût des mathématiques, dont notamment Ada Gallina, Lidia Angeleri, Sisto Baldo, Marco Caliari, Giandomenico Orlandi pour m'avoir encouragée à partir en Erasmus en France et Elena pour m'avoir épaulée au début de ce parcours. Je voudrais dédier cette thèse à tous les membres de ma famille qui m'ont toujours accompagnée même de loin et à Luis qui, avec son soutien, son écoute et ses conseils, m'a encouragée tout au long de cette thèse.

Contents

Chapter 1

Introduction

In this thesis we couple high order finite element discretizations with domain decomposition preconditioners to design a precise and fast solver for the *time-harmonic Maxwell's equation* for the electric field. Maxwell's equations are a system of partial differential equations which model electromagnetic wave propagation. The time-harmonic formulation is derived when the involved fields are sinusoidal (or 'harmonic') in time, varying with an angular frequency ω; thus, the term *frequency domain* is commonly used, in opposition to the time domain. The time-harmonic Maxwell's equations present several difficulties when the frequency ω is large, such as the *sign-indefiniteness* of their variational formulation, the pollution effect which entails particularly fine meshes, and the consequent problematic construction of efficient iterative solvers. Note that, on the contrary, for Maxwell's equations in the time domain, for which an implicit time discretization yields at each step a positive definite problem, there are many good solvers and preconditioners in the literature (e.g. multigrid methods).

This work is motivated by the *application* to brain imaging studied by the project MEDIMAX (ANR-13-MONU-0012, financed by the French National Research Agency, ANR), which gathers Mathematics Laboratories (LJAD in Nice, LJLL and MAP5 in Paris) and the Electrical Engineering Laboratory LEAT (Nice-Sophia Antipolis). The purpose of this project is the full numerical simulation of a microwave imaging system prototype, developed by the medical imaging company EMTensor GmbH (Vienna, Austria) for the diagnosis and monitoring of brain strokes (see Figure 1.1). The data acquired with this device are used as input for an inverse problem associated with the time-harmonic Maxwell's equations, which makes it possible to estimate the complex electric permittivity of the brain tissues of a patient affected by a stroke. Indeed, a stroke results in a variation of the complex electric permittivity inside a region of the brain, thus it can be detected and monitored

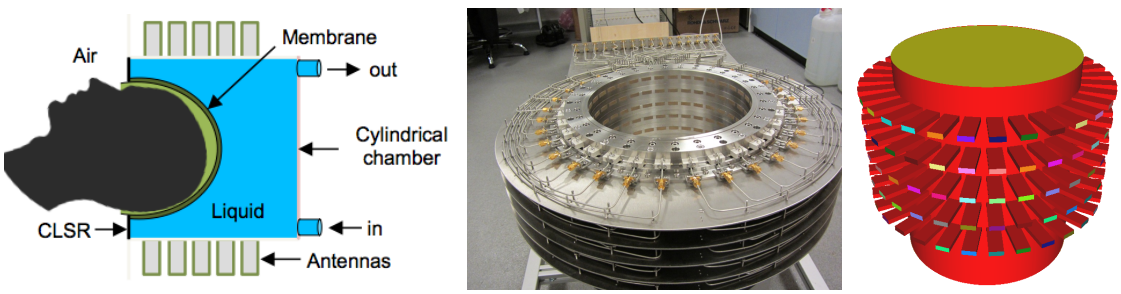


Figure 1.1 – The microwave imaging system prototype developed by EMTensor GmbH and the corresponding computational domain.

by doctors thanks to an image of the brain displaying the values of this property. The solution of the inverse problem requires to solve repeatedly the direct (or forward) problem of the time-harmonic Maxwell's equations, i.e. the more familiar problem in which material properties are given and the unknown is the electric field. Therefore an accurate and fast solver for the direct problem is needed. Accuracy and computing speed are indeed essential in this application, for a precise detection of the stroke and the continuous monitoring of the effects of medical treatment. Nevertheless, the methods studied in this thesis are not specific solely to the imaging problem and have a more general extent of applications. In our strategy, accuracy is provided by high order edge finite elements; the linear system resulting from this discretization is then solved efficiently in parallel with the iterative solver GMRES preconditioned with Schwarz domain decomposition methods. We now briefly introduce these two ingredients, highlighting their advantages and also the difficulties we will face, referring to the first sections of the subsequent Chapters for a more detailed introduction and literature review.

High order finite elements methods make it possible, for a given precision, to reduce significantly the number of unknowns, and they are particularly well suited to discretize wave propagation problems since they can provide a solution with very low dispersion and dissipation errors. For the time-harmonic Maxwell's equation for the electric field in a domain $\Omega \subset \mathbb{R}^3$, the functional space on which the variational formulation is well defined is $H_{\text{curl}} = \{\mathbf{v} \in L^2(\Omega)^3, \nabla \times \mathbf{v} \in L^2(\Omega)^3\}$. Discrete finite element subspaces of H_{curl}, which thus provide *curl-conforming* finite elements, are due to Nédélec [89]. They are often termed *edge elements* because at the lowest order the degrees of freedom are associated with the edges of the mesh, more precisely they are the circulations along the edges. Instead of approximating each component of the field with the usual scalar node-based finite elements, basis functions for edge elements are vector functions, which fit the physical continuity properties of the electric field: its tangential component is continuous across material discontinuities while the normal component can jump. One of the difficulties with edge elements is that basis functions and degrees of freedom are associated with the *oriented* edges of mesh tetrahedra, so that particular attention should be paid to ensure that the contributions coming from tetrahedra sharing edges or faces are assembled properly inside the global matrix of the finite element discretization. Moreover, the *duality* property $\sigma_i(\mathbf{w}_j) = \delta_{ij}$ between basis functions $\mathbf{w}_j$ and degrees of freedom σ_i, on which the standard approach to define a finite element basis is based, is not automatically granted for high order edge elements. Recall that, for the standard node-based finite elements, whose degrees of freedom are the values at the nodes of the mesh, this duality property means that the value of each basis function at the associated node is 1 while it is 0 at the other nodes. This ensures that the expansion coefficients for writing a function in terms of the basis (which are also the unknowns of the algebraic system resulting from the discretization) are given by the degrees of freedom applied to the function.

The algebraic linear system resulting from the high order discretization can be ill conditioned, so that preconditioning becomes mandatory when using iterative solvers. Indeed there are two main families of linear system solvers: iterative solvers and direct solvers. On the one hand, *direct solvers* (such as SuperLU, UMFPACK, MUMPS, ...) are robust, i.e. they find the solution in a finite number of operations no matter how hard the problem is; but because of their high memory cost they are not suited for the very large systems arising for instance from the complex three dimensional model of the imaging system prototype. On the other hand, *iterative solvers* (such as the conjugate gradient method for symmetric positive definite matrices, or GMRES for more general matrices) require less memory and are easy to parallelize since they are based on matrix-vector products. Their drawback is that they lack robustness, so that preconditioning the system is essential to

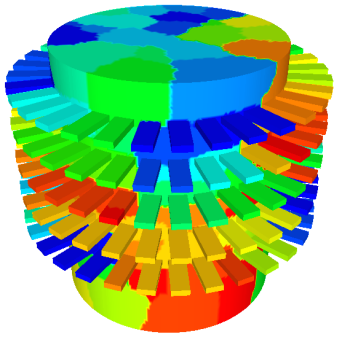


Figure 1.2 – A decomposition of the computational domain into 128 subdomains.

ensure a good convergence. Namely, when one uses an iterative method to solve a linear system $A\mathbf{u} = \mathbf{b}$, it is commonly far more efficient to design a suitable matrix M^{-1}, called *preconditioner*, that approximates A^{-1} and is not too hard to compute, and then solve the system $M^{-1}A\mathbf{u} = M^{-1}\mathbf{b}$, where the condition number of $M^{-1}A$ is much lower than the one of A.

Domain decomposition methods can be viewed as hybrid methods that take the advantages of the two families of solvers: the problem defined on the global domain is decomposed into smaller problems on subdomains (see for instance Figure 1.2), on which direct solvers are applicable, and the solution is coordinated between neighboring subdomains iteratively. Domain decomposition methods are naturally suited to parallel computing because the problems on subdomains can be solved concurrently. Moreover, rather than using these methods as solvers, they can be used, more conveniently, as preconditioners for iterative solvers such as GMRES. For a given problem, an efficient domain decomposition method is defined by two main ingredients: the *transmission conditions*, i.e. the boundary conditions imposed at the interfaces between neighboring subdomains that specify the information exchanged between them; the *coarse-grid correction*, introduced in the so-called two-level methods, that allows information to propagate through the whole domain in one step. The construction and, even more, the convergence analysis of two-level domain decomposition preconditioners for sign-indefinite problems is a challenging open issue.

1.1 Summary and contributions

In the following we detail for each Chapter the contributions of this thesis.

Mathematical models of electromagnetism. Chapter 2 provides an introduction to Maxwell's equations, starting from the full system of four equations in their integral and differential forms. This system needs to be completed with constitutive laws, which model at a macroscopic scale the field-matter interaction. Then the second order (or curl-curl) equation for the electric field is derived, together with its time-harmonic formulation, which is the equation on which we focus in this thesis. We highlight that, as a result of the chosen sign convention in the time-harmonic assumption, the complex valued electric permittivity appearing in the equation has negative imaginary part.

The second part of this Chapter is devoted to the boundary value problem (BVP) describing wave propagation inside a rectangular waveguide. It consists of the second order time-harmonic Maxwell's equation for the electric field together with boundary conditions which are specific to electromagnetism: the metallic or PEC (Perfect Electric Conductor)

boundary condition, which is a Dirichlet-type condition, and the impedance boundary condition, which is a Robin-type and absorbing condition. Again, particular emphasis is placed on the sign appearing in the absorbing condition that depends on the chosen time-harmonic convention. Finally the variational (or weak) formulation of the waveguide boundary value problem is derived, which is the first step to apply the finite element method. Note that, even if the computational domain of the medical imaging application (Figure 1.1, right) is much more complex than a single waveguide, the variational formulation described in §2.2.5 has the same form as the one modeling the imaging system prototype.

A revisitation of high order curl-conforming finite elements. In Chapter 3 we define high order finite elements for H_{curl} on a simplicial mesh $\mathcal{T}_h$ over the computational domain Ω (simplicial means triangular in 2d and tetrahedral in 3d). This work is the object of the publication [20]. Our finite elements are instances of the first family of Nédélec finite elements [89], where the involved functions are vectors with *incomplete* or *trimmed* polynomials as components (that is some of the top-degree monomials are removed to satisfy some constraints).

With regard to basis functions, we adopt the high order generators presented in [97, 98]: their definition is rather simple since it only involves the barycentric coordinates of the simplex and is associated with a geometrical construction, which helps to visualize the list of generators. Here, we revisit the classical degrees of freedom defined by Nédélec [89], in order to obtain a new expression which results to be more friendly in terms of the considered basis functions. Moreover, we propose a general technique to restore duality between degrees of freedom and basis functions for the high order case, thanks to a generalized Vandermonde matrix. In the following we give an overview of these results, referring to Chapter 3 for the detailed presentation.

First of all recall that, for a simplex $T \in \mathcal{T}_h$, the local *lowest order basis functions* for the Nédélec curl-conforming space $V_h \subset H(\mathrm{curl}, \Omega)$ are associated with the oriented edges $e = \{n_i, n_j\}$ of T as follows

$$\mathbf{w}^e = \lambda_{n_i} \nabla \lambda_{n_j} - \lambda_{n_j} \nabla \lambda_{n_i},$$

where the λ_{n_ℓ} are the barycentric coordinates of a point $\mathbf{x} \in T$ with respect to the node n_ℓ of T of Cartesian coordinates $\mathbf{x}_\ell$. The degrees of freedom over T are the functionals

$$\sigma_e \colon \mathbf{w} \mapsto \frac{1}{|e|} \int_e \mathbf{w} \cdot \mathbf{t}_e, \quad \forall\, e \in \mathcal{E}(T),$$

where $\mathbf{t}_e = \mathbf{x}_j - \mathbf{x}_i$ is the tangent vector to the edge e, $|e| = |\mathbf{t}_e|$ the length of e and $\mathcal{E}(T)$ the set of edges of T. At the lowest order, the basis functions are in duality with the degrees of freedom, that is $\sigma_e(\mathbf{w}^{e'}) = 1$, resp. 0, if $e = e'$, resp. if $e \neq e'$.

To state the definition of the *high order generators* of [97, 98], we need to introduce multi-index notations. A multi-index is an array $\mathbf{k} = (k_1, \ldots, k_\nu)$ of ν integers $k_i \geq 0$, and its weight k is $\sum_{i=1}^{\nu} k_i$. The set of multi-indices $\mathbf{k}$ with ν components and of weight k is denoted $\mathcal{I}(\nu, k)$. If $d = 2, 3$ is the ambient space dimension, we consider $\nu \leq d + 1$ and, given $\mathbf{k} \in \mathcal{I}(\nu, k)$, we set $\lambda^{\mathbf{k}} = \prod_{i=1}^{\nu} (\lambda_{n_i})^{k_i}$, where the n_i are ν nodes of the $d + 1$ nodes of T. Now, in the generators definition we take $\nu = d + 1$ and $k = r - 1$, with r the polynomial degree of the generators.

Definition of the generators. The generators for Nédélec edge element spaces $W^1_{h,r}(T)$ of degree $r \geq 1$ in a simplex $T \in \mathcal{T}_h$ are the $\lambda^{\mathbf{k}} \mathbf{w}^e$, with $\mathbf{k} \in \mathcal{I}(d + 1, k)$, $k = r - 1$ and $e \in \mathcal{E}(T)$.

We recast the degrees of freedom in [89] in a new more friendly form as follows.

The revisited degrees of freedom. For $r \geq 1$, the functionals

$$\sigma_e : \mathbf{w} \mapsto \frac{1}{|e|} \int_e (\mathbf{w} \cdot \mathbf{t}_e)\, q, \qquad \forall\, q \in \mathbb{P}_{r-1}(e),\ \forall\, e \in \mathcal{E}(T), \tag{1.1}$$

$$\sigma_f : \mathbf{w} \mapsto \frac{1}{|f|} \int_f (\mathbf{w} \cdot \mathbf{t}_{f,i})\, q, \qquad \forall\, q \in \mathbb{P}_{r-2}(f),\ \forall\, f \in \mathcal{F}(T), \tag{1.2}$$

$$\mathbf{t}_{f,i} \text{ two independent sides of } f,\ i = 1, 2,$$

$$\sigma_T : \mathbf{w} \mapsto \frac{1}{|T|} \int_T (\mathbf{w} \cdot \mathbf{t}_{T,i})\, q, \qquad \forall\, q \in \mathbb{P}_{r-3}(T), \tag{1.3}$$

$$\mathbf{t}_{T,i} \text{ three independent sides of } T,\ i = 1, 2, 3,$$

are the degrees of freedom for a function $\mathbf{w} \in W^1_{h,r}(T)$. $\mathcal{F}(T)$ is the set of faces of T, the norm of the vectors $\mathbf{t}_e, \mathbf{t}_{f,i}, \mathbf{t}_{T,i}$ is the length of the associated edge. We say that e, f, T are the *supports* of the degrees of freedom $\sigma_e, \sigma_f, \sigma_T$. To make the computation of degrees of freedom easier, a convenient choice for the polynomials q spanning the polynomial spaces over (sub)simplices e, f, T is given by suitable products of the barycentric coordinates associated with the nodes of the considered (sub)simplex. Indeed, the space $\mathbb{P}_\rho(S)$ of polynomials of degree $\leq \rho$ over a p-simplex S (i.e. a simplex of dimension $1 \leq p \leq d$) can be generated by the products $\lambda^{\mathbf{k}} = \prod_{i=1}^{p+1} (\lambda_{n_i})^{k_i}$, with $\mathbf{k} \in \mathcal{I}(p+1, \rho)$ and n_i being the nodes of S.

Classification and selection of linearly independent generators. The classification of degrees of freedom into edge-type, face-type, volume-type degrees of freedom can be done also for generators: volume-type generators contain (inside $\lambda^{\mathbf{k}}$ or $\mathbf{w}^e$) the barycentric coordinates w.r.t. all the nodes of a tetrahedron T, face-type generators contain the ones w.r.t. all and only the nodes of a face f, edge-type generators contain the ones w.r.t. only the nodes of an edge e. Note that face-type (resp. volume-type) generators appear for $r > 1$ (resp. $r > 2$) (and the same happens for face-type and volume-type degrees of freedom).

For the high order case ($r > 1$), the fields $\lambda^{\mathbf{k}} \mathbf{w}^e$ are generators for $W^1_{h,r}(T)$, but some of the face-type or volume-type generators are *linearly dependent*. The selection of generators that constitute an actual basis of $W^1_{h,r}(T)$ can be guided by the revisited degrees of freedom. More precisely, as face-type (resp. volume-type) generators keep the ones associated with the two (resp. three) edges e *chosen as* the two sides $\mathbf{t}_{f,1}, \mathbf{t}_{f,2}$ (resp. three sides $\mathbf{t}_{T,1}, \mathbf{t}_{T,2}, \mathbf{t}_{T,3}$) of face-type degrees of freedom (resp. volume-type degrees of freedom).

Restoring duality. The considered basis functions are not in duality with the degrees of freedom when $r > 1$, namely, the matrix V with entries the weights $V_{ij} = \sigma_i(\mathbf{w}_j)$, $1 \leq i, j \leq n_{\text{dofs}} = \dim(W^1_{h,r}(T))$ after a suitable renumbering of degrees of freedom, is not the identity matrix for $r > 1$. Duality can be re-established by considering new basis functions $\tilde{\mathbf{w}}_j$, $1 \leq i, j \leq n_{\text{dofs}}$: $\tilde{\mathbf{w}}_j$ is built as a *linear combination* of the previous basis functions with coefficients given by the entries in the j-th column of V^{-1}. The matrix V is a sort of generalized Vandermonde matrix and has some nice properties: its entries do not depend on the metrics of the tetrahedron T for which they are calculated; V, and hence its inverse V^{-1}, are block lower triangular if we list generators and degrees of freedom in the order dictated by increasing the dimension of the support of degrees of freedom; the entries of V^{-1} are *integer* numbers.

The example in Section 3.7 illustrates explicitly all these notions for $d = 3$, $r = 2$.

Implementation of high order curl-conforming finite elements. The implementation of high order curl-conforming finite elements is quite delicate, especially in the

three-dimensional case. In Chapter 4 (part of the submitted paper [18]) we explicitly describe an implementation strategy, which we embedded in the open source domain specific language FreeFem++ (`http://www.freefem.org/ff++/`). Thus our high order edge finite elements in 3d of degree $2, 3$ are available for the scientific community. To add them to FreeFem++ we define in a C++ plugin various ingredients, among which the principal ones are: the basis functions (and their derivatives) in a simplex; an interpolation operator, which requires degrees of freedom in duality with the basis functions. Indeed, in FreeFem++ the basis functions (and in some cases the coefficients of the interpolation operator) are constructed *locally*, i.e. in each simplex T of the mesh $\mathcal{T}_h$, without the need of a transformation from the reference simplex. Note that the chosen definition of high order generators, which involves only the barycentric coordinates of the simplex, fits perfectly this local construction feature of FreeFem++. Nevertheless, the local construction should be done in such a way that the contributions coming from simplices sharing edges or faces can be then assembled properly inside the *global* matrix of the finite element discretization. Moreover, for the definition of the interpolation operator, in the high order edge elements case we need the generalized Vandermonde matrix V to restore duality between degrees of freedom and basis functions. Here, we carefully address the problem of applying the same Vandermonde matrix to possibly differently oriented simplices of the whole mesh, in order to be able to use in numerical experiments the concepts presented for just one simplex in the previous Chapter. The strategy to fulfill these two requirements is implemented using two permutations, based on the global numbers of the mesh nodes. We also describe in detail the implementation of the *interpolation operator*. Section 4.4 shows how to use these new finite elements in a FreeFem++ script.

Schwarz domain decomposition preconditioners. In Chapter 5 (also part of the submitted paper [18]) we investigate the preconditioning for Maxwell's equations in the time-harmonic regime, which is an underdeveloped issue in the literature, particularly for high order discretizations. We focus on the (one-level) Optimized Restricted Additive Schwarz (ORAS) *overlapping* preconditioner, where the term 'optimized' refers to the use of impedance boundary conditions as transmission conditions, proposed by Després in [38]; the algebraic formulation of optimized overlapping Schwarz methods like ORAS was introduced in [35]. Note that we do not consider more sophisticated transmission conditions because our aim in the next Chapter is to treat general decompositions into overlapping subdomains, with quite rough interfaces, see Figure 1.2. We perform extensive experiments to validate the ORAS preconditioner (and its symmetric version OAS) for different values of physical and numerical parameters, both for 2d and 3d waveguide configurations; in particular we study the effect of their variation on the spectrum of the preconditioned matrix. This numerical investigation shows that Schwarz preconditioning significantly improves GMRES convergence, and that the ORAS preconditioner always performs much better than the OAS preconditioner. Moreover, in all the considered test cases, the number of iterations for convergence using the ORAS preconditioner does not vary when the polynomial degree of the adopted high order finite elements increases. We see that it is necessary to take an overlap of at least one layer of simplices from *both* subdomains of a neighbors pair. All these convergence qualities are reflected by the spectrum of the preconditioned matrix. Finally, the experiments varying the number of subdomains and the frequency exhibit the need for a two-level preconditioner.

In the following we briefly report the algebraic definition of the ORAS preconditioner, when applied to the time-harmonic Maxwell's equation discretized with high order edge finite elements. Consider a decomposition of the domain Ω into N_{sub} overlapping subdo-

mains Ω_s that consist of a union of simplices of the mesh $\mathcal{T}_h$. Let $\mathcal{N}$ be an ordered set of the degrees of freedom of the whole domain, and let $\mathcal{N} = \bigcup_{s=1}^{N_{\text{sub}}} \mathcal{N}_s$ be its decomposition into the (non disjoint) ordered subsets corresponding to the different overlapping subdomains Ω_s: a degree of freedom belongs to $\mathcal{N}_s$ if its support (edge, face or volume) is contained in Ω_s. For edge finite elements, it is important to ensure that the orientation of the degrees of freedom is the same in the domain and in the subdomains. Define the matrix R_s as the *restriction* matrix from Ω to the subdomain Ω_s: it is a $\#\mathcal{N}_s \times \#\mathcal{N}$ Boolean matrix, whose (i, j) entry equals 1 if the i-th degree of freedom in $\mathcal{N}_s$ is the j-th one in $\mathcal{N}$. Note that the *extension* matrix from the subdomain Ω_s to Ω is given by R_s^T. To deal with the unknowns that belong to the overlap between subdomains, define for each subdomain a $\#\mathcal{N}_s \times \#\mathcal{N}_s$ diagonal matrix D_s that gives a discrete *partition of unity*, i.e.

$$\sum_{s=1}^{N_{\text{sub}}} R_s^T D_s R_s = I.$$

Then the *Optimized Restricted Additive Schwarz* (ORAS) preconditioner can be expressed as

$$M_{\text{ORAS}}^{-1} = \sum_{s=1}^{N_{\text{sub}}} R_s^T D_s A_{s,\text{Opt}}^{-1} R_s,$$

where the matrices $A_{s,\text{Opt}}$ are the local matrices, stemmed from the discretization by high order edge finite elements, of the subproblems with impedance boundary conditions $(\nabla \times \mathbf{E}) \times \mathbf{n} + \mathtt{i}\tilde{\omega}\mathbf{n} \times (\mathbf{E} \times \mathbf{n})$ as transmission conditions at the interfaces between subdomains (note that now the term 'local' refers to a subdomain and not to a mesh simplex). The term 'restricted' corresponds to the presence of the partition of unity matrices D_s. The construction of the partition of unity is intricate, especially for (high order) edge finite elements. Here, suitable piecewise linear functions χ_s giving a continuous partition of unity ($\sum_{s=1}^{N_{\text{sub}}} \chi_s = 1$) are interpolated at the barycenters of the support (edge, face, volume) of each degree of freedom of the (high order) edge finite elements. This interpolation is obtained thanks to an auxiliary FreeFem++ *scalar* finite element space that has only the interpolation operator and no basis functions. Note that when optimized conditions are chosen as transmission conditions at the interfaces, it is essential that not only the function χ_s but also its derivative are equal to zero on the border of the subdomain Ω_s.

Application to brain microwave imaging. Chapter 6 shows the benefits of using a discretization based on the high order edge finite elements coupled with the parallel domain decomposition preconditioner, to simulate the microwave imaging system prototype of EMTensor GmbH for the detection and monitoring of brain strokes. We have merged the contributions [12, 112] into this Chapter and a related work is the invited paper [113].

We first introduce the medical context, the characteristics of microwave imaging, which is a novel promising technique, and the operating principle of EMTensor GmbH imaging system, which is composed of 5 rings of 32 antennas around a metallic cylindrical chamber (see Figure 1.1). Each antenna is a ceramic-loaded rectangular waveguide. Then, the direct problem that models this imaging system is described: there is one boundary value problem for each transmitting antenna. We also explain how to compute the so-called scattering coefficients, which are the data acquired by the imaging system, indeed the electric field is not a measurable quantity. For the sake of completeness the inverse problem is presented, as well as to clearly show where the solution of the direct problem intervenes in the inversion tool.

In the numerical results section, in order to validate our numerical modeling, we first compare the scattering coefficients given by the simulation with the measured values obtained by EMTensor: they result to be in very good agreement. Then we demonstrate the advantage, in terms of accuracy and computing time, of using high order edge finite elements compared to the standard lowest order edge elements: for instance, to attain a given accuracy of ≈ 0.1 (see the details in the complete Chapter), the finite element discretization of degree $r = 1$ requires 21 million unknowns and a computing time of 130 seconds, while the high order finite element discretization ($r = 2$) only needs 5 million unknowns, with a corresponding computing time of 62 seconds. The parallel implementation of the domain decomposition ORAS preconditioner in HPDDM [71] (`https://github.com/hpddm/hpddm`, a High-Performance unified framework for Domain Decomposition Methods) is essential to be able to solve the arising linear systems of up to 96 million complex-valued unknowns considered here. To assess the efficiency of the parallel domain decomposition preconditioner we perform a strong scaling analysis: even if the number of iterations increases with the number of subdomains, this experiment exhibits very good time speedups up to 2048 subdomains. Here we studied efficient techniques for the solution of the direct problem, that have been embedded in the inversion tool developed by the ANR MEDIMAX team: in the conclusion section we give a brief account of the results obtained by the team for the inverse problem, showing the feasibility of this microwave imaging technique for detection and monitoring of brain strokes.

Two-level preconditioners for the Helmholtz equation. Chapter 7 is based on [17], which has been submitted to the proceedings of the DD24 International Conference on Domain Decomposition Methods. The time-harmonic Maxwell's equation presents similar difficulties to those encountered with the (scalar) *Helmholtz equation* when the wavenumber $\tilde{\omega}$ is large, namely the sign-indefiniteness of their (standard) variational formulation, the pollution effect, and the consequent problematic construction of fast iterative solvers [50]. Since, even for this scalar equation, there is no established and robust preconditioner, whose behavior is independent of $\tilde{\omega}$ for general decompositions into subdomains, before facing the time-harmonic Maxwell's equation we focus in Chapter 7 on the Helmholtz equation.

In order to achieve independence of the iteration count on the number of subdomains or, for wave propagation problems, on the wavenumber $\tilde{\omega}$, *two-level domain decomposition preconditioners* are generally introduced. One should define two ingredients: an algebraic *formula* to combine the coarse grid correction with the one-level preconditioner (e.g. in a additive or in a hybrid way); a rectangular full column rank matrix Z, whose columns span what is called the *coarse space*. Our purpose is to compare numerically two different coarse space definitions for the Helmholtz equation, which are currently the most robust available in literature. In [34] the coarse space is built by solving local eigenproblems on the interfaces involving the Dirichlet-to-Neumann (DtN) operator. In [62, 63] two-level domain decomposition approximations of the Helmholtz equation with absorption $-\Delta u-(\tilde{\omega}^2+\mathtt{i}\xi)u = f$ are used as preconditioners for the pure Helmholtz equation without absorption; the coarse space is based on a coarser mesh, with diameter constrained by $\tilde{\omega}$, thus here we refer to it as "grid coarse space".

In our numerical experiments, which are in two and three dimensions, we reach more than 28 million complex-valued unknowns in the linear systems, resulting from a discretization (with piecewise linear finite elements) of the pure Helmholtz equation without absorption, with an increasing wavenumber. For both coarse space definitions, the preconditioners built with absorption $\xi_{\text{prec}} = \tilde{\omega}^2$ appear to perform much worse than those with

absorption $\xi_{\text{prec}} = \tilde{\omega}$. We see that, in most cases, for smaller coarse space sizes the grid coarse space gives fewer iterations than the DtN coarse space, while for larger coarse space sizes the grid coarse space gives generally more iterations than the DtN coarse space. Both for the coarse grid space and the DtN coarse space, for appropriate choices of the method parameters we obtain iteration counts which grow quite slowly with the wavenumber $\tilde{\omega}$. Further experiments to compare the two definitions of coarse space should be carried out in the heterogenous case.

Two-level preconditioners for Maxwell's equations. Chapter 8 is based on the forthcoming paper [16] and on [15], also submitted to the proceedings of the DD24 International Conference on Domain Decomposition Methods. We investigate how the two-level domain decomposition preconditioners analyzed for the Helmholtz equation in [62] work in the Maxwell case, both from the theoretical and numerical points of view. We aim at finding a "good" preconditioner, which, in the present context, means that the number of iterations needed to solve the preconditioned system should be independent of the wavenumber $\tilde{\omega}$.

We present a new theory for the time-harmonic Maxwell's equation *with absorption* that provides rates of convergence for GMRES with a two-level Additive Schwarz (AS) preconditioner, *explicit in* the wavenumber $\tilde{\omega}$, the absorption ξ, the coarse-grid diameter H_{cs}, the subdomain diameter H_{sub} and the overlap size δ. These theory uses a $\tilde{\omega}$- and ξ-explicit coercivity result for the underlying sesquilinear form and the main theorems give an upper bound on the norm of the preconditioned matrix and a lower bound on its field of values, so that Elman-type estimates for the convergence of GMRES can be applied. Note that analyzing the convergence of GMRES is hard, since no convergence estimates in terms of the condition number, as those we are used to with the conjugate gradient method, are available for GMRES. An important particular case of the final convergence estimate is the following. If the problem has absorption $\xi \sim \tilde{\omega}^2$, $\delta \sim H_{\text{cs}}$ (which is called generous overlap), $H_{\text{sub}} \sim H_{\text{cs}} \sim \tilde{\omega}^{-1}$, then GMRES preconditioned with the two-level Additive Schwarz method will converge with the number of iterations *independent of the wavenumber*. Note that the detailed theory development is mainly due to Euan Spence.

Large scale numerical experiments are carried out not only in the setting covered by the theory, but also for the time-harmonic Maxwell's equation *without absorption*. This extensive numerical study of the convergence of GMRES examines several versions of the two-level preconditioner: additive and hybrid coarse correction formulas, standard and optimized local solves, generous and minimal overlap, different scalings of the subdomain diameter and the coarse grid diameter with respect to the wavenumber $\tilde{\omega}$. Moreover, we test various levels of absorption in the problem and in the preconditioner, and we consider discretizations of the problem with degree 1 and 2 curl-conforming finite elements.

As a conclusion, some directions for future research are presented.

This thesis led to the following publications and presentations.

Journal and conference papers

- [12] M. Bonazzoli, V. Dolean, F. Rapetti, P.-H. Tournier. "Parallel preconditioners for high order discretizations arising from full system modeling for brain microwave imaging". International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 2017 <hal-01328197>

- [20] M. Bonazzoli, F. Rapetti. "High-order finite elements in numerical electromagnetism: degrees of freedom and generators in duality". Numerical Algorithms, Springer, 2017 <hal-01260354>

- [21] M. Bonazzoli, F. Rapetti, C. Venturini. "Dispersion analysis of triangle-based Whitney element methods for electromagnetic wave propagation". Proceedings of ESCO 2016, 5th European Seminar on Computing. Applied Mathematics and Computation, 2017.

- [14] M. Bonazzoli, F. Rapetti, P.-H. Tournier, C. Venturini. "High order edge elements for electromagnetic waves: remarks on numerical dispersion". Proceedings of ICOSAHOM 2016 International Conference on Spectral and High Order Methods. *Accepted for publication*

- [113] P.-H. Tournier, M. Bonazzoli, V. Dolean, F. Rapetti, F. Hecht, F. Nataf, I. Aliferis, I. El Kanfoud, C. Migliaccio, M. de Buhan, M. Darbas, S. Semenov, C. Pichot. "Numerical modelling and high speed parallel computing: new perspectives for brain strokes detection and monitoring". IEEE Antennas and Propagation Magazine, Special issue on Electromagnetic Inverse Problems for Sensing and Imaging, invited paper. *Accepted for publication*

- [19] M. Bonazzoli, V. Dolean, R. Pasquetti, and F. Rapetti. "Schwarz preconditioning for high order edge element discretizations of the time-harmonic Maxwell's equations". Proceedings of DD23 International Conference on Domain Decomposition Methods 2015. Lecture Notes in Computational Science and Engineering, Springer, 2017 <hal-01250761>

- [13] M. Bonazzoli, E. Gaburro, V. Dolean, F. Rapetti. "High order edge finite element approximations for the time-harmonic Maxwell's equations". Proceedings of 2014 IEEE Conference on Antenna Measurements and Applications (CAMA)

Preprints

- [15] M. Bonazzoli, V. Dolean, I.G. Graham, E.A. Spence, P.-H. Tournier. "A two-level domain-decomposition preconditioner for the time-harmonic Maxwell's equations". Proceedings of DD24 International Conference on Domain Decomposition Methods 2017. *Submitted* <hal-01525438>. These results will appear in full in [16], in preparation.

- [17] M. Bonazzoli, V. Dolean, I.G. Graham, E.A. Spence, P.-H. Tournier. "Two-level preconditioners for the Helmholtz equation". Proceedings of DD24 International Conference on Domain Decomposition Methods 2017. *Submitted* <hal-01525424>

- [18] M. Bonazzoli, V. Dolean, F. Hecht, F. Rapetti. "Explicit implementation strategy of high order edge finite elements and Schwarz preconditioning for the time-harmonic Maxwell's equations". *Submitted* <hal-01298938>

- [112] P.-H. Tournier, I. Aliferis, M. Bonazzoli, M. De Buhan, M. Darbas, V. Dolean, F. Hecht, P. Jolivet, I. El Kanfoud, C. Migliaccio, F. Nataf, C. Pichot, S. Semenov. "Microwave tomographic imaging of cerebrovascular accidents by using High-Performance Computing". *Submitted* <hal-01343687>

- [7] I. Ben Gharbia, M. Bonazzoli, X. Claeys, P. Marchand, P.-H. Tournier. "Fast solution of boundary integral equations for elasticity around a crack network: a comparative study". Proceedings of the summer school CEMRACS 2016 (CIRM, Marseille, France, 6 weeks long), fruit of the project "Boundary integral methods for elasticity around a crack network", financed by LJLL-UPMC and IFPEN (Institut Français du Pétrole Énergies Nouvelles). *Submitted*

Conference and workshop presentations

- "Solving numerically large scale electromagnetism problems using FreeFem++: high order methods and parallel computing", The 27th Biennial Numerical Analysis conference 2017, Glasgow, UK

- "Two-level preconditioners for the Helmholtz equation", DD24 International Conference on Domain Decomposition Methods 2017, Svalbard, Norway

- "High order edge element discretizations and preconditioning of the time-harmonic Maxwell's equations", ICOSAHOM 2016, International Conference on Spectral and High Order Methods, Rio de Janeiro, Brazil

- "High order edge elements and domain decomposition preconditioning for the time-harmonic Maxwell's equations", MAFELAP 2016, 15th Conference on the Mathematics of Finite Elements and Applications, London, UK

- "High order finite elements and domain decomposition methods for Maxwell's equations", Colloque des doctorants du Laboratoire J.A. Dieudonné 2016, Barcelonnette, France

- "High performance computing for brain stroke imaging", Athena Days 2016, Institut Fresnel, Marseille, France

- "Domain decomposition preconditioning for high order finite element discretizations of the time-harmonic Maxwell's equations", EMF 2016, 10th International Symposium on Electric and Magnetic Fields, Lyon, France

- "High order edge finite elements for the time-harmonic Maxwell's equations", 2015, Laboratoire J.A. Dieudonné, Nice, France

- "Schwarz preconditioning of high order edge elements type discretisations for the time-harmonic Maxwell's equations", DD23 International Conference on Domain Decomposition Methods 2015, Jeju Island, South Korea

- "High order edge elements for Maxwell's equations: construction and properties", International CAE Conference 2014, Pacengo del Garda, Italy

Chapter 2

Mathematical models of electromagnetism

Contents

2.1 Maxwell's equations

The foundations of the electromagnetic theory are the four Maxwell's equations [83]: Gauss' theorem, Gauss' theorem for magnetism, Ampère-Maxwell's theorem and Faraday's law. In this section, we first present their integral form and deduce their differential form, obtaining a system of differential equations involving the following physical quantities:

- $\boldsymbol{\mathcal{E}}$, the *electric field intensity*, also referred to as the *electric field* (measured in $\mathrm{V\,m^{-1}}$),
- $\boldsymbol{\mathcal{H}}$, the *magnetic field intensity*, also referred to as the *magnetic field* (in $\mathrm{A\,m^{-1}}$),
- $\boldsymbol{\mathcal{D}}$, the *electric induction*, also called the *electric displacement* (in $\mathrm{A\,s\,m^{-2}}$, i.e. $\mathrm{C\,m^{-2}}$),
- $\boldsymbol{\mathcal{B}}$, the *magnetic induction*, also called the *magnetic flux density* (in $\mathrm{V\,s\,m^{-2}}$, i.e. T),
- ρ, the *electric charge density* (in $\mathrm{C\,m^{-3}}$),

- $\boldsymbol{\mathcal{J}}$, the *electric current density* (in $\mathrm{A\,m^{-2}}$),

(where V is the symbol for volt, A for ampère, C for coulomb, T for tesla [1]). Then, using suitable constitutive laws, we will get a system containing only $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{H}}$ as unknowns. We will also derive a second order formulation for the electric field $\boldsymbol{\mathcal{E}}$. Finally, we will present its time-harmonic formulation, which is the focus of this work. Interesting introductions about Maxwell's equations can be found in the first chapter of the books [27, 85].

2.1.1 Gauss' theorem

A distribution of static electric charges is one of the sources that come into play to create an electromagnetic field. Gauss' theorem states that the flux of the electric induction through a closed surface S is equal to the enclosed electric charge Q:

$$\int_S \boldsymbol{\mathcal{D}} \cdot \mathbf{n} = Q, \tag{2.1}$$

where $\mathbf{n}$ is the outward unit normal to S.

Denoting by V the volume bounded by the surface S, if we write the total charge Q in V as a function of the electric charge density ρ, $Q = \int_V \rho \, dv$, and we apply the divergence theorem

$$\int_S \boldsymbol{\mathcal{D}} \cdot \mathbf{n} = \int_V \nabla \cdot \boldsymbol{\mathcal{D}} \,,$$

we get:

$$\int_V \nabla \cdot \boldsymbol{\mathcal{D}} = \int_V \rho \,.$$

Since the volume V is arbitrary, this implies the differential form of *Gauss' theorem*:

$$\nabla \cdot \boldsymbol{\mathcal{D}} = \rho. \tag{2.2}$$

2.1.2 Gauss' theorem for magnetism

Gauss' theorem for magnetism states that the flux of the magnetic induction through a closed surface S is always zero:

$$\int_S \boldsymbol{\mathcal{B}} \cdot \mathbf{n} = 0, \tag{2.3}$$

which formally expresses the fact that there is currently no experimental evidence of the existence of magnetic charges or magnetic monopoles.

Again, by applying the divergence theorem we get

$$\int_V \nabla \cdot \boldsymbol{\mathcal{B}} = 0,$$

where V is the volume bounded by the surface S, and, since the volume V is arbitrary, we obtain the differential form of *Gauss' theorem for magnetism*:

$$\nabla \cdot \boldsymbol{\mathcal{B}} = 0. \tag{2.4}$$

1. Note that §5.2 of the SI brochure (8th edition), which defines the International System of Units, states that in English the names of units start with a lower-case letter, even when the symbol for the unit begins with a capital letter.

2.1.3 Ampère's theorem

Ampère's original theorem

Ampère's theorem in its original form expresses the relationship between magnetic fields and electric currents that produce them, in absence of time-changing electric field. In particular, it states that the circulation of the magnetic field $\boldsymbol{\mathcal{H}}$ along a closed curve C is equal to the current which crosses a surface S insisting on the curve C:

$$\int_C \boldsymbol{\mathcal{H}} \cdot \mathbf{t} = \int_S \boldsymbol{\mathcal{J}} \cdot \mathbf{n}, \tag{2.5}$$

where $\mathbf{t}$ is the unit tangent to the oriented curve C and the normal $\mathbf{n}$ is oriented according to the orientation of C following the right-hand rule.

Applying Stokes' theorem to the left-hand side of this integral form, we get:

$$\int_S (\nabla \times \boldsymbol{\mathcal{H}}) \cdot \mathbf{n} = \int_S \boldsymbol{\mathcal{J}} \cdot \mathbf{n},$$

and, since the surface S is arbitrary, we obtain the differential form of *Ampère's original theorem*, namely

$$\nabla \times \boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{J}}. \tag{2.6}$$

Ampère-Maxwell theorem

Unless the electric charge density ρ does not vary in time (as in the magnetostatic case), Ampère's original theorem is not consistent with the *law of conservation of the electric charge*. Indeed, this law states that the total current into a volume V with surface S must be equal to the charge variation within the volume:

$$\int_S \boldsymbol{\mathcal{J}} \cdot \mathbf{n} = -\frac{d}{dt} \int_V \rho \,;$$

equivalently, applying the divergence theorem on the left, and exchanging the order of differentiation and integration on the right, we may write

$$\int_V \nabla \cdot \boldsymbol{\mathcal{J}} = -\int_V \frac{\partial \rho}{\partial t} \,,$$

that as usual implies:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{J}} = 0. \tag{2.7}$$

In order to see why equation (2.6) is in contradiction with the conservation of the electric charge, we apply to it the divergence operator and, since $\nabla \cdot (\nabla \times \boldsymbol{\mathcal{H}}) = 0$, we find $\nabla \cdot \boldsymbol{\mathcal{J}} = 0$, which is not consistent with (2.7) if $\partial\rho/\partial t \neq 0$.

Therefore, Maxwell's main contribution was to introduce a correction to (2.6), by adding what he called the *displacement current* $\boldsymbol{\mathcal{J}}_D = \frac{\partial \boldsymbol{\mathcal{D}}}{\partial t}$:

$$\nabla \times \boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{J}} + \boldsymbol{\mathcal{J}}_D.$$

This equation is now coherent with the charge conservation: applying the divergence operator we get

$$0 = \nabla \cdot (\boldsymbol{\mathcal{J}} + \boldsymbol{\mathcal{J}}_D) = \nabla \cdot \boldsymbol{\mathcal{J}} + \frac{\partial (\nabla \cdot \boldsymbol{\mathcal{D}})}{\partial t} = \nabla \cdot \boldsymbol{\mathcal{J}} + \frac{\partial \rho}{\partial t},$$

thus relation (2.7) holds. The corrected equation is now often called the *Ampère-Maxwell theorem*:

$$-\frac{\partial \boldsymbol{\mathcal{D}}}{\partial t} + \nabla \times \boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{J}}. \tag{2.8}$$

In particular, a variation of the flux of the electric induction creates a magnetic field.

2.1.4 Faraday's law

Faraday's law says that an electric current is induced in any closed circuit C when the flux of the magnetic induction through a surface S bounded by the circuit changes in time. The induced electromotive force is opposed to the variation of the magnetic flux:

$$\int_C \boldsymbol{\mathcal{E}} \cdot \mathbf{t} = -\frac{d}{dt} \int_S \boldsymbol{\mathcal{B}} \cdot \mathbf{n}. \tag{2.9}$$

By applying Stokes' theorem on the left and exchanging the order of differentiation and integration on the right, we get:

$$\int_S (\nabla \times \boldsymbol{\mathcal{E}}) \cdot \mathbf{n} = -\int_S \frac{\partial \boldsymbol{\mathcal{B}}}{\partial t} \cdot \mathbf{n}.$$

Using again the fact that the surface S is arbitrary, we can write *Faraday's law* in differential form:

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial t} + \nabla \times \boldsymbol{\mathcal{E}} = \mathbf{0}. \tag{2.10}$$

2.1.5 Maxwell's system and constitutive laws

Equations (2.8), (2.10), (2.2) and (2.4) form the complete Maxwell's system of electromagnetism:

$$-\frac{\partial \boldsymbol{\mathcal{D}}}{\partial t} + \nabla \times \boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{J}}, \qquad \text{(Ampère-Maxwell theorem)} \tag{2.11a}$$

$$\frac{\partial \boldsymbol{\mathcal{B}}}{\partial t} + \nabla \times \boldsymbol{\mathcal{E}} = \mathbf{0}, \qquad \text{(Faraday's law)} \tag{2.11b}$$

$$\nabla \cdot \boldsymbol{\mathcal{D}} = \rho, \qquad \text{(Gauss' theorem)} \tag{2.11c}$$

$$\nabla \cdot \boldsymbol{\mathcal{B}} = 0, \qquad \text{(Gauss' theorem for magnetism)} \tag{2.11d}$$

relating four vector fields $\boldsymbol{\mathcal{E}}$, $\boldsymbol{\mathcal{H}}$, $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{B}}$, and source terms ρ, $\boldsymbol{\mathcal{J}}$ (which are also related by the equation of charge conservation (2.7)). It turns out that these equations are not sufficient to uniquely determine the electromagnetic field and that additional *constitutive laws* are needed, which model at a macroscopic scale the field-matter interaction. They depend on the properties of the materials in the domain occupied by the electromagnetic field.

Here, we consider the case of *linear isotropic* materials (i.e. whose properties do not depend on the direction of the field), for which the constitutive laws are

$$\boldsymbol{\mathcal{D}} = \varepsilon \boldsymbol{\mathcal{E}}, \quad \boldsymbol{\mathcal{B}} = \mu \boldsymbol{\mathcal{H}}, \tag{2.12}$$

where the coefficients ε, called *electric permittivity* or *dielectric constant*, and μ, called *magnetic permeability*, are positive, bounded, scalar functions of position. If one considered instead an anisotropic material (e.g. a finely layered medium), ε and μ would be 3×3 positive definite matrix functions of position. In a vacuum we have $\varepsilon = \varepsilon_0 = 8.85 \cdot 10^{-12}\,\mathrm{F/m}$, $\mu = \mu_0 = 1.26 \cdot 10^{-6}\,\mathrm{H/m}$ (F for farad and H for henry), and the speed of light is given by $c_0 = 1/\sqrt{\varepsilon_0 \mu_0}$.

Another constitutive law is a generalized Ohm's law, namely

$$\boldsymbol{\mathcal{J}} = \sigma \boldsymbol{\mathcal{E}} + \boldsymbol{\mathcal{J}}_g, \tag{2.13}$$

which is composed of *Ohm's law* $\boldsymbol{\mathcal{J}} = \sigma \boldsymbol{\mathcal{E}}$, valid for the regions of space occupied by *conductors*, and of an *applied* current density $\boldsymbol{\mathcal{J}}_g$, imposed independently of the local

electromagnetic field, in the regions of space called *generators*. The coefficient σ is a material dependent non negative function of position, called *electrical conductivity* and measured in siemens (S) per meter (the siemens unit is also called mho because it is defined as Ω^{-1}). One has $\sigma = 0$ in *insulators*, as in a vacuum.

Using constitutive laws (2.12), (2.13) (and considering time-independent coefficients), we can rewrite Maxwell's equations in the following form:

$$\varepsilon \frac{\partial \boldsymbol{\mathcal{E}}}{\partial t} - \nabla \times \boldsymbol{\mathcal{H}} + \sigma \boldsymbol{\mathcal{E}} = -\boldsymbol{\mathcal{J}}_g, \tag{2.14}$$

$$\mu \frac{\partial \boldsymbol{\mathcal{H}}}{\partial t} + \nabla \times \boldsymbol{\mathcal{E}} = \mathbf{0}, \tag{2.15}$$

$$\nabla \cdot (\varepsilon \boldsymbol{\mathcal{E}}) = \rho, \tag{2.16}$$

$$\nabla \cdot (\mu \boldsymbol{\mathcal{H}}) = 0. \tag{2.17}$$

2.1.6 First and second order formulations

In order to obtain the first and second order formulations of Maxwell's equations, we assume that the initial conditions already satisfy the two Gauss' theorems: in this way they are automatically satisfied for all time. Indeed, by applying the divergence operator to (2.15), we obtain

$$\frac{\partial}{\partial t}(\nabla \cdot (\mu \boldsymbol{\mathcal{H}})) = 0,$$

which means that if (2.17) is verified at the initial time, then it is verified for all time t. Similarly, by applying the divergence operator to (2.14), we obtain

$$\frac{\partial}{\partial t}(\nabla \cdot (\varepsilon \boldsymbol{\mathcal{E}})) + \nabla \cdot \boldsymbol{\mathcal{J}} = 0,$$

which gives, using (2.7),

$$\frac{\partial}{\partial t}(\nabla \cdot (\varepsilon \boldsymbol{\mathcal{E}}) - \rho) = 0,$$

hence if (2.16) is verified at the initial time, then it is verified for all time t. Nevertheless, a successful numerical scheme to discretize Maxwell's system must produce an approximation that satisfies discrete analogs of (2.16), (2.17).

Therefore, if we don't include Gauss' theorems in the system of Maxwell's equations, we obtain the classical *first order formulation of Maxwell's equations*:

$$\varepsilon \frac{\partial \boldsymbol{\mathcal{E}}}{\partial t} - \nabla \times \boldsymbol{\mathcal{H}} + \sigma \boldsymbol{\mathcal{E}} = -\boldsymbol{\mathcal{J}}_g, \quad \mu \frac{\partial \boldsymbol{\mathcal{H}}}{\partial t} + \nabla \times \boldsymbol{\mathcal{E}} = \mathbf{0}. \tag{2.18}$$

In component form, for $\boldsymbol{\mathcal{E}} = (\mathcal{E}_x, \mathcal{E}_y, \mathcal{E}_z)$, $\boldsymbol{\mathcal{H}} = (\mathcal{H}_x, \mathcal{H}_y, \mathcal{H}_z)$, $\boldsymbol{\mathcal{J}}_g = (\mathcal{J}_{g_x}, \mathcal{J}_{g_y}, \mathcal{J}_{g_z})$, we can rewrite (2.18) as:

$$\begin{cases} \varepsilon \dfrac{\partial \mathcal{E}_x}{\partial t} - \dfrac{\partial \mathcal{H}_z}{\partial y} + \dfrac{\partial \mathcal{H}_y}{\partial z} + \sigma \mathcal{E}_x = -\mathcal{J}_{g_x}, \\ \varepsilon \dfrac{\partial \mathcal{E}_y}{\partial t} - \dfrac{\partial \mathcal{H}_x}{\partial z} + \dfrac{\partial \mathcal{H}_z}{\partial x} + \sigma \mathcal{E}_y = -\mathcal{J}_{g_y}, \\ \varepsilon \dfrac{\partial \mathcal{E}_z}{\partial t} - \dfrac{\partial \mathcal{H}_y}{\partial x} + \dfrac{\partial \mathcal{H}_x}{\partial y} + \sigma \mathcal{E}_z = -\mathcal{J}_{g_z}, \\ \mu \dfrac{\partial \mathcal{H}_x}{\partial t} + \dfrac{\partial \mathcal{E}_z}{\partial y} - \dfrac{\partial \mathcal{E}_y}{\partial z} = 0, \\ \mu \dfrac{\partial \mathcal{H}_y}{\partial t} + \dfrac{\partial \mathcal{E}_x}{\partial z} - \dfrac{\partial \mathcal{E}_z}{\partial x} = 0, \\ \mu \dfrac{\partial \mathcal{H}_z}{\partial t} + \dfrac{\partial \mathcal{E}_y}{\partial x} - \dfrac{\partial \mathcal{E}_x}{\partial y} = 0. \end{cases} \tag{2.19}$$

We can also eliminate the magnetic field $\boldsymbol{\mathcal{H}}$ from (2.18), by applying the curl operator to the second equation divided by μ, and then using the first equation to express $\nabla \times \boldsymbol{\mathcal{H}}$. This leads to the classical *second order (or curl-curl) formulation of Maxwell's equations*:

$$\varepsilon \frac{\partial^2 \boldsymbol{\mathcal{E}}}{\partial t^2} + \sigma \frac{\partial \boldsymbol{\mathcal{E}}}{\partial t} + \nabla \times \left(\frac{1}{\mu} \nabla \times \boldsymbol{\mathcal{E}} \right) = \boldsymbol{\mathcal{S}}_g, \tag{2.20}$$

where $\boldsymbol{\mathcal{S}}_g = -\frac{\partial \boldsymbol{\mathcal{J}}_g}{\partial t}$.

2.1.7 Time-harmonic formulations

If we wish to analyze electromagnetic propagation at a single frequency, the time-dependent problem (2.18), also referred to as problem in the *time domain*, can be reduced to a *time-harmonic* problem, or problem in the *frequency domain*. This is the case, for instance, when the applied current density $\boldsymbol{\mathcal{J}}_g$ is sinusoidal in time (one often says 'harmonic' in time), that is, of the form

$$\boldsymbol{\mathcal{J}}_g(\mathbf{x}, t) = \mathrm{Re}(\mathbf{J}_g(\mathbf{x}) e^{\mathtt{i}\omega t}), \tag{2.21}$$

where $\mathbf{J}_g(\mathbf{x})$ is a *complex-valued* vector function of position $\mathbf{x} \in \mathbb{R}^3$ but not of time $t \in \mathbb{R}$, and $\omega > 0$ is the *angular frequency* ($\mathtt{i}$ denotes the imaginary unit). So, we restrict the analysis to a time-harmonic electromagnetic field varying with an angular frequency ω, i.e. we consider the representation of the electric field $\boldsymbol{\mathcal{E}}$ and the magnetic field $\boldsymbol{\mathcal{H}}$ as

$$\boldsymbol{\mathcal{E}}(\mathbf{x}, t) = \mathrm{Re}(\mathbf{E}(\mathbf{x}) e^{\mathtt{i}\omega t}), \quad \boldsymbol{\mathcal{H}}(\mathbf{x}, t) = \mathrm{Re}(\mathbf{H}(\mathbf{x}) e^{\mathtt{i}\omega t}), \tag{2.22}$$

where $\mathbf{E}$, $\mathbf{H}$ are the *complex amplitudes* (usually denoted by $\hat{\mathbf{E}}$, $\hat{\mathbf{H}}$), dependent only on position. Alternatively, the conversion from the time domain to the frequency domain can be done by applying to the equations the Fourier transform with respect to the time variable, defined as (for a function $\mathbf{p}$ in the time domain)

$$(\mathcal{F}\mathbf{p})(\omega) = \hat{\mathbf{p}}(\omega) = \int_{\mathbb{R}} e^{-\mathtt{i}\omega t} \mathbf{p}(t)\, dt,$$

with the inverse Fourier transform given by

$$(\mathcal{F}^{-1}\hat{\mathbf{p}})(t) = \mathbf{p}(t) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{\mathtt{i}\omega t} \hat{\mathbf{p}}(\omega)\, d\omega.$$

Recall its property of linearity and behavior on derivatives:

$$\mathcal{F}\left(\frac{\partial \mathbf{p}}{\partial t} \right) = \mathtt{i}\omega \hat{\mathbf{p}}(\omega).$$

Thus, either by substituting expressions (2.21), (2.22) into (2.18), or by applying the Fourier transform (and omitting the hat symbol), we obtain the *first order time-harmonic formulation* of Maxwell's equations:

$$\mathtt{i}\omega\varepsilon\mathbf{E} - \nabla \times \mathbf{H} + \sigma\mathbf{E} = -\mathbf{J}_g, \quad \mathtt{i}\omega\mu\mathbf{H} + \nabla \times \mathbf{E} = \mathbf{0}. \tag{2.23}$$

Moreover, we can rewrite the first equation in (2.23) by factoring out $\mathtt{i}\omega\mathbf{E}$ and introducing a *complex-valued electric permittivity* ε_σ

$$\varepsilon_\sigma = \varepsilon - \mathtt{i}\frac{\sigma}{\omega}, \tag{2.24}$$

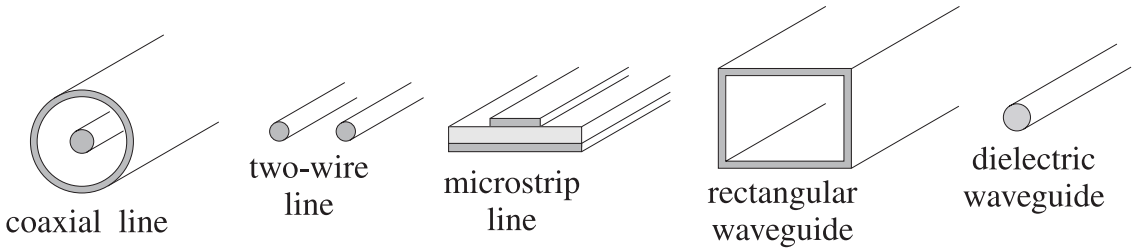


Figure 2.1 – Typical waveguiding structures.

so that system (2.23) with $\mathbf{J}_g = 0$ becomes

$$\nabla \times \mathbf{H} = \mathtt{i}\omega\varepsilon_\sigma \mathbf{E}, \quad \nabla \times \mathbf{E} = -\mathtt{i}\omega\mu\mathbf{H}. \tag{2.25}$$

Now, we can eliminate the field $\mathbf{H}$ (by solving the second equation for $\mathbf{H}$ and substituting into the first equation), and, supposing that μ is constant, we obtain the *second order (or curl-curl) time-harmonic formulation* for the electric field:

$$\nabla \times (\nabla \times \mathbf{E}) - \gamma^2 \mathbf{E} = \mathbf{0}, \tag{2.26}$$

where the (complex-valued) coefficient γ is related to the physical parameters as follows

$$\gamma = \omega\sqrt{\mu\varepsilon_\sigma} = \sqrt{\omega^2\mu\varepsilon - \mathtt{i}\omega\mu\sigma}, \qquad \varepsilon_\sigma = \varepsilon - \mathtt{i}\frac{\sigma}{\omega}.$$

Note that if $\sigma = 0$, we have $\gamma = \tilde{\omega}$, $\tilde{\omega} = \omega\sqrt{\mu\varepsilon}$ being the *wavenumber*. We can also write $\tilde{\omega} = \omega/c$ introducing the *propagation speed* $c = 1/\sqrt{\varepsilon\mu}$. Working with a complex-valued electric permittivity ε_σ makes it possible to include dissipative effects when $\sigma > 0$.

Remark 2.1 (Sign convention!)**.** Note that in this work we chose the *sign convention* with $e^{+\mathtt{i}\omega t}$ in the time-harmonic assumption, which results in a negative imaginary part in the complex valued electric permittivity $\varepsilon_\sigma = \varepsilon - \mathtt{i}\sigma/\omega$. Some authors instead choose a time dependence of $e^{-\mathtt{i}\omega t}$, which gives, repeating similar calculations, a positive imaginary part in ε_σ.

Remark 2.2. In the hypothesis $\nabla\cdot\mathbf{E} = 0$, using the vector calculus identity $\nabla\times(\nabla\times\mathbf{A}) = -\Delta\mathbf{A} + \nabla(\nabla\cdot\mathbf{A})$, we would get that (2.26) is equivalent to a *vector Helmholtz equation with absorption (or damping)*:

$$-\Delta\mathbf{E} - (\tilde{\omega}^2 - \mathtt{i}\omega\mu\sigma)\mathbf{E} = \mathbf{0}. \tag{2.27}$$

2.2 The waveguide problem

Waveguides are used to transfer electromagnetic power efficiently from one point in space to another. Some common guiding structures are shown in Figure 2.1 [93], including coaxial cables, two-wire and microstrip transmission lines, hollow metallic waveguides, and dielectric waveguides as optical fibers. Here, we are interested in hollow metallic *rectangular waveguides*, of cross section given by the rectangle $(0, a) \times (0, b)$. The guide is typically filled with air, but any other dielectric material, with permittivity ε, permeability μ, conductivity σ, can be considered.

To simulate numerically the electromagnetic wave propagation in such waveguide structures, first of all consider as computational domain a bounded section $\Omega = (0, a) \times (0, b) \times$

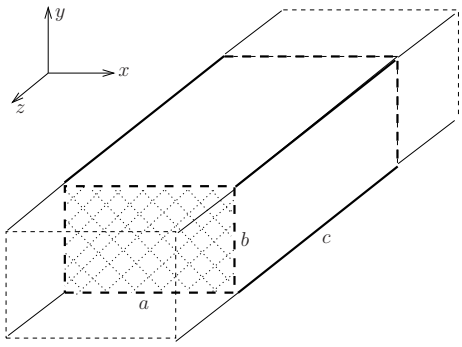


Figure 2.2 – Rectangular waveguide configuration with wave propagation in the z direction. The physical domain $\mathcal{D}$ is in thin line, with dashed style for those boundaries that should be extended to infinity. The computational domain Ω is in thick line, with dashed style for those boundaries where suitable absorbing conditions are imposed.

$(0, c)$ of the physical domain $\mathcal{D} \subset \mathbb{R}^3$, which is an infinite 'parallelepiped' parallel to the z direction, as shown in Figure 2.2. Then, we need to solve the following boundary value problem, consisting of equation (2.26) with suitable boundary conditions (described in the subsections below):

$$\begin{cases} \nabla \times (\nabla \times \mathbf{E}) - \gamma^2 \mathbf{E} = \mathbf{0}, \text{ in } \Omega, & \text{(2.28a)} \\ \mathbf{E} \times \mathbf{n} = \mathbf{0}, \text{ on } \Gamma_{\mathrm{w}}, & \text{(2.28b)} \\ (\nabla \times \mathbf{E}) \times \mathbf{n} + \mathtt{i}\eta\, \mathbf{n} \times (\mathbf{E} \times \mathbf{n}) = \mathbf{g}^{\mathrm{in}}, \text{ on } \Gamma_{\mathrm{in}}, & \text{(2.28c)} \\ (\nabla \times \mathbf{E}) \times \mathbf{n} + \mathtt{i}\eta\, \mathbf{n} \times (\mathbf{E} \times \mathbf{n}) = \mathbf{g}^{\mathrm{out}}, \text{ on } \Gamma_{\mathrm{out}}, & \text{(2.28d)} \end{cases}$$

where $\mathbf{n}$ is the unit outward normal to $\Gamma = \partial\Omega$. Considering the vector $\mathbf{e}_z = (0, 0, 1)^t$, the border Γ is composed of $\Gamma_{\mathrm{w}} = \{\mathbf{x} \in \partial\Omega,\ \mathbf{n}(\mathbf{x}) \cdot \mathbf{e}_z = 0\}$ representing the border of the waveguide walls, $\Gamma_{\mathrm{in}} = \{\mathbf{x} \in \partial\Omega,\ \mathbf{n}(\mathbf{x}) \cdot \mathbf{e}_z < 0\}$ which is the waveguide entrance, and $\Gamma_{\mathrm{out}} = \{\mathbf{x} \in \partial\Omega,\ \mathbf{n}(\mathbf{x}) \cdot \mathbf{e}_z > 0\}$, the waveguide exit. The parameter η is a positive real number, and the vector functions $\mathbf{g}^{\mathrm{in}}$, $\mathbf{g}^{\mathrm{out}}$ depend on the incident wave.

In the following, we discuss the boundary conditions appearing in the boundary value problem, which are specific to electromagnetism. Then we find particular exact solutions (called *modes* of the waveguide) in the case $\sigma = 0$, for an infinitely long waveguide. We consider also a two-dimensional configuration of the problem. Finally, we derive the variational (or weak) formulation of the problem.

2.2.1 Metallic boundary conditions

Boundary conditions (2.28b) express the fact that the metallic walls of the waveguide are modeled as a *perfect electric conductor* (PEC).

Indeed, inside a perfect conductor the electric field vanishes: heuristically, from Ohm's law ($\mathbf{J} = \sigma\mathbf{E}$) we see that if the conductivity $\sigma \to \infty$ and if the current density $\mathbf{J}$ is to remain bounded, then $\mathbf{E} \to 0$.

Moreover, the electric field has *continuous tangential component* across a surface S (Γ_{w} in our case) separating two regions with different materials. More precisely, if $\mathbf{n}$ is the unit normal to S pointing from region 1 to region 2, and if $\mathbf{E}_1$ denotes the limiting value of the electric field as S is approached from region 1 and $\mathbf{E}_2$ denotes the limiting value of the field from the other region, then we have:

$$\mathbf{E}_1 \times \mathbf{n} = \mathbf{E}_2 \times \mathbf{n} \quad \text{on } S. \tag{2.29}$$

This property of tangential continuity of the electric field can be derived from the integral form of Faraday's law (2.9) [2]; furthermore, (2.29) is a requirement for $\nabla \times \mathbf{E}$ to be well defined in a least squares sense, see Lemma 5.3 of [85]. Note that the normal component can jump across material discontinuities instead. This must be taken into account when designing a numerical scheme for approximating Maxwell's equations.

Now, if the material on one side (say region 2) of the interface S is a perfect conductor, like the waveguide walls in our case, then $\mathbf{E}_2 = 0$ in (2.29), that is

$$\mathbf{E}_1 \times \mathbf{n} = \mathbf{0} \quad \text{on } S,$$

where region 1 is the domain Ω and S is the border Γ_{w}.

2.2.2 Impedance boundary conditions

Boundary conditions (2.28c), (2.28d) on the artificial boundaries Γ_{in}, Γ_{out} are called *impedance boundary conditions*, which are Robin-type boundary conditions. On one hand, they model the fact that the waveguide is connected to electronic components such as antennas. On the other hand, they are *absorbing* boundary conditions, first order approximations of *transparent* boundary conditions. Transparent boundary conditions are defined to let outgoing waves pass through the artificial boundaries of the domain unaffected, to obtain in the truncated domain Ω the same solution as in the infinite domain $\mathcal{D}$; however, transparent boundary conditions involve non local operators, so they need to be approximated in order to be used in practice.

Consider the first order absorbing boundary condition

$$\frac{1}{Z}\,\mathbf{n} \times \mathbf{E} - \mathbf{n} \times (\mathbf{H} \times \mathbf{n}) = \mathbf{0}, \tag{2.30}$$

where $Z = \sqrt{\mu/\varepsilon}$ is the *wave impedance*. This condition mimics the *Silver-Müller* radiation condition for the field scattered by an object. Now, by substituting $\mathbf{H} = -\frac{1}{\mathtt{i}\omega\mu}\nabla \times \mathbf{E}$ from the second equation of the time-harmonic system (2.23) inside (2.30) we get:

$$\frac{1}{Z}\,\mathbf{n} \times \mathbf{E} + \frac{1}{\mathtt{i}\omega\mu}\,\mathbf{n} \times ((\nabla \times \mathbf{E}) \times \mathbf{n}) = \mathbf{0},$$

that is, multiplying by $\mathtt{i}\omega\mu$ and using $\mathtt{i}\omega\mu/Z = \mathtt{i}\omega\mu\sqrt{\mu/\varepsilon} = \mathtt{i}\omega\sqrt{\mu\varepsilon} = \mathtt{i}\tilde{\omega}$,

$$\mathtt{i}\tilde{\omega}\,\mathbf{n} \times \mathbf{E} + \mathbf{n} \times ((\nabla \times \mathbf{E}) \times \mathbf{n}) = \mathbf{0}.$$

Therefore we have

$$((\nabla \times \mathbf{E}) \times \mathbf{n}) \times \mathbf{n} + \mathtt{i}\tilde{\omega}\,\mathbf{E} \times \mathbf{n} = \mathbf{0},$$

and, after vector product multiplication by $\mathbf{n}$,

$$(\nabla \times \mathbf{E}) \times \mathbf{n} + \mathtt{i}\tilde{\omega}\,\mathbf{n} \times (\mathbf{E} \times \mathbf{n}) = \mathbf{0}. \tag{2.31}$$

Remark 2.3 (Sign convention!)**.** Note that the adopted *sign convention* with $e^{+\mathtt{i}\omega t}$ in the time-harmonic assumption (2.22) results in a positive sign $+\mathtt{i}\tilde{\omega}$ in the absorbing impedance boundary condition written as in (2.31). The opposite sign choice of $e^{-\mathtt{i}\omega t}$ would give, repeating similar calculations, the parameter $-\mathtt{i}\tilde{\omega}$ in the condition.

2. Thanks to (2.9), it can be proven that $\mathbf{E}_1 \cdot \mathbf{t} = \mathbf{E}_2 \cdot \mathbf{t}$ on S for any tangential direction $\mathbf{t}$. Note that $\mathbf{E} \times \mathbf{n}$ is actually the *tangential trace* of $\mathbf{E}$, while the *tangential component* is $\mathbf{n} \times (\mathbf{E} \times \mathbf{n}) = \mathbf{E} - (\mathbf{E} \cdot \mathbf{n})\mathbf{n}$.

2.2.3 Waveguide modes

We look for particular exact solutions (also called *modes*) of Maxwell's equations, in the non dissipative case ($\sigma = 0$), that propagate along the (infinitely long) waveguide principal axis (the z direction). The complex amplitudes $\mathbf{E}$, $\mathbf{H}$ in the time-harmonic representation (2.22) (of angular frequency ω) are assumed to have the form:

$$\mathbf{E}(\mathbf{x}) = \mathbf{E}(x, y, z) = \widetilde{\mathbf{E}}(x, y)e^{-\mathtt{i}\beta z}, \quad \mathbf{H}(\mathbf{x}) = \mathbf{H}(x, y, z) = \widetilde{\mathbf{H}}(x, y)e^{-\mathtt{i}\beta z}, \tag{2.32}$$

where $\beta > 0$ is the propagation wavenumber along the guide direction. With these assumptions, the component form of the first order formulation of Maxwell's equations (2.19), with $\sigma = 0$, becomes (dropping the tilde symbol):

$$\begin{cases} \mathtt{i}\omega\varepsilon E_x - \dfrac{\partial H_z}{\partial y} - \mathtt{i}\beta H_y = 0, \\ \mathtt{i}\omega\varepsilon E_y + \mathtt{i}\beta H_x + \dfrac{\partial H_z}{\partial x} = 0, \\ \mathtt{i}\omega\varepsilon E_z - \dfrac{\partial H_y}{\partial x} + \dfrac{\partial H_x}{\partial y} = 0, \\ \mathtt{i}\omega\mu H_x + \dfrac{\partial E_z}{\partial y} + \mathtt{i}\beta E_y = 0, \\ \mathtt{i}\omega\mu H_y - \mathtt{i}\beta E_x - \dfrac{\partial E_z}{\partial x} = 0, \\ \mathtt{i}\omega\mu H_z + \dfrac{\partial E_y}{\partial x} - \dfrac{\partial E_x}{\partial y} = 0. \end{cases} \tag{2.33}$$

The metallic boundary condition (2.28b) on Γ_{w} translates into

$$\begin{aligned} E_y = E_z = 0, \ \text{on } x = 0 \text{ and } x = a, \\ E_x = E_z = 0, \ \text{on } y = 0 \text{ and } y = b. \end{aligned} \tag{2.34}$$

We are interested in finding the exact solutions in two particular cases: the *transverse electric* case (TE) in which the longitudinal component $E_z = 0$, and the *transverse magnetic* case (TM) in which the longitudinal component $H_z = 0$.

TE modes

If we take $E_z = 0$, equations (2.33) become

$$\begin{cases} \mathtt{i}\omega\varepsilon E_x - \dfrac{\partial H_z}{\partial y} - \mathtt{i}\beta H_y = 0, \\ \mathtt{i}\omega\varepsilon E_y + \mathtt{i}\beta H_x + \dfrac{\partial H_z}{\partial x} = 0, \\ -\dfrac{\partial H_y}{\partial x} + \dfrac{\partial H_x}{\partial y} = 0, \\ \mathtt{i}\omega\mu H_x + \mathtt{i}\beta E_y = 0, \\ \mathtt{i}\omega\mu H_y - \mathtt{i}\beta E_x = 0, \\ \mathtt{i}\omega\mu H_z + \dfrac{\partial E_y}{\partial x} - \dfrac{\partial E_x}{\partial y} = 0. \end{cases} \tag{2.35}$$

The last three equations in (2.35) yield:

$$H_x = -\frac{\beta}{\mu\omega}E_y, \quad H_y = \frac{\beta}{\mu\omega}E_x, \quad H_z = \frac{1}{\mathtt{i}\omega\mu}\left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}\right), \tag{2.36}$$

and if we substitute these expressions of H_x, H_y, H_z into the first three equations in (2.35) (and we multiply the first two by $\mathtt{i}\omega\mu$), we get:

$$\begin{aligned}
&\left(\beta^2 - \omega^2\mu\varepsilon\right) E_x - \frac{\partial^2 E_x}{\partial x^2} - \frac{\partial^2 E_x}{\partial y^2} = 0,\\
&\left(\beta^2 - \omega^2\mu\varepsilon\right) E_y - \frac{\partial^2 E_y}{\partial x^2} - \frac{\partial^2 E_y}{\partial y^2} = 0,\\
&\frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} = 0.
\end{aligned}$$

To summarize, if we consider the boundary conditions given by (2.34) and we recall that $\tilde{\omega} = \omega\sqrt{\varepsilon\mu}$ is the wavenumber, it is enough to solve the following *Helmholtz problems* in two dimensions:

$$\begin{cases} \left(\beta^2 - \tilde{\omega}^2\right) E_x - \Delta E_x = 0 \\ E_x(x,0) = E_x(x,b) = 0 \end{cases} \qquad \begin{cases} \left(\beta^2 - \tilde{\omega}^2\right) E_y - \Delta E_y = 0 \\ E_y(0,y) = E_y(a,y) = 0 \end{cases}$$

where E_x and E_y are related by the divergence condition $\frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} = 0$. The technique of separation of variables gives the following solutions of the above equations

$$E_x = C\frac{n\pi}{b}\cos\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right), \quad E_y = -C\frac{m\pi}{a}\sin\left(\frac{m\pi x}{a}\right)\cos\left(\frac{n\pi y}{b}\right),$$

where $m, n \in \mathbb{N}$, provided that

$$\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2 = \tilde{\omega}^2 - \beta^2, \tag{2.37}$$

which is called *dispersion relation.*

If we plug the expressions we have just obtained of E_x and E_y in equation (2.36), we see that:

$$\begin{aligned}
H_x &= C\frac{\beta}{\omega\mu}\frac{m\pi}{a}\sin\left(\frac{m\pi x}{a}\right)\cos\left(\frac{n\pi y}{b}\right),\\
H_y &= C\frac{\beta}{\omega\mu}\frac{n\pi}{b}\cos\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right),\\
H_z &= C\frac{1}{\mathtt{i}\omega\mu}\left[\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2\right]\cos\left(\frac{m\pi x}{a}\right)\cos\left(\frac{n\pi y}{b}\right).
\end{aligned}$$

To sum up, writing $C = H_0\,\mathtt{i}\omega\mu/(\tilde{\omega}^2 - \beta^2)$, we conclude that the general form of the TE modes is (after multiplication by $e^{-\mathtt{i}\beta z}$ of the previous solutions):

$$\begin{cases}
E_x^{TE} = H_0\dfrac{\mathtt{i}\omega\mu}{\tilde{\omega}^2-\beta^2}\dfrac{n\pi}{b}\cos\left(\dfrac{m\pi x}{a}\right)\sin\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z},\\
E_y^{TE} = -H_0\dfrac{\mathtt{i}\omega\mu}{\tilde{\omega}^2-\beta^2}\dfrac{m\pi}{a}\sin\left(\dfrac{m\pi x}{a}\right)\cos\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z},\\
E_z^{TE} = 0,\\
H_x^{TE} = H_0\dfrac{\mathtt{i}\beta}{\tilde{\omega}^2-\beta^2}\dfrac{m\pi}{a}\sin\left(\dfrac{m\pi x}{a}\right)\cos\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z},\\
H_y^{TE} = H_0\dfrac{\mathtt{i}\beta}{\tilde{\omega}^2-\beta^2}\dfrac{n\pi}{b}\cos\left(\dfrac{m\pi x}{a}\right)\sin\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z},\\
H_z^{TE} = H_0\cos\left(\dfrac{m\pi x}{a}\right)\cos\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z},
\end{cases} \tag{2.38}$$

and given $m, n \in \mathbb{N}$ the corresponding TE mode is called TE_{mn} *mode.*

Remark 2.4. In the calculations to obtain the field $\mathbf{E}^{TE} = (E_x^{TE}, E_y^{TE}, E_z^{TE})$ we have imposed the metallic boundary conditions on Γ_{w}. Moreover, it satisfies impedance boundary conditions (2.28c), (2.28d) on Γ_{in}, Γ_{out} with parameter $\eta = \beta$ and vector functions $\mathbf{g}^{\mathrm{in}} = (\mathtt{i}\beta + \mathtt{i}\beta)\mathbf{E}^{TE} = 2\mathtt{i}\beta\mathbf{E}^{TE}$ and $\mathbf{g}^{\mathrm{out}} = (-\mathtt{i}\beta + \mathtt{i}\beta)\mathbf{E}^{TE} = \mathbf{0}$.

Remark 2.5. If we consider the definition of the *cutoff frequency*

$$\omega_c = \frac{1}{\sqrt{\varepsilon\mu}}\sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2},$$

from relation (2.37) we have

$$\beta^2 = \tilde{\omega}^2 - \left(\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2\right) = \varepsilon\mu\left(\omega^2 - \omega_c^2\right).$$

So, for $\omega > \omega_c$ the propagation wavenumber β is real, i.e. the corresponding TE_{mn} mode can propagate in the waveguide; for $\omega < \omega_c$, β is imaginary, i.e. the corresponding TE_{mn} mode can not propagate, it attenuates exponentially along the waveguide direction. Usually, waveguide systems are operated in a frequency range that ensures that only the mode with the lowest cutoff frequency can propagate. If the frequency ω is greater than the cutoff frequencies of several modes, then all of these modes can propagate. Conversely, if ω is less than all cutoff frequencies, then none of the modes can propagate. If we arrange the cutoff frequencies in increasing order, $\omega_{c1} < \omega_{c2} < \omega_{c3} < \ldots$, then, to ensure single-mode operation, the frequency should be restricted to the interval $\omega_{c1} < \omega < \omega_{c2}$, so that only the lowest mode will propagate. This interval defines the *operating bandwidth* of the guide [93]. If $a > b$, the mode with the lowest cutoff frequency (called the dominant mode) is the TE_{10} mode, i.e. the one with $m = 1$, $n = 0$.

TM modes

Suppose that now we take $H_z = 0$. In a similar way we can derive the TM solutions:

$$\begin{cases} E_x^{TM} = -E_0 \dfrac{\mathtt{i}\beta}{\tilde{\omega}^2 - \beta^2}\dfrac{m\pi}{a}\cos\left(\dfrac{m\pi x}{a}\right)\sin\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z}, \\ E_y^{TM} = -E_0 \dfrac{\mathtt{i}\beta}{\tilde{\omega}^2 - \beta^2}\dfrac{n\pi}{b}\sin\left(\dfrac{m\pi x}{a}\right)\cos\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z}, \\ E_z^{TM} = E_0 \sin\left(\dfrac{m\pi x}{a}\right)\sin\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z}, \\ H_x^{TM} = E_0 \dfrac{\mathtt{i}\omega\varepsilon}{\tilde{\omega}^2 - \beta^2}\dfrac{n\pi}{b}\sin\left(\dfrac{m\pi x}{a}\right)\cos\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z}, \\ H_y^{TM} = -E_0 \dfrac{\mathtt{i}\omega\varepsilon}{\tilde{\omega}^2 - \beta^2}\dfrac{m\pi}{a}\cos\left(\dfrac{m\pi x}{a}\right)\sin\left(\dfrac{n\pi y}{b}\right)e^{-\mathtt{i}\beta z}, \\ H_z^{TM} = 0. \end{cases} \tag{2.39}$$

2.2.4 Two-dimensional problem

To write a simplified model in 2d, used for preliminary studies and computations, we consider the physical domain $\mathcal{D} \subset \mathbb{R}^3$ given by the space contained between two infinite parallel metallic plates, say $y = 0$, $y = b$ (see Figure 2.3): the wave propagates in the x direction, and all physical parameters μ, σ, ε have to be assumed invariant in the third direction, which is neglected. The computational domain $\Omega \subset \mathbb{R}^2$ is a bounded section, say $\Omega = (0, c) \times (0, b)$, of $\mathcal{D}$.

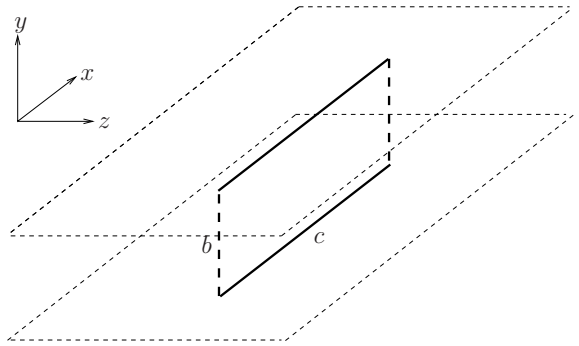


Figure 2.3 – Simplified two-dimensional configuration with wave propagation in the x direction. The physical domain $\mathcal{D}$ is in thin line, with dashed style for those boundaries that should be extended to infinity. The computational domain Ω is in thick line, with dashed style for those boundaries where suitable absorbing conditions are imposed.

By writing $\mathbf{E} = (E_x, E_y, 0)$, we obtain the expression of the curl operator in the two-dimensional setting: $\nabla \times \mathbf{E} = (0, 0, \partial_x E_y - \partial_y E_x)$. Similarly, the vector product $\times$ is calculated by writing $\mathbf{E} = (E_x, E_y, 0)$, $\mathbf{n} = (n_x, n_y, 0)$.

The function $\mathbf{E}_{\text{ex}} = (0, e^{-\mathtt{i}\gamma x})$ verifies the equation of boundary value problem (2.28), the metallic boundary conditions on $\Gamma_{\text{w}} = \{\mathbf{x} \in \partial\Omega, y = 0 \text{ or } y = b\}$, and the impedance boundary conditions on $\Gamma_{\text{in}} = \{\mathbf{x} \in \partial\Omega, x = 0\}$, $\Gamma_{\text{out}} = \{\mathbf{x} \in \partial\Omega, x = c\}$ with parameter $\eta = \tilde{\omega}$ and vector functions $\mathbf{g}^{\text{in}} = (\mathtt{i}\gamma + \mathtt{i}\tilde{\omega})\mathbf{E}_{\text{ex}}$, $\mathbf{g}^{\text{out}} = (-\mathtt{i}\gamma + \mathtt{i}\tilde{\omega})\mathbf{E}_{\text{ex}}$; when $\sigma = 0$ we get $\mathbf{g}^{\text{in}} = 2\mathtt{i}\tilde{\omega}\mathbf{E}_{\text{ex}}$ and $\mathbf{g}^{\text{out}} = \mathbf{0}$. The real part of the propagation constant $-\mathtt{i}\gamma$ gives the rate at which the amplitude changes as the wave propagates, which corresponds to *wave dissipation* (note that if $\sigma > 0$, $\text{Re}(-\mathtt{i}\gamma) < 0$, while if $\sigma = 0$, $\text{Re}(-\mathtt{i}\gamma) = 0$).

2.2.5 Variational formulation

To cast in the variational (or weak) form the boundary value problem (2.28), one has to multiply equation (2.28a) by the complex conjugate of a complex-valued test function $\mathbf{v}$ of a suitable functional space V (specified below), and then integrate over the computational domain Ω using for the first term the integration by parts formula

$$\int_\Omega (\nabla \times \mathbf{w}) \cdot \mathbf{v} = \int_\Omega \mathbf{w} \cdot (\nabla \times \mathbf{v}) - \int_{\partial\Omega} (\mathbf{w} \times \mathbf{n}) \cdot \mathbf{v}. \tag{2.40}$$

Thus we obtain:

$$\int_\Omega \Big[(\nabla \times \mathbf{E}) \cdot (\nabla \times \overline{\mathbf{v}}) - \gamma^2 \mathbf{E} \cdot \overline{\mathbf{v}} \Big] - \int_{\partial\Omega} \Big((\nabla \times \mathbf{E}) \times \mathbf{n} \Big) \cdot \overline{\mathbf{v}} = 0.$$

Now, we can rewrite the boundary term by splitting it into the integrals on the different parts of the boundary:

$$\int_{\partial\Omega} \Big((\nabla \times \mathbf{E}) \times \mathbf{n} \Big) \cdot \overline{\mathbf{v}} = \int_{\Gamma_{\text{w}}} \Big((\nabla \times \mathbf{E}) \times \mathbf{n} \Big) \cdot \overline{\mathbf{v}} + \int_{\Gamma_{\text{in}} \cup \Gamma_{\text{out}}} \Big((\nabla \times \mathbf{E}) \times \mathbf{n} \Big) \cdot \overline{\mathbf{v}}.$$

On Γ_{w}, we choose test functions $\mathbf{v}$ such that $\mathbf{v} \times \mathbf{n} = \mathbf{0}$, so we have

$$\int_{\Gamma_{\text{w}}} \Big((\nabla \times \mathbf{E}) \times \mathbf{n} \Big) \cdot \overline{\mathbf{v}} = - \int_{\Gamma_{\text{w}}} (\nabla \times \mathbf{E}) \cdot (\overline{\mathbf{v}} \times \mathbf{n}) = 0,$$

and on $\Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ we have

$$\begin{aligned}
&\int_{\Gamma_{\text{in}}\cup\Gamma_{\text{out}}} \Big((\nabla \times \mathbf{E}) \times \mathbf{n}\Big) \cdot \overline{\mathbf{v}} = \\
&= -\int_{\Gamma_{\text{in}}\cup\Gamma_{\text{out}}} \mathtt{i}\eta(\mathbf{n} \times (\mathbf{E} \times \mathbf{n})) \cdot \overline{\mathbf{v}} + \int_{\Gamma_{\text{in}}} \mathbf{g}^{\text{in}} \cdot \overline{\mathbf{v}} + \int_{\Gamma_{\text{out}}} \mathbf{g}^{\text{out}} \cdot \overline{\mathbf{v}} \\
&= -\int_{\Gamma_{\text{in}}\cup\Gamma_{\text{out}}} \mathtt{i}\eta(\mathbf{E} \times \mathbf{n}) \cdot (\overline{\mathbf{v}} \times \mathbf{n}) + \int_{\Gamma_{\text{in}}} \mathbf{g}^{\text{in}} \cdot \overline{\mathbf{v}} + \int_{\Gamma_{\text{out}}} \mathbf{g}^{\text{out}} \cdot \overline{\mathbf{v}}.
\end{aligned}$$

Therefore, the weak problem reads: find $\mathbf{E} \in V$ such that

$$\begin{aligned}
\int_\Omega \Big[(\nabla \times \mathbf{E}) \cdot (\nabla \times \overline{\mathbf{v}}) - \gamma^2 \mathbf{E} \cdot \overline{\mathbf{v}}\Big] + \int_{\Gamma_{\text{in}}\cup\Gamma_{\text{out}}} \mathtt{i}\eta(\mathbf{E} \times \mathbf{n}) \cdot (\overline{\mathbf{v}} \times \mathbf{n}) \\
= \int_{\Gamma_{\text{in}}} \mathbf{g}^{\text{in}} \cdot \overline{\mathbf{v}} + \int_{\Gamma_{\text{out}}} \mathbf{g}^{\text{out}} \cdot \overline{\mathbf{v}} \quad \forall \mathbf{v} \in V, \quad (2.41)
\end{aligned}$$

with $V = \{\mathbf{v} \in H(\text{curl}, \Omega), \mathbf{v} \times \mathbf{n} = \mathbf{0} \text{ on } \Gamma_{\text{w}}\}$. For a detailed discussion about existence and uniqueness of solutions we refer to [85] (note that the sign convention therein adopted is the opposite to ours, see Remarks 2.1, 2.3). The functional space $H(\text{curl}, \Omega)$ is a Sobolev space of vector-valued functions appropriate for analyzing Maxwell's equations for the electric field. It is the space of square integrable vector functions whose curl is also square integrable:

$$H(\text{curl}, \Omega) = \{\mathbf{v} \in L^2(\Omega)^3, \nabla \times \mathbf{v} \in L^2(\Omega)^3\},$$

where the derivatives are understood in the weak sense, so that $\nabla \times \mathbf{v}$ satisfies

$$\int_\Omega (\nabla \times \mathbf{v}) \cdot \boldsymbol{\phi} = \int_\Omega \mathbf{v} \cdot (\nabla \times \boldsymbol{\phi}) \quad \forall \boldsymbol{\phi} \in (C_0^\infty(\Omega))^3.$$

The norm and the associated inner product on the space $H(\text{curl}, \Omega)$ are defined as follows:

$$\begin{aligned}
\|\mathbf{v}\|_{H(\text{curl},\Omega)} &= \big(\|\mathbf{v}\|^2_{L^2(\Omega)^3} + \|\nabla \times \mathbf{v}\|^2_{L^2(\Omega)^3}\big)^{1/2}, \\
(\mathbf{u}, \mathbf{v})_{H(\text{curl},\Omega)} &= (\mathbf{u}, \mathbf{v})_{L^2(\Omega)^3} + (\nabla \times \mathbf{u}, \nabla \times \mathbf{v})_{L^2(\Omega)^3}.
\end{aligned}$$

Chapter 3

A revisitation of high order curl-conforming finite elements

The content of this chapter is partially extracted from [20], in collaboration with Francesca Rapetti, published in *Numerical Algorithms (Springer)*.

Contents

3.1 Introduction

When applying the finite element (FE) method to a given problem arising in science and engineering, one writes the variational (or weak) formulation of the problem to be discretized, defines the suitable discrete FE space on a mesh covering the computational domain, and selects the algorithm to solve the final algebraic system in an efficient way [96]. In Section 2.2.5 we have derived the variational formulation of the boundary value problem (2.28) on which we focus in this work, and where the equation is the second order time-harmonic formulation of Maxwell's equations *for the electric field* (2.26); we have introduced the functional space H_{curl} on which the variational formulation of this problem is well defined. In this chapter, we will specify discrete FE spaces in H_{curl} suitable for discretizing the considered problem.

3.1.1 De Rham complex

Nevertheless, one should be aware that H_{curl} belongs to a *complex* of functional spaces $H_{\mathcal{L}} = \{u \in L^2,\, \mathcal{L}\, u \in L^2\}$, where $\mathcal{L}$ is one of the differential operators (the gradient, the curl and the divergence) appearing in the full Maxwell's system (2.11), and acting on scalar or vector fields:

$$H_{\mathrm{grad}} \xrightarrow{\nabla} H_{\mathrm{curl}} \xrightarrow{\nabla\times} H_{\mathrm{div}} \xrightarrow{\nabla\cdot} L^2,$$

where the composition of two consecutive operators is 0. This *de Rham complex* summarizes, for example, the fact that if $q \in H_{\mathrm{grad}}$ then $\nabla q \in H_{\mathrm{curl}}$ (indeed, $\nabla \times (\nabla q) = 0 \in L^2$ and $\nabla q \in L^2$). Note that the space H_{grad} is usually denoted H^1. Finite element spaces in H_{curl} on suitable meshes are used to discretize the electric and magnetic fields, in H_{div} to discretize the electric and magnetic inductions, in L^2 to discretize the electric charge density, in H_{grad} to discretize for example the electric potential $\mathcal{V}_E$ defined by $\boldsymbol{\mathcal{E}} = -\nabla\mathcal{V}_E$. Therefore, one should consider not only discrete FE subspaces W_h^1 of H_{curl}, but also subspaces W_h^0 of H_{grad}, W_h^2 of H_{div} and W_h^3 of L^2, which also form for each h a complex with the same operators, $h > 0$ being the maximal diameter of the elements constituting a mesh $\mathcal{T}_h$ over the computational domain Ω. Moreover, these two complexes are related by interpolation operators Π_h^p, $0 \le p \le 3$, onto the discrete subspaces, forming a diagram

$$\begin{array}{ccccccc}
H_{\mathrm{grad}} & \xrightarrow{\nabla} & H_{\mathrm{curl}} & \xrightarrow{\nabla\times} & H_{\mathrm{div}} & \xrightarrow{\nabla\cdot} & L^2 \\
\Pi_h^0\big\downarrow & & \Pi_h^1\big\downarrow & & \Pi_h^2\big\downarrow & & \Pi_h^3\big\downarrow \\
W_h^0 & \xrightarrow{\nabla} & W_h^1 & \xrightarrow{\nabla\times} & W_h^2 & \xrightarrow{\nabla\cdot} & W_h^3
\end{array}$$

that *commutes*, that is one can follow the arrows along any path between two spaces and obtain the same operator between these two spaces. After the use of the commuting diagram in the approximation of second order elliptic problems in mixed form [43], the central importance of this structure in the approximation of electromagnetism problems was first noticed by Bossavit [24, 27]. Then several works followed, e.g. [68, 69, 9, 3], see [11, §2.1.4] for more references.

For the construction of the generic FE subspace W_h we follow the classical approach of Ciarlet [32]. One introduces first the triple (K, P, Σ), representing a *finite element*, where

- K is the kind of geometrical "tile" of the mesh $\mathcal{T}_h$ over $\bar{\Omega}$ (in our case a simplex, i.e. a triangle in $2d$ and a tetrahedron in $3d$),
- P is a finite-dimensional space of functions (usually polynomials) defined on K,
- Σ is a set of linear functionals σ_i, called *degrees of freedom (dofs)*, acting on P.

The finite element (K, P, Σ) is said to be *unisolvent* if any element z of P is determined once the values $\sigma_i(z)$ are known. One then defines the *FE space* as

$$W_{h,r}^p = \{u \in H_{\mathcal{L}},\, u_{|K} \in P,\, \forall K \in \mathcal{T}_h\},$$

where the integer r denotes the maximal polynomial degree of $u_{|K}$ (resp. of the components of $u_{|K}$) for scalar (resp. vector) fields $u \in H_{\mathcal{L}}$, and the integer $0 \le p \le 3$ refers to the geometrical dimension involved to define the dofs for $u_{|K}$ at the lowest degree. Note that P is a suitable approximation of $H_{\mathcal{L}}$ locally, in each element K, and $W_{h,r}^p$ respects globally, on the whole domain, the *smoothness requirements* of the underlying functional space $H_{\mathcal{L}}$, associated with the boundary value problem to be approximated:

- for H_{grad} (i.e. H^1) the continuity of a global finite element function u is required,

- for H_{curl} the continuity of the *tangential* trace $\mathbf{u}\times\mathbf{n}$ of a global finite element function $\mathbf{u}$ is required (see the proof in e.g. [85], Lemma 5.3),
- for H_{div} the continuity of the *normal* component $\mathbf{u}\cdot\mathbf{n}$ of a global finite element function $\mathbf{u}$ is required.

Indeed, here for a given functional space $H_{\mathcal{L}}$ we consider finite elements (K, P, Σ) which are termed $H_{\mathcal{L}}$-*conforming*, that is, the corresponding global FE space is a subspace of $H_{\mathcal{L}}$. In the case of H_{curl} and H_{div} one often says curl-conforming and div-conforming. The dofs line up to guarantee the needed global smoothness. Note that the tangential continuity is a physical property of the electric field, which is indeed discretized with curl-conforming finite elements.

3.1.2 Characteristics of Nédélec finite elements

FE subspaces $W^0_{h,r} \subset H_{\text{grad}}$, and $W^3_{h,r} \subset L^2$, are well documented in the literature, sets of basis functions of arbitrary order r on tetrahedra are explicitly detailed in books [109, 73]. FE subspaces $W^1_{h,r} \subset H_{\text{curl}}$ and $W^2_{h,r} \subset H_{\text{div}}$ are due to Nédélec [89], where, for a simplex K, the space P is spanned by vectors with *incomplete* or *trimmed* polynomials of degree $\leq r$ as components: this means that some of the top-degree monomials are removed to satisfy some constraints. A second family of Nédélec FEs was introduced in [88], where the definition of P on a simplex K is simpler, as it is a set of vectors with *complete* polynomials of degree $\leq r$ as components. The second family offers superior error estimates for the interpolant compared to the first family, but at the cost of using more dofs for a mesh with a given h. Among the vast literature on high order Nédélec FEs on simplicial meshes we cite [1, 101, 4, 55].

Degrees of freedom for Nédélec FEs are not of Lagrangian type (i.e., they are not the functionals giving the values at mesh nodes): they are functionals involving integrals along a curve or across a surface, with orientation to make things more complicated. For the lowest degree ($r = 1$), dofs are circulations for H_{curl}, or fluxes for H_{div}, of the considered field, but for higher orders ($r > 1$) dofs are *moments*, that are integrals over sub-simplices of the field (or of a component of it) against some function. Moreover, the *duality* property $\sigma_i(\mathbf{w}_j) = \delta_{ij}$ between dofs σ_i and basis functions $\mathbf{w}_j$ of P is not automatically granted for $r > 1$, which is instead the case for the usual H^1-conforming FEs with their nodal dofs. Duality would ensure, for instance, that dofs give the expansion coefficients for writing in terms of the basis a general function in the FE space; so, in particular, the coefficients of the solution vector $\mathbf{u}$ (of the algebraic system) for writing the FE approximation $\mathbf{E}_h$ of the field could be interpreted as the dofs applied to $\mathbf{E}_h$.

In this chapter, we focus on high order finite elements for H_{curl}, since it is the functional space needed for the considered boundary value problem (2.28). We published in [20] the complete study for all the spaces of De Rham complex. Nédélec curl-conforming FEs are often termed *edge elements* because at the lowest degree ($r = 1$) dofs and basis functions are associated with edges of the mesh. There are several reasons to rely on edge elements rather than on classical node-based vector-valued elements [22]. For instance, besides fitting the continuity properties of the electric field, edge elements are known to avoid the pollution of the numerical solution by spurious modes (see §9.3.3 of [27] and [25, 10]).

Here, we adopt the high order basis functions presented in [97, 98] (which belong to the first family of Nédélec FEs), whose definition is rather simple since it only involves the barycentric coordinates of the simplex (see also [59] for previous work in this direction). Their construction, characterized by a geometrical approach, is described in Section 3.4, after introducing in Section 3.2 some notation concerning mesh components and after

recalling in Section 3.3 the definition and properties of the low order FEs. In Section 3.5 we revisit the definition of classical dofs by moments, obtaining a more friendly expression in terms of the considered basis functions. In Section 3.6 we propose a general technique to restore duality between dofs and basis functions for the high order case, thanks to a generalized Vandermonde matrix. We conclude with an example in Section 3.7 illustrating the notions introduced in this Chapter, and with a numerical study in Section 3.8 about the order of convergence of the high order finite element method.

3.2 Notation for mesh components and incidence matrices

Let $\Omega \subset \mathbb{R}^3$ be a bounded domain with piecewise smooth boundary Γ. We consider a *simplicial* mesh of $\bar{\Omega}$, that is a tessellation of $\bar{\Omega}$ by tetrahedra, subject to the condition that any two of them may intersect along a common face, edge or node, but in no other way. We denote by $\mathcal{N}$, $\mathcal{E}$, $\mathcal{F}$, $\mathcal{T}$ (resp. $N_{\mathcal{N}}$, $N_{\mathcal{E}}$, $N_{\mathcal{F}}$, $N_{\mathcal{T}}$) the sets (resp. the cardinality of the sets) of nodes (0-simplices), edges (1-simplices), faces (2-simplices), and tetrahedra (3-simplices) thus obtained, and by $\mathcal{T}_h$ the mesh itself, with $h > 0$ standing for the maximal diameter of the tetrahedra of $\mathcal{T}$. Note that if a simplex s belongs to the mesh $\mathcal{T}_h$, all simplices that form the boundary of s also belong to $\mathcal{T}_h$; each simplex appears only once in $\mathcal{T}_h$. Labels n, e, f, v are used for nodes, edges, faces, volumes (tetrahedra). The *placement* of the mesh stands for the function from $\mathcal{N}$ to $\bar{\Omega}$, giving for each node n_i its position $\mathbf{x}_i$ in $\bar{\Omega}$. The tetrahedron $v = \{n_1, n_2, n_3, n_4\}$ is defined as the non degenerate convex envelope of four points n_1, n_2, n_3, n_4 in $\mathbb{R}^3$, where non degenerate means $(\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1) \cdot (\mathbf{x}_4 - \mathbf{x}_1)$ different from zero. Similarly, a *p-simplex* s, $0 \leq p \leq 3$, is the non degenerate convex envelope of $p+1$ geometrically distinct points $n_1, \ldots, n_{p+1}$. The points $n_1, \ldots, n_{p+1}$ are called *vertices* of s, and p is the *dimension* of the p-simplex s, which we shall denote $s = \{n_1, \ldots, n_{p+1}\}$. Any $(p-1)$-simplex that is a subset of $\{n_1, \ldots, n_{p+1}\}$ is called $(p-1)$-*face* of s.

To make more compact the layout of a formula, the node $n = n_l$ can be denoted as $e - n$ when $e = \{n_i, n_l\}$. The same node becomes $f - e$ when $e = \{n_i, n_j\}$ and $f = \{n_i, n_j, n_l\}$, or $v - f$ when $v = \{n_i, n_j, n_l, n_q\}$ and $f = \{n_i, n_j, n_q\}$.

Besides the list of nodes and of their positions, the mesh data structure also contains *incidence matrices*, saying which node belongs to which edge, which edge bounds which face, etc., and there is a notion of (inner) *orientation* of the simplices to consider. In short, an edge, face, etc., is not only a two-node, three-node, etc., subset of $\mathcal{N}$, but such a set plus an orientation of the simplex it subtends [27].

For example, $e = \{n_i, n_j\}$, where we assume that $n_i < n_j$ for simplicity at the implementation step, denotes the edge that connects the global vertices n_i and n_j, oriented in such a way that the tangent vector goes from the vertex n_i to the vertex n_j. If $e = \{n_i, n_j\}$, the edge $\{n_j, n_i\}$ is referred to as $-e$. One introduces the so-called incidence numbers $\boldsymbol{\partial}_{e,n_i} = -1$, $\boldsymbol{\partial}_{e,n_j} = 1$, and $\boldsymbol{\partial}_{e,n_k} = 0$ for nodes n_k other than n_i and n_j. They form a rectangular matrix $\mathbf{G} = (\boldsymbol{\partial}_{e,n})$, with $N_{\mathcal{E}}$ rows and $N_{\mathcal{N}}$ columns, which describes how edges connect to nodes.

Faces are also oriented, not merely a collection of 3 nodes. For example, $f = \{n_i, n_j, n_k\}$, where we assume that $n_i < n_j < n_k$ for simplicity, is the face with the three vertices n_i, n_j, n_k, oriented such that the vectors $\mathbf{x}_j - \mathbf{x}_i$, $\mathbf{x}_k - \mathbf{x}_i$, form a reference frame in the plane supporting f. An orientation of f induces an orientation of its boundary and, with respect to this induced orientation, an edge runs along or not. Then one introduces the incidence number $\boldsymbol{\partial}_{f,e}$, as $+1$ if e runs along the boundary of f, -1 otherwise, and 0 if e is not one of the edges of f. They form a rectangular matrix $\mathbf{R} = (\boldsymbol{\partial}_{f,e})$, with $N_{\mathcal{F}}$ rows and $N_{\mathcal{E}}$ columns.

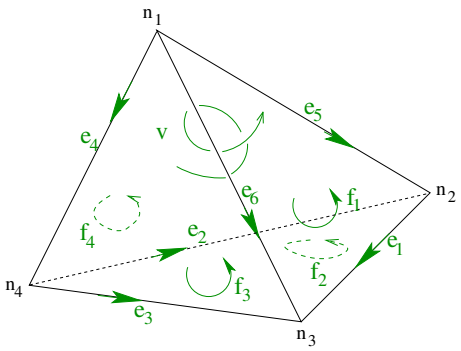


Figure 3.1 – The oriented tetrahedron $v = \{n_1, n_2, n_3, n_4\}$ with oriented p-faces, $0 < p < 3$. The boundary of the face $f_1 = \{n_1, n_3, n_2\}$ is $\partial(f_1) = -e_1 - e_5 + e_6$. It can be identified with the vector $(-1, 0, 0, 0, -1, 1)$ collecting the coefficients in front of each edge, which gives the first line of the incidence matrix $\mathbf{R}$ for v.

A matrix $\mathbf{D} = (\boldsymbol{\partial}_{v,f})$, indexed over $\mathcal{T}$ and $\mathcal{F}$, is similarly defined: $\boldsymbol{\partial}_{v,f} = \pm 1$ if face f bounds tetrahedron v, the sign depending on whether the orientations of f and of the boundary of v match or not. This makes sense only after the tetrahedron v itself has been oriented, and the convention will be that if $v = \{n_i, n_j, n_k, n_l\}$, the vectors $\mathbf{x}_j - \mathbf{x}_i$, $\mathbf{x}_k - \mathbf{x}_i$, and $\mathbf{x}_l - \mathbf{x}_i$, in this order, define a positive frame. So, the incidence number $\boldsymbol{\partial}_{v,f} = 1$ (resp. -1) if the normal of the oriented face f is outward (resp. inward). Implicitly, we have been orienting all nodes the same way, $+1$, up to now.

Example 3.1. For the tetrahedron in Figure 3.1, the three incidence matrices are

$$\mathbf{G} = \begin{pmatrix} 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} -1 & 0 & 0 & 0 & -1 & 1 \\ -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 & 0 \end{pmatrix}, \quad \mathbf{D} = (1 \ -1 \ 1 \ -1).$$

The well known property of incidence matrices is here recalled. (Note that this, together with Remark 3.3, constitutes a sort of discrete version of the de Rham complex.)

Proposition 3.2. *We have* $\mathbf{DR} = \mathbf{0}$ *and* $\mathbf{RG} = \mathbf{0}$.

Proof. Let $e \in \mathcal{E}$ and $v \in \mathcal{T}$. By definition of matrix product, $(\mathbf{DR})_{v,e} = \sum_{f \in \mathcal{F}} \boldsymbol{\partial}_{v,f} \boldsymbol{\partial}_{f,e}$, where the only nonzero terms are for faces f that both contain the edge e and bound the volume v, which means that e is an edge of v. There are exactly two faces f and g of v sharing the edge e. If $\boldsymbol{\partial}_{v,g} = \boldsymbol{\partial}_{v,f}$, then their boundaries are oriented in such a way that e must run along one and counter the other, so $\boldsymbol{\partial}_{g,e} = -\boldsymbol{\partial}_{f,e}$, and the sum is zero. If $\boldsymbol{\partial}_{v,g} = -\boldsymbol{\partial}_{v,f}$, the opposite happens, that is $\boldsymbol{\partial}_{g,e} = \boldsymbol{\partial}_{f,e}$, with the same final result. The proof of $\mathbf{RG} = \mathbf{0}$ is similar. □

3.3 Low order curl-conforming finite elements

Before describing the curl-conforming FE of the lowest degree ($r = 1$), we get familiar with the approach of the triple (K, P, Σ), presented in Section 3.1 to define a finite element, by applying it to the well known continuous nodal FE. For each node $n \in \mathcal{N}$, we denote by

D_n the cluster of tetrahedra in $\mathcal{T}$ which have a vertex at n. With the node $n \in \mathcal{N}$, we associate the continuous, piecewise affine function w^n defined as

$$w^n(\mathbf{x}) = \begin{cases} \lambda_n(\mathbf{x}), & \mathbf{x} \in \bar{D}_n, \\ 0, & \text{otherwise,} \end{cases}$$

where $\lambda_n(\mathbf{x})$ is the *barycentric (or volume) coordinate* of $\mathbf{x}$ with respect to n, computed in the tetrahedron of D_n containing $\mathbf{x}$ (the functions w^n are often called *hat functions*). By construction, $\bar{D}_n$ coincides with the support of w^n and $\sum_{n\in\mathcal{N}} w^n(\mathbf{x}) = 1$, for all $\mathbf{x} \in \bar{D}_n$. Let us denote $\mathbb{P}_1(v)$ the vector space of first order polynomials defined in v: it is generated by the functions w^n, with n a vertex of v. The *nodal* FE (K, P, Σ) for the approximation of scalar fields in v is given by $K = v$, $P = \mathbb{P}_1(v)$, and

$$\Sigma = \{\sigma_n : z \mapsto z(\mathbf{x}_n),\ n \text{ a vertex of } v\}.$$

Then, $W^0_{h,1} = \text{span}\{w^n,\, n \in \mathcal{N}\}$ and its functions are continuous over $\bar{\Omega}$. For a suitable subspace Y^0 [65], the interpolation operator $\Pi^0_h : Y^0 \subset H_{\text{grad}} \to W^0_{h,1}$ associates a function $z \in \mathrm{Y}^0$ with its decomposition on the basis $\{w^n\}$ defined as $\Pi^0_h z(\mathbf{x}) = \sum_{n\in\mathcal{N}} \sigma_n(z) w^n(\mathbf{x})$, for all $\mathbf{x} \in \bar{\Omega}$.

Now, with the (oriented) edge $e = \{n_i, n_j\}$, we associate the vector field

$$\mathbf{w}^e = \lambda_{n_i} \nabla\lambda_{n_j} - \lambda_{n_j} \nabla\lambda_{n_i}. \tag{3.1}$$

It can be shown (see, e.g., Proposition 2 in [28]) that the $\mathbf{w}^e$, varying e among the edges of v, generate the vector space $\mathrm{R}(v) = \{\mathbf{w} \in \mathbb{R}^3,\ \mathbf{w}(\mathbf{x}) = \mathbf{a} \times \mathbf{x} + \mathbf{b},\ \mathbf{a}, \mathbf{b} \in \mathbb{R}^3\}$ of Nédélec first family [89]. The *edge* FE (K, P, Σ) for the approximation of vector fields in v is given by $K = v$, $P = \mathrm{R}(v)$, and

$$\Sigma = \left\{ \sigma_e : \mathbf{z} \mapsto \frac{1}{|e|} \int_e \mathbf{z} \cdot \mathbf{t}_e,\ e \text{ an edge of } v \right\}, \tag{3.2}$$

where $\mathbf{t}_e = \mathbf{x}_j - \mathbf{x}_i$ for $e = \{n_i, n_j\}$ is the tangent vector to the edge e, $|e| = |\mathbf{t}_e|$ the length of e. One has $W^1_{h,1} = \text{span}\{\mathbf{w}^e,\, e \in \mathcal{E}\}$ and its vectors have tangential component continuous across the inter-element faces (see Section 5.2.2, page 141, of [27]). Accordingly, the dofs σ_e are based on a tangential quantity. For a suitable subspace Y^1 [65], the interpolation operator $\Pi^1_h : Y^1 \subset H_{\text{curl}}(\Omega) \to W^1_{h,1}$ assigns to a vector $\mathbf{z} \in \mathrm{Y}^1$ its decomposition on the basis $\{\mathbf{w}^e\}$, defined as $\Pi^1_h \mathbf{z}(\mathbf{x}) = \sum_{e\in\mathcal{E}} \sigma_e(\mathbf{z}) \mathbf{w}^e(\mathbf{x})$, for all $\mathbf{x} \in \bar{\Omega}$.

These definitions of the interpolation operators Π^p_h, $p = 0, 1$, are justified by the fact that the (scalar or vector) basis functions w^j are in *duality* with the dofs σ_i, that is $\sigma_i(w^j) = \delta_{ij}$ [1], $1 \le i, j \le N_{\text{dofs}} = \dim(W^p_{h,1})$. Indeed, the *interpolant* of a suitably smooth (scalar or vector) function z is defined to be the unique function $\Pi^p_h z \in W^p_{h,1}$ such that

$$\sigma_i(\Pi^p_h z - z) = 0, \quad 1 \le i \le N_{\text{dofs}},$$

(note that uniqueness holds because the FE is unisolvent), and by writing it as a linear combination with coefficients c_j of the basis functions w^j

$$\Pi^p_h z = \sum_{j=1}^{N_{\text{dofs}}} c_j w^j,$$

1. The Kronecker delta δ_{ij} is the function whose value is $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ otherwise. For the nodal FE, the duality property $\sigma_{n_i}(w^{n_j}) = \lambda_{n_j}(\mathbf{x}_i) = \delta_{n_i n_j}$ is satisfied thanks to properties of the barycentric coordinates λ_n. For the edge FE, the proof of the duality property $\sigma_{e_i}(\mathbf{w}^{e_j}) = \frac{1}{|e_i|} \int_{e_i} \mathbf{w}^{e_j} \cdot \mathbf{t}_{e_i} = \delta_{e_i e_j}$ can be found, e.g., in Section 5.2.2, page 140, of [27].

we get by linearity

$$\sum_{j=1}^{N_{\text{dofs}}} c_j \sigma_i(w^j) = \sigma_i(z), \quad 1 \le i \le N_{\text{dofs}},$$

that is, by duality,

$$c_i = \sigma_i(z), \quad 1 \le i \le N_{\text{dofs}}.$$

So, for $W^1_{h,1}$ the coefficients are the *circulations* of $\mathbf{z}$ along the oriented edges e of the mesh. Another way to state the property of duality is by introducing a matrix of weights, a square matrix V with entries defined as

$$V_{ij} = \sigma_i(w^j), \quad 1 \le i, j \le n_{\text{dofs}} = \dim(W^1_{h,1}(v)),$$

in a tetrahedron v. When duality holds, the matrix of weights V is the identity. Duality won't be automatically granted when moving to higher order in the case of $W^1_{h,r}$, so in Section 3.6 we will propose a simple technique to restore duality.

Remark 3.3. The edge-to-node incidence matrix $\mathbf{G}$ is as a discrete analogue of the gradient operator. Indeed, if one sets, for example, $\mathbf{z} = \nabla\varphi$, where $\varphi = \sum_{n\in\mathcal{N}} \varphi_n w^n$ is an element of $W^0_{h,1}$, then $\mathbf{z} \in W^1_{h,1}$ is expressed as $\sum_{e\in\mathcal{E}} z_e \mathbf{w}^e$, with

$$z_e = \sigma_e(\mathbf{z}) = \sigma_e(\nabla\varphi) = \frac{1}{|e|}\int_e \nabla\varphi \cdot \mathbf{t}_e = \varphi(\mathbf{x}_j) - \varphi(\mathbf{x}_i) = \sigma_{n_j}(\varphi) - \sigma_{n_i}(\varphi) = \varphi_{n_j} - \varphi_{n_i},$$

where $e = \{n_i, n_j\}$ and we have used Stokes theorem. Therefore, remembering the definition of $\mathbf{G}$, the relation between the vectors of coefficients (z_e) for $\mathbf{z}$ and (φ_n) for φ, is $(z_e) = \mathbf{G}(\varphi_n)$.

3.3.1 Correspondence with Whitney forms

As pointed out in [27], the generators of FE subspace $W^p_{h,1}$ correspond to constructs in algebraic topology known as *Whitney forms* [114] (see [28] for a short presentation). The following recursive definition of Whitney p-forms of lower degree in a simplex v has been firstly stated in [23].

Definition 3.4. For $p = 0$, we set $w^n = \lambda_n$, for all 0-simplices $n \in \mathcal{N}$. For any integer $0 < p \le 3$, where 3 is the ambient dimension in Ω, the *Whitney p-form* w^s associated with the p-simplex s of a mesh $\mathcal{T}_h$ in $\bar{\Omega}$ is

$$w^s = \sum_{\sigma\in\{(p-1)-\text{simplices}\}} \boldsymbol{\partial}_{s,\sigma} \lambda_{s-\sigma} \mathrm{d}w^\sigma \tag{3.3}$$

where $\boldsymbol{\partial}_{s,\sigma}$ is the incidence matrix entry linking σ to s, w^σ is the $(p-1)$-form associated with σ, and d is the exterior derivative operator from $(p-1)$-forms to p-forms.

To recover from this definition the expression of the basis functions for $W^1_{h,1}(v)$, it is sufficient to replace the exterior derivative operator d by the gradient operator ∇. For the edge $e = \{l, m\}$, the Whitney 1-form $w^e = \sum_{n\in\mathcal{N}} \boldsymbol{\partial}_{e,n} \lambda_{e-n} \mathrm{d}w^n$ becomes $w^e = \lambda_l \mathrm{d}w^m - \lambda_m \mathrm{d}w^l$, thus the vector function $\mathbf{w}^e = \lambda_l \nabla\lambda_m - \lambda_m \nabla\lambda_l$ of (3.1).

3.4 Generators for high order curl-conforming finite elements

We recall the definition of the high order generators for $W^1_{h,r}$ presented in [97, 98], which are indeed Whitney 1-forms of higher degree ($r > 1$). Adopting a geometrical approach to define higher order Whitney forms, we have to construct a finer description of sub-simplices in the volume v.

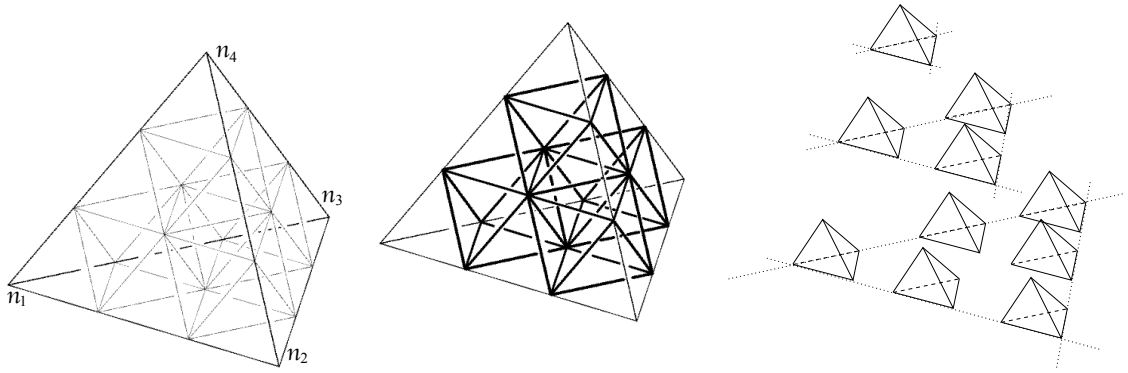


Figure 3.2 – For $d = 3$ and $r = 3$, the edges connecting the points of $T_3(v)$ (left), the holes in thick line (center), the small d-simplices in the exploded configuration (right). The small 0-simplices are nothing else than the nodes of $T_3(v)$. Starting from the nodes of $T_3(v)$, we may construct 60 small edges, 40 small faces, 10 small tetrahedra.

3.4.1 Small simplices

An analogous procedure is classically used to define higher order nodal FEs for scalar fields [32]: it goes through the introduction of the *principal lattice* of order $r \geq 1$ in the volume $v \subset \mathbb{R}^d$. It consists in the set of points

$$T_r(v) = \left\{ \mathbf{x} \in v,\, \lambda_j(\mathbf{x}) \in \left\{ 0,\ \frac{1}{r},\ \frac{2}{r},\ \ldots,\ \frac{(r-1)}{r},\ 1 \right\},\ 1 \leq j \leq d+1 \right\}.$$

Recall that the cardinality of $T_r(v)$ is equal to the dimension of $\mathbb{P}_r(v)$, the space of real-valued polynomials defined in $\mathbb{R}^d$, restricted to the volume v, of degree $\leq r$ (the proof is by recurrence on the ambient space dimension d). The functionals $\sigma_l : z \mapsto z(\mathbf{x}_l)$, with $\mathbf{x}_l \in T_r(v)$, are classically taken as dofs for functions in $\mathbb{P}_r(v)$.

Here, since we deal with fields whose physical meaning is specified by circulations along curves, we need to define dofs accordingly, even when we consider a high order approximation of those fields. Thus we have to create a sort of principal lattice of edges in v for $r > 1$. This is realized by connecting the points of $T_r(v)$ with planes parallel to the faces of v, (as in Figure 3.2, left). One thus obtains a partition of v including d-simplices homothetic to v (the so-called "small" d-simplices, visible in Figure 3.2, right) and other objects (the *"holes"*, the objects with thick boundary in Figure 3.2, center) that can only be octahedra and reversed tetrahedra when v is a tetrahedron. Any p-simplex, $0 \leq p < d$, belonging to the boundary of a small d-simplex is called *small p-simplex*. For the case $p = 1$ in which we are interested here, we consider small 1-simplices, i.e. *small edges*.

To formalize the construction defined right above, we need to introduce multi-index notations. A *multi-index* is an array $\mathbf{k} = (k_1, \ldots, k_\nu)$ of ν integers $k_i \geq 0$, and its weight k is $\sum_{i=1}^{\nu} k_i$. The set of multi-indices $\mathbf{k}$ with ν components and of weight k is denoted $\mathcal{I}(\nu, k)$. If $d = 2, 3$ is the ambient space dimension, we consider $\nu \leq d + 1$. Now, in the following definition we take $\nu = d + 1$ and $k = r - 1$.

Definition 3.5 (Small edges)**.** Let us consider the principal lattice $T_r(v)$ of order $r > 1$ in the simplex $v = \{n_1, ..., n_{d+1}\}$. The *small 1-simplex* or *small edge* $\{\mathbf{k}, e\}$, with $\mathbf{k} \in \mathcal{I}(d + 1, r - 1)$, is a 1-simplex parallel and $1/r$-homothetic to the (big) edge e of v, with vertices in $T_r(v)$. It belongs to the boundary of the small tetrahedron whose barycenter $\mathbf{g}$ has barycentric coordinates

$$\lambda_{n_j}(\mathbf{g}) = \frac{\left(\frac{1}{d+1} + k_j\right)}{k+1}, \qquad 1 \leq j \leq d+1.$$

Example 3.6. The barycenters of the small tetrahedra for a degree $r > 1$ can be used to localize the small edges. Referring to Figure 3.2, left and right, where $v = \{n_1, n_2, n_3, n_4\}$ and $r = 3$, the small simplex $\{(0, 0, 2, 0),\ e = \{n_3, n_4\}\}$ is the small edge parallel to e that belongs to the small tetrahedron in the bottom layer, corner position and touching the edge $\{n_3, n_4\}$. The barycenter $\mathbf{g}$ of this small tetrahedron verifies

$$\lambda_{n_1}(\mathbf{g}) = \frac{1}{12}, \quad \lambda_{n_2}(\mathbf{g}) = \frac{1}{12}, \quad \lambda_{n_3}(\mathbf{g}) = \frac{1}{12} + \frac{2}{3}, \quad \lambda_{n_4}(\mathbf{g}) = \frac{1}{12}.$$

The small simplex $\{(1, 1, 0, 0),\ e = \{n_3, n_4\}\}$ is the small edge parallel to e that belongs to the small tetrahedron in the bottom layer, middle position, touching the edge $\{n_1, n_2\}$. The barycenter $\mathbf{g}$ of this latter small tetrahedron verifies

$$\lambda_{n_1}(\mathbf{g}) = \frac{1}{12} + \frac{1}{3}, \quad \lambda_{n_2}(\mathbf{g}) = \frac{1}{12} + \frac{1}{3}, \quad \lambda_{n_3}(\mathbf{g}) = \frac{1}{12}, \quad \lambda_{n_4}(\mathbf{g}) = \frac{1}{12}.$$

In practice, we can think that each component of the multi-index $\mathbf{k}$ says how close the small tetrahedron is to each vertex of the big tetrahedron (the higher k_i, the closer the small tetrahedron to n_i).

3.4.2 High order generators

The complex of small edges is not created in the reality, it is drawn to help to *visualize* the high order construction. The high order generators for $W^1_{h,r}(v)$ in a tetrahedron v will be associated with the small edges identified in the principal lattice $T_r(v)$. Before defining these generators, which are Whitney 1-forms of high order, we need to introduce also the following notation for products of barycentric coordinates.

Definition 3.7. Given $\mathbf{k} \in \mathcal{I}(d+1, k)$, we set $\lambda^{\mathbf{k}} = \prod_{i=1}^{d+1} (\lambda_{n_i})^{k_i}$.

These homogeneous polynomials of degree k in barycentric coordinates are in one-to-one correspondence with polynomials of degree $\leq k$ in Cartesian coordinates. For this reason, we can say that $\mathbb{P}_k(v) = \text{span}(\lambda^{\mathbf{k}})_{\mathbf{k} \in \mathcal{I}(d+1,k)}$ on each volume v. We recall that $\dim(\mathbb{P}_k(v)) = \binom{k+d}{d}$. When revisiting dofs in the next section, it will occur that we adopt another version of Definition 3.7. Let $s = \{n_1, ..., n_{p'+1}\}$ be a p'-simplex, $0 \leq p' \leq d$, and let $\mathbf{k}_s \in \mathcal{I}(p'+1, k')$ be a multi-index of weight k'; we introduce the notation

$$\lambda_s^{\mathbf{k}_s} = \prod_{i=1}^{p'+1} (\lambda_{n_i})^{(\mathbf{k}_s)_i}. \tag{3.4}$$

For the same reason as before, we have that the space of polynomials of degree $\leq k'$ on a p'-simplex s can be spanned by the $\lambda_s^{\mathbf{k}_s}$ with $\mathbf{k}_s \in \mathcal{I}(p'+1, k')$:

$$\mathbb{P}_{k'}(s) = \text{span}(\lambda_s^{\mathbf{k}_s})_{\mathbf{k}_s \in \mathcal{I}(p'+1,k')}. \tag{3.5}$$

Now we can state the definition of the generators adopted here for $W^1_{h,r}(v)$ in a tetrahedron v, firstly introduced in [98]. This definition is rather simple since it only involves the barycentric coordinates of the simplex.

Definition 3.8 (Generators for $W^1_{h,r}(v)$)**.** *Whitney 1-forms of high order* $r = k+1$ (where $k \geq 0$) in a volume v are the

$$\lambda^{\mathbf{k}} \mathbf{w}^e,$$

for all small edges $\{\mathbf{k}, e\}$, with $\mathbf{k} \in \mathcal{I}(d+1, k)$ and e an edge of v. The $\mathbf{w}^e$ are the Whitney 1-forms of polynomial degree 1 as stated in Definition 3.4 ($p = 1$), corresponding to (3.1).

Note that these generators for $W^1_{h,r}(v)$ enjoy the same conformity properties as those for $W^1_{h,1}(v)$ since they are defined as products between $\mathbf{w}^e \in W^1_{h,1}(v)$ and the continuous function $\lambda^{\mathbf{k}}$ (product of barycentric coordinates). They are indexed on the basis of the small edges $\{\mathbf{k}, e\}$ of Definition 3.5. If one makes the list of the small edges for a given k, this immediately and explicitly yields the list of all generators of $W^1_{h,r}(v)$, with $r = k + 1$.

However, the products $\lambda^{\mathbf{k}}\mathbf{w}^e$ generate the space $W^1_{h,r}$, *but* do not actually constitute a basis as they are not all linearly independent when $r > 1$. This result is stated in [98], Proposition 3.5, which is recalled here.

Proposition 3.9. *For any face f we have*

$$\sum_{e\in\mathcal{E}} \boldsymbol{\partial}_{f,e}\, \lambda_{f-e}\, \mathbf{w}^e = \mathbf{0}, \tag{3.6}$$

Proof. Replacing w^e by its expression given in Definition 3.4 we get $\sum_e \boldsymbol{\partial}_{f,e}\, \lambda_{f-e}\, \mathbf{w}^e = \sum_{n,e} \lambda_{f-e}\lambda_{e-n}\, \boldsymbol{\partial}_{f,e}\, \boldsymbol{\partial}_{e,n}\, \mathrm{d}w^n$. This equals $\mathbf{0}$ since for a fixed n vertex of f, $\lambda_{f-e}\lambda_{e-n}$ is the same for all e in ∂f and $(\boldsymbol{\partial}_{f,e})(\boldsymbol{\partial}_{e,n}) = \mathbf{0}$ (Proposition 3.2). □

Due to (3.6), for each face there exists a combination with nonzero coefficients (± 1) of forms $\lambda^{\mathbf{k}}\mathbf{w}^e$ with certain multi-indices $\mathbf{k} \in \mathcal{I}(d + 1, 1)$ that gives zero. These are the *relations* among the generators for $r = 2$. To get the relations for $r > 2$, it is sufficient to multiply these relations by the products $\lambda^{\mathbf{k}}$ with $\mathbf{k} \in \mathcal{I}(d + 1, r - 2)$. As detailed later, the selection of generators that constitute an actual basis of $W^1_{h,r}(v)$ can be guided by the degrees of freedom introduced in the next section. See the explicit list of the generators for $d = 3$, $r = 2$ in Example 3.22.

3.5 Dofs for high order curl-conforming finite elements

We now deal with possible degrees of freedom (dofs) for fields in $W^1_{h,r}(v)$. Different sets of unisolvent dofs exist for high order elements. In [98] dofs associated with the small simplices have been analyzed: for $W^1_{h,r}(v)$ they were defined as circulations along the small edges. Here, we revisit the classical dofs, referred to as *moments*, defined by Nédélec [89], in order to obtain a new expression which results to be more friendly in terms of the considered generators.

We first recall the definition stated in [89] (Definition 4 therein). We denote by $\mathcal{E}(v)$ the set of edges of v and by $\mathcal{F}(v)$ the set of faces of v.

Definition 3.10 (Classical dofs)**.** The dofs for a vector function $\mathbf{w} \in W^1_{h,r}(v)$, for $r \geq 1$, are the functionals

$$\sigma_e : \mathbf{w} \mapsto \frac{1}{|e|}\int_e (\mathbf{w}\cdot\mathbf{t}_e)\, u, \qquad \forall\, u \in \mathbb{P}_{r-1}(e),\ \forall\, e \in \mathcal{E}(v), \tag{3.7}$$

$$\sigma_f : \mathbf{w} \mapsto \frac{1}{|f|}\int_f (\mathbf{w}\times\mathbf{n}_f)\cdot\mathbf{q}, \qquad \forall\, \mathbf{q} \in (\mathbb{P}_{r-2}(f))^2,\ \forall\, f \in \mathcal{F}(v), \tag{3.8}$$

$$\sigma_v : \mathbf{w} \mapsto \frac{1}{|v|}\int_v \mathbf{w}\cdot\mathbf{z}, \qquad \forall\, \mathbf{z} \in (\mathbb{P}_{r-3}(v))^3, \tag{3.9}$$

with $\mathbf{t}_e$ (resp. $\mathbf{n}_f$) the vector of length $|e|$ (resp. 1), tangent to e (resp. normal to f).

In Section 1.2 of [89] it is proved that dofs (3.7)-(3.9) are unisolvent. Note that in Definition 3.10 if $r < 3$, dofs given by (3.9) are not used, which is implicitly stated by the fact that it is not possible to define the elements of $\mathbb{P}_{r-3}(v)$ when $r < 3$, namely

polynomials with negative degree. For the same reason, if $r < 2$, dofs given by (3.8) and (3.9) are not used. Note that if $r = 1$, dofs given by (3.7) reduce to the circulations (3.2).

Now in the following Propositions we recast these dofs in a new more friendly form.

Proposition 3.11. *Let us consider a vector $\mathbf{w} \in W^1_{h,r}(v)$ for $r \geq 1$. Its moments on faces given by (3.8) are equivalent to*

$$\sigma_f : \mathbf{w} \mapsto \tfrac{1}{|f|} \textstyle\int_f (\mathbf{w} \cdot \mathbf{t}_{f,i})\, q, \quad \forall\, q \in \mathbb{P}_{r-2}(f),\ \forall\, f \in \mathcal{F}(v), \quad \mathbf{t}_{f,i} \text{ two independent sides of } f,\ i = 1, 2. \tag{3.10}$$

Proof. Any vector $\mathbf{q} \in (\mathbb{P}_{r-2}(f))^2$ for the face $f = \{n_i, n_j, n_l\}$ can be written as a linear combination of the two independent vectors

$$\mathbf{q}_1 = \lambda_f^{\mathbf{k}_f} (\mathbf{t}_{f,1} \times \mathbf{n}_f) \quad \text{and} \quad \mathbf{q}_2 = \lambda_f^{\mathbf{k}_f} (\mathbf{t}_{f,2} \times \mathbf{n}_f)$$

with $\mathbf{t}_{f,1} = \mathbf{x}_j - \mathbf{x}_i,\ \mathbf{t}_{f,2} = \mathbf{x}_l - \mathbf{x}_i$ two independent sides of the face f, $\mathbf{n}_f$ its unit normal, $\mathbf{k}_f \in \mathcal{I}(3, r-2)$ and $\lambda_f^{\mathbf{k}_f}$ defined in (3.4), recalling (3.5). We use the vector identity

$$(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c})$$

to see that

$$\begin{aligned}(\mathbf{w} \times \mathbf{n}_f) \cdot \mathbf{q}_1 &= \lambda_f^{\mathbf{k}_f} (\mathbf{w} \times \mathbf{n}_f) \cdot (\mathbf{t}_{f,1} \times \mathbf{n}_f) \\ &= \lambda_f^{\mathbf{k}_f} \left[(\mathbf{w} \cdot \mathbf{t}_{f,1})(\mathbf{n}_f \cdot \mathbf{n}_f) - (\mathbf{w} \cdot \mathbf{n}_f)(\mathbf{n}_f \cdot \mathbf{t}_{f,1}) \right] \\ &= \lambda_f^{\mathbf{k}_f} \left[(\mathbf{w} \cdot \mathbf{t}_{f,1})(1) - (\mathbf{w} \cdot \mathbf{n}_f)(0) \right] = \lambda_f^{\mathbf{k}_f} (\mathbf{w} \cdot \mathbf{t}_{f,1})\end{aligned}$$

and similarly with $\mathbf{q}_2$. The proof ends as $\mathbb{P}_{r-2}(f) = \operatorname{span}(\lambda_f^{\mathbf{k}_f})_{\mathbf{k} \in \mathcal{I}(3, r-2)}$. □

Proposition 3.12. *Let us consider a vector $\mathbf{w} \in W^1_{h,r}(v)$ for $r \geq 1$. Its moments in the volume given by (3.9) are equivalent to*

$$\sigma_v : \mathbf{w} \mapsto \tfrac{1}{|v|} \textstyle\int_v (\mathbf{w} \cdot \mathbf{t}_{v,i})\, q, \quad \forall\, q \in \mathbb{P}_{r-3}(v), \quad \mathbf{t}_{v,i} \text{ three independent sides of } v,\ i = 1, 2, 3. \tag{3.11}$$

Proof. Any vector $\mathbf{z} \in (\mathbb{P}_{r-3}(v))^3$ for the volume $v = \{n_1, n_2, n_3, n_4\}$ can be written as a linear combination of the three independent vectors

$$\mathbf{q}_1 = \lambda^{\mathbf{k}}\, \mathbf{t}_{v,1}, \qquad \mathbf{q}_2 = \lambda^{\mathbf{k}}\, \mathbf{t}_{v,2}, \quad \text{and} \quad \mathbf{q}_3 = \lambda^{\mathbf{k}}\, \mathbf{t}_{v,3},$$

where $\mathbf{t}_{v,\ell} = \mathbf{x}_{\ell+1} - \mathbf{x}_1$, for $\ell = 1, 2, 3$, and $\mathbf{k} \in \mathcal{I}(3+1, r-3)$. In the definition of the $\mathbf{q}_i$, the part $\lambda^{\mathbf{k}}$ allows to specify the polynomial degree $r-3$ and the polynomial variables (barycentric coordinates with respect to vertices of v), whereas the (constant) $\mathbf{t}_{v,\ell}$ determine the vector nature. Recall that $\mathbb{P}_{r-3}(v) = \operatorname{span}(\lambda^{\mathbf{k}})_{\mathbf{k} \in \mathcal{I}(3+1, r-3)}$. □

Summing up the content of the two Propositions to rewrite σ_f and σ_v, we state the new definition of dofs.

Definition 3.13 (Revisited dofs)**.** The dofs for a vector function $\mathbf{w} \in W^1_{h,r}(v)$, for $r \geq 1$, are the functionals

$$\sigma_e \colon \mathbf{w} \mapsto \frac{1}{|e|} \int_e (\mathbf{w} \cdot \mathbf{t}_e)\, q, \qquad \forall\, q \in \mathbb{P}_{r-1}(e),\ \forall\, e \in \mathcal{E}(v), \tag{3.12}$$

$$\sigma_f \colon \mathbf{w} \mapsto \frac{1}{|f|} \int_f (\mathbf{w} \cdot \mathbf{t}_{f,i})\, q, \qquad \forall\, q \in \mathbb{P}_{r-2}(f),\ \forall\, f \in \mathcal{F}(v), \tag{3.13}$$

$$\mathbf{t}_{f,i} \text{ two independent sides of } f,\ i = 1, 2,$$

$$\sigma_v \colon \mathbf{w} \mapsto \frac{1}{|v|} \int_v (\mathbf{w} \cdot \mathbf{t}_{v,i})\, q, \qquad \forall\, q \in \mathbb{P}_{r-3}(v), \tag{3.14}$$

$$\mathbf{t}_{v,i} \text{ three independent sides of } v,\ i = 1, 2, 3,$$

where the norm of the vectors $\mathbf{t}_e, \mathbf{t}_{f,i}, \mathbf{t}_{v,i}$ is the length of the associated edge. We say that e, f, v are the *supports* of the dofs $\sigma_e, \sigma_f, \sigma_v$.

Remark 3.14. To make the computation of dofs easier, a *convenient choice for the polynomials* q spanning the polynomial spaces over (sub)simplices e, f, v that appear in Definition 3.13 is given by suitable products of the barycentric coordinates associated with the vertices of the considered (sub)simplex. Remember indeed (3.5).

3.5.1 Selection of linearly independent generators

The *classification* of dofs into edge-type, face-type, volume-type dofs can be done also for generators: volume-type generators contain (inside $\lambda^{\mathbf{k}}$ or $\mathbf{w}^e$) the barycentric coordinates w.r.t. all the nodes of a tetrahedron v, face-type generators contain the ones w.r.t. all and only the nodes of a face f, edge-type generators contain the ones w.r.t. only the nodes of an edge e. Note that face-type (resp. volume-type) generators appear for $r > 1$ (resp. $r > 2$) (and the same happens for face-type and volume-type dofs). See the explicit list of generators and dofs for the case $d = 3, r = 2$ in Example 3.22. It turns out that dofs σ_e are 0 on face-type and volume-type generators, and dofs σ_f are 0 on volume-type generators.

As mentioned at the end of the previous Section, for the high order case ($r > 1$) the fields $\lambda^{\mathbf{k}}\mathbf{w}^e$ in Definition 3.8 are generators for $W^1_{h,r}(v)$, but some of them are linearly dependent. Note that all the relations are among face-type or volume-type generators. The *selection of generators* that constitute an actual basis of $W^1_{h,r}(v)$ can be guided by the dofs in Definition 3.13. More precisely, as face-type (resp. volume-type) generators keep the ones associated with the two (resp. three) edges e *chosen as* the two sides $\mathbf{t}_{f,1}, \mathbf{t}_{f,2}$ (resp. three sides $\mathbf{t}_{v,1}, \mathbf{t}_{v,2}, \mathbf{t}_{v,3}$) of face-type dofs (3.13) (resp. volume-type dofs (3.14)). A convenient choice of sides is described in Section 4.2 and is the one adopted in Example 3.22. One can check that the total number of dofs $\sigma_e, \sigma_f, \sigma_v$ in a simplex v is equal to $\dim(W^1_{h,r}(v)) = (r + d)(r + d - 1) \cdots (r + 2)r/(d - 1)!$.

3.6 Restoring duality between generators and dofs

The considered basis functions are not in *duality* with the dofs in Definition 3.13 when $r > 1$. This means that, after a suitable renumbering of dofs, the matrix V with entries the weights

$$V_{ij} = \sigma_i(\mathbf{w}_j), \quad 1 \leq i, j \leq n_{\text{dofs}} = \dim(W^1_{h,r}(v))$$

is not the identity matrix for $r > 1$, that is $\sigma_i(\mathbf{w}_j) \neq \delta_{ij}$. Here the $\{\mathbf{w}_j\}$ are a linearly independent subset of the generators given in Definition 3.8. In the following we propose

a technique to restore duality, considering new basis functions $\tilde{\mathbf{w}}_j$ built as suitable *linear combinations* of the $\mathbf{w}_j$, such that $\sigma_i(\tilde{\mathbf{w}}_j) = \delta_{ij}$.

This technique is related with the well known *polynomial fitting* of a scalar function f at $n = r+1$ points x_i of a real interval I. Considering the canonical basis $\{x^{j-1}\}_{j=1,...,n}$ of $\mathbb{P}_r(I)$, it consists in finding a polynomial $I_r f(x) = \sum_{j=1}^n a_j x^{j-1}$ such that $I_r f(x_i) = f(x_i)$ for all $i = 1, ..., n$. The coefficients a_j results to be the entries of $\mathbf{a}$, solution of the linear system $V\mathbf{a} = \mathbf{f}$ where V is the *Vandermonde matrix* with entries $V_{ij} = x_i^{j-1}$, and $\mathbf{f}_i = f(x_i)$. More generally, considering a basis $\{\psi_j\}$ for $\mathbb{P}_r(I)$ (different from the canonical one) and dofs $\sigma_i \colon \mathbb{P}_r(I) \to \mathbb{R}$ (e.g. $\sigma_i(f) = f(x_i)$), we have the following result.

Proposition 3.15. *Let $\{\psi_j\}_j$ be a basis for $\mathbb{P}_r(I)$ and $\{\sigma_i\}_i$ suitable dofs for functions in $\mathbb{P}_r(I)$. Writing*

$$I_r f(x) = \sum_{j=1}^{n} u_j \psi_j(x),$$

the vector $\mathbf{u}$ such that $\sigma_i(I_r f) = \sigma_i(f)$, for all $i = 1, ..., n$, is solution of the algebraic system

$$V\mathbf{u} = \mathbf{f},$$

where V is a generalized Vandermonde matrix *with entries $V_{ij} = \sigma_i(\psi_j)$ and $\mathbf{f}$ has (known) components $\mathbf{f}_i = \sigma_i(f)$.*

Proof. Let us apply σ_i on both sides of the equality $I_r f = \sum_{j=1}^n u_j \psi_j$. Since σ_i is linear, we obtain

$$\sigma_i(I_r f) = \sum_{j=1}^{n} u_j \sigma_i(\psi_j),$$

which gives, using $\sigma_i(I_r f) = \sigma_i(f) = \mathbf{f}_i$, and $V_{ij} = \sigma_i(\psi_j)$,

$$\sum_{j=1}^{n} u_j V_{ij} = \mathbf{f}_i, \quad \text{for all } i = 1, ..., n,$$

that is $V\mathbf{u} = \mathbf{f}$ in matrix form. □

In particular, if we consider *cardinal (dual) functions* $\{\phi_j\}$ in $\mathbb{P}_r(I)$ defined by $\sigma_i(\phi_j) = \delta_{ij}$, we have $\mathbf{u} = \mathbf{f}$, thus

$$I_r f(x) = \sum_{j=1}^{n} \sigma_j(f)\, \phi_j(x).$$

The ϕ_j have to be determined as linear combinations of chosen basis functions ψ_j. The same reasoning applies for vector field interpolation, replacing the ψ_j with $\mathbf{w}_j$, and ϕ_j with $\tilde{\mathbf{w}}_j$: thus the interpolation operator $\Pi^1_{h,r}$ is defined by

$$\Pi^1_{h,r} \colon Y^1 \subset H(\mathrm{curl}, v) \to W^1_{h,r}(v), \quad \mathbf{u} \mapsto \mathbf{u}_h = \sum_{i=1}^{n_{\mathrm{dofs}}} c_i \tilde{\mathbf{w}}_i, \quad \text{with } c_i := \sigma_i(\mathbf{u}), \tag{3.15}$$

provided $\sigma_i(\tilde{\mathbf{w}}_k) = \delta_{ik}$ (Y^1 is a suitable subspace of $H(\mathrm{curl}, v)$, see [65]).

Now, to determine the $\tilde{\mathbf{w}}_k$, written as linear combinations $\tilde{\mathbf{w}}_k = \sum_{j=1}^{n_{\mathrm{dofs}}} c_j^k \mathbf{w}_j$, we need to find coefficients c_j^k such that $\sigma_i(\tilde{\mathbf{w}}_k) = \delta_{ik}$. For a given k, the coefficients c_j^k turn out to be the entries of the k-th *column* of V^{-1}, the inverse of the generalized Vandermonde matrix V with entries $V_{ij} = \sigma_i(\mathbf{w}_j)$. This is proved in the following Proposition.

Proposition 3.16. *Let $\{\mathbf{w}_j\}_j$ be a basis for $W^1_{h,r}(v)$ and $\{\sigma_i\}$ suitable dofs for functions in $W^1_{h,r}(v)$, as the ones in Definition 3.13. The vector $\mathbf{c}^k = (c^k_1, c^k_2, ..., c^k_{n_{dofs}})^\top$, one per each function $\tilde{\mathbf{w}}_k$, such that*

$$\sigma_i(\tilde{\mathbf{w}}_k) = \delta_{ik}, \qquad \tilde{\mathbf{w}}_k(x) = \sum_{j=1}^{n_{dofs}} c^k_j \mathbf{w}_j(x),$$

is solution of the algebraic system

$$V\,\mathbf{c}^k = \mathbf{e}^k, \tag{3.16}$$

where V is the generalized Vandermonde matrix *with entries $V_{ij} = \sigma_i(\mathbf{w}_j)$ and $\mathbf{e}^k$ is the k-th column of the n_{dofs}-identity matrix (i.e., $(\mathbf{e}^k)_i = \delta_{ik}$). Thus $\mathbf{c}^k = V^{-1}\mathbf{e}^k$, which determines $\mathbf{c}^k$ as the k-th column of V^{-1}.*

Proof. Let us apply σ_i on both sides of the equality $\tilde{\mathbf{w}}_k = \sum_{j=1}^{n_{\text{dofs}}} c^k_j \mathbf{w}_j$. Using the linearity of σ_i and the duality property of $\tilde{\mathbf{w}}_k$ (that is $\sigma_i(\tilde{\mathbf{w}}_k) = \delta_{ik}$), we obtain

$$\sum_{j=1}^{n_{\text{dofs}}} c^k_j \sigma_i(\mathbf{w}_j) = \delta_{ik} \qquad \text{i.e.} \qquad \sum_{j=1}^{n_{\text{dofs}}} V_{ij} c^k_j = \delta_{ik},$$

which gives, fixing k and varying i, the relation $V\mathbf{c}^k = \mathbf{e}^k$. □

Remark 3.17. Unisolvence of the finite element is equivalent to the unique invertibility of the linear systems (3.16).

Remark 3.18. If M is the local mass matrix on a tetrahedron v for the basis functions $\mathbf{w}_j$, i.e. $M_{ij} = \int_v \mathbf{w}_i \cdot \mathbf{w}_j$, then the local mass matrix $\tilde{M}$ for the basis functions $\tilde{\mathbf{w}}_j$ is $\tilde{M} = V^{-T} M V$. Indeed, calling $B = V^{-1}$, for given i, j, since $\tilde{\mathbf{w}}_i = \sum_{k=1}^{n_{\text{dofs}}} B_{ki}\mathbf{w}_k$ and $\tilde{\mathbf{w}}_j = \sum_{\ell=1}^{n_{\text{dofs}}} B_{\ell j}\mathbf{w}_\ell$, we have

$$\tilde{M}_{ij} = \int_v \tilde{\mathbf{w}}_i \cdot \tilde{\mathbf{w}}_j = \sum_{k=1}^{n_{\text{dofs}}}\sum_{\ell=1}^{n_{\text{dofs}}} B_{ki}B_{\ell j} \int_v \mathbf{w}_k \cdot \mathbf{w}_\ell = \sum_{k=1}^{n_{\text{dofs}}}\sum_{\ell=1}^{n_{\text{dofs}}} (B^T)_{ik} M_{k\ell} B_{\ell j} = (B^T M B)_{ij}.$$

3.6.1 Properties of the generalized Vandermonde matrix

Some nice properties characterize the square matrix V of size n_{dofs} (which depends on the values of r and d).

Property 3.19. *The entries of V can be calculated explicitly by a combinatorial formula and do not depend on the metrics of the tetrahedron v for which they are computed.*

Indeed, first of all, note that dofs in Definition 3.13 are conveniently normalized. Moreover, the $\sigma_i(\mathbf{w}_j)$ are integrals of two addends of the type $\lambda^{\mathbf{k}'}\nabla\lambda_{n_i} \cdot \mathbf{t}_e$ (here $\lambda^{\mathbf{k}'}$ gathers the products of barycentric coordinates appearing in the basis functions and in q, and $\mathbf{t}_e$ stands also for $\mathbf{t}_{f,i}, \mathbf{t}_{v,i}$). Now, we have $\nabla\lambda_{n_i} \cdot \mathbf{t}_e = -1$ if n_i is the first node of e, $+1$ if it is its second node, 0 if it isn't a node of e. So, in the end, only terms of the type $\lambda^{\mathbf{k}'}$ survive in the integral and the value of $\sigma_i(\mathbf{w}_j)$ can be calculated using the well known 'magic formula' (see for instance [97] for a proof): if s is a p'-simplex,

$$\frac{1}{|s|}\int_s \prod_{i=1}^{p'+1} (\lambda_{n_i})^{k_i} = \frac{p'!\Big(\prod_{i=1}^{p'+1} k_i!\Big)}{\Big(p' + \sum_{i=1}^{p'+1} k_i\Big)!}.$$

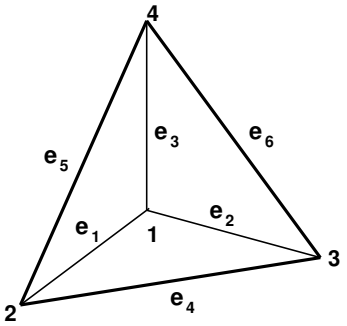


Figure 3.3 – For the tetrahedron in the figure, the edges are $e_1 = \{1,2\}$, $e_2 = \{1,3\}$, $e_3 = \{1,4\}$, $e_4 = \{2,3\}$, $e_5 = \{2,4\}$, $e_6 = \{3,4\}$, the faces are $f_1 = \{2,3,4\}$, $f_2 = \{1,3,4\}$, $f_3 = \{1,2,4\}$, $f_4 = \{1,2,3\}$ (note that the face f_i is the one opposite the node i).

This value is clearly independent of the metrics of s. The independence from the metrics of the p'-face s of v for which they are calculated means that the entries of V can be computed *once* on a generic volume v and are valid in any other volume v' different from v, up to a suitable orientation of the edges and choice of independent sides in v' (see Section 4.2). For a quicker but still precise calculation, high order quadrature formula can also be used to compute $\sigma_i(\mathbf{w}_j)$.

Property 3.20. *The matrix V, and hence its inverse V^{-1}, are block lower triangular.*

The matrix V can be written in a block lower triangular form, if dofs, indexing the rows i, and generators, indexing the columns j, are suitably numbered. On the one hand, dofs can be ordered block-wisely, depending on the dimension p' of their support (domain of integration), from the lowest allowed ($p' = p = 1$) to the highest one (which depends on the degree r). On the other hand, as already mentioned, also the generators can be classified into edge-type, face-type, volume-type generators: edge-type generators contain (inside $\lambda^{\mathbf{k}}$ or $\mathbf{w}^e$) the barycentric coordinates w.r.t. only the nodes of an edge e, face-type generators contain the ones w.r.t. all and only the nodes of a face f, volume-type generators contain the ones w.r.t. all the nodes of a tetrahedron v. Face-type (resp. volume-type) generators appear for $r > 1$ (resp. $r > 2$) and the same happens for face-type and volume-type dofs. It turns out that dofs σ_e are 0 on face-type and volume-type generators, and dofs σ_f are 0 on volume-type generators. If we list generators and dofs in the order dictated by increasing the dimension of the support of dofs, we may use a unique array of integers to list (thus to order) both dofs and generators. Following this order, the matrix V with entries $V_{ij} = \sigma_i(\mathbf{w}_j)$ is block triangular.

Property 3.21. *The entries of the matrix V^{-1} are* integer *numbers.*

This is related to the meaning of the entries of V^{-1}, to the considered generators, relying on linear combinations of products $\lambda^{\mathbf{k}}\nabla\lambda_i$, with $\mathbf{k}$ a multi-index and λ_i a barycentric coordinate, and to the considered dofs, integrals over *entire* simplices of dimension $0 \leq p' \leq d$. With dofs on small-simplices, the entries of V^{-1} would not be integers, indeed small-simplices are *portions* of mesh (big) simplices.

3.7 Illustration of the notions with an example

In the following example we illustrate the notions introduced in this Chapter for the case $d = 3$, $r = 2$.

Example 3.22 (Generators, dofs, 'dualizing' matrix for $d = 3$, $r = 2$)**.** Recalling Definition 3.8 of generators, for $r = 2$ the product of barycentric coordinates $\lambda^{\mathbf{k}}$ reduces to one barycentric coordinate because each multi-index $\mathbf{k} \in \mathcal{I}(4, 1)$ has just one nonzero component, equal to 1. If the edges and the faces of a tetrahedron are numbered as in Figure 3.3, the basis functions are

$$\mathbf{w}_j = \lambda_{n_{r_j}} \mathbf{w}^{e_{s_j}}, \quad 1 \le j \le 20,$$

where the 12 edge-type basis functions have

$$(r_j)_{j=1}^{12} = (1,2,\ 1,3,\ 1,4,\ 2,3,\ 2,4,\ 3,4) \ \text{ and } \ (s_j)_{j=1}^{12} = (1,1,\ 2,2,\ 3,3,\ 4,4,\ 5,5,\ 6,6),$$

and the 8 face-type basis functions have

$$(r_j)_{j=13}^{20} = (4,3,\ 4,3,\ 4,2,\ 3,2) \ \text{ and } \ (s_j)_{j=13}^{20} = (4,5,\ 2,3,\ 1,3,\ 1,2).$$

Note that in order to get a basis, i.e. a set of linearly independent generators, we have chosen to eliminate the (face-type) generators $\mathbf{w}_{21} = \lambda_{n_2}\mathbf{w}^{e_6}$, $\mathbf{w}_{22} = \lambda_{n_1}\mathbf{w}^{e_6}$, $\mathbf{w}_{23} = \lambda_{n_1}\mathbf{w}^{e_5}$, $\mathbf{w}_{24} = \lambda_{n_1}\mathbf{w}^{e_4}$. The corresponding edge-type dofs are:

$$\sigma_1 \colon \mathbf{w} \mapsto \frac{1}{|e_1|}\int_{e_1} (\mathbf{w}\cdot\mathbf{t}_{e_1})\,\lambda_{n_1}, \quad \sigma_2 \colon \mathbf{w} \mapsto \frac{1}{|e_1|}\int_{e_1} (\mathbf{w}\cdot\mathbf{t}_{e_1})\,\lambda_{n_2}, \ \ldots$$
$$\sigma_{11} \colon \mathbf{w} \mapsto \frac{1}{|e_6|}\int_{e_6} (\mathbf{w}\cdot\mathbf{t}_{e_6})\,\lambda_{n_3}, \quad \sigma_{12} \colon \mathbf{w} \mapsto \frac{1}{|e_6|}\int_{e_6} (\mathbf{w}\cdot\mathbf{t}_{e_6})\,\lambda_{n_4},$$

and the face-type dofs are:

$$\sigma_{13} \colon \mathbf{w} \mapsto \frac{1}{|f_1|}\int_{f_1} (\mathbf{w}\cdot\mathbf{t}_{e_4}), \quad \sigma_{14} \colon \mathbf{w} \mapsto \frac{1}{|f_1|}\int_{f_1} (\mathbf{w}\cdot\mathbf{t}_{e_5}), \ \ldots$$
$$\sigma_{19} \colon \mathbf{w} \mapsto \frac{1}{|f_4|}\int_{f_4} (\mathbf{w}\cdot\mathbf{t}_{e_1}), \quad \sigma_{20} \colon \mathbf{w} \mapsto \frac{1}{|f_4|}\int_{f_4} (\mathbf{w}\cdot\mathbf{t}_{e_2}).$$

For this ordering and choice of generators and dofs, the inverse of the generalized Vandermonde matrix, whose columns give the coefficients to restore duality between generators and dofs as in Proposition 3.16, is:

$$V^{-1} = \begin{bmatrix}
4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -4 & -2 & 2 & -2 & 2 & 4 & 8 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & -4 & -2 & -4 & -2 & -4 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -4 & -2 & 2 & -2 & 0 & 0 & 0 & 0 & 2 & 4 & 0 & 0 & 8 & -4 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & -2 & -4 & -2 & 0 & 0 & 0 & 0 & -4 & -2 & 0 & 0 & -4 & 8 & 0 & 0 & 0 & 0 \\
-4 & -2 & 0 & 0 & 2 & -2 & 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & -4 & 0 & 0 \\
2 & -2 & 0 & 0 & -4 & -2 & 0 & 0 & -4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 8 & 0 & 0 \\
-4 & -2 & 2 & -2 & 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & -4 \\
2 & -2 & -4 & -2 & 0 & 0 & -4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 8
\end{bmatrix}.$$

Number of dofs	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
h_1	17	58	123	212	325
h_2	33	114	243	420	645
h_3	65	226	483	836	1285
h_4	129	450	963	1668	2565
h_5	450	1668	3654	6408	9930

Table 3.1 – Total number of dofs for the chosen values of h and $r = k + 1$.

3.8 Convergence order

To complete the presentation, we report the preliminary numerical study of the proceedings paper [13] about the order of (h- and r-) convergence of the high order edge finite element method. This was joint work with Victorita Dolean, Elena Gaburro and Francesca Rapetti and the routines were coded in Matlab. Note that in [13] we did not consider basis functions in duality with the degrees of freedom, which were those of [98] associated with small edges and not the revisited moments studied here.

The test case is the two-dimensional waveguide problem described in Paragraph 2.2.4, for which the function $\mathbf{E}_{\text{ex}} = (0, e^{-\mathbf{i}\gamma x})$ is the exact solution, with suitable data in the impedance boundary conditions. We take a two-dimensional waveguide with $b = 0.00127\,\text{m}$, $c = 0.0502\,\text{m}$ and physical parameters $\varepsilon = \varepsilon_0 = 8.85 \cdot 10^{-12}\,\text{F m}^{-1}$, $\mu = \mu_0 = 1.26 \cdot 10^{-6}\,\text{H m}^{-1}$ and $\sigma = 0\,\text{S m}^{-1}$. We consider three high angular frequencies $\omega_1 = 75\,\text{Ghz}$, $\omega_2 = 95\,\text{GHz}$ and $\omega_3 = 110\,\text{GHz}$, and for each frequency we take $k = r - 1 = 0, 1, 2, 3, 4$ and five discretization triangle diameters $h_1 = 1.2614 \cdot 10^{-2}$, $h_2 = 6.4022 \cdot 10^{-3}$, $h_3 = 3.3848 \cdot 10^{-3}$, $h_4 = 2.0184 \cdot 10^{-3}$ and $h_5 = 1.0092 \cdot 10^{-3}$; these diameters have been obtained by doubling the discretization points over the long side of the rectangle. The corresponding total number of dofs is reported in Table 3.1.

To analyze the numerical error on the real part of the numerical solution $\mathbf{E}_h$ with respect to the exact solution $\mathbf{E}$, we take the maximum over all the triangles of the modulus of the difference between $\mathbf{E}$ and $\mathbf{E}_h$, considering their values at the triangle barycentres. In Table 3.2 we report the numerical errors for ω_1, ω_2 and ω_3 for the different values of h and k. Looking at the bold numbers along the diagonals we can see that to obtain an error of the same order of magnitude we can take a coarser mesh if we use higher order elements. We notice also that with the *same total number of dofs* we get a *remarkably smaller error* using higher order elements (see the boxed entries in Table 3.1 and Table 3.2).

In Figure 3.4a we show the log-log plot of the error for the considered choices of k: the convergence to the exact solution is of algebraic type and achieved with an order of accuracy equal to $r = k + 1$ with respect to h. In Figure 3.4b we show as well the semi-log plot of the error for the considered choices of h: a super-algebraic convergence is achieved with respect to r. These convergence orders are in agreement with those found in [97] (in which the tested boundary value problem was slightly different).

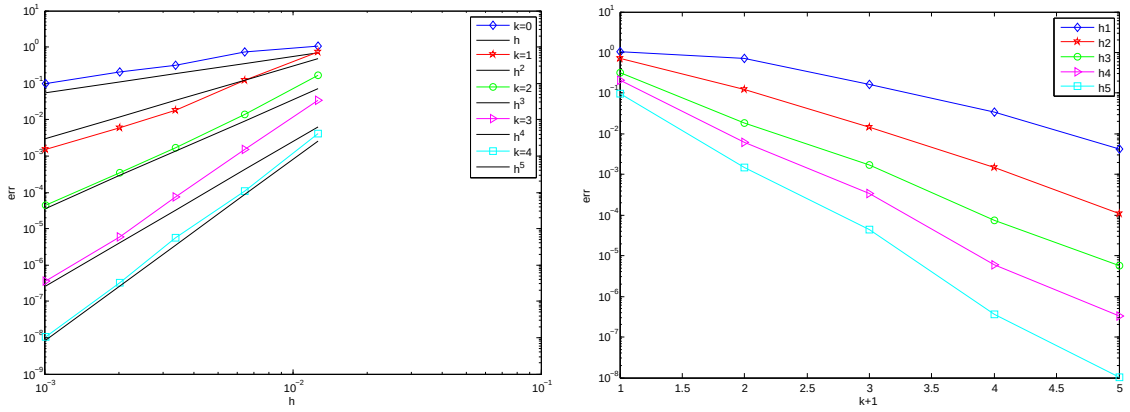
The superiority, in terms of accuracy and running time, of the high order finite elements presented in this Chapter over the lowest order ones will be illustrated in Section 6.3.2 for a large scale three-dimensional problem, arising from the application described in Chapter 6. The implementation of these finite elements is examined in the next Chapter.

ω_1	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
h_1		$3.31 \cdot 10^{-1}$	$2.84 \cdot 10^{-2}$	$5.50 \cdot 10^{-3}$	$\mathbf{4.05 \cdot 10^{-4}}$
h_2	$4.67 \cdot 10^{-1}$	$3.45 \cdot 10^{-2}$	$3.10 \cdot 10^{-3}$	$\mathbf{2.74 \cdot 10^{-4}}$	$1.10 \cdot 10^{-5}$
h_3	$2.07 \cdot 10^{-1}$	$8.10 \cdot 10^{-3}$	$\mathbf{5.13 \cdot 10^{-4}}$	$1.53 \cdot 10^{-5}$	$7.81 \cdot 10^{-7}$
h_4	$1.30 \cdot 10^{-1}$	$\mathbf{2.80 \cdot 10^{-3}}$	$1.09 \cdot 10^{-4}$	$1.27 \cdot 10^{-6}$	$4.80 \cdot 10^{-8}$
h_5	$6.21 \cdot 10^{-2}$	$7.14 \cdot 10^{-4}$	$1.37 \cdot 10^{-5}$	$7.94 \cdot 10^{-8}$	$1.52 \cdot 10^{-9}$

ω_2	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
h_1		$6.21 \cdot 10^{-1}$	$7.99 \cdot 10^{-2}$	$1.78 \cdot 10^{-2}$	$\mathbf{1.70 \cdot 10^{-3}}$
h_2	$6.23 \cdot 10^{-1}$	$7.24 \cdot 10^{-2}$	$7.90 \cdot 10^{-3}$	$\mathbf{7.70 \cdot 10^{-4}}$	$5.06 \cdot 10^{-5}$
h_3	$2.66 \cdot 10^{-1}$	$1.33 \cdot 10^{-2}$	$\mathbf{1.10 \cdot 10^{-3}}$	$4.04 \cdot 10^{-5}$	$2.74 \cdot 10^{-6}$
h_4	$1.75 \cdot 10^{-1}$	$\mathbf{4.50 \cdot 10^{-3}}$	$2.23 \cdot 10^{-4}$	$3.27 \cdot 10^{-6}$	$1.59 \cdot 10^{-7}$
h_5	$8.10 \cdot 10^{-2}$	$1.20 \cdot 10^{-3}$	$2.78 \cdot 10^{-5}$	$2.04 \cdot 10^{-7}$	$4.98 \cdot 10^{-9}$

ω_3	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$
h_1		$7.19 \cdot 10^{-1}$	$1.67 \cdot 10^{-1}$	$3.47 \cdot 10^{-2}$	$\mathbf{4.20 \cdot 10^{-3}}$
h_2	$7.28 \cdot 10^{-1}$	$1.23 \cdot 10^{-1}$	$1.42 \cdot 10^{-2}$	$\mathbf{1.50 \cdot 10^{-3}}$	$1.07 \cdot 10^{-4}$
h_3	$3.23 \cdot 10^{-1}$	$1.86 \cdot 10^{-2}$	$\mathbf{1.70 \cdot 10^{-3}}$	$7.42 \cdot 10^{-5}$	$5.71 \cdot 10^{-6}$
h_4	$2.07 \cdot 10^{-1}$	$\mathbf{6.10 \cdot 10^{-3}}$	$3.46 \cdot 10^{-4}$	$5.86 \cdot 10^{-6}$	$3.31 \cdot 10^{-7}$
h_5	$9.61 \cdot 10^{-2}$	$1.50 \cdot 10^{-3}$	$4.32 \cdot 10^{-5}$	$3.67 \cdot 10^{-7}$	$1.04 \cdot 10^{-8}$

Table 3.2 – Numerical errors for the chosen values of h and $r = k + 1$.



(a) h-convergence of the numerical error. (b) k-convergence of the numerical error.

Figure 3.4 – Convergence orders for $\omega = \omega_3$.

Chapter 4

Implementation of high order curl-conforming finite elements

This Chapter is the result of a collaboration with Victorita Dolean, Frédéric Hecht and Francesca Rapetti. The corresponding submitted preprint [18] is available on arXiv and HAL (<hal-01298938>).

Contents

4.1 Addition of new finite elements to FreeFem++

The implementation of high order curl-conforming finite elements (also called edge finite elements) is quite delicate, especially in the three-dimensional case. Here, we explicitly describe an implementation strategy, which has been embedded in FreeFem++ (`http://www.freefem.org/ff++/`). FreeFem++ is an open source domain specific language (DSL) specialized in solving boundary value problems by using variational methods, and it is based on a natural transcription of the weak formulation of the considered problem [67].

The user can *add* new finite elements to FreeFem++ by writing a C++ plugin that defines various ingredients, among which the principal ones are:

- the basis functions (and their derivatives) in a simplex,
- an interpolation operator, which requires dofs in *duality* with the basis functions.

Indeed, in FreeFem++ the basis functions (and in some cases the coefficients of the interpolation operator) are constructed *locally*, i.e. in each simplex T (triangle in $2d$, tetrahedron in $3d$) of the mesh $\mathcal{T}_h$, without the need of a transformation from the reference simplex. Note that the chosen definition of high order generators (Definition 3.8), which involves only the barycentric coordinates of the simplex, fits perfectly this local construction feature of FreeFem++. Nevertheless, the local construction should be done in such a way that the contributions coming from simplices sharing edges or faces can be then assembled

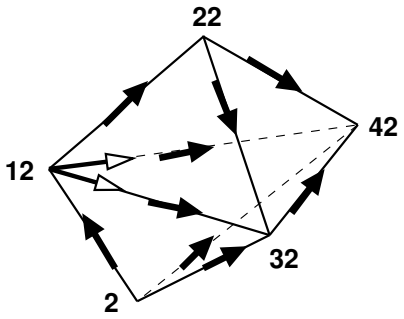


Figure 4.1 – Orientation of edges ('filled' arrows) and choice of 2 edges ('empty' arrows) of the face shared by two adjacent tetrahedra using the numbering of mesh nodes.

properly inside the *global* matrix of the FE discretization. Moreover, for the definition of the interpolation operator, in the high order edge elements case we need the generalized Vandermonde matrix V introduced in Section 3.6 to restore duality between dofs and basis functions. Here, we carefully address the problem of applying the same Vandermonde matrix to possibly differently oriented simplices (triangles, tetrahedra) of the whole mesh, in order to be able to use in numerical experiments the concepts presented for just one simplex in the previous Chapter. The strategy developed to deal with these issues for the high order edge elements is described in Section 4.2. The definition and the implementation of the interpolation operator are detailed in Section 4.3.

We added in this way the edge elements in 3d of degree $2, 3$ presented before. The code of the C++ plugin `Element_Mixte3d.cpp`, in which they are defined, is visible if FreeFem++ sources are downloaded (from `http://www.freefem.org/ff++/`) and is thus found in the folder `examples++-load`. Section 4.4 shows how to use these new finite elements in a FreeFem++ script.

4.2 Local implementation strategy for the global assembling

The implementation of edge finite elements is quite delicate. Indeed, basis functions and dofs are associated with the *oriented* edges of mesh simplices: note that the low order $\mathbf{w}^e$ and the high order $\lambda^{\mathbf{k}}\mathbf{w}^e$ generators change sign if the orientation of the edge e is reversed. Moreover, recall that for $r > 1$, in order to get a set of linearly independent generators, we also have to *choose* 2 edges for each face f. Here we wish to construct basis functions *locally*, i.e. in each simplex T of $\mathcal{T}_h$, in such a way that the contributions coming from simplices sharing edges or faces could be assembled properly inside the *global* matrix of the FE discretization. For this purpose, it is essential to assign the *same* orientation to edges shared by simplices and to choose the *same* 2 edges for faces shared by adjacent tetrahedra. We have this need also to construct dofs giving the coefficients for the interpolation operator.

This need is satisfied using the *global numbers* of the mesh nodes (see Figure 4.1). More precisely, to assign an orientation to the edges e of the basis functions and to the vectors $\mathbf{t}_e, \mathbf{t}_{f,i}, i = 1, 2, \mathbf{t}_{T,i}, i = 1, 2, 3$ of the dofs, we go from the node with the smallest global number to the node with the biggest global number. Similarly, to choose 2 edges per face for the face-type basis functions and dofs, we take the 2 edges going out from the node with the smallest global number in the face (and the 1st edge goes to the node with the 2nd smallest global number, the 2nd edge goes to the node with the biggest global number in the face).

Moreover, when we want basis functions $\tilde{\mathbf{w}}_j$ in duality with the dofs, a *second need*

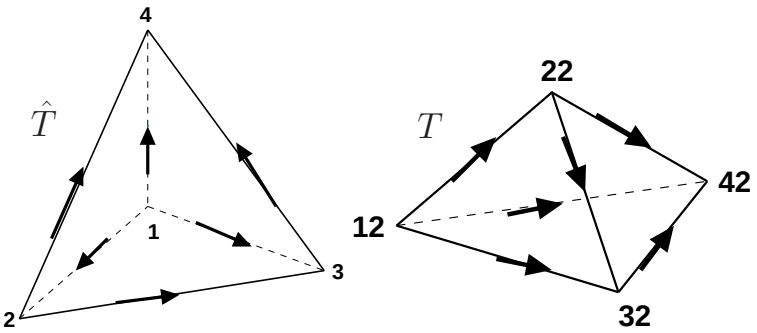


Figure 4.2 – Using global numbers to examine edges and faces, the 'structure of orientation' of $T = \{12, 32, 42, 22\}$ is the one of $\hat{T} = \{1, 2, 3, 4\}$ up to a rotation.

should be satisfied: we wish to use *for all mesh simplices* T the 'dualizing' coefficients of the matrix $\hat{V}^{-1}$ calculated, once for all, for the reference simplex $\hat{T}$ *with* a certain choice of orientation and choice of edges (recall that V^{-1} already does not depend on the metrics of the simplex for which it is calculated, see Property 3.19). To be allowed to do this, it is sufficient to use the nodes *global numbers* to decide the order in which the non dual $\mathbf{w}_j$ (from which we start to then get the $\tilde{\mathbf{w}}_j$) are constructed locally on T. More precisely, for the edge-type (resp. face-type) basis functions the edges (resp. faces) are examined in the order written in the caption of Figure 3.3, but replacing the nodes numbers $1, 2, 3, 4$ with the increasing global numbers of the nodes of T: the 1st examined edge is from the node with the 1st smallest global number to the one with the 2nd smallest global number, the 2nd examined edge is from the node with the 1st smallest global number to the one with the 3rd smallest global number, and so on, then the 1st examined face is the one opposite the node with the smallest global number, and so on. Indeed, in this way the first need is respected *and* the 'structure of orientation' of T is the one of $\hat{T}$ up to a rotation (see Figure 4.2): then we are allowed to use the coefficients of $\hat{V}^{-1}$ for the linear combinations giving the $\tilde{\mathbf{w}}_j$.

Note that in 3d (resp. in 2d), to assemble the global linear system matrix, it is not essential which volume-type (resp. face-type) generators are chosen since they are not shared between tetrahedra (resp. triangles). On the contrary, also this choice is important when we want to use for all mesh simplices the coefficients of $\hat{V}^{-1}$ calculated for a simplex with a certain choice of orientation and choice of edges.

4.2.1 Implementation of the basis functions

To implement the strategy introduced to construct locally the basis functions $\tilde{\mathbf{w}}_j$ while respecting the two requirements just described, two *permutations* can be used; note that in this paragraph the numberings start from 0, and no more from 1, in order to comply with the C++ plugin written for the insertion in FreeFem++ of the new FE space. First, to construct the non dual $\mathbf{w}_j$, we define a permutation p_{d+1} of $d+1$ elements as follows: $p_{d+1}[i]$ is the local number (it takes values among $0, \ldots, d$) of the node with the i-th smallest global number in the simplex T, so we can say that p_{d+1} is the permutation for which the nodes of T are listed with increasing global number. For instance, for the tetrahedron $T = \{12, 32, 42, 22\}$ in Figure 4.2, we have $p_4 = \{0, 3, 1, 2\}$. So, in the first step of construction of the $\mathbf{w}_j$, we replace each λ_i appearing in their expression with $\lambda_{p_{d+1}[i]}$. In the code of the FreeFem++ plugin, the permutation p_4 is called `perm`.

```
int k0=0, k1=1, k2=2, k3=3;
if(tV[k0]>tV[k1]) Exchange(k0,k1);
```

```
if(tV[k1]>tV[k2]) Exchange(k1,k2);
if(tV[k2]>tV[k3]) Exchange(k2,k3);
if(tV[k0]>tV[k1]) Exchange(k0,k1);
if(tV[k1]>tV[k2]) Exchange(k1,k2);
if(tV[k0]>tV[k1]) Exchange(k0,k1);
int perm[4] = {k0,k1,k2,k3};
```

Then, in the second step of construction of the $\tilde{\mathbf{w}}_j$ as linear combinations of the $\mathbf{w}_j$, we use a permutation $P_{n_{\text{dofs}}}$ of $n_{\text{dofs}} = \dim(W^1_{h,r}(T))$ elements to go back to the local order of edges and faces. For instance for the tetrahedron $T = \{12, 32, 42, 22\}$, the order in which edges are examined in the first step is

$$\{\{12, 22\}, \{12, 32\}, \{12, 42\}, \{22, 32\}, \{22, 42\}, \{32, 42\}\},$$

while the local order of edges would be

$$\{\{12, 32\}, \{12, 42\}, \{12, 22\}, \{32, 42\}, \{22, 32\}, \{22, 42\}\}$$

(the local order is given by how the nodes of T are listed); similarly, the order in which faces are examined in the first step is

$$\{\{22, 32, 42\}, \{12, 32, 42\}, \{12, 22, 42\}, \{12, 22, 32\}\},$$

while the local order of faces would be

$$\{\{22, 32, 42\}, \{12, 22, 42\}, \{12, 22, 32\}, \{12, 32, 42\}\}.$$

So for this tetrahedron, if $r = 2$ (for which there are 2 basis functions for each edge and 2 basis functions for each face, 20 basis functions in total listed in Example 3.22), we have

$$P_{20} = \{4, 5,\ 0, 1,\ 2, 3,\ 8, 9,\ 10, 11,\ 6, 7;\ 12, 13,\ 18, 19,\ 14, 15,\ 16, 17\},$$

(note that inside each edge or face the 2 related dofs remain ordered according to the global numbers). This permutation ($r = 2$) is built with the following code. There, `edgesMap` corresponds to a map that associates the pair $\{a, b\}$ of nodes of an edge e_i with its number $0 \leq i \leq 5$; this map is rather implemented with an array defined as `edgesMap`$[(a + 1)(b + 1)] = i$, where $(a + 1)(b + 1)$ results to be unique and symmetric for a pair (a, b), $0 \leq a, b \leq 3$, representing a tetrahedron edge.

```
int edgesMap[13] = {-1,-1,0,1,2,-1,3,-1,4,-1,-1,-1,5};
// static const int nvedge[6][2] = {{0,1},{0,2},{0,3},{1,2},{1,3},{2,3}};
int p20[20];
for(int i=0; i<6; ++i) // edge dofs
{
    int ii0 = Element::nvedge[i][0], ii1 = Element::nvedge[i][1];
    int i0 = perm[ii0]; int i1 = perm[ii1];
    int iEdge = edgesMap[(i0+1)*(i1+1)]; // i of the edge [i0,i1]
    p20[i*2] = iEdge*2;
    p20[i*2+1] = iEdge*2+1;
}
for(int j=0; j<4; ++j) // face dofs
{
    int jFace = perm[j];
    p20[12+j*2] = 12+jFace*2;
    p20[12+j*2+1] = 12+jFace*2+1;
}
```

Then, we will save the linear combinations of the $\mathbf{w}_\ell$, with coefficients given by the j-th column of $\hat{V}^{-1}$ (see Example 3.22), in the final basis functions $\tilde{\mathbf{w}}_{P_{20}[j]}$, thus in duality with the chosen dofs:

```
wtilde[p20[0]] = +4*w[0]-2*w[1]-4*w[16]+2*w[17]-4*w[18]+2*w[19];
wtilde[p20[1]] = -2*w[0]+4*w[1]-2*w[16]-2*w[17]-2*w[18]-2*w[19];
wtilde[p20[2]] = +4*w[2]-2*w[3]-4*w[14]+2*w[15]+2*w[18]-4*w[19];
wtilde[p20[3]] = -2*w[2]+4*w[3]-2*w[14]-2*w[15]-2*w[18]-2*w[19];
wtilde[p20[4]] = +4*w[4]-2*w[5]+2*w[14]-4*w[15]+2*w[16]-4*w[17];
wtilde[p20[5]] = -2*w[4]+4*w[5]-2*w[14]-2*w[15]-2*w[16]-2*w[17];
wtilde[p20[6]] = +4*w[6]-2*w[7]-4*w[12]+2*w[13]+2*w[18]-4*w[19];
wtilde[p20[7]] = -2*w[6]+4*w[7]-2*w[12]-2*w[13]+4*w[18]-2*w[19];
wtilde[p20[8]] = +4*w[8]-2*w[9]+2*w[12]-4*w[13]+2*w[16]-4*w[17];
wtilde[p20[9]] = -2*w[8]+4*w[9]-2*w[12]-2*w[13]+4*w[16]-2*w[17];
wtilde[p20[10]] = +4*w[10]-2*w[11]+2*w[12]-4*w[13]+2*w[14]-4*w[15];
wtilde[p20[11]] = -2*w[10]+4*w[11]+4*w[12]-2*w[13]+4*w[14]-2*w[15];
wtilde[p20[12]] = +8*w[12]-4*w[13];
wtilde[p20[13]] = -4*w[12]+8*w[13];
wtilde[p20[14]] = +8*w[14]-4*w[15];
wtilde[p20[15]] = -4*w[14]+8*w[15];
wtilde[p20[16]] = +8*w[16]-4*w[17];
wtilde[p20[17]] = -4*w[16]+8*w[17];
wtilde[p20[18]] = +8*w[18]-4*w[19];
wtilde[p20[19]] = -4*w[18]+8*w[19];
```

4.3 The interpolation operator

Duality of the basis functions with the dofs is needed in FreeFem++ to provide an *interpolation operator* onto a desired FE space of a function given by its analytical expression (or of a function belonging to another FE space). We define for a (vector) function $\mathbf{u} \in Y^1 \subset H(\mathrm{curl}, T)$ (where Y^1 is a suitable subspace, see [65]) its finite element approximation $\mathbf{u}_h = \Pi_h(\mathbf{u})$ using the interpolation operator (3.15), which is recalled here:

$$\Pi_h \colon Y^1 \subset H(\mathrm{curl}, T) \to W^1_{h,r}(T), \quad \mathbf{u} \mapsto \mathbf{u}_h = \sum_{i=1}^{n_{\mathrm{dofs}}} c_i \tilde{\mathbf{w}}_i, \quad \text{with } c_i := \sigma_i(\mathbf{u}). \tag{4.1}$$

The interpolant coefficients $c_i = \sigma_i(\mathbf{u})$ are computed in FreeFem++ with suitable quadrature formulas (on edges, faces or volumes) to approximate the values of the dofs in Definition 3.13 applied to $\mathbf{u}$.

Now, denote by g the whole integrand inside the dof expression, by n_{QF_i} the number of quadrature points of the suitable quadrature formula (on a segment, triangle or tetrahedron) to compute the integral (of precision high enough so that the integral is computed exactly when the dof is applied to a basis function), and by $\mathbf{x}_p$, a_p, $1 \le p \le n_{\mathrm{QF}_i}$ the quadrature points and their weights. Then we have

$$c_i = \sigma_i(\mathbf{u}) = \sum_{p=1}^{n_{\mathrm{QF}_i}} a_p \, g(\mathbf{x}_p) = \sum_{p=1}^{n_{\mathrm{QF}_i}} a_p \sum_{j=1}^{d} \beta_j(\mathbf{x}_p) \, u_j(\mathbf{x}_p), \tag{4.2}$$

where for the second equality we have factorized $g(\mathbf{x}_p)$ in order to highlight the d components of $\mathbf{u}$, denoted by u_j, $1 \le j \le d$ (see the paragraph below).

Therefore, by substituting the expression of the coefficients (4.2) in the interpolation operator definition (4.1), we have the following expression of the interpolation operator

$$\Pi_h(\mathbf{u}) = \sum_{i=1}^{n_{\mathrm{dofs}}} \sum_{p=1}^{n_{\mathrm{QF}_i}} \sum_{j=1}^{d} a_p \, \beta_j(\mathbf{x}_p) \, u_j(\mathbf{x}_p) \, \tilde{\mathbf{w}}_i = \sum_{\ell=1}^{n_{\mathrm{ind}}} \alpha_\ell \, u_{j_\ell}(\mathbf{x}_{p_\ell}) \, \tilde{\mathbf{w}}_{i_\ell}, \tag{4.3}$$

where we have set α_ℓ equals each $a_p\,\beta_j(\mathbf{x}_p)$ for the right triple $(i,p,j)=(i_\ell,p_\ell,j_\ell)$. Indeed, a FreeFem++ plugin to introduce a new finite element (represented with a C++ class) should implement (4.3) by specifying the quadrature points, the indices i_ℓ (dof indices), p_ℓ (quadrature point indices), j_ℓ (component indices), which do not depend on the simplex and are defined in the class constructor, and the coefficients α_ℓ, which can depend on the simplex (if so, which is in particular the edge elements case, the α_ℓ are defined with the class function `set`).

4.3.1 Implementation of the interpolation operator for $d=3$, $r=2$

We report here the code (extracted from the plugin `Element_Mixte3d.cpp` mentioned before) defining first the indices of (4.3) for the `Edge13d` finite element, i.e. for $d=3$, $r=2$. There `QFe`, `QFf` are the edge, resp. face, quadrature formulas, and `ne`=6, `nf`=4, are the number of edges, resp. faces, of the simplex (tetrahedron); we have $n_{\text{ind}} = d\cdot$`QFe.n`$\cdot$ 2`ne` $+\, d\cdot$`QFf.n`$\cdot 2$`nf`. Note that in the code the numberings start from 0, and no more from 1.

```
int i=0, p=0, e=0; // i is l
for(e=0; e<(Element::ne)*2; e++) // 12 edge dofs
{
  if (e%2==1) {p = p-QFe.n;}
  // if true, the quadrature pts are the ones of the previous dof (same edge)
  for(int q=0; q<QFe.n; ++q,++p) // 2 edge quadrature pts
    for (int c=0; c<3; c++,i++) // 3 components
    {
      this->pInterpolation[i]=p; // p_l
      this->cInterpolation[i]=c; // j_l
      this->dofInterpolation[i]=e; // i_l
      this->coefInterpolation[i]=0.; // alfa_l (filled with the function set)
    }
}
for(int f=0; f<(Element::nf)*2; f++) // 8 face dofs
{
  if (f%2==1) {p = p-QFf.n;}
  // if true, the quadrature pts are the ones of the previous dof (same face)
  for(int q=0; q<QFf.n; ++q,++p) // 3 face quadrature pts
    for (int c=0; c<3; c++,i++) // 3 components
    {
      this->pInterpolation[i]=p; // p_l
      this->cInterpolation[i]=c; // j_l
      this->dofInterpolation[i]=e+f; // i_l
      this->coefInterpolation[i]=0.; // alfa_l (filled with the function set)
    }
}
```

Then, the coefficients α_ℓ are defined as follows. We start by writing (4.2) for one edge-type dof, with $e=\{n_1,n_2\}$:

$$c_i=\sigma_i(\mathbf{u})=\frac{1}{|e|}\int_e(\mathbf{u}\cdot\mathbf{t}_e)\,\lambda_{n_1}=\sum_{p=1}^{\texttt{QFe.n}}a_p\,(\mathbf{u}(\mathbf{x}_p)\cdot\mathbf{t}_e)\,\lambda_{n_1}(\mathbf{x}_p)$$
$$=\sum_{p=1}^{\texttt{QFe.n}}a_p\sum_{j=1}^{d}u_j(\mathbf{x}_p)(x_{n_2\,j}-x_{n_1\,j})\lambda_{n_1}(\mathbf{x}_p)$$

so $\beta_j(\mathbf{x}_p)=(x_{n_2\,j}-x_{n_1\,j})\lambda_{n_1}(\mathbf{x}_p)$ and $\alpha_\ell=a_{p_\ell}\,\beta_{j_\ell}(\mathbf{x}_{p_\ell})=(x_{n_2\,j_\ell}-x_{n_1\,j_\ell})\,a_{p_\ell}\,\lambda_{n_1}(\mathbf{x}_{p_\ell})$.

Similarly for one face-type dof, with $f = \{n_1, n_2, n_3\}$, $e = \{n_1, n_2\}$:

$$c_i = \sigma_i(\mathbf{u}) = \frac{1}{|f|}\int_f (\mathbf{u}\cdot\mathbf{t}_e) = \sum_{p=1}^{\texttt{QFf.n}} a_p\,(\mathbf{u}(\mathbf{x}_p)\cdot\mathbf{t}_e) = \sum_{p=1}^{\texttt{QFf.n}} a_p \sum_{j=1}^{d} u_j(\mathbf{x}_p)(x_{n_2 j} - x_{n_1 j})$$

so $\beta_j(x_p) = (x_{n_2 j} - x_{n_1 j})$ and $\alpha_\ell = a_{p_\ell}\,\beta_{j_\ell}(\mathbf{x}_{p_\ell}) = (x_{n_2 j_\ell} - x_{n_1 j_\ell})\,a_{p_\ell}$. The code that generalizes this calculations for all the dofs is the following, extracted from the function `set` of the plugin (note that also here we have to pay particular attention to the orientation and choice issues).

```
int i=0, p=0;
for(int ee=0; ee<Element::ne; ee++) // loop on the edges
{
  R3 E=K.Edge(ee);
  int eo = K.EdgeOrientation(ee);
  if(!eo) E=-E;
    for(int edof=0; edof<2; edof++) // 2 dofs for each edge
    {
      if (edof==1) {p = p-QFe.n;}
      for(int q=0; q<QFe.n; ++q,++p)
      {
        double ll=QFe[q].x; // value of lambda_0 or lambda_1
        if( (edof+eo) == 1 ) ll = 1-ll;
        for(int c=0; c<3; c++,i++)
        {
          M.coef[i] = E[c]*QFe[q].a*ll;
        }
      }
    }
}
for(int ff=0; ff<Element::nf; ff++) // loop on the faces
{
  const Element::Vertex * fV[3] = {& K.at(Element::nvface[ff][0]), ...
  //  (one unique line with the following)
  ... & K.at(Element::nvface[ff][1]), & K.at(Element::nvface[ff][2])};
  int i0=0, i1=1, i2=2;
  if(fV[i0]>fV[i1]) Exchange(i0,i1);
    if(fV[i1]>fV[i2]) { Exchange(i1,i2);
      if(fV[i0]>fV[i1]) Exchange(i0,i1); }
  // now local numbers in the tetrahedron:
  i0 = Element::nvface[ff][i0], i1 = Element::nvface[ff][i1], ...
  ... i2 = Element::nvface[ff][i2];
  for(int fdof=0; fdof<2; ++fdof) // 2 dofs for each face
  {
    int ie0=i0, ie1 = fdof==0? i1 : i2;
    // edge for the face dof (its endpoints local numbers)
    R3 E(K[ie0],K[ie1]);
    if (fdof==1) {p = p-QFf.n;}
    for(int q=0; q<QFf.n; ++q,++p) // loop on the 3 face quadrature pts
      for (int c=0; c<3; c++,i++) // loop on the 3 components
      {
        M.coef[i] = E[c]*QFf[q].a;
      }
  }
}
```

4.4 Using the new finite elements in a FreeFem++ script

The edge elements in 3d of degree $2, 3$ can be used (since FreeFem++ version 3.44) by loading in the `edp` script the plugin (`load "Element_Mixte3d"`), and using the keywords `Edge13d`, `Edge23d` respectively. The edge elements of the lowest degree 1 were already available and called `Edge03d`. After generating a tetrahedral mesh `Th`, complex vector functions `E`, `v` in, e.g., the `Edge03d` space on `Th` are declared with the commands:

```
fespace Vh(Th,Edge03d);   Vh<complex> [Ex,Ey,Ez], [vx,vy,vz];
```

Then the weak formulation (2.41) of the problem is naturally transcribed as:

```
macro Curl(ux,uy,uz) [dy(uz)-dz(uy),dz(ux)-dx(uz),dx(uy)-dy(ux)] // EOM
macro Nvec(ux,uy,uz) [uy*N.z-uz*N.y,uz*N.x-ux*N.z,ux*N.y-uy*N.x] // EOM

problem waveguide([Ex,Ey,Ez], [vx,vy,vz], solver=sparsesolver) =
                  int3d(Th)(Curl(Ex,Ey,Ez)'*Curl(vx,vy,vz))
                - int3d(Th)(gamma^2*[Ex,Ey,Ez]'*[vx,vy,vz])
                + int2d(Th,in,out)(1i*eta*Nvec(Ex,Ey,Ez)'*Nvec(vx,vy,vz))
                - int2d(Th,in)([vx,vy,vz]'*[Gix,Giy,Giz])
                + on(guide,Ex=0,Ey=0,Ez=0);
```

Note that Dirichlet-type boundary conditions, as in this case the metallic (or PEC) boundary condition, are imposed in FreeFem++ by using the keyword `on`: by the penalty method, it acts on the unknowns corresponding to FE dofs whose support belongs to the boundary. Thus, even if the code writing seems to impose $\mathbf{E} = \mathbf{0}$, in fact, since the present dofs (3.13) have tangential nature, the imposed condition is properly $\mathbf{E} \times \mathbf{n} = \mathbf{0}$.

See more details about the use of the new finite elements in the example `waveguide.edp` available in `examples++-load` folder of every FreeFem++ distribution and in the more elaborated codes of Appendix A.

The interpolation operator in FreeFem++ is simply called with the `=` symbol: for example one can define analytical functions `func f1 = 1+x+2*y+3*z; func f2 = -1-x-2*y+2*z; func f3 = 2-2*x+y-2*z;` and call `[Ex,Ey,Ez]=[f1,f2,f3];`.

Chapter 5

Schwarz domain decomposition preconditioners

This Chapter is the result of a collaboration with Victorita Dolean, Frédéric Hecht and Francesca Rapetti. The corresponding submitted preprint [18] is available on arXiv and HAL (<hal-01298938>). A preliminary study was published in the proceedings paper [19] of the DD23 International Conference on Domain Decomposition Methods, in collaboration with Victorita Dolean, Richard Pasquetti, Francesca Rapetti. Paragraph 5.3.1 comes from a collaboration also with Frédéric Nataf and Pierre-Henri Tournier.

Contents

5.1 Introduction

In this thesis, we are interested in solving the *time-harmonic* Maxwell's equation (2.26), that is we work in the *frequency domain*. The time-harmonic formulation of Maxwell's equations has the same nature and presents the same difficulties as the Helmholtz equation $-\Delta u - \tilde{\omega}^2 u = f$, the simplest possible model of wave propagation [1]. Indeed, when the wavenumber $\tilde{\omega}$ is large, the (standard) variational formulation of these equations is symmetric but *sign-indefinite* [2]. This is one of the reasons why the linear systems arising from FE discretizations of these equations are difficult to solve with classical iterative methods (see the review [50] for more details in the case of the Helmholtz equation). Moreover, as the wavenumber $\tilde{\omega}$ increases the matrix of the linear system becomes very large because

1. The Helmholtz equation can be derived from the wave equation $\frac{1}{c^2}\frac{\partial^2 \mathcal{U}}{\partial t^2} - \Delta \mathcal{U} = \mathcal{F}$ (where c is the propagation speed), by looking for solutions in the form $\mathcal{U}(\mathbf{x}, t) = \mathrm{Re}(u(\mathbf{x})e^{\mathrm{i}\omega t})$ when the source $\mathcal{F}(\mathbf{x}, t) = \mathrm{Re}(f(\mathbf{x})e^{\mathrm{i}\omega t})$ is harmonic in time. Recall that the wavenumber can be expressed as $\tilde{\omega} = \omega/c$.

2. See [84] for a non standard positive definite variational formulation of the Helmholtz equation and also for a complete review of the properties of the equation.

a higher wavenumber entails a finer discretization. Indeed, an accurate approximation of waves that oscillates on a scale of $\tilde{\omega}^{-1}$ requires, at least, the mesh size h to be proportional to $\tilde{\omega}^{-1}$ as $\tilde{\omega}$ increases; furthermore, the *pollution effect* means that this is still not enough to control the finite element method discretization error (a mesh size $h \sim \tilde{\omega}^{-3/2}$ is generally required, see [70]).

Note that in literature there are many good solvers and preconditioners for another class of Maxwell's problems, as the ones arising from an implicit time discretization of the time-dependent Maxwell's equations (see, e.g., the introduction of [40]), which yield positive definite matrices. Such solvers include multigrid or auxiliary space methods, see e.g. [99, 8, 76, 26] for low order finite elements, [77] for high order ones, and domain decomposition methods, see e.g. [111, 39].

For the time-harmonic Maxwell's equations, Domain Decomposition (DD) methods or preconditioners are currently the most promising solution techniques. The first domain decomposition method for these equations was proposed by Després in [38], where impedance boundary conditions are used as transmission conditions at the interfaces between subdomains, instead of the classical Dirichlet conditions. Further improvements can be found in [33] where new transmission conditions are studied. Over the last decade, these *optimized Schwarz methods* were further developed: for the first order formulation (2.23) of the equations complete optimization results are known, also in the case of conductive medium [39, 48], while for the second order formulation (2.26) partial optimization results were obtained in various works. Recently it has been shown in [40] that the convergence factors and the optimization process for the two formulations are the same.

Nevertheless, the development of Schwarz domain decomposition solvers and preconditioners is still an open issue for high order discretizations. A recent work for the non overlapping case is reported in [82], see also [110]. In this chapter, we use overlapping Schwarz DD preconditioners based on impedance transmission conditions to solve the algebraic system resulting from equation (2.41), discretized with the high order edge finite elements defined in Chapters 3 and 4. Recall that equation (2.41) is the variational formulation of the boundary value problem (2.28), which models electromagnetic wave propagation in waveguides (see Section 2.2.4 for the simplified $2d$ model). Before presenting the preconditioners suited to the time-harmonic Maxwell's equations in Section 5.3, we introduce the classical Schwarz domain decomposition methods in Section 5.2. In Section 5.4 numerical experiments, both in two and three dimensions, are performed to validate the preconditioners.

5.2 Classical Schwarz domain decomposition methods

Among the Domain Decomposition (DD) methods, we focus on the family of Schwarz methods, named after Hermann A. Schwarz, who in 1869 invented the first DD method, but rather as an analytical tool [102]. See [56] for a detailed historical presentation of Schwarz methods and book [41] for a complete overview of DD methods.

Schwarz wanted to find a rigorous proof of the Dirichlet principle, which states that, for a bounded domain Ω, a function satisfying Laplace's problem

$$\begin{cases} -\Delta u = 0 \text{ in } \Omega, \\ \quad u = g \text{ on } \partial\Omega, \end{cases} \tag{5.1}$$

gives the infimum of the integral $\int_\Omega |\nabla v|^2$ over all functions v satisfying $v = g$ on $\partial\Omega$; Schwarz had to show that the infimum is attained on arbitrary domains. Thus, he considered a complicated domain Ω, composed of two overlapping simple ones Ω_1 and Ω_2,

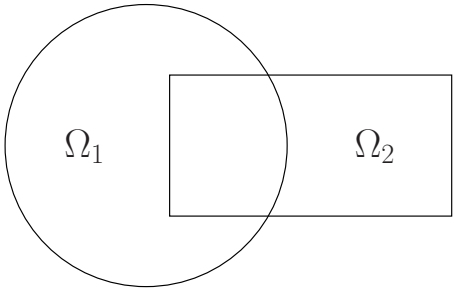


Figure 5.1 – A complicated domain Ω, composed of a disk Ω_1 and a rectangle Ω_2, for which the first domain decomposition method was introduced by Schwarz. Nowadays it is used as the logo of the DD community.

like a disk and a rectangle (see Figure 5.1), where solutions can be obtained using Fourier series. He proposed an iterative method, called the *alternating Schwarz method*, which converges to the solution of problem (5.1). It consists in solving the problem alternately in each subdomain, using at the interfaces ($\partial\Omega_1 \cap \overline{\Omega}_2$ and $\partial\Omega_2 \cap \overline{\Omega}_1$) Dirichlet *transmission conditions* coming from the solution just computed by the neighbour subdomain. More precisely it updates $(u_1^n, u_2^n) \to (u_1^{n+1}, u_2^{n+1})$ by:

$$\begin{aligned} -\Delta u_1^{n+1} &= 0 \text{ in } \Omega_1 & & & -\Delta u_2^{n+1} &= 0 \text{ in } \Omega_2 \\ u_1^{n+1} &= g \text{ on } \partial\Omega_1 \cap \partial\Omega & \text{then} & & u_2^{n+1} &= g \text{ on } \partial\Omega_2 \cap \partial\Omega \\ u_1^{n+1} &= u_2^n \text{ on } \partial\Omega_1 \cap \overline{\Omega}_2 & & & u_2^{n+1} &= u_1^{n+1} \text{ on } \partial\Omega_2 \cap \overline{\Omega}_1. \end{aligned} \tag{5.2}$$

More than a century later, a small modification by Pierre Louis Lions [80] of this algorithm made it suited to parallel architectures, which were becoming more and more available. The *parallel Schwarz method* solves the subdomain problems concurrently, the only change with respect to the alternating Schwarz method is the iteration index in the second transmission condition:

$$\begin{aligned} -\Delta u_1^{n+1} &= 0 \text{ in } \Omega_1 & & -\Delta u_2^{n+1} &= 0 \text{ in } \Omega_2 \\ u_1^{n+1} &= g \text{ on } \partial\Omega_1 \cap \partial\Omega & & u_2^{n+1} &= g \text{ on } \partial\Omega_2 \cap \partial\Omega \\ u_1^{n+1} &= u_2^n \text{ on } \partial\Omega_1 \cap \overline{\Omega}_2 & & u_2^{n+1} &= u_1^n \text{ on } \partial\Omega_2 \cap \overline{\Omega}_1. \end{aligned} \tag{5.3}$$

While the convergence of these methods depends on the underlying partial differential equation (PDE) to be solved, similar methods can be defined for any PDE, their fundamental idea of decomposition and iteration is completely general. They can also be easily extended to a decomposition of the domain Ω into more than two overlapping subdomains Ω_s, $s = 1, \ldots, N_{\text{sub}}$.

If we look at the *algebraic level*, considering the linear system $A\mathbf{u} = \mathbf{f}$ arising from the discretization of the PDE, Theorem 3.5 of [56] shows that a discretization of the parallel Schwarz method is equivalent to the *Restricted Additive Schwarz method* (RAS), introduced by Cai and Sarkis [30]. To define discrete Schwarz methods, we consider an ordered set $\mathcal{N}$ of the unknowns (on the whole domain), and its decomposition $\mathcal{N} = \bigcup_{s=1}^{N_{\text{sub}}} \mathcal{N}_s$ into the (non disjoint) ordered subsets corresponding to the different overlapping subdomains Ω_s. Then one builds the following matrices:

- the *restriction* matrix R_s from Ω to the subdomain Ω_s: it is a $\#\mathcal{N}_s \times \#\mathcal{N}$ Boolean matrix whose (i, j) entry equals 1 if the i-th unknown in $\mathcal{N}_s$ is the j-th one in $\mathcal{N}$;
- the *extension* (by zero) matrix from the subdomain Ω_s to Ω, which is given by R_s^T;

- the matrix $\widetilde{R}_s$, a $\#\mathcal{N}_s \times \#\mathcal{N}$ restriction matrix like R_s, but with some of the unit entries corresponding to the overlap replaced by zeros: this would correspond to a decomposition into *non overlapping* subdomains $\widetilde{\Omega}_s \subset \Omega_s$ (completely non overlapping, not even along their boundaries). Thus we have

$$\sum_{s=1}^{N_{\text{sub}}} \widetilde{R}_s^T R_s = I, \tag{5.4}$$

  that is the matrices $\widetilde{R}_s$ give a discrete *partition of unity*, which properly deals with the unknowns belonging to the overlap between subdomains.

Finally, we can give the definition of the restricted additive Schwarz method: it is the *preconditioned fixed point iteration* defined by

$$\mathbf{u}^{n+1} = \mathbf{u}^n + M_{RAS}^{-1}\,\mathbf{r}^n, \quad \mathbf{r}^n = \mathbf{f} - A\mathbf{u}^n,$$

where the matrix

$$M_{RAS}^{-1} = \sum_{s=1}^{N_{\text{sub}}} \widetilde{R}_s^T (R_s A R_s^T)^{-1} R_s \tag{5.5}$$

is called the *RAS preconditioner*. Note that here the local matrix $A_s \coloneqq R_s A R_s^T$ is the minor of the matrix A corresponding to the subset of unknowns $\mathcal{N}_s$ (in this Chapter the term 'local' refers to a subdomain and not to a mesh simplex).

As explained in the introduction of [30], the RAS method was found by modifying accidentally another discrete Schwarz method, the *Additive Schwarz method* (AS) introduced in [44]:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + M_{AS}^{-1}\,\mathbf{r}^n, \quad \mathbf{r}^n = \mathbf{f} - A\mathbf{u}^n,$$

where the matrix

$$M_{AS}^{-1} = \sum_{s=1}^{N_{\text{sub}}} R_s^T (R_s A R_s^T)^{-1} R_s \tag{5.6}$$

is called the *AS preconditioner*, and is symmetric if A is symmetric, contrarily to the *RAS* preconditioner. However, the additive Schwarz iteration fails to converge in the overlap (see, e.g., Paragraph 3.2 of [56] for more details).

When applying matrices (5.5) or (5.6) to the vector $\mathbf{r}^n$ in the fixed point iteration defining the discrete Schwarz methods, the local linear systems of matrices A_s are solved with a direct solver. Domain decomposition methods can be viewed indeed as hybrid methods that take the advantages of the two families of linear system solvers: *iterative solvers* and *direct solvers*. On the one hand, direct solvers are robust, i.e. they find the solution in a finite number of operations no matter how hard the problem is; but they are not suited for very large systems because of their high memory cost. On the other hand, iterative solvers require less memory and are easy to parallelize since they are based on matrix-vector products; their drawback is that they lack robustness, indeed for ill conditioned problems the use of a preconditioner is essential for fast convergence. Domain decomposition methods are naturally suited to parallel computing and they use the robust direct solvers on subproblems of a smaller size.

Moreover, rather than using Schwarz methods as iterative solvers, the matrices M_{RAS}^{-1} or M_{AS}^{-1} can be conveniently used *as preconditioners* for Krylov type iterative solvers, which are faster than the fixed point iterations. Krylov type iterative solvers include the CG (Conjugate Gradient) method, which can be applied just in the case of symmetric positive definite matrices A, or the GMRES (Generalized Minimal RESidual) method for more general matrices, for which M_{RAS}^{-1} should be always preferred to M_{AS}^{-1}.

5.3 Schwarz preconditioners for Maxwell's equations

A drawback of classical Schwarz methods is that, as iterative solvers, they are not convergent for some PDEs, like the Helmholtz equation: as shown for instance in Paragraph 2.2.1 of [41] or graphically in Paragraph 4.2 of [56], the classical methods remove the high frequency components in the error, but not the low frequency ones. This lack of convergence is one of the reasons why *optimized Schwarz methods* were originally proposed: in these methods the classical Dirichlet *transmission conditions* at the interfaces between subdomains are replaced with more effective conditions, like Robin conditions in the case of the Helmholtz equation [37, 58]. A good choice of transmission conditions is given by absorbing boundary conditions, that approximate the non local operators appearing in transparent boundary conditions. For bibliography references about transmission conditions for the time-harmonic Maxwell's equations see the introduction of this Chapter.

The discrete formulation of optimized Schwarz methods was introduced in [35]: the *Optimized Restricted Additive Schwarz* (ORAS) preconditioner is

$$M_{\mathrm{ORAS}}^{-1} = \sum_{s=1}^{N_{\mathrm{sub}}} \widetilde{R}_s^T A_{s,\mathrm{Opt}}^{-1} R_s, \tag{5.7}$$

where the local matrices $A_{s,\mathrm{Opt}}$ are now the matrices arising from discretizations of the problem in the subdomain Ω_s *with* the optimized transmission conditions as boundary conditions at the interfaces.

In our case, for the time-harmonic Maxwell's equation (2.26), the matrices $A_{s,\mathrm{Opt}}$ are the local matrices of the subproblems with homogeneous *impedance* boundary conditions (2.31) as transmission conditions at the interfaces between subdomains. The sign of the parameter in the impedance transmission conditions is important, see Remark 2.3. These local matrices stem from the discretization of the variational formulation (2.41) by the high order edge finite elements defined in Chapters 3 and 4. To build the subset of unknowns $\mathcal{N}_s$ corresponding to the subdomain Ω_s, note that each unknown corresponds to a degree of freedom (dof), of the type of Definition 3.13; so, an unknown belongs to $\mathcal{N}_s$ if the dof support (edge, face or volume) is contained in Ω_s. For edge finite elements, it is important to ensure that the *orientation* of the dofs is the same in the domain and in the subdomains. As an illustration of the construction, in the following example we write explicitly the restriction matrices $R_s, \widetilde{R}_s$ for the simple two-dimensional case shown in Figure 5.2, considering edge elements of degree $r = 1$.

Example 5.1 ($d = 2$, $r = 1$)**.** The domain Ω is decomposed into two overlapping subdomains Ω_1, Ω_2, and edge finite elements of degree $r = 1$ are considered: the degrees of freedom are in correspondence with the edges of the mesh, so their order inside $\mathcal{N}$, $\mathcal{N}_1$, $\mathcal{N}_2$ is given by the edges numbering of the figure. Here, since $\#\mathcal{N}_1 = \#\mathcal{N}_2 = 13$, $\#\mathcal{N} = 17$, all restriction matrices have dimension 13×17. The matrices $\widetilde{R}_1$, $\widetilde{R}_2$ are constructed using the non overlapping subdomains $\widetilde{\Omega}_1$, $\widetilde{\Omega}_2$ shown in the figure (note that the edge 11 of Ω is contained in $\widetilde{\Omega}_2$ and not in $\widetilde{\Omega}_1$). In the matrices below, if an entry is empty it should be

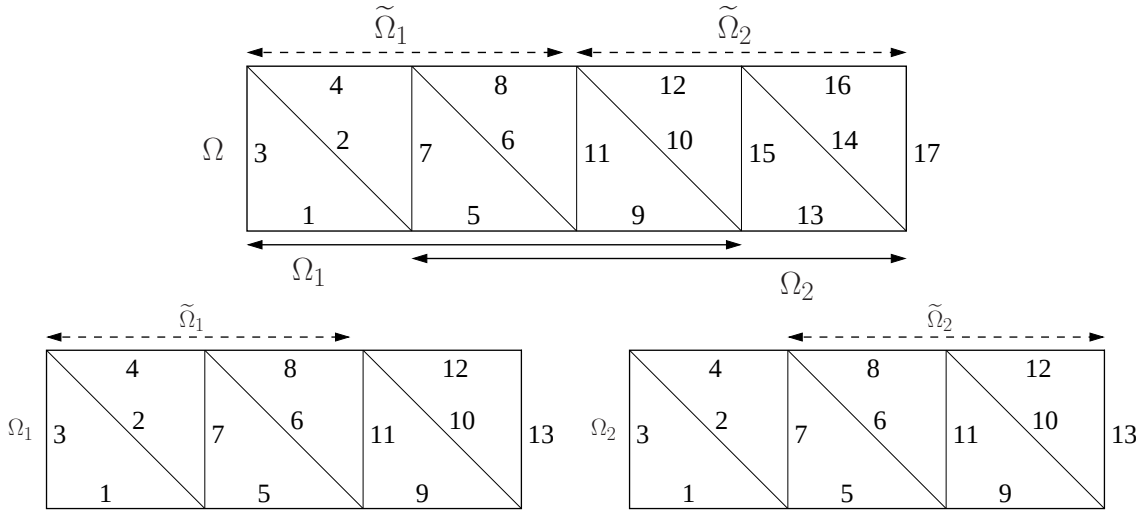


Figure 5.2 – Edges numbering for a domain Ω decomposed into two overlapping subdomains Ω_1, Ω_2 (above), and edges numbering for Ω_1 and Ω_2 (below). The non overlapping decomposition into $\widetilde{\Omega}_1$, $\widetilde{\Omega}_2$ is used to construct the matrices $\widetilde{R}_1$, $\widetilde{R}_2$.

equal to 0, and the bold zeros in $\widetilde{R}_1$, $\widetilde{R}_2$ are the zeros that replace unit entries in R_1, R_2:

$$R_1 = \begin{bmatrix} 1 & & & & & & & & & & & & \\ & 1 & & & & & & & & & & & \\ & & 1 & & & & & & & & & & \\ & & & 1 & & & & & & & & & \\ & & & & 1 & & & & & & & & \\ & & & & & 1 & & & & & & & \\ & & & & & & 1 & & & & & & \\ & & & & & & & 1 & & & & & \\ & & & & & & & & 1 & & & & \\ & & & & & & & & & 1 & & & \\ & & & & & & & & & & 1 & & \\ & & & & & & & & & & & 0 & 0 \\ & & & & & & & & & & & 1 & 0 \\ & & & & & & & & & & & 0 & 0 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & & & & & & & & \\ & & & & & 1 & & & & & & & \\ & & & & & & 1 & & & & & & \\ & & & & & & & 1 & & & & & \\ & & & & & & & & 1 & & & & \\ & & & & & & & & & 1 & & & \\ & & & & & & & & & & 1 & & \\ & & & & & & & & & & & 1 & \\ & & & & & & & & & & & & 1 \end{bmatrix},$$

$$\widetilde{R}_1 = \begin{bmatrix} 1 & & & & & & & & & & & & \\ & 1 & & & & & & & & & & & \\ & & 1 & & & & & & & & & & \\ & & & 1 & & & & & & & & & \\ & & & & 1 & & & & & & & & \\ & & & & & 1 & & & & & & & \\ & & & & & & 1 & & & & & & \\ & & & & & & & \mathbf{0} & & & & & \\ & & & & & & & & \mathbf{0} & & & & \\ & & & & & & & & & \mathbf{0} & & & \\ & & & & & & & & & & \mathbf{0} & & \\ & & & & & & & & & & & 0 & 0 \\ & & & & & & & & & & & 0 & \mathbf{0} \\ & & & & & & & & & & & 0 & 0 \end{bmatrix}, \quad \widetilde{R}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{0} & & & & & & & & \\ & & & & & \mathbf{0} & & & & & & & \\ & & & & & & \mathbf{0} & & & & & & \\ & & & & & & & \mathbf{0} & & & & & \\ & & & & & & & & 1 & & & & \\ & & & & & & & & & 1 & & & \\ & & & & & & & & & & 1 & & \\ & & & & & & & & & & & 1 & \\ & & & & & & & & & & & & 1 \end{bmatrix}.$$

5.3.1 Partition of unity

In FreeFem++ we implemented a slightly different form of the ORAS preconditioner, writing the matrices $\widetilde{R}_s$ in (5.4) as $\widetilde{R}_s = D_s R_s$, where we have highlighted diagonal matrices D_s (of size $\#\mathcal{N}_s \times \#\mathcal{N}_s$), such that

$$\sum_{s=1}^{N_{\text{sub}}} R_s^T D_s R_s = I. \tag{5.8}$$

Then preconditioner (5.7) becomes

$$M_{\mathrm{ORAS}}^{-1} = \sum_{s=1}^{N_{\mathrm{sub}}} R_s^T D_s A_{s,\mathrm{Opt}}^{-1} R_s. \tag{5.9}$$

The construction of the partition of unity matrices D_s is intricate, especially for (high order) edge finite elements. The starting point is considering continuous partition of unity functions $\{\chi_i\}_{1\leqslant i\leqslant N_{\mathrm{sub}}}$ for the classical piecewise linear nodal finite element, whose dofs are (the functional giving) the values at the nodes of the mesh. Denote by $\{\mathcal{T}_i^0\}_{1\leqslant i\leqslant N_{\mathrm{sub}}}$ the meshes of the auxiliary non overlapping subdomains, and $\{\mathcal{T}_i^\delta\}_{1\leqslant i\leqslant N_{\mathrm{sub}}}$ the meshes of the overlapping subdomains. In order to define the function χ_i, for $i = 1, \ldots, N_{\mathrm{sub}}$, we define first the function $\widetilde{\chi}_i$ as the continuous piecewise linear function on the global mesh, with support contained in $\mathcal{T}_i^\delta$, such that

$$\widetilde{\chi}_i = \begin{cases} 1 & \text{at all nodes of } \mathcal{T}_i^0, \\ 0 & \text{at all nodes of } \mathcal{T}_i^\delta \setminus \mathcal{T}_i^0. \end{cases}$$

The function χ_i can be then defined as the continuous piecewise linear function on the global mesh, with support contained in $\mathcal{T}_i^\delta$, such that its *discrete* value for each dof is evaluated by

$$\chi_i = \frac{\widetilde{\chi}_i}{\sum_{j=1}^{N_{\mathrm{sub}}} \widetilde{\chi}_j}. \tag{5.10}$$

Thus, we have $\sum_{i=1}^{N_{\mathrm{sub}}} \chi_i = 1$, both at the discrete and continuous level. Note that in the practical implementation the functions $\widetilde{\chi}_i$ and χ_i are constructed locally on $\mathcal{T}_i^\delta$, the relevant contribution of the $\widetilde{\chi}_j$ in (5.10) being on $\mathcal{T}_j^\delta \cap \mathcal{T}_i^\delta$. This removes all dependency on the global mesh, which could be otherwise problematic at large scales.

Now, for the high order edge finite elements, we can build a geometric partition of unity based on the support of the dofs (given in Definition 3.13): the entries of the diagonal matrix D_i, $i = 1, \ldots, N_{\mathrm{sub}}$, are obtained by *interpolating* the piecewise linear function χ_i at the *barycenters* of the support (edge, face, volume) of each dof. The partition of unity property (5.8) is then satisfied since $\sum_{i=1}^{N_{\mathrm{sub}}} \chi_i = 1$.

This interpolation is obtained thanks to an auxiliary FreeFem++ *scalar* FE space that has only the interpolation operator and no basis functions, available in the plugin `Element_Mixte3d` already mentioned in Section 4.1: these scalar FE spaces are called `Edge03ds0`, `Edge13ds0`, `Edge23ds0`, respectively for the FE spaces `Edge03d`, `Edge13d`, `Edge23d` of degree $r = 1, 2, 3$.

Remark 5.2. When impedance conditions are chosen as transmission conditions at the interfaces, it is essential that not only the function χ_s but also its derivative are equal to zero on the border of the subdomain Ω_s. Indeed, if this property is satisfied, the continuous version of the ORAS method is equivalent to the continuous method where the datum of the optimized transmission conditions comes from the neighbor subdomain (see the details in [41] §2.3.2).

5.4 Numerical experiments

We validate the ORAS preconditioner (5.9) for different values of physical and numerical parameters, and compare it with a symmetric variant without the partition of unity (called

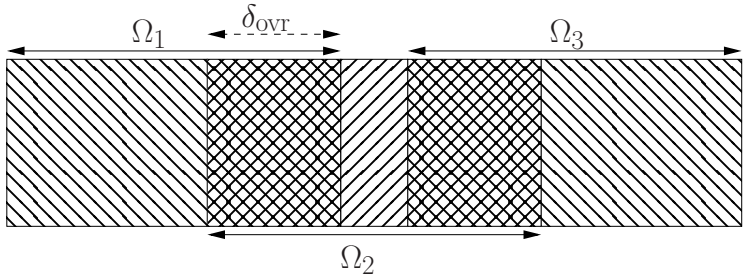


Figure 5.3 – The stripwise decomposition of the two-dimensional domain.

k	N_{dofs}	N_{iterNp}	N_{iter}	$\max\lvert\lambda-(1,0)\rvert$	$\#\{\lambda\in\mathbb{C}\setminus\bar{\mathcal{D}}_1\}$	$\#\{\lambda\in\partial\mathcal{D}_1\}$
0	282	179	5(10)	$1.04e{-}1(1.38e{+}1)$	0(4)	0(12)
1	884	559	6(15)	$1.05e{-}1(1.63e{+}1)$	0(8)	0(40)
2	1806	1138	6(17)	$1.05e{-}1(1.96e{+}1)$	0(12)	0(84)
3	3048	1946	6(21)	$1.05e{-}1(8.36e{+}2)$	0(16)	0(144)
4	4610	2950	6(26)	$1.05e{-}1(1.57e{+}3)$	0(20)	0(220)

Table 5.1 – Influence of the polynomial degree $r = k+1$ on the convergence of ORAS(OAS) preconditioner for $\omega = \omega_2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 2h$.

Optimized Additive Schwarz):

$$M_{\text{OAS}}^{-1} = \sum_{s=1}^{N_{\text{sub}}} R_s^T A_{s,\text{Opt}}^{-1} R_s.$$

The numerical experiments are performed for a waveguide configuration in $2d$ and $3d$.

5.4.1 Results for the two-dimensional problem

We present the results obtained for a two-dimensional waveguide as in Figure 2.3 with $c = 0.0502\,\text{m}$, $b = 0.00254\,\text{m}$, with the physical parameters: $\varepsilon = 8.85 \cdot 10^{-12}\,\text{F m}^{-1}$, $\mu = 1.26 \cdot 10^{-6}\,\text{H m}^{-1}$ and $\sigma = 0.15\,\text{S m}^{-1}$. We consider three angular frequencies $\omega_1 = 16\,\text{GHz}$, $\omega_2 = 32\,\text{GHz}$, and $\omega_3 = 64\,\text{GHz}$, varying the mesh size h according to the relation $h^2 \cdot \tilde{\omega}^3 = 2$ (which is generally believed to remove the pollution effect, see the introduction of this Chapter).

Here we solve the linear system resulting from the finite element discretization with GMRES (with a stopping criterion based on the relative residual and a tolerance of 10^{-6}), starting with a *random* initial guess, which ensures, unlike a zero initial guess, that all frequencies are present in the error. We compare the ORAS and OAS preconditioners, taking a stripwise subdomains decomposition, along the wave propagation, as shown in Figure 5.3.

To study the convergence of GMRES preconditioned by ORAS or OAS we vary first the polynomial degree $r = k + 1$ of the basis functions given in Definition 3.8 (Table 5.1, Figures 5.4–5.5, Figure 5.13), then the angular frequency ω (Table 5.2, Figures 5.6–5.7, Figure 5.14), the number of subdomains N_{sub} (Table 5.3, Figures 5.8–5.9, Figure 5.15) and finally the overlap size δ_{ovr} (Table 5.4, Figures 5.10–5.12, Figure 5.16). Here, $\delta_{\text{ovr}} = 1h, 2h, 4h$ means that we consider a total overlap between two subdomains of $1, 2, 4$ mesh triangles along the horizontal direction (see Figure 5.3). Note that usually in literature the overlap size indicates just half of δ_{ovr}, since that would say of how many layers each

ω	N_{dofs}	N_{iterNp}	N_{iter}	$\max\lvert\lambda-(1,0)\rvert$	$\#\{\lambda\in\mathbb{C}\setminus\bar{\mathcal{D}}_1\}$	$\#\{\lambda\in\partial\mathcal{D}_1\}$
ω_1	339	232	5(11)	2.46e−1(1.33e+1)	0(6)	0(45)
ω_2	1806	1138	6(17)	1.05e−1(1.96e+1)	0(12)	0(84)
ω_3	7335	4068	9(24)	3.03e−1(2.73e+1)	0(18)	0(123)

Table 5.2 – Influence of the angular frequency ω on the convergence of ORAS(OAS) preconditioner for $k=2$, $N_{\text{sub}}=2$, $\delta_{\text{ovr}}=2h$.

N_{sub}	N_{iter}	$\max\lvert\lambda-(1,0)\rvert$	$\#\{\lambda\in\mathbb{C}\setminus\bar{\mathcal{D}}_1\}$	$\#\{\lambda\in\partial\mathcal{D}_1\}$
2	6(17)	1.05e−1(1.96e+1)	0(12)	0(84)
4	10(27)	5.33e−1(1.96e+1)	0(38)	0(252)
8	19(49)	7.73e−1(1.96e+1)	0(87)	0(588)

Table 5.3 – Influence of the number of subdomains N_{sub} on the convergence of ORAS(OAS) preconditioner for $k=2$, $\omega=\omega_2$, $\delta_{\text{ovr}}=2h$.

auxiliary non overlapping subdomain has been extended in each direction to obtain the overlapping subdomain.

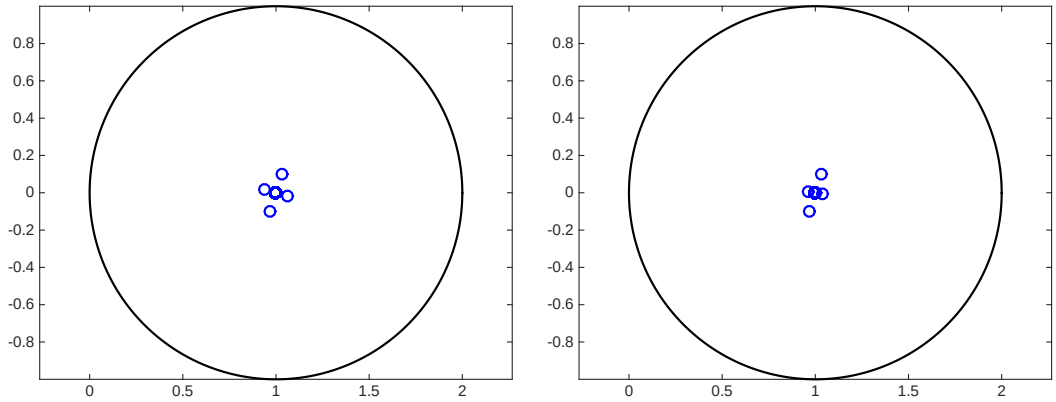
In Tables 5.1–5.4, N_{dofs} is the total number of degrees of freedom, N_{iterNp} is the number of iterations necessary to attain the prescribed convergence for GMRES without any preconditioner, and N_{iter} is the number of iterations for GMRES preconditioned by ORAS (OAS). Moreover, denoting by

$$\mathcal{D}_1=\{z\in\mathbb{C}:|z-z_0|<1\}$$

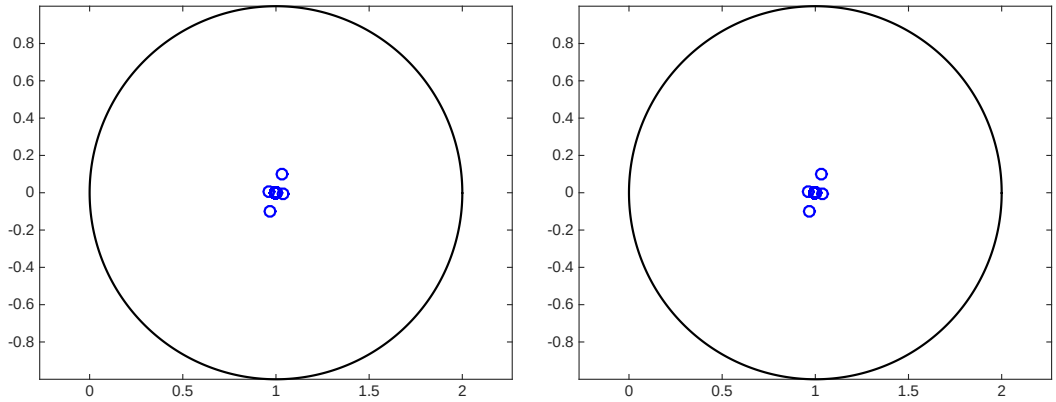
the unit disk centered at $z_0=(1,0)$ in the complex plane, we measure also the maximum distance to $(1,0)$ of the eigenvalues λ of the preconditioned matrix, the number of eigenvalues that have distance greater than 1, and the number of eigenvalues that have distance equal to 1 (up to a tolerance of 10^{-10}). This information is useful to characterize the convergence. Indeed, if A is the matrix of the system to solve and M^{-1} is the domain decomposition preconditioner, then $I-M^{-1}A$ is the iteration matrix of the Schwarz method used as an iterative solver: indeed, the preconditioned fixed point iteration can be written as $\mathbf{u}^{n+1}=\mathbf{u}^n-M^{-1}A\mathbf{u}^n+M^{-1}\mathbf{f}$. So, a measure of the convergence of the Schwarz solver would be to check whether the eigenvalues of the preconditioned matrix $M^{-1}A$ are contained in $\mathcal{D}_1$. When the Schwarz method is used, like here, as a preconditioner, the distribution of the spectrum remains qualitatively a good indicator of the convergence. Note that the matrix of the linear system doesn't change when N_{sub} or δ_{ovr} vary, therefore in Tables 5.3–5.4 (where $k=2$, $\omega=\omega_2$) we don't report $N_{\text{dofs}}=1806$ and $N_{\text{iterNp}}=1138$ again. In all Tables 5.1–5.4, we don't mention the condition number of the preconditioned matrix: indeed, no convergence rate estimates in terms of the condition number of the

δ_{ovr}	N_{iter}	$\max\lvert\lambda-(1,0)\rvert$	$\#\{\lambda\in\mathbb{C}\setminus\bar{\mathcal{D}}_1\}$	$\#\{\lambda\in\partial\mathcal{D}_1\}$
$1h$	10(20)	1.95e+1(1.96e+1)	3(12)	0(39)
$2h$	6(17)	1.05e−1(1.96e+1)	0(12)	0(84)
$4h$	5(14)	1.06e−1(1.96e+1)	0(12)	0(174)

Table 5.4 – Influence of the overlap size δ_{ovr} on the convergence of ORAS(OAS) preconditioner for $k=2$, $\omega=\omega_2$, $N_{\text{sub}}=2$.



(a) $k = 0$ (b) $k = 1$



(c) $k = 2$ (d) $k = 3$

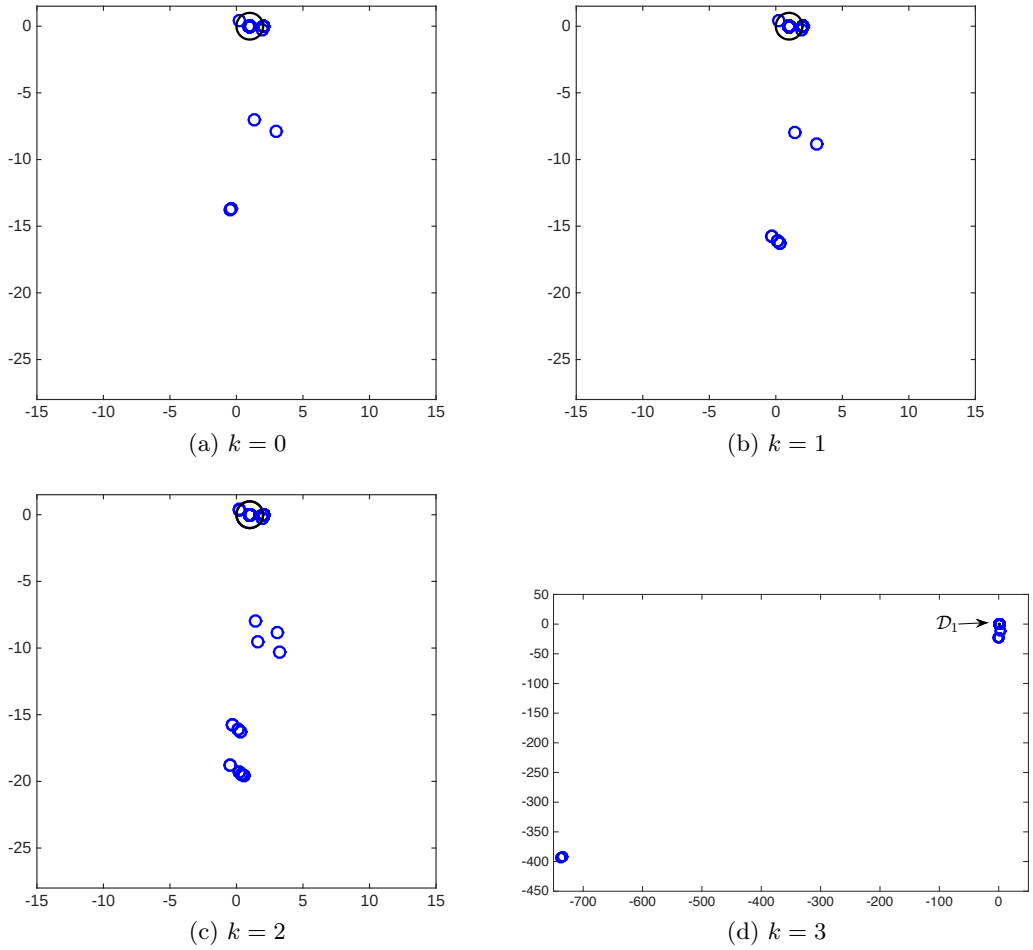
Figure 5.4 – Influence of the polynomial degree $r = k + 1$ on the spectrum of the ORAS-preconditioned matrix for $\omega = \omega_2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 2h$.

matrix, as those we are used to with the conjugate gradient method, are available for the GMRES method.

Figures 5.4, 5.6, 5.8, 5.10, respectively Figures 5.5, 5.7, 5.9, 5.11, show the whole spectrum in the complex plane of the matrix preconditioned by ORAS, respectively by OAS (note that many eigenvalues are multiple), together with $\partial\mathcal{D}_1$. Figures 5.13–5.16 show the evolution of the relative residual during the iterations of GMRES preconditioned with ORAS (left) and OAS (right).

Looking at the tables and figures, we can see that the non preconditioned GMRES method is very slow, and the ORAS preconditioner gives much faster convergence than the OAS preconditioner. As expected, the convergence becomes slower when ω or N_{sub} increase, or when δ_{ovr} decreases. In these tests, when varying k (which here gives the polynomial degree $r = k + 1$ of the FE basis functions), the number of iterations for convergence using the ORAS preconditioner is equal to 5 for $k = 0$ and then it stays equal to 6 for $k > 0$; this is reflected by the corresponding spectra in Figure 5.4, which indeed remain quite similar when k varies, and by the convergence history in Figure 5.13, left.

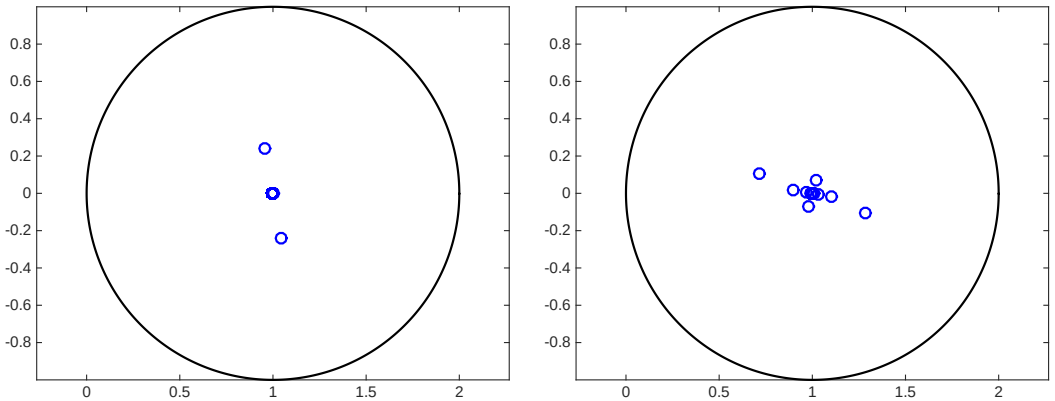
Note also that, when using the ORAS preconditioner, for 2 subdomains the spectrum is always well clustered inside the unit disk, except for the case with $\delta_{\text{ovr}} = 1h$ (see Figure 5.12), in which 3 eigenvalues are outside with distances from $(1, 0)$ equal to $19.5, 19.4, 14.4$. This case $\delta_{\text{ovr}} = 1h$ corresponds to adding a layer of triangles just to one of the two non overlapping subdomains to obtain the overlapping decomposition; hence



(a) $k = 0$ (b) $k = 1$

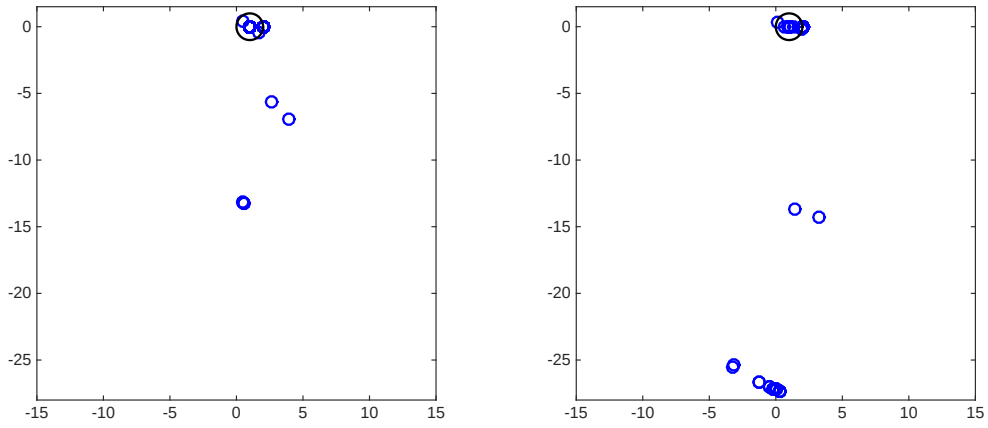
(c) $k = 2$ (d) $k = 3$

Figure 5.5 – Influence of the polynomial degree $r = k + 1$ on the spectrum of the OAS-preconditioned matrix for $\omega = \omega_2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 2h$.



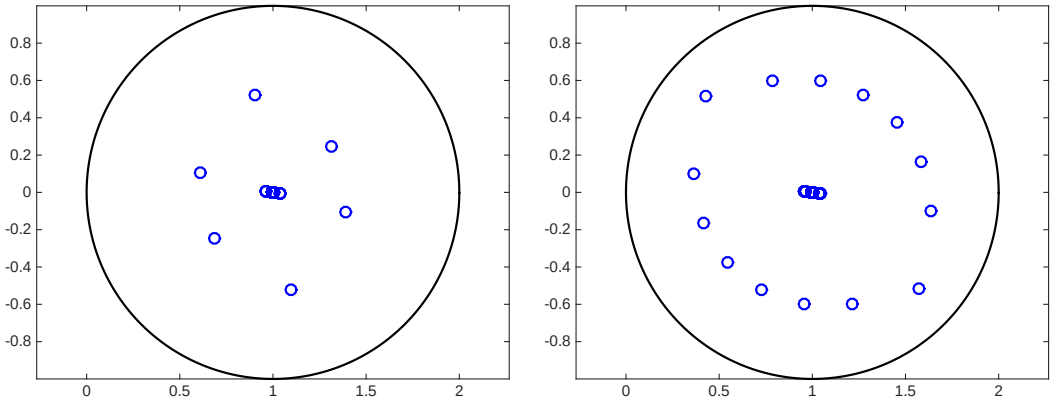
(a) $\omega = \omega_1$ (b) $\omega = \omega_3$

Figure 5.6 – Influence of the angular frequency ω on the spectrum of the ORAS-preconditioned matrix for $k = 2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 2h$.



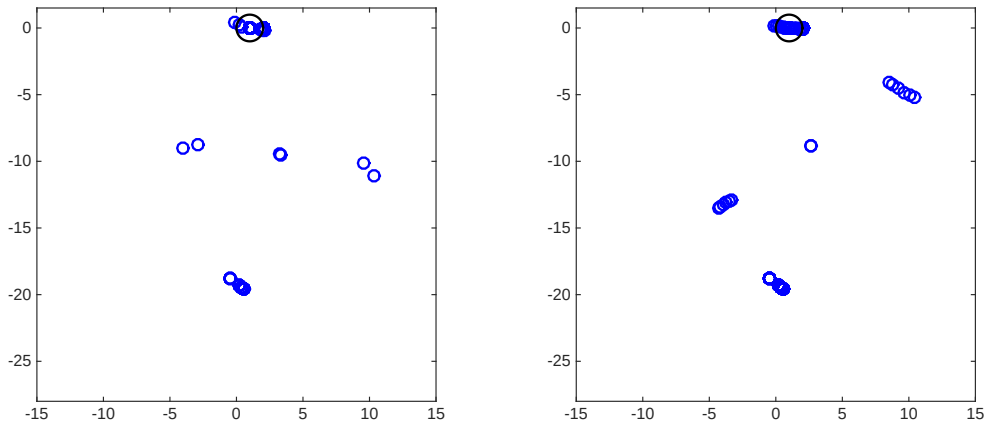
(a) $\omega = \omega_1$ (b) $\omega = \omega_3$

Figure 5.7 – Influence of the angular frequency ω on the spectrum of the OAS-preconditioned matrix for $k = 2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 2h$.



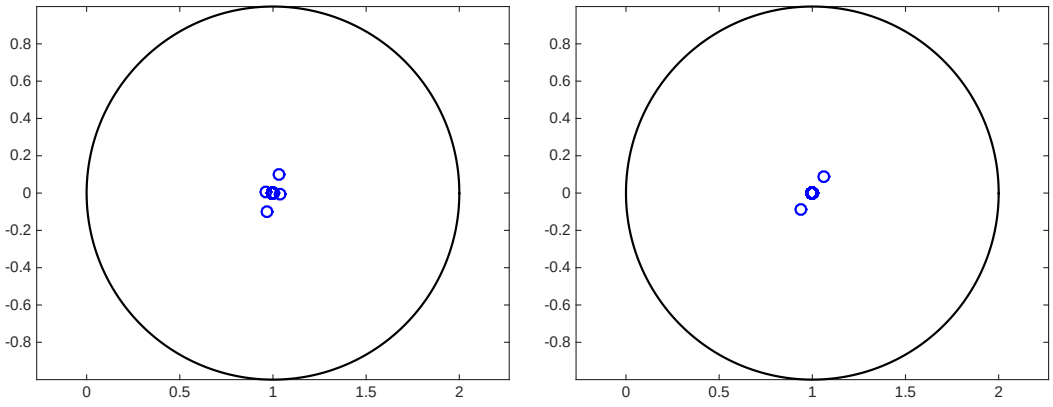
(a) $N_{\text{sub}} = 4$ (b) $N_{\text{sub}} = 8$

Figure 5.8 – Influence of the number of subdomains N_{sub} on the spectrum of the ORAS-preconditioned matrix for $k = 2$, $\omega = \omega_2$, $\delta_{\text{ovr}} = 2h$.



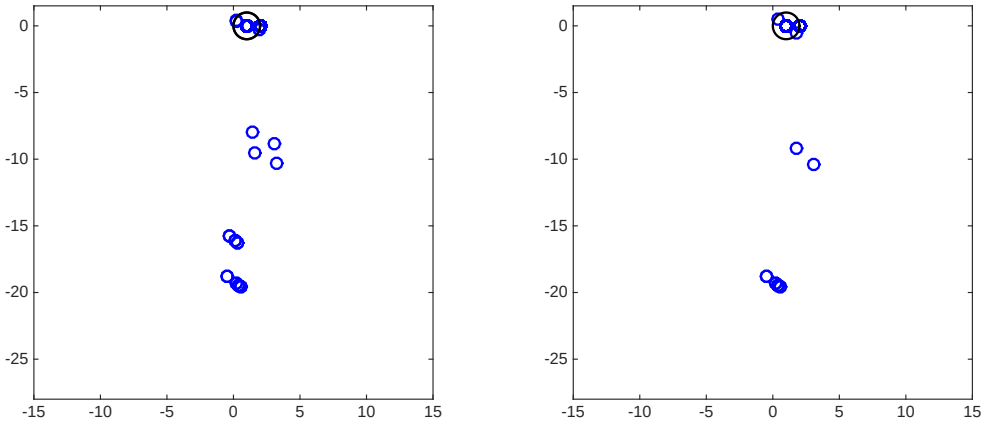
(a) $N_{\text{sub}} = 4$ (b) $N_{\text{sub}} = 8$

Figure 5.9 – Influence of the number of subdomains N_{sub} on the spectrum of the OAS-preconditioned matrix for $k = 2$, $\omega = \omega_2$, $\delta_{\text{ovr}} = 2h$.



(a) $\delta_{\text{ovr}} = 2h$ (b) $\delta_{\text{ovr}} = 4h$

Figure 5.10 – Influence of the overlap size δ_{ovr} on the spectrum of the ORAS-preconditioned matrix for $k = 2$, $\omega = \omega_2$, $N_{\text{sub}} = 2$.



(a) $\delta_{\text{ovr}} = 2h$ (b) $\delta_{\text{ovr}} = 4h$

Figure 5.11 – Influence of the overlap size δ_{ovr} on the spectrum of the OAS-preconditioned matrix for $k = 2$, $\omega = \omega_2$, $N_{\text{sub}} = 2$.

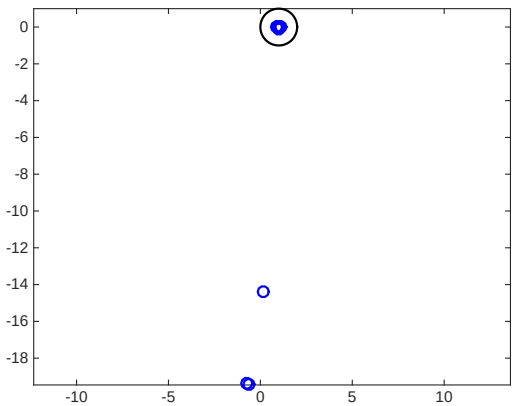


Figure 5.12 – The spectrum of the ORAS-preconditioned matrix for $k = 2$, $\omega = \omega_2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 1h$.

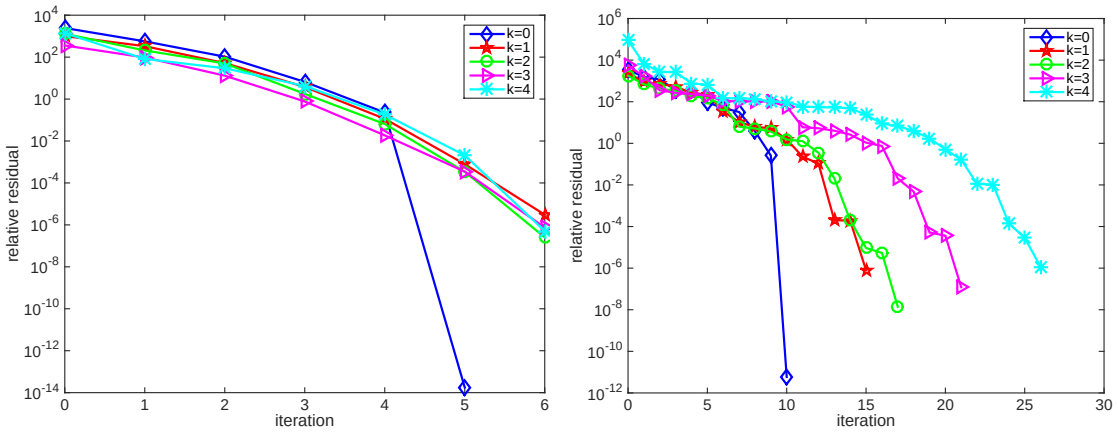


Figure 5.13 – Convergence history of GMRES preconditioned with ORAS (left) and OAS (right), for different polynomial degrees $r = k + 1$ ($\omega = \omega_2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 2h$).

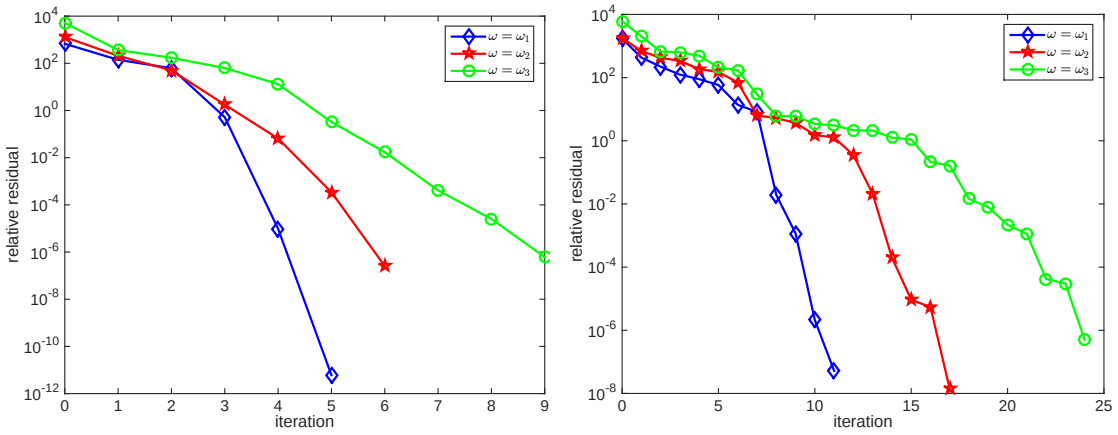


Figure 5.14 – Convergence history of GMRES preconditioned with ORAS (left) and OAS (right), for different angular frequencies ω ($k = 2$, $N_{\text{sub}} = 2$, $\delta_{\text{ovr}} = 2h$).

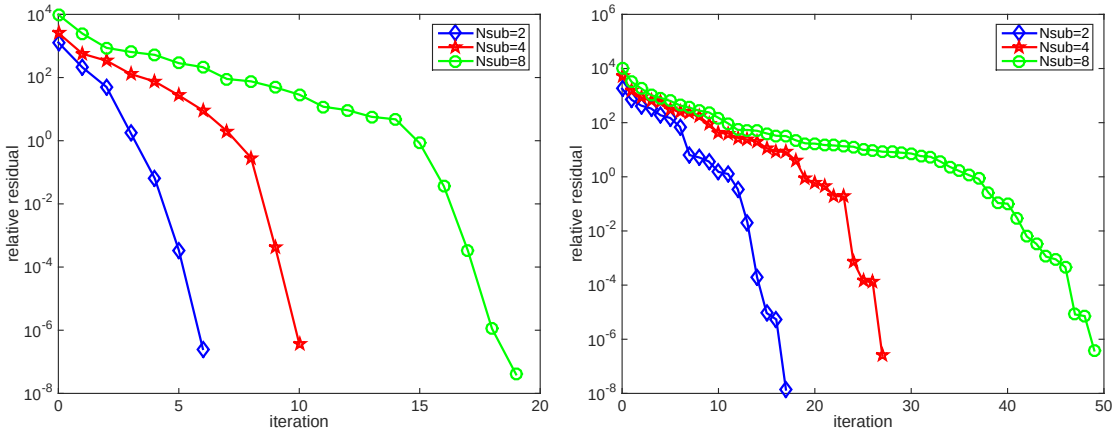


Figure 5.15 – Convergence history of GMRES preconditioned with ORAS (left) and OAS (right), for different numbers of subdomains N_{sub} ($k = 2$, $\omega = \omega_2$, $\delta_{\text{ovr}} = 2h$).

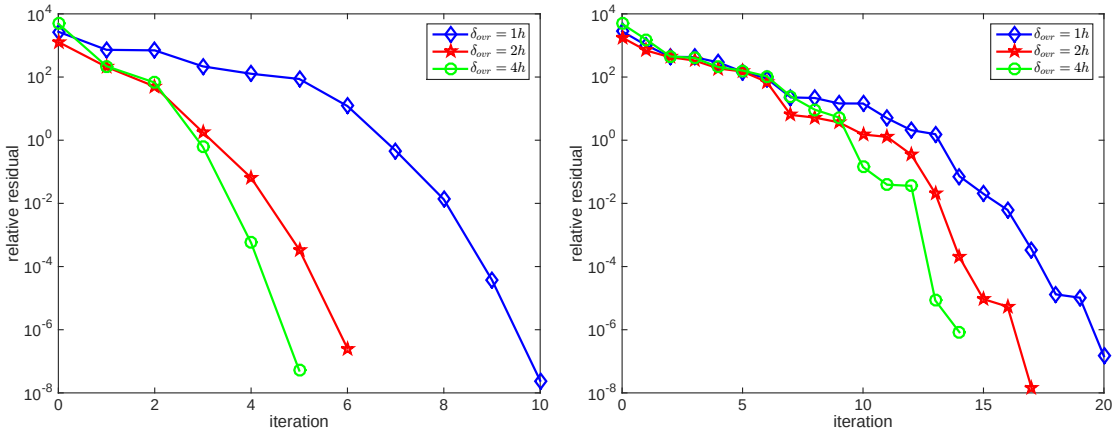


Figure 5.16 – Convergence history of GMRES preconditioned with ORAS (left) and OAS (right), for different overlap sizes δ_{ovr} ($k = 2$, $\omega = \omega_2$, $N_{\text{sub}} = 2$).

it appears necessary to add at least one layer from both subdomains. Then we see that for 4 and 8 subdomains the spectrum becomes less well clustered. In the convergence history plot in Figure 5.15 where N_{sub} varies, the typical plateaux appear, whose length is proportional to the number of subdomains in one direction. With the OAS preconditioner we can see that there are always eigenvalues outside the unit disk. For all the considered cases, we observe that the less clustered the spectrum, the slower the convergence.

5.4.2 Results for the three-dimensional problem

We complete the presentation showing some results for the full 3d simulation, for a waveguide as in Figure 2.2 of dimensions $a = 0.01016\,\text{m}$, $b = 0.00508\,\text{m}$, and $c = 0.1004\,\text{m}$. The physical parameters are: $\varepsilon = 8.85 \cdot 10^{-12}\,\text{F m}^{-1}$, $\mu = 1.26 \cdot 10^{-6}\,\text{H m}^{-1}$ and $\sigma = 0.15\,\text{S m}^{-1}$ or $\sigma = 0\,\text{S m}^{-1}$. We take a stripwise subdomains decomposition along the wave propagation, with $\delta_{\text{ovr}} = 2h$; however, note that in FreeFem++ very general subdomains decompositions can be considered, as the ones obtained with the automatic graph partitioners SCOTCH [94] or METIS [74] (see Chapter 6).

The data in the impedance boundary conditions (2.28c)–(2.28d) of the considered BVP are those given in Remark 2.4. Since the propagation constant in 3d is β and no more $\tilde{\omega}$, we compute the mesh size h using the relation $h^2 \cdot \beta^3 = 1$ (to avoid the pollution effect), taking $\beta = \omega_\beta \sqrt{\mu\varepsilon}$, with $\omega_\beta = 32\,\text{GHz}$. Then the dispersion relation (2.37) gives $\tilde{\omega} = \sqrt{\beta^2 + (m\pi/a)^2 + (n\pi/b)^2}$ (where we choose $m = 1, n = 0$), and we get $\omega = \tilde{\omega}/\sqrt{\mu\varepsilon}$.

Again, the linear system is solved with preconditioned GMRES, with a stopping criterion based on the relative residual and a tolerance of 10^{-6}, starting with a random initial guess. To apply the preconditioner, the local problems in each subdomain of matrices $A_{s,\text{Opt}}$ are solved with the direct solver MUMPS [2].

In Tables 5.5, 5.6 we show the number of iterations N_{iter} for convergence, for the problem with $\sigma = 0.15\,\text{S m}^{-1}$ and $\sigma = 0\,\text{S m}^{-1}$ respectively, varying first the polynomial degree $r = k + 1$ (for $N_{\text{sub}} = 2$), and then the number of subdomains N_{sub} (for $k = 1$). Like in the 2d case, the number of iterations using the ORAS preconditioner does not vary with the polynomial degree of the FE basis functions, while using the OAS preconditioner it varies and is much higher. Again, the convergence becomes slower when the number of subdomains increases, both with ORAS and OAS. We see that for more than 2 subdomains the number of iterations for the non dissipative problem ($\sigma = 0$) is higher than for the

k	N_{dofs}	N_{iter}	N_{sub}	N_{dofs}	N_{iter}
0	62283	8(40)	2	324654	8(70)
1	324654	8(70)	4	324654	11(106)
2	930969	8(99)	8	324654	17(168)

Table 5.5 – Results in 3d, $\sigma = 0.15\,\mathrm{S\,m^{-1}}$: influence of the polynomial degree $r = k + 1$ (for $N_{\text{sub}} = 2$), and of the number of subdomains N_{sub} (for $k = 1$), on the convergence of ORAS(OAS) preconditioner ($\beta = \omega_\beta\sqrt{\mu\varepsilon}$ with $\omega_\beta = 32\,\mathrm{GHz}$, $\delta_{\text{ovr}} = 2h$).

k	N_{dofs}	N_{iter}	N_{sub}	N_{dofs}	N_{iter}
0	62283	7(40)	2	324654	8(67)
1	324654	8(67)	4	324654	13(114)
2	930969	8(97)	8	324654	23(201)

Table 5.6 – Results in 3d, $\sigma = 0\,\mathrm{S\,m^{-1}}$: influence of the polynomial degree $r = k + 1$ (for $N_{\text{sub}} = 2$), and of the number of subdomains N_{sub} (for $k = 1$), on the convergence of ORAS(OAS) preconditioner ($\beta = \omega_\beta\sqrt{\mu\varepsilon}$ with $\omega_\beta = 32\,\mathrm{GHz}$, $\delta_{\text{ovr}} = 2h$).

problem with $\sigma = 0.15\,\mathrm{S\,m^{-1}}$.

In Figure 5.17 we plot the norm of the real part of the solution, which decreases as the wave propagates since there $\sigma = 0.15\,\mathrm{S\,m^{-1}}$ is different from zero. See Appendix A for the FreeFem++ scripts giving these results.

5.5 Conclusion

Numerical experiments have shown that Schwarz preconditioning significantly improves GMRES convergence for different values of physical and numerical parameters, and that the ORAS preconditioner always performs much better than the OAS preconditioner. Indeed, the only advantage of the OAS method is to preserve symmetry for symmetric problems: that is why it should be used only for symmetric positive definite matrices as a preconditioner for the conjugate gradient method. Moreover, in all the considered test cases, the number of iterations for convergence using the ORAS preconditioner does not vary when the polynomial degree of the adopted high order finite elements increases. We have also seen that it is necessary to take an overlap of at least one layer of simplices from *both* subdomains of a neighbors pair. All these convergence qualities are reflected by the spectrum of the preconditioned matrix.

For higher order discretizations the computational cost per iteration grows since matrices become very large, therefore a parallel implementation as the one of HPDDM [71] (a high-performance unified framework for domain decomposition methods which is interfaced with FreeFem++) is considered for large scale problems, as those arising from the application described in Chapter 6. A two-level preconditioner via a coarse space correction will be studied in Chapter 8 in order to fix the dependence on the number of subdomains or on the frequency of the iteration count.

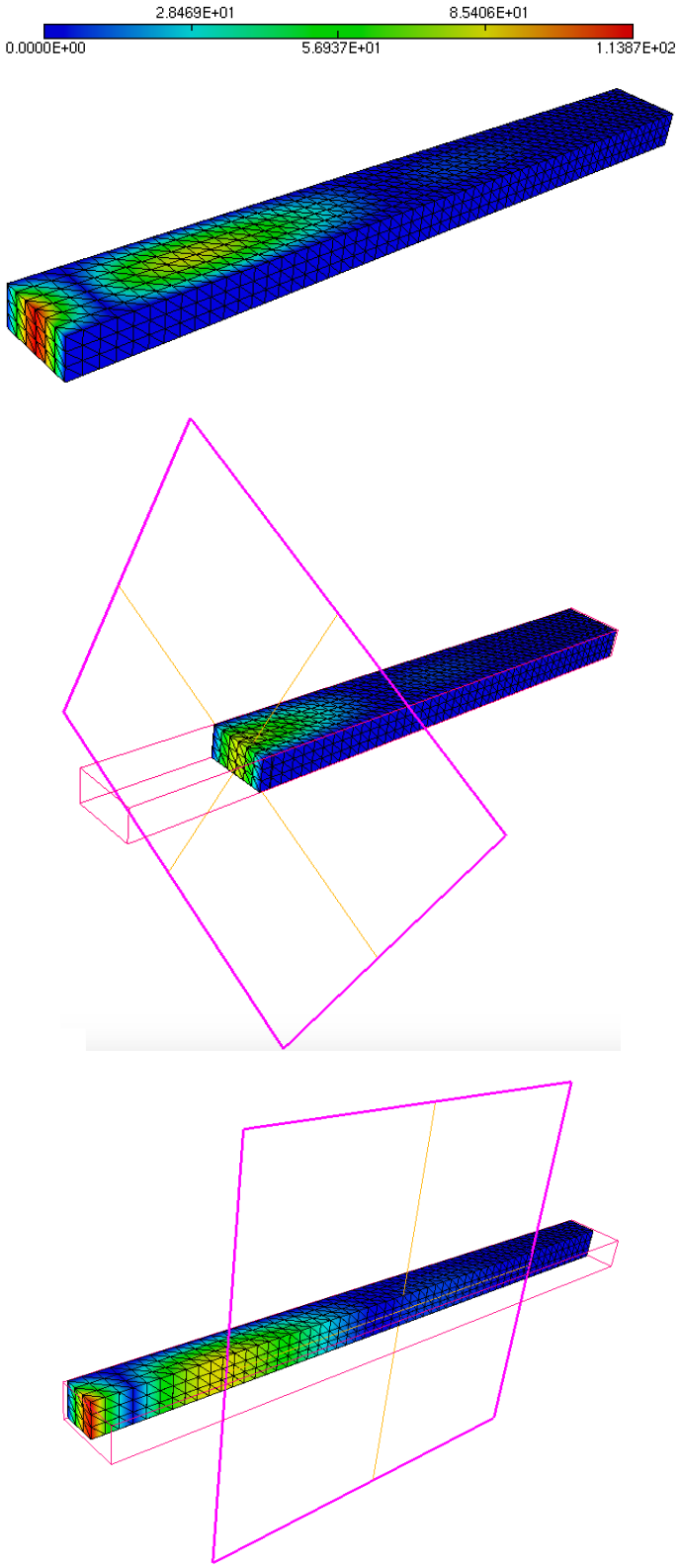


Figure 5.17 – The norm of the real part of the solution for $\sigma = 0.15\,\mathrm{S\,m^{-1}}$, with two sections of the waveguide.

Chapter 6

Application to brain microwave imaging

We have merged the following contributions into this Chapter:

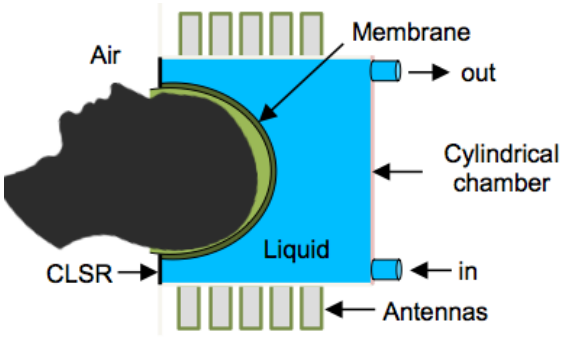
- [12] in collaboration with Victorita Dolean, Francesca Rapetti, Pierre-Henri Tournier, published in *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*;
- the submitted paper [112] *Microwave Tomographic Imaging of Cerebrovascular Accidents by Using High-Performance Computing*, in collaboration with Pierre-Henri Tournier, Iannis Aliferis, Maya de Buhan, Marion Darbas, Victorita Dolean, Frédéric Hecht, Pierre Jolivet, Ibtissam El Kanfoud, Claire Migliaccio, Frédéric Nataf, Christian Pichot, Serguei Semenov, available on arXiv and HAL (<hal-01343687>).

The related work [113] has been accepted for publication as an invited paper in the *Special issue on "Electromagnetic Inverse Problems for Sensing and Imaging"* of *IEEE Antennas and Propagation Magazine*, in collaboration with Pierre-Henri Tournier, Iannis Aliferis, Maya de Buhan, Marion Darbas, Victorita Dolean, Frédéric Hecht, Ibtissam El Kanfoud, Claire Migliaccio, Frédéric Nataf, Christian Pichot, Francesca Rapetti, Serguei Semenov.

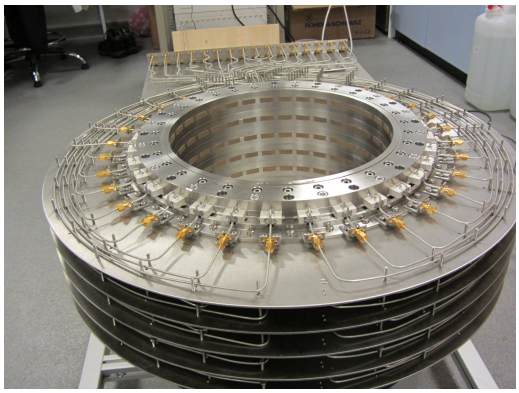
Contents

6.1 Introduction

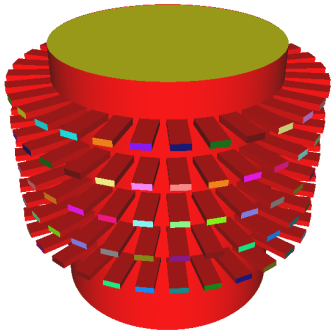
We apply the methods studied in the previous Chapters to simulate the microwave imaging system prototype shown in Figure 6.1, developed by the medical imaging company EMTen-



(a) Operating principle.



(b) Imaging system prototype.



(c) Computational domain.

Figure 6.1 – The microwave imaging system prototype (courtesy of EMTensor GmbH) and the corresponding computational domain.

sor GmbH for the detection and diagnosis of brain strokes. In the following, we describe the medical context as well as the application motivating the numerical simulation.

Strokes (also referred to as cerebrovascular accidents, CVA) can be classified into two major categories, *ischemic* (80% of strokes) and *hemorrhagic* (20% of strokes). During an ischemic stroke the blood supply to a part of the brain is interrupted by the formation of a blood clot inside a vessel, while a hemorrhagic stroke occurs when a blood vessel bursts inside the brain. It is essential to determine the type of stroke in the shortest possible time in order to start the correct treatment, which is opposite in the two situations: in the first case the blood flow should be restored, while in the second one the blood pressure should be lowered. Note that it is vital to make a clear distinction between the two types of stroke before treating the patient: the treatment that suits an ischemic stroke would be fatal if applied to a hemorrhagic stroke and vice versa. Moreover, it is desirable to be able to monitor continuously the effect of the treatment on the evolution of the stroke during the hospitalization. Strokes can be detected by estimating the variable complex-valued electric permittivity ε_σ (2.24) of the brain tissues of the patient. Indeed, ischemic and hemorrhagic strokes result in *opposite variations* of the real and imaginary parts of ε_σ in a region of the brain (more precisely, in a reduction, respectively increase, of about 10% of the baseline tissue values for ischemic, respectively hemorrhagic, strokes). An image here consists in a map of the electric permittivity at different points of the brain.

Usually stroke diagnosis relies mainly on two types of imaging techniques: MRI (magnetic resonance imaging) and CT scan (computerized tomography scan). These are very precise techniques, especially the MRI with a spatial resolution of 1 mm. However, a MRI

Figure 6.2 – A representation of the diagnosis technology (courtesy of EMTensor GmbH).

machine is too big to be carried in ambulance vehicles and it is also too expensive; a CT scan, which consists in measuring the absorption of X-rays by the brain, is harmful and thus cannot be used to monitor continuously the patient in hospital.

A novel competitive technique with these traditional imaging modalities is microwave tomography. With *microwave imaging*, at frequencies of the order of 1 GHz, the tissues are well differentiated and they can be imaged on the basis of their dielectric properties. The electromagnetic emissions are lower than the ones from mobile phones and the spatial resolution is good. The first works on microwave imaging date back to 1982, when Lin and Clarke [79] tested experimentally the detection of cerebral edema (an excessive accumulation of water in the brain) using a signal of frequency 2.4 GHz in a head phantom. Other works followed, almost always on phantoms or synthetic simplified models [103]. Despite these encouraging results, there is still no microwave device for medical diagnosis. The new techniques designed by the University of Chalmers (Gothenburg, Sweden) [95] and by EMTensor GmbH [104] rely on technologies and softwares developed only in recent years: in both cases the improvement in terms of reliability, price and miniaturization of electromagnetic sensors is a key factor. In the approach of EMTensor GmbH, the microwave measurement system is lightweight and can be carried in ambulance vehicles. The acquired measurements are transferred wirelessly to a remote computing center, where a HPC (High-Performance Computing) machine computes the images of the patient's brain. Once obtained, these images can be quickly transmitted from the computing center to the hospital, see Figure 6.2.

The simulation results presented in this work have been obtained on the microwave *imaging system* prototype shown in Figure 6.1, developed by EMTensor GmbH. It is composed of 5 rings of 32 antennas around a metallic cylindrical chamber of diameter 28.5 cm and total height 28 cm, into which the patient's head is inserted. The antennas are ceramic-loaded rectangular waveguides. The metallic chamber is filled with a matching solution and a membrane is used to isolate the head. Each of the 160 antennas *alternately* transmits a signal at a fixed frequency, typically 1 GHz. The electromagnetic wave propagates inside the chamber and in the object to be imaged according to its electromagnetic properties. The retrieved data then consist in the reflection and transmission coefficients measured by the 160 receiving antennas (see Section 6.2.2).

These coefficients are used as input for the *inverse problem* associated with the time-harmonic Maxwell's equations: the unknown of the inverse problem is the complex-valued electric permittivity ε_σ at the points of the computational domain, knowing the measured reflection and transmission coefficients. The solution of the inverse problem requires to solve repeatedly the *direct (or forward) problem* of the time-harmonic Maxwell's equations, i.e. the problem in which ε_σ is given and one wants to determine the electric field and then the reflection and transmission coefficients. Therefore an accurate and fast solver

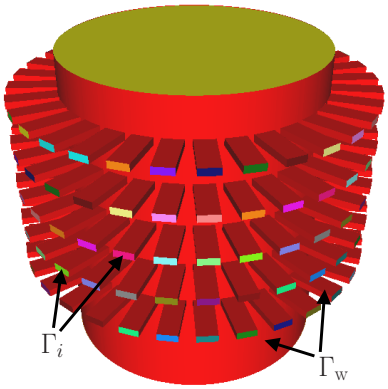


Figure 6.3 – The computational domain with the (red) metallic boundary Γ_w and the ports Γ_i, $i = 1, \ldots, 160$, of the waveguides, on which impedance boundary conditions are imposed.

for the direct problem is needed. Accuracy is provided by the high order edge finite elements defined in Chapters 3 and 4; the linear system resulting from this discretization is then solved efficiently in parallel with GMRES preconditioned with the Schwarz domain decomposition methods presented in Chapter 5.

This thesis is focused on the direct problem and the Chapter is organized as follows. In Section 6.2 the boundary value problems that model the EMTensor imaging system are first described (there is one boundary value problem for each transmitting antenna), then we explain how to compute the reflection and transmission coefficients (which are the measurable quantities), and finally, for the sake of completeness, the inverse problem is presented. Section 6.3 is dedicated to numerical results for the direct problem. In order to validate our numerical modeling, we first compare the coefficients given by the simulation with the measured values obtained by EMTensor (for the simple configuration where the chamber contains just the matching solution). Then we demonstrate the advantage, in terms of accuracy and computing time, of using high order edge finite elements compared to the standard lowest order edge elements. In the last part of the numerical results section, we perform a strong scaling analysis to assess the efficiency of the parallel domain decomposition preconditioner. Finally, we conclude in Section 6.4, giving also a brief account of the results obtained by the ANR MEDIMAX team for the inverse problem, showing the feasibility of this microwave imaging technique for detection and monitoring of brain strokes.

6.2 Mathematical model

6.2.1 The direct problem

Consider the second order time-harmonic Maxwell's equation (2.26) in the computational domain $\Omega \subset \mathbb{R}^3$ shown in Figure 6.3, with variable complex-valued electric permittivity $\varepsilon_\sigma(\mathbf{x}) = \varepsilon(\mathbf{x}) - \mathtt{i}\,\sigma(\mathbf{x})/\omega$ at each point $\mathbf{x} \in \Omega$, and a constant magnetic permeability μ equal to the free space magnetic permeability μ_0. Note that the object inserted in the imaging chamber (the head in the complete application) does not need to be represented in the domain Ω, because it is described by the variable coefficient $\varepsilon_\sigma(\mathbf{x})$.

Since *alternately* each waveguide $j = 1, \ldots, 160$ transmits a signal, we denote by $\mathbf{E}^j$ the solution of the corresponding boundary value problem, which differs only in the boundary

conditions on the waveguides ports. Indeed, for each $j = 1, \dots, 160$, we solve the equation

$$\nabla \times \left(\nabla \times \mathbf{E}^j\right) - \gamma^2 \mathbf{E}^j = \mathbf{0}, \tag{6.1}$$

with metallic boundary conditions

$$\mathbf{E}^j \times \mathbf{n} = \mathbf{0} \text{ on } \Gamma_{\mathrm{w}}, \tag{6.2}$$

on the cylinder and waveguides walls Γ_{w}, and with the following impedance boundary conditions on the port Γ_j of the j-th waveguide, which *transmits* the signal, and on the ports Γ_i of the *receiving* waveguides, $i = 1, \dots, 160$, $i \neq j$:

$$(\nabla \times \mathbf{E}^j) \times \mathbf{n} + \mathtt{i}\beta \mathbf{n} \times (\mathbf{E}^j \times \mathbf{n}) = \mathbf{g}^j \text{ on } \Gamma_j, \tag{6.3}$$

$$(\nabla \times \mathbf{E}^j) \times \mathbf{n} + \mathtt{i}\beta \mathbf{n} \times (\mathbf{E}^j \times \mathbf{n}) = \mathbf{0} \quad \text{on } \Gamma_i \text{ , } i \neq j. \tag{6.4}$$

Here $\mathbf{n}$ is the unit outward normal to $\partial\Omega$ and $\beta > 0$ is the propagation wavenumber along the waveguides. Equation (6.3) imposes an incident wave which corresponds to the excitation of the TE_{10} fundamental mode $\mathbf{E}_0^j$ of the j-th waveguide, with $\mathbf{g}^j = (\nabla \times \mathbf{E}_0^j) \times \mathbf{n} + \mathtt{i}\beta \mathbf{n} \times (\mathbf{E}_0^j \times \mathbf{n})$. Equation (6.4) is an absorbing boundary condition on the outer port of the receiving waveguides $i = 1, \dots, 160$, with $i \neq j$. The bottom of the chamber is considered metallic, and we impose an impedance boundary condition on the top of the chamber. Therefore we end up with the following boundary value problem for each transmitting waveguide j:

$$\begin{cases} \nabla \times \left(\nabla \times \mathbf{E}^j\right) - \gamma^2 \mathbf{E}^j = \mathbf{0}, \text{ in } \Omega, \\ \mathbf{E}^j \times \mathbf{n} = \mathbf{0}, \text{ on } \Gamma_{\mathrm{w}}, \\ (\nabla \times \mathbf{E}^j) \times \mathbf{n} + \mathtt{i}\beta \mathbf{n} \times (\mathbf{E}^j \times \mathbf{n}) = \mathbf{g}^j, \text{ on } \Gamma_j, \\ (\nabla \times \mathbf{E}^j) \times \mathbf{n} + \mathtt{i}\beta \mathbf{n} \times (\mathbf{E}^j \times \mathbf{n}) = \mathbf{0}, \text{ on } \Gamma_i \text{ , } i \neq j. \end{cases} \tag{6.5}$$

To derive the variational formulation of (6.5) we proceed as in Section 2.2.5, and we obtain: find $\mathbf{E}^j \in V$ such that

$$\int_\Omega \left[(\nabla \times \mathbf{E}^j) \cdot (\nabla \times \overline{\mathbf{v}}) - \gamma^2 \mathbf{E}^j \cdot \overline{\mathbf{v}} \right] + \int_{\bigcup_{i=1}^{160} \Gamma_i} \mathtt{i}\beta (\mathbf{E}^j \times \mathbf{n}) \cdot (\overline{\mathbf{v}} \times \mathbf{n}) = \int_{\Gamma_j} \mathbf{g}^j \cdot \overline{\mathbf{v}} \quad \forall \mathbf{v} \in V,$$

with $V = \{\mathbf{v} \in H(\mathrm{curl}, \Omega), \mathbf{v} \times \mathbf{n} = \mathbf{0} \text{ on } \Gamma_{\mathrm{w}}\}$. After discretizing these variational problems with the finite element method, we obtain 160 linear systems, one for each transmitting waveguide j and differing only in the right-hand side:

$$A\mathbf{u}^j = \mathbf{b}^j. \tag{6.6}$$

For the discretization we use the high order edge finite elements defined in Chapters 3–4. Direct solvers are not suited for such large linear systems arising from complex three dimensional models because of their high memory cost. Therefore, we use an iterative solver (GMRES), which, on the other hand, is not robust and requires preconditioning. Domain decomposition preconditioners are naturally suited to parallel computing and make it possible to deal with smaller subproblems, on which direct solvers are applicable. Here we use the Optimized Restricted Additive Schwarz preconditioner (ORAS) in equation (5.9).

6.2.2 Reflection and transmission coefficients

The physical quantity that can be acquired by the measurement system of the imaging system shown in Figures 6.1–6.3 is the *scattering matrix* (also referred to as S matrix), which gathers the complex-valued *reflection and transmission coefficients* measured by the 160 receiving antennas for a signal transmitted by one of these 160 antennas successively. A set of measurements then consists in a complex matrix of size 160×160. In order to compute the numerical counterparts of these reflection and transmission coefficients, we use the following formula, which is appropriate in the case of open-ended waveguides:

$$S_{ij} = \frac{\int_{\Gamma_i} \overline{\mathbf{E}^j} \cdot \mathbf{E}_0^i}{\int_{\Gamma_i} |\mathbf{E}_0^i|^2}, \quad i,j = 1,\dots,160, \tag{6.7}$$

where $\mathbf{E}^j$ is the solution of the boundary value problem (6.5) where the j-th waveguide transmits the signal, and $\mathbf{E}_0^i$ is the TE_{10} fundamental mode of the i-th receiving waveguide ($\overline{\mathbf{E}^j}$ denotes the complex conjugate of $\mathbf{E}^j$). The S_{ij} with $i \neq j$ are the transmission coefficients, and the S_{jj} are the reflection coefficients.

6.2.3 The inverse problem

Even if this thesis is focused on the direct problem, we present now the inverse problem, in order to provide a complete description of the mathematical model for the imaging application, as well as to clearly show where the solution of the direct problem intervenes in the inversion tool.

The inverse problem that we consider consists in finding the unknown complex-valued electric permittivity $\varepsilon_\sigma(\mathbf{x}) = \varepsilon(\mathbf{x}) - \mathtt{i}\,\sigma(\mathbf{x})/\omega$ in Ω, such that the solutions $\mathbf{E}^j, j = 1,\dots,N$ of problem (6.5) lead to corresponding scattering coefficients S_{ij} (6.7) that coincide with the measured scattering coefficients S_{ij}^{mes}, for $i,j = 1,\dots,N$. Here N denotes the number of antennas.

Let $\kappa := \gamma^2 = \omega^2\mu\varepsilon_\sigma$ be the unknown complex parameter of our inverse problem. Let us denote by $\mathbf{E}^j(\kappa)$ the solution of the direct problem (6.5) with complex electric permittivity ε_σ. The corresponding scattering coefficients will be denoted by $S_{ij}(\kappa)$:

$$S_{ij}(\kappa) = \frac{\displaystyle\int_{\Gamma_i} \overline{\mathbf{E}^j(\kappa)} \cdot \mathbf{E}_0^i}{\displaystyle\int_{\Gamma_i} |\mathbf{E}_0^i|^2}, \quad i,j = 1,\dots,N.$$

The misfit of the parameter κ to the data can be defined through the following functional:

$$J(\kappa) = \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{N} \left|S_{ij}(\kappa) - S_{ij}^{\mathrm{mes}}\right|^2 = \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{N} \left| \frac{\displaystyle\int_{\Gamma_i} \overline{\mathbf{E}^j(\kappa)} \cdot \mathbf{E}_0^i}{\displaystyle\int_{\Gamma_i} |\mathbf{E}_0^i|^2} - S_{ij}^{\mathrm{mes}} \right|^2. \tag{6.8}$$

In a classical way, solving the inverse problem consists in minimizing the functional J with respect to the parameter κ. Computing the differential of J in a given arbitrary direction $\delta\kappa$ yields:

$$DJ(\kappa,\delta\kappa) = \sum_{j=1}^{N}\sum_{i=1}^{N} \mathrm{Re}\left[\overline{\left(S_{ij}(\kappa) - S_{ij}^{\mathrm{mes}}\right)} \frac{\displaystyle\int_{\Gamma_i} \overline{\delta\mathbf{E}^j(\kappa)} \cdot \mathbf{E}_0^i}{\displaystyle\int_{\Gamma_i} |\mathbf{E}_0^i|^2} \right], \quad \delta\kappa \in \mathbb{C},$$

where $\delta\mathbf{E}^j(\kappa)$ is the solution of the following linearized problem:

$$\begin{cases} \nabla\times(\nabla\times\delta\mathbf{E}^j)-\kappa\delta\mathbf{E}^j=\delta\kappa\mathbf{E}^j & \text{in } \Omega,\\ \delta\mathbf{E}^j\times\mathbf{n}=0 & \text{on } \Gamma_{\mathrm{w}},\\ (\nabla\times\delta\mathbf{E}^j)\times\mathbf{n}+\mathrm{i}\beta\mathbf{n}\times(\delta\mathbf{E}^j\times\mathbf{n})=0 & \text{on } \Gamma_i\ ,\ i=1,\dots,N. \end{cases} \tag{6.9}$$

We now use the adjoint approach in order to simplify the expression of DJ. This will allow us to compute the gradient efficiently after discretization, with a number of computations independent of the size of the parameter space. Considering the variational formulation of problem (6.9) with a test function $\mathbf{F}$ and integrating by parts, we get

$$\begin{aligned} \int_\Omega \delta\kappa\mathbf{E}^j\cdot\mathbf{F} &= \int_\Omega \big(\nabla\times(\nabla\times\delta\mathbf{E}^j)-\kappa\delta\mathbf{E}^j\big)\cdot\mathbf{F}\\ &= \int_\Omega (\nabla\times(\nabla\times\mathbf{F})-\kappa\mathbf{F})\cdot\delta\mathbf{E}^j-\int_{\partial\Omega}((\nabla\times\delta\mathbf{E}^j)\times\mathbf{n})\cdot\mathbf{F}\\ &\quad+\int_{\partial\Omega}((\nabla\times\mathbf{F})\times\mathbf{n})\cdot\delta\mathbf{E}^j\\ &= \int_\Omega (\nabla\times(\nabla\times\mathbf{F})-\kappa\mathbf{F})\cdot\delta\mathbf{E}^j+\sum_{i=1}^N\int_{\Gamma_i}\mathrm{i}\beta(\mathbf{n}\times(\mathbf{F}\times\mathbf{n}))\cdot\delta\mathbf{E}^j\\ &\quad+\int_{\Gamma_{\mathrm{w}}}(\nabla\times\delta\mathbf{E}^j)\cdot(\mathbf{F}\times\mathbf{n})+\sum_{i=1}^N\int_{\Gamma_i}((\nabla\times\mathbf{F})\times\mathbf{n})\cdot\delta\mathbf{E}^j. \end{aligned}$$

Introducing the solution $\mathbf{F}^j(\kappa)$ of the following adjoint problem

$$\begin{cases} \nabla\times(\nabla\times\mathbf{F}^j)-\kappa\mathbf{F}^j=0 & \text{in } \Omega,\\ \mathbf{F}^j\times\mathbf{n}=0 & \text{on } \Gamma_{\mathrm{w}},\\ (\nabla\times\mathbf{F}^j)\times\mathbf{n}+\mathrm{i}\beta\mathbf{n}\times(\mathbf{F}^j\times\mathbf{n})=\dfrac{(S_{ij}(\kappa)-S_{ij}^{\mathrm{mes}})}{\displaystyle\int_{\Gamma_i}|\mathbf{E}_0^i|^2}\overline{\mathbf{E}_0^i} & \text{on } \Gamma_i\ ,\ i=1,\dots,N, \end{cases} \tag{6.10}$$

we get

$$\int_\Omega \delta\kappa\mathbf{E}^j\cdot\mathbf{F}^j=\sum_{i=1}^N(S_{ij}(\kappa)-S_{ij}^{\mathrm{mes}})\frac{\displaystyle\int_{\Gamma_i}\overline{\mathbf{E}_0^i}\cdot\delta\mathbf{E}^j}{\displaystyle\int_{\Gamma_i}|\mathbf{E}_0^i|^2}.$$

Finally, the differential of J can be computed as

$$DJ(\kappa,\delta\kappa)=\sum_{j=1}^N\mathrm{Re}\left[\int_\Omega\overline{\delta\kappa\,\mathbf{E}^j\cdot\mathbf{F}^j}\right].$$

We can then compute the gradient to use in a gradient-based local optimization algorithm. The images in Figure 6.10 are obtained using a limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm. Note that every evaluation of J requires the solution of the direct (or state) problem (6.5), while the computation of the gradient requires the solution of (6.5) as well as the solution of the adjoint problem (6.10). Moreover, the state and adjoint problems use the same operator. Therefore, the computation of the gradient only needs the assembly of one matrix and its associated domain decomposition preconditioner.

The functional J considered in the numerical simulation of the inverse problem is slightly different from (6.8), as we add a normalization term for each pair (i, j) as well as a Tikhonov regularizing term:

$$J(\kappa) = \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \frac{\left|S_{ij}(\kappa) - S_{ij}^{\text{mes}}\right|^2}{\left|S_{ij}^{\text{empty}}\right|^2} + \frac{\alpha}{2} \int_{\Omega} |\nabla \kappa|^2, \tag{6.11}$$

where S_{ij}^{empty} refers to the coefficients computed from the simulation of the chamber filled only with the homogeneous matching solution. In this way, the contribution of each pair (i, j) in the misfit functional is normalized and does not depend on the amplitude of the coefficient, which can greatly vary between pairs (i, j) as displayed in Figure 6.5. The Tikhonov regularizing term aims at reducing the effects of noise in the data. For now, the regularization parameter α is chosen empirically so as to obtain a visually good compromise between reducing the effects of noise and keeping the reconstructed image pertinent. All calculations carried out in this section can be accommodated in a straightforward manner to definition (6.11) of the functional.

In tomographic imaging the reconstruction is done layer by layer. For the imaging chamber of EMTensor considered here, one layer corresponds to one of the five rings of 32 antennas, and we solve an inverse problem independently for each of the five rings. More precisely, each of these inverse problems is solved in a domain truncated around the corresponding ring of antennas, containing at most two other rings (one ring above and one ring below). We impose absorbing boundary conditions on the artificial boundaries of the truncated computational domain. For each inverse problem, only the coefficients S_{ij} with transmitting antennas j in the corresponding ring are taken into account: we consider 32 antennas as transmitters and at most 96 antennas as receivers.

6.3 Numerical results

The linear systems (6.6), resulting from the high order edge finite elements discretizations, are solved by GMRES preconditioned with the ORAS preconditioner (5.9), as implemented in HPDDM [71] (`https://github.com/hpddm/hpddm`), a High-Performance unified framework for Domain Decomposition Methods. Domain decomposition methods naturally offer good parallel properties on distributed architectures. The computational domain is decomposed into subdomains in which concurrent computations are performed. The coupling between subdomains requires communications between computing nodes, which are based on the Message Passing Interface (MPI) in HPDDM. The assembly of the preconditioner involves the concurrent factorization of the local matrices $A_{s,\text{Opt}}$, which are stored on different processes in the distributed computing context. Likewise, applying the preconditioner to a distributed vector only requires peer-to-peer communications between neighboring subdomains, and a local forward elimination and backward substitution. See Chapter 8 of [41] for more details about the parallel implementation.

We solve for multiple right-hand sides (corresponding to the different transmitting antennas) simultaneously using a pseudo-block method implemented inside GMRES: this consists in fusing the multiple arithmetic operations associated with each right-hand side (matrix-vector products, dot products), in order to achieve higher arithmetic intensity (see [72, §V.B.1] for more details). The GMRES algorithm is stopped once the relative residual is lower than 10^{-8}.

All the simulations are performed in FreeFem++, which is interfaced with HPDDM. The overlapping decomposition into subdomains of the domain Ω is obtained by partitioning the global mesh into non-overlapping submeshes with the automatic graph partitioner

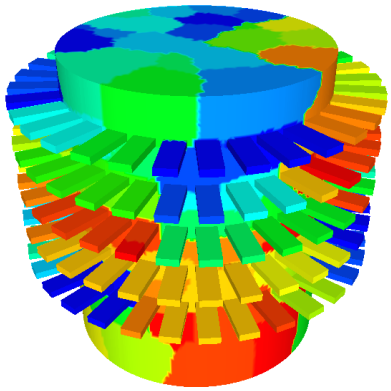


Figure 6.4 – The decomposition of the computational domain into 128 subdomains.

SCOTCH [94], and by adding then layers of adjacent tetrahedra. The resulting decomposition is quite general, like the one in Figure 6.4, and interfaces between subdomains are rough. The construction of the partition of unity matrices appearing in preconditioner (5.9) for (high order) edge finite elements is described in Paragraph 5.3.1.

In the following subsections, we first validate our numerical modeling of the imaging system prototype by comparing the results of the simulation with experimental measurements obtained by EMTensor. Then, we illustrate the superiority of the high order finite elements presented in Chapters 3–4 over the classical lowest order ones in terms of running time and accuracy. Finally, we perform a strong scaling analysis in order to assess the efficiency of preconditioner (5.9). Results were obtained on the Curie supercomputer (at TGCC-CEA, `http://www-hpc.cea.fr/en/complexe/tgcc-curie.htm`).

6.3.1 Comparison with experimental measurements

Recall that the measurable quantity is not the electric field but the reflection and transmission coefficients S_{ij} defined in Section 6.2.2. In this Section we compare the coefficients (6.7) given by the simulation of the direct problem with the measured values obtained by EMTensor, for a configuration in which the imaging chamber is filled with a homogenous matching solution. The electric permittivity ε of the matching solution is chosen by EMTensor in order to minimize contrasts with the ceramic-loaded waveguides and with the different brain tissues. The choice of the conductivity σ of the matching solution is a compromise between the minimization of reflection artifacts from metallic boundaries and the desire to have best possible signal-to-noise ratio. Here the relative complex permittivity of the matching solution at frequency $f = 1\,\mathrm{GHz}$ is $\varepsilon_r^{\mathrm{gel}} = 44 - 20\mathtt{i}$. The relative complex permittivity inside the ceramic-loaded waveguides is $\varepsilon_r^{\mathrm{cer}} = 59 - 0\mathtt{i}$. Here with ε_r we mean the ratio between the complex permittivity $\varepsilon_\sigma = \varepsilon - \mathtt{i}\,\sigma/\omega$ and the permittivity of free space ε_0.

For this test case, the set of experimental data given by EMTensor consists in transmission coefficients for transmitting antennas in the second ring from the top. Figure 6.5 shows the normalized magnitude (dB) and phase (degree) of the complex-valued coefficients S_{ij} corresponding to a transmitting antenna in the second ring from the top and to the 31 receiving antennas in the middle ring (note that the measured coefficients are available only for 17 receiving antennas). The magnitude in dB is calculated as $20\log_{10}(|S_{ij}|)$. The computed coefficients are obtained by solving the direct problem with edge finite elements of polynomial degree $r = 2$. We can see that the computed transmission coefficients are in very good agreement with the measurements.

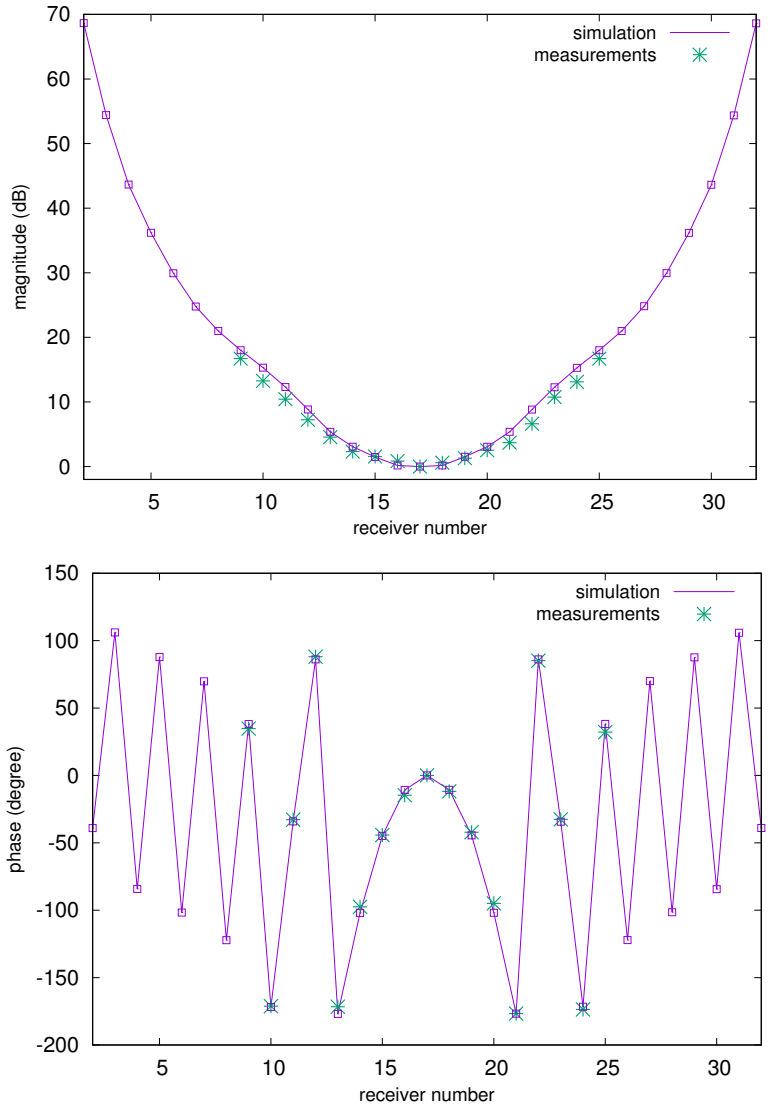


Figure 6.5 – The normalized magnitude (top) and phase (bottom) of the transmission coefficients computed with the simulation and measured experimentally.

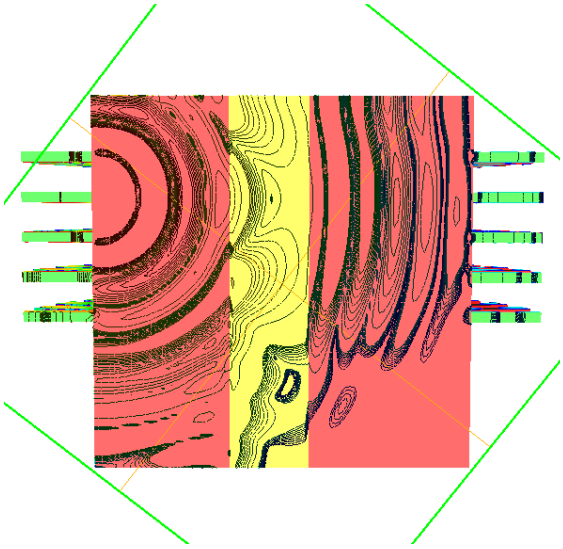


Figure 6.6 – Slice of the imaging chamber, showing the non-dissipative plastic-filled cylinder and some isolines of the norm of the real part of the total field $\mathbf{E}^j$.

6.3.2 Efficiency of high order finite elements

The goal of the following numerical experiments is to assess the efficiency of the high order finite elements described in Chapters 3–4 compared to the classical lowest order edge elements in terms of accuracy and computing time, which are of great importance for such an application in brain imaging.

For this test case, a non-dissipative plastic-filled cylinder of diameter 6 cm and relative permittivity $\varepsilon_r^{\text{cyl}} = 3$ is inserted in the imaging chamber and surrounded by matching solution of relative complex permittivity $\varepsilon_r^{\text{gel}} = 44 - 20\mathtt{i}$ (see Figure 6.6). We consider the 32 antennas of the second ring from the top as transmitting antennas at frequency $f = 1\,\text{GHz}$, and all 160 antennas are receiving. Slices in Figures 6.6 and 6.7 show the computational domain and the solution $\mathbf{E}^j$ for one transmitting antenna j in the second ring from the top.

We evaluate the *relative error* on the reflection and transmission coefficients S_{ij} with respect to the coefficients S_{ij}^{ref} computed from a reference solution. The relative error is calculated with the following formula:

$$E = \frac{\sqrt{\sum_{j,i} |S_{ij} - S_{ij}^{\text{ref}}|^2}}{\sqrt{\sum_{j,i} |S_{ij}^{\text{ref}}|^2}}. \tag{6.12}$$

The reference solution is computed on a fine mesh of approximately 18 million tetrahedra, which corresponds to 20 points per wavelength, and using edge finite elements of degree $r = 2$, resulting in 114 million unknowns.

We compare the computing time and the relative error (6.12) for different numbers of unknowns corresponding to several mesh sizes, for approximation degrees $r = 1$ and $r = 2$. All these simulations are done using 512 subdomains with one MPI process and two OpenMP threads per subdomain, for a total of 1024 cores on the Curie supercomputer. We summarize all the results in Table 6.1 and in Figure 6.8. In the plot, each bullet corresponds to one simulation with a certain mesh size and degree: we report next to each bullet the corresponding total number of unknowns, on the vertical axis the relative error, and on the horizontal axis the computing time. As we can see, the high order approximation ($r = 2$)

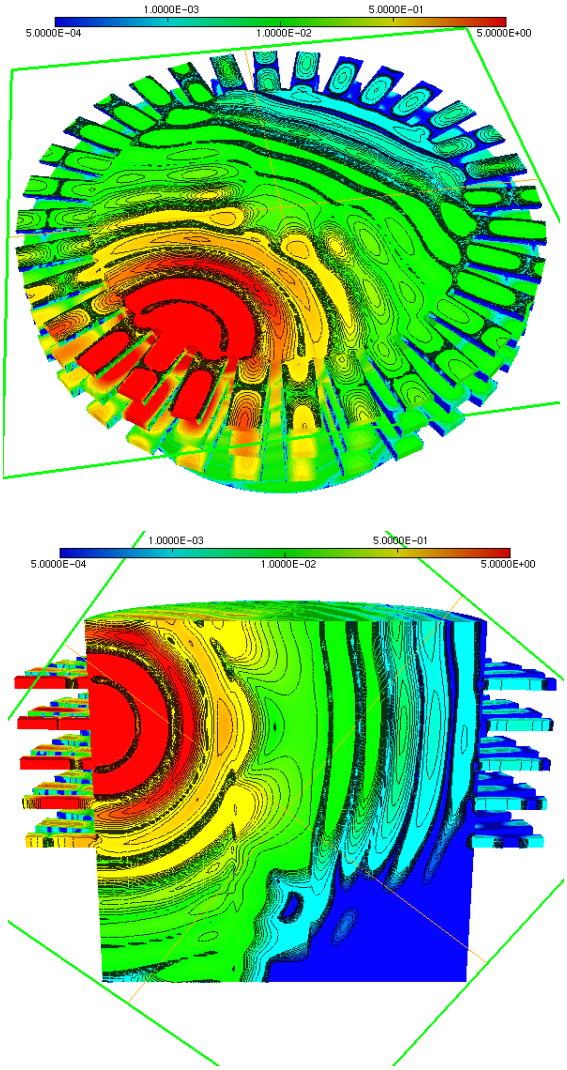


Figure 6.7 – Slices showing the norm of the real part of the total field $\mathbf{E}^j$ in the imaging chamber with the plastic-filled cylinder inside, for one transmitting antenna j in the second ring from the top.

Degree 1			Degree 2		
# unknowns	time (s)	error	# unknowns	time (s)	error
2 373 214	22	0.384	1 508 916	39	0.243
8 513 191	53	0.184	5 181 678	62	0.099
21 146 710	130	0.117	12 693 924	122	0.057
42 538 268	268	0.083	26 896 130	236	0.036
73 889 953	519	0.068	45 781 986	396	0.019

Table 6.1 – Total number of unknowns, time to solution (seconds) and relative error on the computed S_{ij} with respect to the reference solution for edge finite elements of degree 1 and 2 on different meshes.

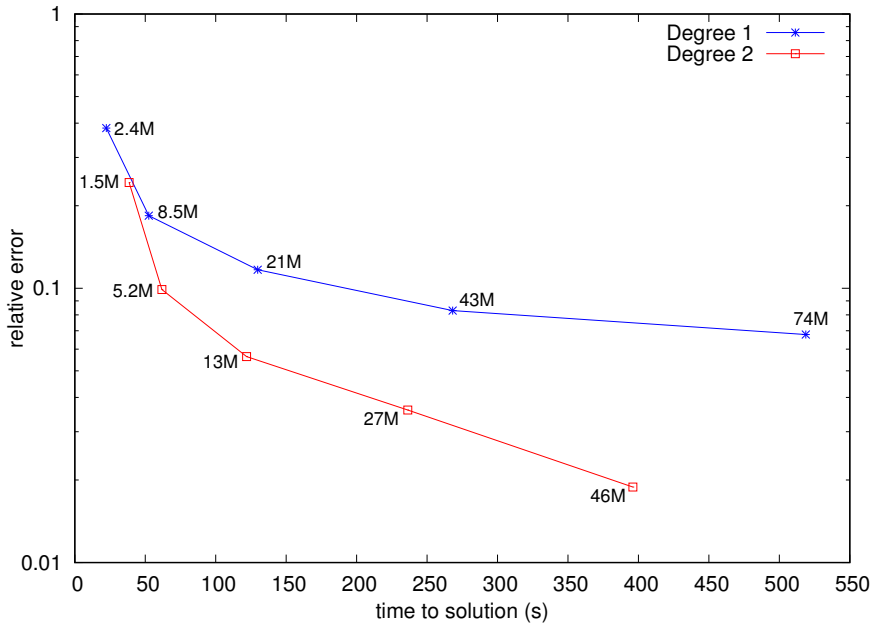


Figure 6.8 – Time to solution (seconds) and relative error on the computed S_{ij} with respect to the reference solution, using edge finite elements of degree 1 and degree 2 for different mesh sizes. The total number of unknowns in millions is also reported for each simulation.

allows to attain a given accuracy with much fewer unknowns and much less computing time than the lowest order approximation ($r = 1$). For example, at a given accuracy of $E \approx 0.1$, the finite element discretization of degree $r = 1$ requires 21 million unknowns and a computing time of 130 seconds, while the high order finite element discretization ($r = 2$) only needs 5 million unknowns, with a corresponding computing time of 62 seconds.

6.3.3 Strong scaling analysis

There are two common metrics to evaluate the performances of a parallel code: strong scaling and weak scaling. *Strong scaling* shows how a code performs when the number of processing units is increased for solving a fixed size global problem: ideally the elapsed time should be inversely proportional to the number of processing units. *Weak scaling* shows how a code behaves when the number of processing units increases, while maintaining local problems with constant size: ideally the elapsed time should be constant.

Here we consider the setting of Section 6.3.1, where the chamber is filled with a ho-

N_{sub}	Setup time	Solve time	N_{iter}	Speedup
256	293.36	73.06	43	1
512	95.11	36.92	53	2.8
1024	35.13	20.55	64	6.6
2048	25.89	12.77	81	9.5

Table 6.2 – Strong scaling experiment. For N_{sub} subdomains, timings (in seconds) of the setup and solution phases, number of iterations N_{iter}, and speedup.

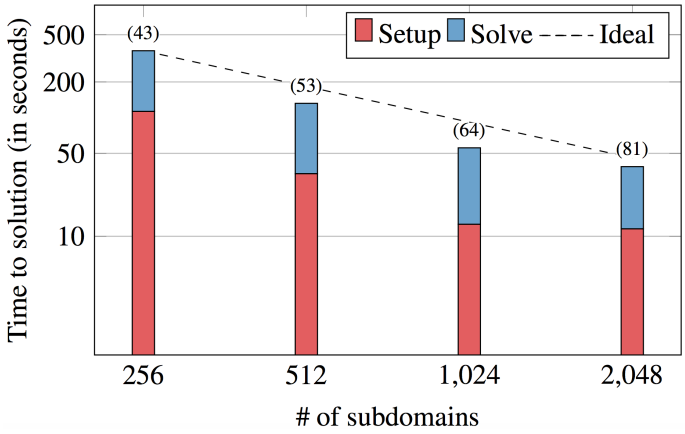


Figure 6.9 – Strong scaling experiment. Colors indicate the fraction of the total time spent in the setup and solution phases. The number of GMRES iterations is reported in parentheses.

mogeneous matching solution, and a transmitting antenna in the second ring from the top. For a strong scaling analysis, given a fine mesh of the domain composed of 82 million tetrahedra, we increase the number of MPI processes to solve a linear system of 96 million double-precision complex unknowns. The direct solver for the local problems in the subdomains is PARDISO [100] from Intel MKL. We use one subdomain and two OpenMP threads per MPI process.

Results are reported in Table 6.2 and illustrated in Figure 6.9, with a plot of the time to solution including both the setup and solution phases, on 256 up to 2048 subdomains. The setup time corresponds to the maximum time spent for the factorization of the local matrices in the preconditioner over all subdomains, while the solve time corresponds to the time needed to solve the linear system with GMRES. In the plot the number of GMRES iterations is reported in parentheses. In the table, the *speedup* for N_{sub} subdomains is the ratio between the total time with 256 subdomains and the total time with N_{sub} subdomains. Even if the number of iterations increases with the number of subdomains, we are able to obtain very good speedups up to 4096 cores (2048 subdomains) on Curie, with a superlinear speedup of 9.5 between 256 and 2048 subdomains (with 'superlinear' we mean that $9.5 > 8 = 2048/256$).

6.4 Conclusion

This work shows the benefits of using a discretization based on high order edge finite elements coupled with a parallel domain decomposition preconditioner, for the simulation of the EMTensor microwave imaging system in Figure 6.1. In such complex systems, accu-

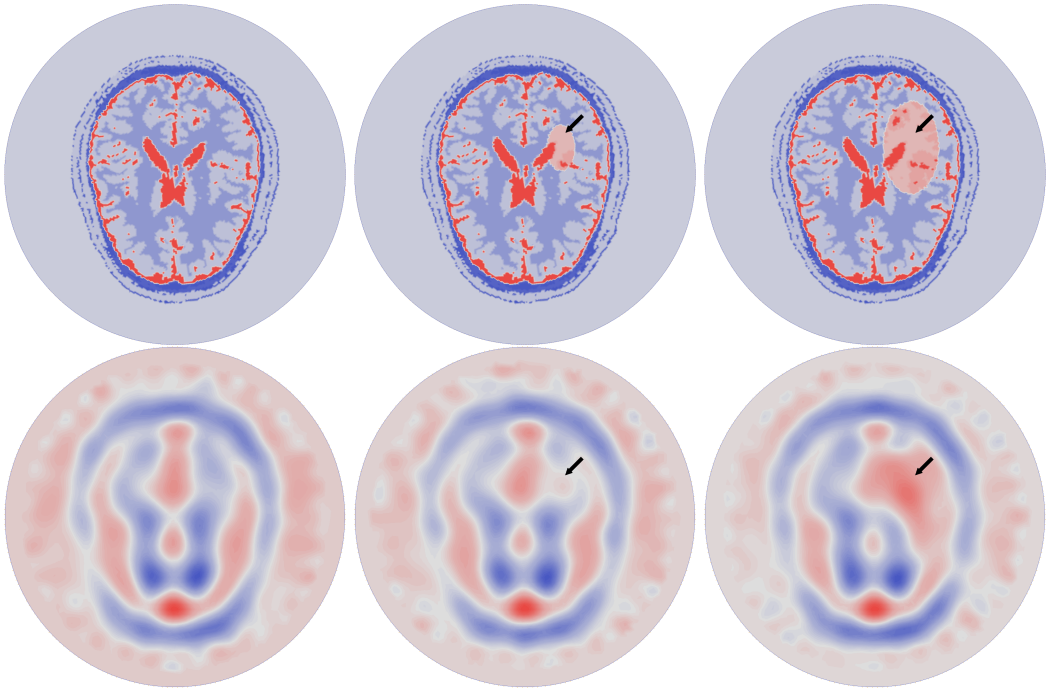


Figure 6.10 – Top row: imaginary part of the exact permittivity used to produce noisy data as input for the inverse problem, for three phases of the evolution of a simulated hemorrhagic stroke, from the healthy brain (left column) to a large stroke (right column). Bottom row: corresponding reconstructions obtained with the inversion tool developed by the ANR MEDIMAX team.

racy and computing speed are of paramount importance, especially for the application of diagnosis and monitoring of brain strokes. The high order approximation makes it possible to attain a given accuracy with much fewer unknowns and much less computing time than the lowest order approximation. The parallel implementation in HPDDM of the domain decomposition preconditioner is essential to be able to solve the arising linear systems of up to 96 million complex-valued unknowns considered here. Even if the number of iterations increases with the number of subdomains, the strong scaling experiment exhibits very good time speedups up to 2048 subdomains.

Here we studied efficient techniques for the solution of the *direct problem*, that have been applied in the *inversion tool* developed by the ANR MEDIMAX team (recall that the solution of the inverse problem requires repeated solves of the direct problem). The team thus managed to reconstruct a microwave tomographic image of the brain (the third one in Figure 6.10) in less than 2 minutes using 4096 cores. This computational time corresponds to clinician acceptance for rapid diagnosis or medical monitoring in hospital. Each reconstructed image in Figure 6.10 corresponds to the solution of an inverse problem with *noisy synthetic data* as input. More precisely, these data were obtained as follows: starting from a very accurate model of the complex-valued permittivity $\varepsilon_\sigma(\mathbf{x})$ of a healthy brain, a hemorrhagic stroke was simulated by adding an ellipsoid in which the value of $\varepsilon_\sigma(\mathbf{x})$ was increased; then for this $\varepsilon_\sigma(\mathbf{x})$ the direct problem was solved to compute the corresponding coefficients S_{ij} using (6.7) and finally a 10% noise was added to these synthetic coefficients. The evolution of the stroke was simulated by increasing the size of the ellipsoid. Although the reconstructed images do not feature the complex heterogeneities of the brain, which is in accordance with what we expect from microwave imaging methods, they allow the

characterization of the stroke and its monitoring. The next step would be the validation of the inversion tool on clinical measured data.

The domain decomposition preconditioner employed here is a one-level method, which cannot scale well beyond thousands of subdomains since the number of iterations deteriorates. The introduction of a *two-level preconditioner* with an adequate *coarse space* for Maxwell's equations would maintain very good speedups even for decompositions into a larger number of subdomains. This is a challenging open problem for equations yielding indefinite matrices, which is investigated in the next Chapters.

Chapter 7

Two-level preconditioners for the Helmholtz equation

This Chapter is based on [17], in collaboration with Victorita Dolean, Ivan G. Graham, Euan A. Spence, and Pierre-Henri Tournier, which has been submitted to the proceedings of the *DD24 International Conference on Domain Decomposition Methods*. A preprint is available on arXiv and HAL (<hal-01525424>).

Contents

7.1 Introduction

We have already pointed out in Section 5.1 that the time-harmonic Maxwell's equation (2.26) presents similar difficulties to those encountered with the (scalar) *Helmholtz equation*

$$-\Delta u - \tilde{\omega}^2 u = f \tag{7.1}$$

when the wavenumber $\tilde{\omega}$ is large, namely the sign-indefiniteness of their (standard) variational formulation, the pollution effect, and the consequent problematic construction of fast iterative solvers [50]. Although there have been different attempts to solve the high-frequency Helmholtz equation efficiently, we believe that there is no established and robust preconditioner, whose behavior is independent of $\tilde{\omega}$, for general decompositions into subdomains. Therefore, before facing the time-harmonic Maxwell's equation, in this Chapter we focus on the Helmholtz equation (7.1).

In order to achieve independence of the iteration count on the number of subdomains or, for wave propagation problems, on the wavenumber $\tilde{\omega}$, *two-level* domain decomposition

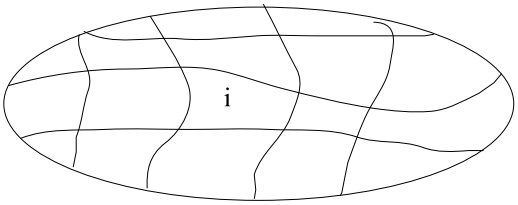


Figure 7.1 – Decomposition into several subdomains.

preconditioners are generally introduced: a so-called *coarse-grid correction* is combined with the one-level preconditioners (5.5), (5.6), (5.7) presented in Chapter 5. A two-level preconditioner is defined by two ingredients:

- an algebraic *formula* to combine the coarse grid correction with the one-level preconditioner (e.g. in a additive or in a hybrid way),
- a rectangular full column rank matrix Z, whose columns span what is called the *coarse space*.

In addition to the local solves in the subdomains, the application of a two-level domain decomposition preconditioner implies solving a reduced size problem built with Z called the *coarse problem*. The construction of effective two-level preconditioners and coarse spaces is a difficult task. After introducing in Section 7.2 these methods for Symmetric Positive Definite (SPD) problems, in the rest of the Chapter we concentrate on the Helmholtz equation. Our purpose is to compare numerically, both in two and three dimensions, two different coarse space definitions for this equation, which are currently the most robust available in literature.

7.2 Two-level preconditioners for positive definite problems

For one-level domain decomposition preconditioners, which are based solely on local solves in the subdomains, the iteration count grows with the number of subdomains and plateaux appear in the convergence history plots. These plateaux can be observed for instance in Figure 5.15 of Chapter 5, but this is the case even for a simple model such as the Poisson problem:

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases}$$

The problem of one-level methods is a *lack of global communication*. Data are exchanged only from one subdomain to its direct neighbors, but the solution in each subdomain depends on the right-hand side in all subdomains. If we denote by N_d the number of subdomains in one direction, then the leftmost domain of Figure 7.1 needs at least N_d iterations before being aware of the value of the right-hand side f in the rightmost subdomain. The length of the plateau is thus typically related to the number of subdomains in one direction (see for instance §4.1 of [41]).

In order to achieve scalability with respect to the number of subdomains, two-level domain decomposition preconditioners with a coarse-grid correction are introduced: the coarse problem, built using the coarse space matrix Z, couples all subdomains at each iteration. Defining $R_0 = Z^T$, the most natural correction for the Additive Schwarz preconditioner (5.6) is of additive type: the *two-level additive Schwarz preconditioner* is defined

as

$$M_{2,AS}^{-1} = \sum_{j=1}^{N_{\text{sub}}} R_j^T (R_j A R_j^T)^{-1} R_j + R_0^T (R_0 A R_0^T)^{-1} R_0,$$

where $R_0 A R_0^T$ is the coarse problem matrix. For the Poisson problem, an adequate coarse space is the *Nicolaides* coarse space [92]: the columns of the matrix Z are vectors that have local support in the subdomains and such that the constant function $\mathbf{1}$ belongs to the vector space spanned by them, i.e. the i-th column of Z is $R_i^T D_i R_i \mathbf{1}$ for $1 \leq i \leq N_{\text{sub}}$. Indeed, the constant functions are the modes, lying in the kernel of the Laplace operator, that hamper the convergence of the one-level method.

This classical coarse space should be enriched with more than one vector per subdomain when the problem to be solved gets more complex, for instance as a result of highly heterogeneous coefficients. Jumps in the coefficients along subdomain interfaces (rather than across the interfaces or inside the subdomains far from their boundaries) are particularly problematic. In [86, 87, 42] a robust coarse space was introduced and analyzed for the scalar elliptic equation $-\nabla \cdot (\alpha(\mathbf{x}) \nabla u) = f$ (also known as the Darcy equation), based on local generalized eigenvalue problems involving the *Dirichlet-to-Neumann* (DtN) operator on the subdomains interfaces. This method is efficient for arbitrary domain decompositions and jumps in the coefficients, leading to an automatic preconditioning method for SPD scalar heterogenous problems.

The arguments of the analysis in [42] cannot be easily generalized to the case of SPD systems of partial differential equations. Thus in [107, 106] a robust coarse space based on Generalized Eigenproblems in the Overlap (referred to as the *GenEO* coarse space) was studied. Its convergence can be proved by reformulating the domain decomposition method in an abstract setting in order to apply the fictitious space lemma of [91, 90, 64] (see for instance Chapter 7 of [41]). The considered generalized eigenvalue problems are closely related but different from those proposed in [45].

7.3 Two-level preconditioners for the Helmholtz equation

The construction and, even more, the convergence analysis of two-level domain decomposition preconditioners for *sign-indefinite* problems are a challenging open issue. In this work we compare numerically two different definitions of the coarse space matrix Z for the Helmholtz equation (7.1), both in two and three dimensions. More precisely, we are interested in solving the interior Helmholtz problem of the following form: let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be a polyhedral, bounded domain; find $u \colon \Omega \to \mathbb{C}$ such that

$$\begin{cases} -\Delta u - \tilde{\omega}^2 u = f, & \text{in } \Omega, \quad (7.2a) \\ \dfrac{\partial u}{\partial n} - \mathtt{i}\tilde{\omega} u = 0, & \text{on } \Gamma = \partial\Omega. \quad (7.2b) \end{cases}$$

The two-level domain decomposition preconditioners compared here are both built from the corresponding problem with *absorption* (or *damping*), given by a parameter ξ [1]:

$$\begin{cases} -\Delta u - (\tilde{\omega}^2 + \mathtt{i}\xi) u = f, & \text{in } \Omega, \quad (7.3a) \\ \dfrac{\partial u}{\partial n} - \mathtt{i}\eta u = 0, & \text{on } \Gamma = \partial\Omega. \quad (7.3b) \end{cases}$$

1. Note that the usual notation for the absorption parameter ξ is ε, but in this thesis ε already denotes the electric permittivity; the wavenumber $\tilde{\omega}$ is often called k.

The variational formulation of this problem is: find $u \in V = H^1(\Omega)$ such that

$$a_\xi(u, v) = F(v), \, \forall v \in V,$$

where the sesquilinear form[2] $a_\xi \colon V \times V \to \mathbb{C}$ and $F \colon V \to \mathbb{C}$ are defined by

$$a_\xi(u, v) = \int_\Omega \left(\nabla u \cdot \overline{\nabla v} - (\tilde{\omega}^2 + \mathtt{i}\xi) u \overline{v} \right) - \int_\Gamma \mathtt{i}\eta u \overline{v}, \quad F(v) = \int_\Omega f \overline{v}.$$

Note that if $\xi \neq 0$ and the Robin parameter $\eta = \text{sign}(\xi)\tilde{\omega}$ in (7.3b), then a_ξ is *coercive* (see §2 in [62]). We consider a discretization of the variational problem using piecewise linear finite elements on a uniform simplicial mesh $\mathcal{T}_h$ of Ω. Denoting by $V_h \subset V$ the corresponding finite element space and by $\{\phi_i\}_{i=1}^n$ its basis functions, $n := \dim(V_h)$, the discretized problem reads: find $u_h \in V_h$ such that

$$a_\xi(u_h, v_h) = F(v_h), \, \forall v_h \in V_h,$$

that is, in matrix form,

$$A_\xi \mathbf{u} = \mathbf{f}, \tag{7.4}$$

where the coefficients of the matrix $A_\xi \in \mathbb{C}^{n \times n}$ and the right-hand side $\mathbf{f} \in \mathbb{C}^n$ are given by $(A_\xi)_{i,j} = a(\phi_i, \phi_j)$ and $(\mathbf{f})_i = F(\phi_i)$. The matrix A_ξ is complex, symmetric (but not Hermitian), and sign-indefinite if $\xi = 0$.

Consider a decomposition of the domain Ω into a set of overlapping subdomains $\{\Omega_j\}_{j=1}^{N_{\text{sub}}}$, with each subdomain consisting of a union of elements of the mesh $\mathcal{T}_h$. The *one-level* preconditioner, on which the tested two-level preconditioners are based, is the Optimized Restricted Additive Schwarz (ORAS) preconditioner (5.7) presented in detail in Chapter 5, but built from the problem with absorption:

$$M_{1,\xi}^{-1} = \sum_{j=1}^{N_{\text{sub}}} \widetilde{R}_j^T A_{j,\xi}^{-1} R_j \tag{7.5}$$

(in the subscript, 1 stands for one-level and ξ indicates the presence of absorption in the preconditioner). In (7.5) the local matrix $A_{j,\xi}$ is the matrix stemming from the discretization of the following Robin boundary value problem with absorption in the subdomain Ω_j:

$$\begin{cases} -\Delta u_j - (\tilde{\omega}^2 + \mathtt{i}\xi) u_j = f, & \text{in } \Omega_j, \\ \dfrac{\partial u_j}{\partial n_j} - \mathtt{i}\eta u_j = 0, & \text{on } \partial\Omega_j. \end{cases}$$

In order to achieve weak dependence on the wavenumber $\tilde{\omega}$, we combine the one-level preconditioner (7.5) with a coarse correction. The *two-level* preconditioners considered in this work can be written in a generic way as follows:

$$M_{2,\xi}^{-1} = Q M_{1,\xi}^{-1} P + Z E^{-1} Z^*, \tag{7.6}$$

where $*$ denotes the conjugate transpose. The matrices appearing in (7.6) are:

- $M_{1,\xi}^{-1}$ is the one-level preconditioner,

2. A map $\varphi \colon V \times V \to \mathbb{C}$ over a complex vector space V is sesquilinear if $\varphi(x + y, z + w) = \varphi(x, z) + \varphi(x, w) + \varphi(y, z) + \varphi(y, w)$, and $\varphi(ax, by) = a\bar{b}\,\varphi(x, y)$, i.e. it is linear in one variable and semilinear in the other (the Latin prefix sesqui means "one and a half").

- Z is a rectangular matrix with full column rank, whose columns span the coarse space (abbreviated CS in the numerical experiments tables),
- $E = Z^* A_\xi Z$ is the so-called coarse grid matrix (note that here it is built from the problem with absorption),
- $\Xi = Z E^{-1} Z^*$ is the so-called coarse grid correction matrix,

and the matrices P, Q are defined according to the chosen correction *formula*:

- if $P = Q = I$ this is an *additive* two-level preconditioner,
- if $P = I - A_\xi \Xi$ and $Q = I - \Xi A_\xi$, this is a *hybrid* two-level preconditioner; for special choices of M_1^{-1} and Z it is also known as the Balancing Neumann-Neumann (BNN) preconditioner introduced in [81].

The two-level preconditioner is characterized by the choice of Z: we will consider the coarse space definitions of [62] and [34], which are currently the most robust available in literature.

7.3.1 The grid coarse space

The most natural coarse space would be one based on a *coarser mesh*, in this work we subsequently call it "grid coarse space". Let us consider $\mathcal{T}_{H_{\text{cs}}}$ a simplicial mesh of Ω with mesh diameter H_{cs} and $V_{H_{\text{cs}}} \subset V$ the corresponding piecewise linear finite element space. Let $\mathcal{I}_0 \colon V_{H_{\text{cs}}} \to V_h$ be the nodal interpolation operator from the coarse grid finite element space to the fine grid finite element space and define Z as the corresponding matrix. Then in this case $E = Z^* A_\xi Z$ is really the matrix of the problem (with absorption) discretized on the coarse mesh.

This definition was studied in [62, 63], where, in the numerical experiments, two-level domain decomposition approximations of the Helmholtz problem with absorption (7.3) were used as preconditioners for the pure Helmholtz problem *without* absorption (7.2); in this method the coarse mesh diameter is constrained by the wavenumber $\tilde{\omega}$. The *theory* developed in [62] concerns the solution of the problem *with* absorption (7.3), using GMRES with a two-level Additive Schwarz preconditioner: it provides rigorous convergence estimates, explicit in the wavenumber $\tilde{\omega}$, the absorption ξ, the coarse-grid diameter H_{cs}, the subdomain diameter H_{sub}, the overlap size δ. We report here the final theorem:

Theorem 7.1 (GMRES convergence for left preconditioning in [62])**.** *Consider the weighted GMRES method where the residual is minimized in a suitable $\tilde{\omega}$-weighted norm $\|\cdot\|_{D_{\tilde{\omega}}}$. Let $\mathbf{r}^m$ denote the mth iterate of GMRES applied to the system A_ξ, left preconditioned with the two-level Additive Schwarz preconditioner. Then*

$$\frac{\|\mathbf{r}^m\|_{D_{\tilde{\omega}}}}{\|\mathbf{r}^0\|_{D_{\tilde{\omega}}}} \lesssim \left(1 - \left(1 + \frac{H_{cs}}{\delta}\right)^{-2} \left(\frac{|\xi|}{\tilde{\omega}^2}\right)^6\right)^{m/2},$$

provided the following condition holds

$$\max\left\{\tilde{\omega} H_{\text{sub}},\ \tilde{\omega} H_{\text{cs}} \left(1 + \frac{H_{\text{cs}}}{\delta}\right) \left(\frac{\tilde{\omega}^2}{|\xi|}\right)^2\right\} \leq \mathcal{C}_1 \left(1 + \frac{H_{\text{cs}}}{\delta}\right)^{-1} \left(\frac{|\xi|}{\tilde{\omega}^2}\right).$$

An important special case of this theory is that, for the problem with absorption $\xi \sim \tilde{\omega}^2$, the number of GMRES iterates is bounded *independently* of $\tilde{\omega}$, provided $H_{\text{cs}} \sim H_{\text{sub}} \sim \tilde{\omega}^{-1}$ and $\delta \sim H_{\text{cs}}$ ("generous overlap").

The rigorous analysis of the earlier work [57] focused on the choice of the absorption parameter ξ when discretizations of (7.3) are used as preconditioners for problem (7.2). This technique is called *shifted Laplacian preconditioning* (see §1.1 of [57] for a complete literature survey), and is motivated by the fact that, as ξ increases, the problem with absorption (the "shifted" problem) becomes easier to solve iteratively. In [57] it was shown that if $\xi/\tilde{\omega}$ is bounded above by a sufficiently small constant, then GMRES with the shifted Laplacian preconditioner converges in a $\tilde{\omega}$-independent number of iterations. Note that this sufficient condition on ξ (for the shifted Laplacian to be a good preconditioner for problem (7.2)) does not overlap with the one of [62] (for the domain decomposition method to be a good preconditioner for the shifted problem). Thus the combination of [57] with [62] does not provide a complete theory for preconditioning problem (7.2); nevertheless, these are among the few rigorous convergence theory results available in literature about preconditioners for wave propagation problems.

7.3.2 The DtN coarse space

The second definition of Z we consider is the one introduced in [34], where the coarse space is built by solving local eigenproblems involving the *Dirichlet-to-Neumann* (DtN) operator on the subdomains interfaces. This is an adaptation of the idea for scalar elliptic problems developed in [86, 87, 42]. This method proved to be very robust, with respect to heterogeneous coefficients, compared to the reference coarse space based on plane waves. Plane waves were originally used in the multigrid context [29] and later applied to domain decomposition methods [53, 52, 75, 78], but mainly for homogeneous problems. Moreover, contrarily to the plane waves coarse space, the DtN coarse space construction in [34] is completely automatic, refraining from the need for parameter tuning, which is crucial for indefinite problems since even slight deviations from the optimal choice can be fatal [54]. However, note that a complete convergence theory for the DtN coarse space for the Helmholtz equation is missing.

We recall now in detail the definition of this coarse space but note that here, contrarily to the original definition in [34], it is built from problems *with absorption* in order to compare it with the grid coarse space under the same conditions.

On each interface $\Gamma_i = \partial\Omega_i \setminus \partial\Omega$, we solve the *local DtN eigenproblem*: find the eigenvalues λ and the eigenfunctions u_{Γ_i} such that

$$\text{DtN}_{\Omega_i}(u_{\Gamma_i}) = \lambda u_{\Gamma_i}, \tag{7.7}$$

where the operator DtN_{Ω_i} is defined as follows. For $v_{\Gamma_i} : \Gamma_i \to \mathbb{C}$, define

$$\text{DtN}_{\Omega_i}(v_{\Gamma_i}) = \left.\frac{\partial u}{\partial n}\right|_{\Gamma_i},$$

where $u : \Omega_i \to \mathbb{C}$ is the *Helmholtz extension* to Ω_i of v_{Γ_i} given by

$$\begin{cases} -\Delta u - (\tilde{\omega}^2 + \mathtt{i}\xi)u = 0, & \text{in } \Omega_i, \\ \partial u/\partial n - \mathtt{i}\eta u = 0, & \text{on } \partial\Omega_i \cap \partial\Omega, \\ u = v_{\Gamma_i}, & \text{on } \Gamma_i. \end{cases}$$

In other words, given a function defined on Γ_i, the operator DtN_{Ω_i} extends it to the interior of the subdomain using the Helmholtz extension, and then returns the normal derivative

of the extension at Γ_i. This is indeed a map between Dirichlet and Neumann data. To constitute the coarse space, for each subdomain Ω_i we choose m_i eigenfunctions of the DtN eigenproblem (7.7) according to the following *automatic selection criterion*: choose all eigenfunctions for which the associated eigenvalue λ satisfies

$$\mathrm{Re}(\lambda) < \max_{\mathbf{x}\in\Omega_i} \tilde{\omega}(\mathbf{x}). \tag{7.8}$$

Note that this criterion respects local variations in the wavenumber and is hence suited also for heterogeneous problems.

Now, in order to define the discrete formulation of the DtN eigenproblem, for a subdomain Ω_i, first of all consider $a^{(i)}\colon H^1(\Omega_i)\times H^1(\Omega_i)\to\mathbb{R}$

$$a^{(i)}(v,w) = \int_{\Omega_i} \left(\nabla v\cdot\overline{\nabla w} - (\tilde{\omega}^2 + \mathtt{i}\xi)v\overline{w}\right) - \int_{\partial\Omega_i\cap\partial\Omega} \mathtt{i}\eta u\overline{v},$$

and the associated matrix $(A^{(i)})_{kl} = a^{(i)}(\phi_k,\phi_l)$. Let I and Γ_i be the sets of indices corresponding, respectively, to the interior and boundary degrees of freedom on Ω_i, with n_I and n_{Γ_i} their cardinalities. With the usual block notation, the subscripts I and Γ_i for the matrices A and $A^{(i)}$ denote the entries of these matrices associated with the respective degrees of freedom. Let

$$M_{\Gamma_i} = \left(\int_{\Gamma_i} \phi_k\phi_l\right)_{k,l\in\Gamma_i}$$

be the mass matrix on the interface Γ_i of subdomain Ω_i. Following a procedure analogous to the one in [42], the *discrete formulation of the DtN eigenproblem* (7.7) is: find $(\lambda,\mathbf{u})\in\mathbb{C}^{n_{\Gamma_i}}\times\mathbb{C}$, such that

$$(A^{(i)}_{\Gamma_i\Gamma_i} - A_{\Gamma_i I}A_{II}^{-1}A_{I\Gamma_i})\mathbf{u} = \lambda M_{\Gamma_i}\mathbf{u}. \tag{7.9}$$

Now, the matrix Z of the DtN coarse space is a rectangular, *block-diagonal* matrix with blocks W_i, associated with the subdomain Ω_i, $1\le i\le N_{\mathrm{sub}}$, given by Algorithm 7.3.1. If m_i is the number of eigenvectors selected by the automatic criterion (7.8) and n_i is the number of dofs in the subdomain Ω_i, the block W_i has dimensions $n_i\times m_i$, and the matrix Z has dimensions $n\times\sum_{j=1}^{N_{\mathrm{sub}}} m_i$. Due to the overlap in the decomposition, the blocks may share some rows inside the matrix Z.

Algorithm 7.3.1 Construction of the block W_i of the DtN coarse space matrix Z

1: Solve the discrete DtN eigenproblem (7.9) on subdomain Ω_i for the eigenpairs $(\lambda_j,\mathbf{g}_i^j)$.
2: Choose m_i eigenvectors $\mathbf{g}_i^j\in\mathbb{C}^{n_{\Gamma_i}}$ using criterion (7.8).
3: **for** $j=1$ to m_i **do**
4: Compute the discrete Helmholtz extension $\mathbf{u}_i^j\in\mathbb{C}^{n_i}$ to Ω_i of $\mathbf{g}_i^j$ as $\mathbf{u}_i^j = [-A_{II}^{-1}A_{I\Gamma_i}\mathbf{g}_i^j,\quad \mathbf{g}_i^j]^T$.
5: **end for**
6: Define the matrix $W_i\in\mathbb{C}^{n_i\times m_i}$ as $W_i = \left(D_i\mathbf{u}_i^1,\quad \ldots,\quad D_i\mathbf{u}_i^{m_i}\right)$.

Since this construction is based on local problems only, it is possible to construct the coarse space efficiently in parallel. Note that the sparsity of the coarse grid matrix $E = Z^*A_\xi Z$ results from the sparsity of Z, the non zero components of E corresponding to adjacent subdomains.

7.4 Numerical experiments

We solve the pure Helmholtz problem (7.2) on the unit square ($d = 2$) or cube ($d = 3$). We consider a uniform simplicial (triangular in $2d$, tetrahedral in $3d$) mesh of diameter $h \sim \tilde{\omega}^{-3/2}$, thus the number of degrees of freedom grows quite rapidly with the wavenumber (recall that this is the discretization level which is generally believed to remove the pollution effect). The right-hand side is given by $f = -\exp(-100((x-0.5)^2 + (y-0.5)^2))$ for $d = 2$, $f = -\exp(-400((x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2))$ for $d = 3$.

We use GMRES with right preconditioning, starting with a *random initial guess*, which ensures, unlike a zero initial guess, that all frequencies are present in the error; the stopping criterion is based on the relative residual with a tolerance $\tau = 10^{-6}$. The maximum number of iterations allowed is 500 in $2d$, 200 in $3d$. We consider a regular decomposition into subdomains (squares/cubes), the overlap for each subdomain is of size $\mathcal{O}(2h)$ in all directions and the two-level preconditioner (7.6) is used with the hybrid formula, which results to be more effective than the additive one. This hybrid two-level preconditioner, based on the one-level ORAS preconditioner, is called ImpHRAS in [62]: the prefix O for Optimized is replaced with Imp, which stands for impedance (i.e. Robin) transmission conditions, and H stands for hybrid.

All the computations are done in FreeFem++. The code for $3d$ computations is parallelized and run on the Curie supercomputer (at TGCC-CEA, `http://www-hpc.cea.fr/en/complexe/tgcc-curie.htm`). We assign each subdomain to one processor. So the number of processors increases if the number of subdomains increases. To apply the preconditioner, the local problems in each subdomain (with matrices $A_{j,\xi}$ in (7.5)) and the coarse space problem (with matrix E in (7.6)) are solved with a direct solver (UMFPACK [36] in $2d$, MUMPS [2] on one processor in $3d$).

As in [62, 63], in the experiments we take the subdomain diameter H_{sub} and the coarse mesh diameter H_{cs} constrained by $\tilde{\omega}$:

$$H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}, \qquad H_{\text{cs}} \sim \tilde{\omega}^{-\alpha'},$$

for some choices of $0 < \alpha, \alpha' <= 1$ detailed in the following; if not differently specified, we take $\alpha = \alpha'$, which is the setting of all numerical experiments in [62]. For a given $\tilde{\omega}$, the smaller is α the more (and the smaller) are the subdomains; the smaller is α', the coarser is the mesh for the grid coarse space. Note that H_{cs} does not appear as a parameter in the DtN coarse space!

We denote by n_{CS} the size of the coarse space i.e. the dimension of the matrix E in (7.6). For the grid coarse space since E is really the matrix of the problem (with absorption) discretized on the coarse mesh, we have $n_{\text{CS}} = (1/H_{\text{cs}} + 1)^d$, the number of dofs for the nodal linear finite elements in the unit square/cube. For the DtN coarse space we have $n_{\text{CS}} = \sum_{j=1}^{N_{\text{sub}}} m_i$, the total number of computed eigenvectors for all the subdomains. While we solve the pure Helmholtz problem without absorption, both the one-level preconditioner (7.5) and the two-level preconditioner (7.6) are built from problems which have non zero absorption given by

$$\xi_{\text{prec}} = \tilde{\omega}^{\beta}.$$

In the experiments we put $\beta = 1$ or $\beta = 2$.

In the following tables we compare the one-level preconditioner, the two-level preconditioner with the grid coarse space and the two-level preconditioner with the DtN coarse space in terms of number of iterations of GMRES and size of the coarse space (n_{CS}), for different values of the wavenumber $\tilde{\omega}$ and of the parameters α, β. We also report the

number of subdomains N_{sub}, which is controlled by $\tilde{\omega}$ and α as mentioned above. Since $h \sim \tilde{\omega}^{-3/2}$, the size n of the linear system matrix is of order $\tilde{\omega}^{3d/2}$; for 3d experiments we report n explicitly. Tables 7.1, 7.2 concern the $2d$ problem, Table 7.3 the $3d$ problem.

Note that the points of view about the aim of a two-level preconditioner adopted by [34] and [62] are different. The DtN coarse space in [34] is built to reach the classical scalability with respect to the number of subdomains, while the grid coarse space in [62] seeks independence with respect to the wavenumber $\tilde{\omega}$ and for that considers a number of subdomains constrained by $\tilde{\omega}$. In the following comparison we adopt the second point of view.

7.4.1 Experiment 1

In Table 7.1, we let the DtN coarse space size be determined by the automatic selection criterion (7.8) and the grid coarse space size by $H_{\text{cs}} \sim \tilde{\omega}^{-\alpha}$.

We see that the DtN coarse space is much larger than the grid coarse space and gives fewer iterations. The preconditioners with absorption $\xi_{\text{prec}} = \tilde{\omega}^2$ ($\beta = 2$) perform much worse than those with absorption $\xi_{\text{prec}} = \tilde{\omega}$ ($\beta = 1$) independently of the coarse space size. For $\xi_{\text{prec}} = \tilde{\omega}$, when $\alpha = 1$ the number of iterations grows mildly with the wavenumber $\tilde{\omega}$ for both coarse spaces (but at the cost of an increasing coarse space size), while the one-level preconditioner performs poorly. When $\alpha < 1$, i.e. for coarser coarse meshes, the growth with $\tilde{\omega}$ is higher, and for $\alpha = 0.6$ the two-level preconditioner is not much better than the one-level preconditioner because the coarse grid problem is too coarse; for $\alpha = 0.8$ with the DtN coarse space the growth with $\tilde{\omega}$ degrades less than with the grid coarse space.

7.4.2 Experiment 2

We have seen in Table 7.1 that the DtN coarse space gives fewer iterations than the grid coarse space, but their sizes differed significantly. Therefore, in Table 7.2 we compare the two methods *forcing n_{CS} to be similar*.

On the left, we force the DtN coarse space to have a smaller size, similar to the one of the grid coarse space, by taking just $m_i = 2$ eigenvectors for each subdomain. On the right, we do the opposite, we force the grid coarse space to have the size of the DtN coarse space obtained in Table 7.1, by prescribing a smaller coarse mesh diameter H_{cs}, while keeping the same number of subdomains as in Table 7.1 with $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}$. In this experiment we take $\xi_{\text{prec}} = \tilde{\omega}$, which proved to give better iteration counts than $\xi_{\text{prec}} = \tilde{\omega}^2$ in the previous experiment. We can observe that for smaller coarse space sizes (left) the grid coarse space gives fewer iterations than the DtN coarse space, while for larger coarse space sizes (right) the result is reversed.

7.4.3 Experiment 3

We have seen that the coarse mesh obtained with $H_{\text{cs}} \sim \tilde{\omega}^{-\alpha'}$, $\alpha' = \alpha$ can be too coarse if $\alpha = 0.6$. At the same time, for $\alpha = 1$ the number of subdomains gets quite large since $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}$, especially in $3d$; this is not desirable because in our parallel implementation we assign each subdomain to one processor, so communication among them would prevail and each processor would not be fully exploited since the subdomains would become very small. Therefore, in this $3d$ experiment, to improve convergence with the grid coarse space while maintaining a reasonable number of subdomains, we consider *separate coarse mesh diameter and subdomain diameter*, taking $\alpha' \neq \alpha$.

For load balancing, meant as local problems having the same size as the *grid coarse space* problem, in $3d$ we choose $\alpha' = 3/2 - \alpha$. Indeed, the number of degrees of freedom in

		$\beta = 1$				
		$\alpha = 0.6$				
$\tilde{\omega}$	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	9	22	19	16	11	39
20	36	48	46	49	26	204
40	81	78	98	100	37	531
60	121	109	114	144	94	875
		$\alpha = 0.8$				
$\tilde{\omega}$	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	36	35	19	49	10	122
20	100	71	35	121	13	394
40	361	158	88	400	22	1440
60	676	230	187	729	39	2700
		$\alpha = 1$				
$\tilde{\omega}$	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	100	65	26	121	11	324
20	400	122	26	441	14	1120
40	1600	286	33	1681	20	4640
60	3600	445	45	3721	29	10560

		$\beta = 2$				
		$\alpha = 0.6$				
$\tilde{\omega}$	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	9	28	27	16	23	40
20	36	67	56	49	40	220
40	81	121	114	100	72	578
60	121	169	165	144	135	758
		$\alpha = 0.8$				
$\tilde{\omega}$	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	36	39	27	49	28	86
20	100	83	51	121	41	362
40	361	182	95	400	71	1370
60	676	268	150	729	103	2698
		$\alpha = 1$				
$\tilde{\omega}$	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	100	57	30	121	23	324
20	400	130	49	441	42	1120
40	1600	296	80	1681	72	4640
60	3600	455	112	3721	101	10560

Table 7.1 – (d=2) Number of iterations (and coarse space size n_{CS}) for the one-level preconditioner and the two-level preconditioners with the grid coarse space/DtN coarse space, with $H_{\text{sub}} = H_{\text{cs}} \sim \tilde{\omega}^{-\alpha}$, $\xi_{\text{prec}} = \tilde{\omega}^{\beta}$.

		n_{CS} forced by grid CS				n_{CS} forced by DtN CS			
		$\alpha = 0.6$				$\alpha = 0.6$			
$\tilde{\omega}$	N_{sub}	grid CS	n_{CS}	DtN CS	n_{CS}	grid CS	n_{CS}	DtN CS	n_{CS}
10	9	19	16	18	18	17	36	11	39
20	36	46	49	44	72	24	196	26	204
40	81	98	100	85	162	50	529	37	531
60	121	114	144	109	242	104	841	94	875
		$\alpha = 0.8$				$\alpha = 0.8$			
$\tilde{\omega}$	N_{sub}	grid CS	n_{CS}	DtN CS	n_{CS}	grid CS	n_{CS}	DtN CS	n_{CS}
10	36	19	49	26	72	15	121	10	122
20	100	35	121	61	200	20	361	13	394
40	361	88	400	139	722	35	1369	22	1440
60	676	187	729	191	1352	52	2601	39	2700
		$\alpha = 1$				$\alpha = 1$			
$\tilde{\omega}$	N_{sub}	grid CS	n_{CS}	DtN CS	n_{CS}	grid CS	n_{CS}	DtN CS	n_{CS}
10	100	26	121	52	200	17	324	11	324
20	400	26	441	43	800	23	1089	14	1120
40	1600	33	1681	157	3200	22	4624	20	4640
60	3600	45	3721	338	7200	26	10404	29	10560

Table 7.2 – (d=2) Number of iterations (and coarse space size n_{CS}) for the two-level preconditioners with the grid coarse space/DtN coarse space *forcing similar* n_{CS}, with $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}$, $\xi_{\text{prec}} = \tilde{\omega}$.

each subdomain ignoring the overlap is around $(h^{-1}/H_{\text{sub}}^{-1})^3 \sim \tilde{\omega}^{9/2-3\alpha}$ and the number of degrees of freedom in the grid coarse space problem is $(H_{\text{cs}}^{-1})^3 \sim \tilde{\omega}^{3\alpha'}$, so for load balancing in the sense described above we require $3\alpha' = 9/2 - 3\alpha$. The DtN coarse space size is still determined by the automatic choice criterion (among 20 computed local eigenvectors) in each subdomain.

In Table 7.3 we report the results of this experiment. Note that the size n of the global matrix, which should depend just on the wavenumber $\tilde{\omega}$ (with $h \sim \tilde{\omega}^{-3/2}$), in fact appears to vary also for different α: this is due to the fact that we modify the number of points in each direction to be an exact multiple of the number of subdomains in that direction, so as to have smooth cubes as subdomains.

As expected, for the grid coarse space the best iteration counts are obtained for $\alpha = 0.5$ because then $\alpha' = 1$ gives the coarse mesh with the smallest diameter among the experimented ones: the number of iterations grows slowly, with $\mathcal{O}(\tilde{\omega}^{0.61}) \cong \mathcal{O}(n^{0.13})$. With higher α the iteration counts get worse quickly, and $\alpha = 0.8$ is not usable.

For the DtN coarse space, the larger coarse space size is obtained by taking α bigger (recall that α' is not a parameter in the DtN case): for $\alpha = 0.8$ the number of iterations grows slowly, with $\mathcal{O}(\tilde{\omega}^{0.2}) \cong \mathcal{O}(n^{0.04})$, but this value may be optimistic, there is a decrease in iteration number between $\tilde{\omega} = 20$ and 30. We believe that for the other values of α, where the iteration counts are not much better or worse than with the one-level preconditioner, we did not compute enough eigenvectors in each subdomain to build the DtN coarse space.

			$\alpha = 0.5$, $\alpha' = 1$				
$\tilde{\omega}$	n	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	39304	27	25	12	1331	14	316
20	704969	64	39	17	9261	31	1240
30	5000211	125	55	21	29791	54	2482
40	16194277	216	74	29	68921	80	4318
			$\alpha = 0.6$, $\alpha' = 0.9$				
$\tilde{\omega}$	n	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	39304	27	25	15	512	14	316
20	912673	216	61	24	3375	41	2946
30	4826809	343	73	34	10648	65	6226
40	16194277	729	98	48	21952	108	13653
			$\alpha = 0.7$, $\alpha' = 0.8$				
$\tilde{\omega}$	n	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	46656	125	34	19	343	11	896
20	912673	512	73	35	1331	18	4567
30	5929741	1000	103	57	4096	65	12756
40	17779581	2197	139	89	8000	116	30603
			$\alpha = 0.8$, $\alpha' = 0.7$				
$\tilde{\omega}$	n	N_{sub}	1-level	grid CS	n_{CS}	DtN CS	n_{CS}
10	50653	216	39	23	216	19	1354
20	1030301	1000	46	86	729	23	7323
30	5929741	3375	137	116	1331	21	26645
40	28372625	6859	189	200	2744	27	54418

Table 7.3 – (d=3) Number of iterations (and coarse space size n_{CS}) for the one-level preconditioner and the two-level preconditioners with the grid coarse space/DtN coarse space, with $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}$, $H_{\text{cs}} \sim \tilde{\omega}^{-\alpha'}$, $\xi_{\text{prec}} = \tilde{\omega}$.

7.5 Conclusion

We tested numerically two different coarse space definitions for two-level domain decomposition preconditioners for the pure Helmholtz equation (discretized with piecewise linear finite elements), both in $2d$ and $3d$, reaching more than 28 million complex-valued unknowns in the resulting linear systems.

The preconditioners built with absorption $\xi_{\text{prec}} = \tilde{\omega}^2$ appear to perform much worse than those with absorption $\xi_{\text{prec}} = \tilde{\omega}$. We have seen that in most cases for smaller coarse space sizes the grid coarse space gives fewer iterations than the DtN coarse space, while for larger coarse space sizes the grid coarse space gives generally more iterations than the DtN coarse space. The best iterations counts for the grid coarse space are obtained by separating the coarse mesh diameter $H_{\text{cs}} \sim \tilde{\omega}^{-\alpha'}$ from the subdomain diameter $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}$, taking $\alpha' > \alpha$. Both for the coarse grid space and the DtN coarse space, for appropriate choices of the method parameters we have obtained iteration counts which grow quite slowly with the wavenumber $\tilde{\omega}$.

Further experiments to compare the two definitions of coarse space should be carried out in the heterogenous case. Note that the DtN coarse space construction naturally respects local variations in the wavenumber, and paragraphs 6.6 – 6.7 of [34] present numerical experiments for some simple heterogenous configurations. Also for the grid coarse space the heterogeneous case is under investigation and preliminary results can be found in paragraph 5.3 of [63].

Chapter 8

Two-level preconditioners for Maxwell's equations

This Chapter is based on the forthcoming paper [16] and on [15], in collaboration with Victorita Dolean, Ivan G. Graham, Euan A. Spence, and Pierre-Henri Tournier. The preparatory paper [15] has been submitted to the proceedings of the *DD24 International Conference on Domain Decomposition Methods* and a preprint is available on arXiv and HAL (<hal-01525438>).

Contents

8.1 Introduction

After focusing on the Helmholtz equation in Chapter 7, we now face the time-harmonic Maxwell's equation (2.26). The construction of fast iterative solvers for both these equations in the high-frequency regime is a challenging problem. Here we investigate how the two-level domain decomposition preconditioners rigorously analyzed in [62] for the Helmholtz equation (see §7.3.1) work in the Maxwell case, both from the theoretical and numerical points of view. In the context considered here, we wish to find a "good" preconditioner in the sense that the number of iterations needed to solve the preconditioned system should be independent of the wavenumber $\tilde{\omega}$.

We present a new theory for the time-harmonic Maxwell's equation *with absorption*, which physically corresponds to the case of dissipative materials with non zero conductivity $\sigma > 0$. This theory provides rates of convergence for GMRES with a two-level Additive Schwarz (AS) preconditioner, *explicit in* the wavenumber $\tilde{\omega}$, the absorption ξ, the coarse-grid diameter, the subdomain diameter and the overlap size. It uses a $\tilde{\omega}$- and ξ-explicit coercivity result for the underlying sesquilinear form and the main theorems give an upper bound on the norm of the preconditioned matrix and a lower bound on its field of values, so that Elman-type estimates for the convergence of GMRES can be applied. Note that analyzing the convergence of GMRES is hard, since no convergence estimates in terms of the condition number, as those we are used to with the conjugate gradient method, are available for GMRES.

Extensive large scale numerical experiments are carried out not only in the setting covered by the theory, but also for the time-harmonic Maxwell's equation *without absorption*, and with more efficient two-level preconditioners, considering for instance impedance transmission conditions at interfaces between subdomains.

In paragraph 8.1.1 below we specify which are the boundary value problems (BVPs) considered in this work; then the Chapter is organized as follows. Section 8.2 introduces the variational formulation of the BVP studied by the theory, and also of its adjoint, which intervenes in the analysis; then it provides the continuity and coercivity properties of the corresponding sesquilinear form, using a norm induced by a wavenumber-dependent inner product $(\cdot,\cdot)_{\mathrm{curl},\tilde{\omega}}$, and states regularity properties of their solutions. Section 8.3 fixes the notation for the domain decomposition set-up and identifies the assumptions on the subdomains; it recalls the discrete Helmholtz decomposition of the (global and local) curl-conforming finite element spaces, and the Poincaré–Friedrichs type-inequality of [60], which holds just for a term of the direct sum.

In Section 8.4 we state all the intermediate theorems for the theory about the two-level Additive Schwarz method. We report only partially the proofs which will be part of the forthcoming paper [16]. The analysis in based on expressing the action of the domain decomposition preconditioner on the matrix as an operator $\mathbf{T}_\xi$, sum of projection operators onto the local and coarse finite element spaces. Our goal is to bound from above the norm of the operator $\mathbf{T}_\xi$ and from below its field of values. Then in Section 8.5 we convert these bounds into estimates for the norm and field of values of the preconditioned matrix (in the induced weighted Euclidean inner product). Thus the Elman-type estimates for convergence of GMRES recalled in §8.5.2 can be applied to obtain the final convergence rate for GMRES (left- or right-) preconditioned with the two-level Additive Schwarz method. Note that the detailed theory development is mainly due to Euan Spence.

Finally, in Section 8.6 we report an extensive numerical study of the convergence of GMRES, with several versions of the two-level preconditioner: additive and hybrid coarse correction formulas, standard and optimized local solves, generous and minimal overlap, different scalings of the subdomain diameter and the coarse grid diameter with respect to

the wavenumber $\tilde{\omega}$. Moreover, we test various levels of absorption in the problem and in the preconditioner, and discretizations with degree 1 and 2 curl-conforming finite elements.

8.1.1 Maxwell boundary value problems

The time-harmonic Maxwell's equation (2.26) with a source term $\mathbf{F}$ is

$$\nabla \times (\nabla \times \mathbf{E}) - (\omega^2 \mu\varepsilon - \mathtt{i}\omega\mu\sigma)\mathbf{E} = \mathbf{F}. \tag{8.1}$$

In the case of propagation through a homogeneous medium, $\mu = \mu_0$, $\varepsilon = \varepsilon_0$, and $\sigma = \sigma_0$, where μ_0, ε_0, and σ_0 are all positive constants. Then, with the wavenumber $\tilde{\omega}$ defined as usual by $\tilde{\omega} = \omega\sqrt{\varepsilon_0\mu_0} > 0$, (8.1) becomes

$$\nabla \times (\nabla \times \mathbf{E}) - \left(\tilde{\omega}^2 - \mathtt{i}\tilde{\omega}\sigma_0\sqrt{\frac{\mu_0}{\varepsilon_0}}\right)\mathbf{E} = \mathbf{F}. \tag{8.2}$$

In this work, we consider domain decomposition preconditioning for finite element discretizations of the boundary value problems involving the PDE

$$\nabla \times (\nabla \times \mathbf{E}) - (\tilde{\omega}^2 + \mathtt{i}\xi)\mathbf{E} = \mathbf{F} \tag{8.3}$$

with the *absorption* parameter $\xi \in \mathbb{R} \setminus \{0\}$. Our theory is for the PDE (8.3) posed in a bounded Lipschitz polyhedron Ω with the PEC boundary conditions

$$\mathbf{E} \times \mathbf{n} = \mathbf{0}, \quad \text{on } \partial\Omega. \tag{8.4}$$

Nevertheless, we give numerical experiments in the case that the BVP has impedance boundary conditions

$$(\nabla \times \mathbf{E}) \times \mathbf{n} - \mathtt{i}\,\mathrm{sign}(\xi)\,\tilde{\omega}\,\mathbf{n} \times (\mathbf{E} \times \mathbf{n}) = \mathbf{0}, \quad \text{on } \partial\Omega, \tag{8.5}$$

and we discuss in Remark 8.13 the prospects of extending our theory to this case.

In the case that the absorption parameter ξ equals $-\tilde{\omega}\sigma_0\sqrt{\mu_0/\varepsilon_0}$, (8.3) becomes (8.2). However, our main motivation for considering discretizations of (8.3) is as preconditioners for the indefinite Maxwell problem

$$\nabla \times (\nabla \times \mathbf{E}) - \tilde{\omega}^2\mathbf{E} = \mathbf{F}, \tag{8.6}$$

with the same boundary conditions as prescribed for (8.3).

In the case of PEC boundary conditions, the solution of the BVP with the PDE (8.6) (i.e. with $\xi = 0$) is not unique for a countable set of values of $\tilde{\omega}$ (this values are called cavity eigenvalues or resonances of Ω, see, e.g, [85, Corollary 4.19]), whereas the solution of the BVP with the PDE (8.3) (i.e. with $\xi \neq 0$) is unique for every $\tilde{\omega}$ (see Corollary 8.5). In the case of impedance boundary conditions, the solutions to (8.6) and (8.3) are unique for all $\tilde{\omega}$.

8.2 The variational formulation and some preliminary results

8.2.1 Variational formulation

Let Ω be a bounded Lipschitz domain in $\mathbb{R}^3$; the vast majority of our results will be for the particular case that Ω is a Lipschitz polyhedron, but we indicate below when we make this assumption. Recall that

$$\mathbf{H}(\mathrm{curl}, \Omega) = \left\{\mathbf{v} \in \mathbf{L}^2(\Omega) : \nabla \times \mathbf{v} \in \mathbf{L}^2(\Omega)\right\}.$$

The theory concerns the PDE (8.3) in Ω with the PEC boundary condition $\mathbf{E} \times \mathbf{n} = \mathbf{0}$ on $\partial\Omega$. We therefore work in the space

$$\mathbf{H}_0(\mathrm{curl}, \Omega) = \{\mathbf{v} \in \mathbf{L}^2(\Omega), \nabla \times \mathbf{v} \in \mathbf{L}^2(\Omega), \mathbf{v} \times \mathbf{n} = \mathbf{0}\}.$$

We define the $\tilde{\omega}$-*weighted* inner product and norm on $\mathbf{H}_0(\mathrm{curl}, \Omega)$ by

$$(\mathbf{v}, \mathbf{w})_{\mathrm{curl},\tilde{\omega}} = (\nabla \times \mathbf{v}, \nabla \times \mathbf{w})_{\mathbf{L}^2(\Omega)} + \tilde{\omega}^2 (\mathbf{v}, \mathbf{w})_{\mathbf{L}^2(\Omega)} \quad \text{and} \quad \|\mathbf{v}\|_{\mathrm{curl},\tilde{\omega}} = (\mathbf{v}, \mathbf{v})^{1/2}_{\mathrm{curl},\tilde{\omega}}. \tag{8.7}$$

As in Section 2.2.5, the standard variational formulation of the BVP

$$\begin{cases} \nabla \times (\nabla \times \mathbf{E}) - (\tilde{\omega}^2 + \mathtt{i}\xi)\mathbf{E} &= \mathbf{F}, \quad \text{in } \Omega, \\ \mathbf{E} \times \mathbf{n} &= \mathbf{0}, \quad \text{on } \partial\Omega \end{cases} \tag{8.8}$$

is: given $\mathbf{F} \in \mathbf{L}^2(\Omega)$, $\xi \in \mathbb{R}$ and $\tilde{\omega} > 0$, find $\mathbf{E} \in \mathbf{H}_0(\mathrm{curl}, \Omega)$ such that

$$a_\xi(\mathbf{E}, \mathbf{v}) = F(\mathbf{v}) \ \text{ for all } \ \mathbf{v} \in \mathbf{H}_0(\mathrm{curl}, \Omega), \tag{8.9}$$

where

$$a_\xi(\mathbf{E}, \mathbf{v}) = \int_\Omega \nabla \times \mathbf{E} \cdot \overline{\nabla \times \mathbf{v}} - \int_\Omega (\tilde{\omega}^2 + \mathtt{i}\xi)\mathbf{E} \cdot \overline{\mathbf{v}} \tag{8.10}$$

and

$$F(\mathbf{v}) = \int_\Omega \mathbf{F} \cdot \overline{\mathbf{v}}. \tag{8.11}$$

When $\xi = 0$ and the PDE is (8.6), we simply write $a(\cdot, \cdot)$ instead of $a_\xi(\cdot, \cdot)$.

We will also need the *adjoint* of the sesquilinear form $a_\xi(\cdot, \cdot)$, denoted by $a^*_\xi(\cdot, \cdot)$, which is given by

$$a^*_\xi(\mathbf{E}, \mathbf{v}) = \int_\Omega \nabla \times \mathbf{E} \cdot \overline{\nabla \times \mathbf{v}} - \int_\Omega (\tilde{\omega}^2 - \mathtt{i}\xi)\mathbf{E} \cdot \overline{\mathbf{v}}, \tag{8.12}$$

and one can check that the variational problem (8.9) with $a_\xi(\cdot, \cdot)$ replaced by $a^*_\xi(\cdot, \cdot)$ is equivalent to the BVP

$$\begin{cases} \nabla \times (\nabla \times \mathbf{E}) - (\tilde{\omega}^2 - \mathtt{i}\xi)\mathbf{E} &= \mathbf{F}, \quad \text{in } \Omega, \\ \mathbf{E} \times \mathbf{n} &= \mathbf{0}, \quad \text{on } \partial\Omega, \end{cases} \tag{8.13}$$

we refer to this BVP as the *adjoint BVP*.

Now, with Ω a Lipschitz polyhedron, let $\mathcal{T}^h$ be a family of conforming tetrahedral meshes that are shape-regular as the mesh diameter $h \to 0$. We define our approximation space $\mathbf{Q}_h \subset \mathbf{H}_0(\mathrm{curl}, \Omega)$ as the curl-conforming finite element space defined in Chapter 3, of some fixed degree r, on the mesh $\mathcal{T}^h$ with functions whose tangential trace is zero on $\partial\Omega$. The Galerkin method applied to the variational problem (8.9) is

$$\text{find } \mathbf{E}_h \in \mathbf{Q}_h \ \text{ such that } \ a_\xi(\mathbf{E}_h, \mathbf{v}_h) = F(\mathbf{v}_h) \ \text{ for all } \ \mathbf{v}_h \in \mathbf{Q}_h.$$

Let $\mathcal{I}^h$ be a set of indices associated with the degrees of freedom for the subspace $\mathbf{Q}_h$. Then the Galerkin matrix A_ξ is defined by

$$(A_\xi)_{ij} = a_\xi(\mathbf{w}_i, \mathbf{w}_j), \quad i, j \in \mathcal{I}^h, \tag{8.14}$$

and the Galerkin method is equivalent to solving the linear system $A_\xi \mathbf{U} = \mathbf{G}$, where $G_i = F(\mathbf{w}_i)$.

8.2.2 Properties of the sesquilinear form

In this section we provide the key properties of the sesquilinear form a_ξ given in (8.10). This form depends on both parameters ξ and $\tilde{\omega}$, but only the first of these is reflected in the notation. We will assume throughout that

$$|\xi| \lesssim \tilde{\omega}^2. \tag{8.15}$$

Here the notation $A \lesssim B$ means that A/B is bounded above by a constant independent of $\tilde{\omega}$, ξ, and mesh diameters $h, H_{\text{sub}}, H_{\text{cs}}$ (the latter two introduced below). We write $A \sim B$ when $A \lesssim B$ and $B \lesssim A$.

The continuity result follows from the Cauchy-Schwarz inequality.

Lemma 8.1 (Continuity of $a_\xi(\cdot,\cdot)$ and $a_\xi^*(\cdot,\cdot)$)**.**

$$|a_\xi(\mathbf{v},\mathbf{w})| \lesssim \|\mathbf{v}\|_{\text{curl},\tilde{\omega}} \, \|\mathbf{w}\|_{\text{curl},\tilde{\omega}} \tag{8.16}$$

for all $\tilde{\omega} > 0$ and $\mathbf{v}, \mathbf{w} \in \mathbf{H}_0(\text{curl}, \Omega)$. Furthermore, the inequality (8.16) *holds with a_ξ replaced by a_ξ^*.*

We now give a result about the coercivity of $a_\xi(\cdot,\cdot)$. Before stating this result, we need to define $\sqrt{\tilde{\omega}^2 + \mathtt{i}\xi}$, taking care to allow ξ to be positive or negative. Indeed, we need to consider both positive and negative ξ since, whichever choice we make for the problem (8.8), the other forms the adjoint problem, and we need estimates on the solutions and sesquilinear forms for both problems (in particular, this is essential for analyzing both left and right preconditioning).

Definition 8.2. $z(\tilde{\omega}, \xi) := \sqrt{\tilde{\omega}^2 + \mathtt{i}\xi}$ where the square root is defined with the branch cut on the positive real axis. Note that this definition implies that, when $\xi \neq 0$,

$$\text{Im}(z) > 0, \quad \text{sign}(\xi)\,\text{Re}(z) > 0, \quad \text{and} \quad z(\tilde{\omega}, -\xi) = -\overline{z(\tilde{\omega}, \xi)}. \tag{8.17}$$

Lemma 8.3. [62, Proposition 2.3] *With $z(\tilde{\omega}, \xi)$ defined above, for all $\tilde{\omega} > 0$,*

$$|z| \sim \tilde{\omega} \quad \textit{and} \quad \frac{\text{Im}(z)}{|z|} \sim \frac{|\xi|}{\tilde{\omega}^2}. \tag{8.18}$$

Lemma 8.4 (Coercivity of $a_\xi(\cdot,\cdot)$ and $a_\xi^*(\cdot,\cdot)$)**.** *There exists a constant $\rho > 0$ independent of $\tilde{\omega}$ and ξ such that*

$$|a_\xi(\mathbf{v},\mathbf{v})| \;\geq\; \text{Im}\left(\Theta a_\xi(\mathbf{v},\mathbf{v})\right) \;\geq\; \rho\left(\frac{|\xi|}{\tilde{\omega}^2}\right) \|\mathbf{v}\|^2_{\text{curl},\tilde{\omega}} \tag{8.19}$$

for all $\tilde{\omega} > 0$ and $v \in \mathbf{H}_0(\text{curl}, \Omega)$, where $\Theta = -\overline{z}/|z|$. Furthermore the inequality (8.19) *holds with $a_\xi(\cdot,\cdot)$ replaced by $a_\xi^*(\cdot,\cdot)$.*

Proof. Writing $z = p + iq$ and using the definition of a_ξ, we have

$$a_\xi(\mathbf{v},\mathbf{v}) = \big\|\nabla \times \mathbf{v}\big\|^2_{\mathbf{L}^2(\Omega)} - (p + \mathtt{i}q)^2 \big\|\mathbf{v}\big\|^2_{\mathbf{L}^2(\Omega)}.$$

Therefore

$$\text{Im}\left[-(p - \mathtt{i}q) a_\xi(\mathbf{v},\mathbf{v})\right] = q\big\|\nabla \times \mathbf{v}\big\|^2_{L^2(\Omega)} + q(p^2 + q^2)\big\|\mathbf{v}\big\|^2_{\mathbf{L}^2(\Omega)}.$$

Hence, dividing through by $|z| = \sqrt{p^2+q^2}$ and setting $\Theta = -\overline{z}/|z|$, we have

$$\operatorname{Im}\left[\Theta a_\xi(\mathbf{v},\mathbf{v})\right] \;=\; \frac{\operatorname{Im}(z)}{|z|}\left[\left\|\nabla\times\mathbf{v}\right\|^2_{\mathbf{L}^2(\Omega)} + |z|^2\left\|\mathbf{v}\right\|^2_{\mathbf{L}^2(\Omega)}\right].$$

The second inequality in (8.19) then follows from the two estimates in (8.18). The first one holds since $|\Theta| = 1$.

The result about the adjoint form $a^*_\xi(\cdot,\cdot)$ follows immediately after noticing that the third equation in (8.17) implies that $\operatorname{Im} z(\tilde\omega,-\xi) = \operatorname{Im} z(\tilde\omega,\xi)$. □

Corollary 8.5 (Bound on the solutions of (8.8) and (8.13) via Lax–Milgram)**.** *The solution of the variational problem* (8.9) *exists, is unique, and satisfies the bound.*

$$\|\mathbf{E}\|_{\operatorname{curl},\tilde\omega} \lesssim \left(\frac{\tilde\omega}{|\xi|}\right)\left\|\mathbf{F}\right\|_{\mathbf{L}^2(\Omega)} \tag{8.20}$$

for all $\tilde\omega > 0$*. The same is true if the sesquilinear form* $a_\xi(\cdot,\cdot)$ *in* (8.9) *is replaced by* $a^*_\xi(\cdot,\cdot)$ *given by* (8.12)*.*

Proof. The Lax–Milgram theorem, the continuity result of Lemma 8.1, and the coercivity result of Lemma 8.4 imply that the solution of the variational problem (8.9) satisfies

$$\|\mathbf{E}\|_{\operatorname{curl},\tilde\omega} \lesssim \left(\frac{\tilde\omega^2}{|\xi|}\right)\|F\|_{(\operatorname{curl},\tilde\omega)'},$$

where $\|\cdot\|_{(\operatorname{curl},\tilde\omega)'}$ denotes the norm on the dual-space of $\mathbf{H}_0(\operatorname{curl},\Omega)$ defined by

$$\|F\|_{(\operatorname{curl},\tilde\omega)'} := \sup_{\mathbf{v}\in\mathbf{H}_0(\operatorname{curl},\Omega)\setminus\{\mathbf{0}\}}\frac{|F(\mathbf{v})|}{\|\mathbf{v}\|_{\operatorname{curl},\tilde\omega}}.$$

From the definition (8.11),

$$|F(\mathbf{v})| \le \left\|\mathbf{F}\right\|_{\mathbf{L}^2(\Omega)}\left\|\mathbf{v}\right\|_{\mathbf{L}^2(\Omega)} \le \frac{1}{\tilde\omega}\left\|\mathbf{F}\right\|_{\mathbf{L}^2(\Omega)}\|\mathbf{v}\|_{\operatorname{curl},\tilde\omega}$$

and the inequality (8.20) follows. The result about the solution to the adjoint problem follows in a similar way. □

8.2.3 Regularity of the BVP and its adjoint

In order to estimate the approximation properties of the coarse grid operator in Lemma 8.17 below, we need $\mathbf{H}^1$-regularity of both $\mathbf{E}$ and $\nabla\times\mathbf{E}$.

Assumption 8.6 ($\tilde\omega$- and ξ-explicit $\mathbf{H}^1$ regularity)**.** The domain Ω is such that, given $\mathbf{F}\in L^2(\Omega)\cap\mathbf{H}(\operatorname{div}^0,\Omega) := \left\{\mathbf{u}\in\mathbf{L}^2(\Omega) : \nabla\cdot\mathbf{u} = 0\right\}$, $\xi\in\mathbb{R}\setminus\{0\}$, and $\tilde\omega > 0$, the solution of the BVP (8.8) is such that $\nabla\times\mathbf{E}\in\mathbf{H}^1(\Omega)$ and $\mathbf{E}\in\mathbf{H}^1(\Omega)$. Moreover, if ξ satisfies (8.15) then, given $\tilde\omega_0 > 0$,

$$\|\nabla\times\mathbf{E}\|_{\mathbf{H}^1(\Omega)} + \tilde\omega\,\|\mathbf{E}\|_{\mathbf{H}^1(\Omega)} \lesssim \tilde\omega\left(\left\|\nabla\times\mathbf{E}\right\|_{\mathbf{L}^2(\Omega)} + \tilde\omega\left\|\mathbf{E}\right\|_{\mathbf{L}^2(\Omega)}\right) + \|\mathbf{F}\|_{\mathbf{L}^2(\Omega)}, \tag{8.21}$$

for all $\tilde\omega \ge \tilde\omega_0$.

The bound (8.21) can be viewed as a rigorous expression of the idea that taking a derivative of a solution of the PDE (8.6) incurs a power of $\tilde\omega$. A quite long and involved proof shows that Assumption 8.6 holds in the following situations.

Lemma 8.7. *If Ω is* either *a bounded $C^{1,1}$ domain* or *a convex polyhedron, then Assumption 8.6 holds.*

If Assumption 8.6 holds, the bound (8.20) from the Lax–Milgram theorem immediately implies the following corollary.

Corollary 8.8. *If Assumption 8.6 holds, and $\mathbf{E}$ is the solution to* either *the BVP* (8.8) or *the adjoint BVP* (8.13) *with $\mathbf{F} \in \mathbf{L}^2(\Omega) \cap \mathbf{H}(\mathrm{div}^0, \Omega)$, $\xi \in \mathbb{R} \setminus \{0\}$ satisfying* (8.15), *and $\tilde{\omega} > 0$, then, given $\tilde{\omega}_0 > 0$,*

$$\|\nabla \times \mathbf{E}\|_{\mathbf{H}^1(\Omega)} + \tilde{\omega}\, \|\mathbf{E}\|_{\mathbf{H}^1(\Omega)} \lesssim \left(\frac{\tilde{\omega}^2}{|\xi|}\right) \|\mathbf{F}\|_{\mathbf{L}^2(\Omega)}\,, \tag{8.22}$$

for all $\tilde{\omega} \geq \tilde{\omega}_0$.

8.3 Domain decomposition set-up

To define appropriate subspaces of the edge finite element space $\mathbf{Q}_h \subset \mathbf{H}_0(\mathrm{curl}, \Omega)$, we start with a collection of open subsets $\{\hat{\Omega}_\ell : \ell = 1, \ldots, N\}$ of $\mathbb{R}^d$ that form an overlapping cover of $\overline{\Omega}$, and we set $\Omega_\ell = \hat{\Omega}_\ell \cap \overline{\Omega}$. Each $\overline{\Omega}_\ell$ is assumed to be non-empty and to consist of a union of elements of the mesh $\mathcal{T}_h$. Then, for each $\ell = 1, \ldots, N$, we set

$$\mathbf{Q}_h^\ell := \mathbf{Q}_h \cap \mathbf{H}_0(\mathrm{curl}, \Omega_\ell), \tag{8.23}$$

i.e. the tangential traces of elements of $\mathbf{Q}_h^\ell$ vanish on the internal boundary $\partial\Omega_\ell \backslash \partial\Omega$ (as well as on $\partial\Omega_\ell \cap \partial\Omega$). In writing the definition of $\mathbf{Q}_h^\ell$, we are using the fact that $\mathbf{H}_0(\mathrm{curl}, \Omega_\ell)$ can be considered as a subset of $\mathbf{H}_0(\mathrm{curl}, \Omega)$ by *extending* functions in $\mathbf{H}_0(\mathrm{curl}, \Omega_\ell)$ *by zero* (such extensions are in $\mathbf{H}_0(\mathrm{curl}, \Omega)$ by, e.g., [85, Lemma 5.3]). We make the following assumptions on the subdomains:

1. *Shape regularity:* the subdomains are shape-regular Lipschitz polyhedra of diameter H_{sub}, in the sense that the volume is of order H_{sub}^3 and surface area of order H_{sub}^2 (with omitted constants independent of all parameters).

2. *Uniform overlap of order δ:* For each $\ell = 1, \ldots, N$, let $\mathring{\Omega}_\ell$ denote the part of Ω_ℓ that is not overlapped by any other subdomains, and for $\mu > 0$ let $\Omega_{\ell,\mu}$ denote the set of points in Ω_ℓ that are a distance no more than μ from the boundary $\partial\Omega_\ell$. Then we assume that for some $\delta > 0$ and some $0 < c < 1$ fixed,

$$\Omega_{\ell,c\delta} \subset \Omega_\ell \backslash \mathring{\Omega}_\ell \subset \Omega_{\ell,\delta};$$

   the case $\delta \sim H_{\mathrm{sub}}$ is called *generous overlap.*

3. *Finite overlap assumption:* as $h, H_{\mathrm{sub}} \to 0$,

$$\#\Lambda(\ell) \lesssim 1, \quad \text{where} \quad \Lambda(\ell) = \left\{\ell' : \Omega_\ell \cap \Omega_{\ell'} \neq \emptyset\right\}. \tag{8.24}$$

Let $\mathcal{I}^h(\Omega_\ell)$ be the set of indices of the degrees of freedom whose support is contained in Ω_ℓ. We then have that $\mathcal{I}^h = \bigcup_{\ell=1}^N \mathcal{I}^h(\Omega_\ell)$. We define the restriction matrices $(R^\ell)_{\ell=1}^N$ and partition of unity matrices $(D^\ell)_{\ell=1}^N$ as in §5.2 and §5.3.1.

For the coarse space, let $\{\mathcal{T}^{H_{\mathrm{cs}}}\}$ be a sequence of shape-regular, tetrahedral meshes on $\overline{\Omega}$, with mesh diameter H_{cs}. Let $\mathcal{I}^H$ be an index set for the degrees of freedom on the coarse mesh. The space $\mathbf{Q}^0$ is the curl-conforming finite element space on the mesh $\mathcal{T}^{H_{\mathrm{cs}}}$

with functions whose tangential trace is zero on $\partial\mathcal{T}^{H_{\rm cs}}$. Similarly to §7.3.1, the coarse space matrix Z is the matrix corresponding to the interpolation operator $\mathcal{I}^0\colon \mathbf{Q}^0 \to \mathbf{Q}_h$ from the coarse grid finite element space to the fine grid finite element space, and we set the "restriction" matrix $R^0 = Z^T$.

With the restriction matrices $(R^\ell)_{\ell=0}^N$ we define

$$A_\xi^\ell := R^\ell A_\xi (R^\ell)^T. \tag{8.25}$$

For $\ell = 1, \ldots, N$, the matrix A_ξ^ℓ is then just the minor of A_ξ corresponding to rows and columns taken from $\mathcal{I}^h(\Omega_\ell)$ and the matrix A_ξ^0 is the Galerkin matrix for the variational problem (8.9) discretized in $\mathbf{Q}^0$. The coercivity result Lemma 8.4 implies that the matrices A_ξ^ℓ, $\ell = 0, \ldots, N$, are invertible for all mesh sizes h and all choices of $\xi \neq 0$. Indeed, if $A_\xi^0 \mathbf{V} = \mathbf{0}$, where $\mathbf{V}$ is a vector such that $\mathbf{v}_H := \sum_{p\in\mathcal{I}^H} V_p \mathbf{w}_p^H \in \mathbf{Q}^0$, then $0 = \mathbf{V}^* A_\xi^0 \mathbf{V} = a_\xi(\mathbf{v}_H, \mathbf{v}_H)$. Therefore,

$$0 = |a_\xi(\mathbf{v}_H, \mathbf{v}_H)| \geq \rho\left(\frac{|\xi|}{\tilde{\omega}^2}\right) \|\mathbf{v}_H\|^2_{\mathrm{curl},\tilde{\omega}},$$

which immediately implies $\mathbf{v}_H = \mathbf{0}$, and thus $\mathbf{V} = \mathbf{0}$. Similar arguments apply to A_ξ^ℓ and to the adjoints $(A_\xi^\ell)^*$, $\ell = 0, \ldots, N$.

As in §7.3 we combine the one-level preconditioner with the coarse correction using an additive or hybrid formula to write a two-level preconditioner. The theory concerns the *two-level Additive Schwarz preconditioner* for A_ξ, based on the one-level AS preconditioner with an additive correction:

$$M_{\xi,\mathrm{AS}}^{-1} = \sum_{\ell=0}^{N} (R^\ell)^T (A_\xi^\ell)^{-1} R^\ell \tag{8.26}$$

(note that in this Chapter we omit the subscript 2 standing for two-level).

In the numerical experiments we will also consider:

- the two-level Restricted Additive Schwarz preconditioner based on the one-level RAS preconditioner with an additive correction:

$$M_{\xi,\mathrm{RAS}}^{-1} = \sum_{\ell=1}^{N} (R^\ell)^T D_\ell (A_\xi^\ell)^{-1} R^\ell + \Xi, \quad \Xi = (R^0)^T (A_\xi^0)^{-1} R^0, \tag{8.27}$$

- $M_{\xi,\mathrm{ImpRAS}}^{-1}$, which is similar to $M_{\xi,\mathrm{RAS}}^{-1}$ but it is based on the one-level ORAS preconditioner (in the terminology of the Helmholtz paper [62] the prefix O for Optimized is replaced with Imp, for impedance transmission conditions),

- the two-level hybrid version of RAS

$$M_{\xi,\mathrm{HRAS}}^{-1} = (I - \Xi A_\xi)\left(\sum_{\ell=1}^{N} (R^\ell)^T D_\ell (A_\xi^\ell)^{-1} R^\ell\right)(I - A_\xi \Xi) + \Xi, \tag{8.28}$$

- $M_{\xi,\mathrm{ImpHRAS}}^{-1}$, the two-level hybrid version of ImpRAS.

8.3.1 Discrete Helmholtz decomposition and associated results

Recall that $\mathbf{H}(\mathrm{div},\Omega)$ is defined by

$$\mathbf{H}(\mathrm{div},\Omega) = \left\{\mathbf{u} \in \mathbf{L}^2(\Omega) : \nabla\cdot\mathbf{u} \in L^2(\Omega)\right\}$$

and $\mathbf{H}_0(\mathrm{div},\Omega)$ by

$$\mathbf{H}_0(\mathrm{div},\Omega) = \left\{\mathbf{u} \in \mathbf{L}^2(\Omega) : \nabla\cdot\mathbf{u} \in L^2(\Omega) \text{ and } \mathbf{u}\cdot\mathbf{n} = 0 \text{ on } \partial\Omega\right\}$$

(recall that the normal trace is well-defined on $\mathbf{H}(\mathrm{div},\Omega)$ by, e.g., [85, Thereom 3.24]).

Let $\mathbf{V}_h$ denote the Raviart-Thomas finite element subspaces of $\mathbf{H}_0(\mathrm{div},\Omega)$ of index r based on the fine mesh $\mathcal{T}^h$. Let W_h denote the subspace of $H^1_0(\Omega)$ consisting of piecewise polynomials of degree $r+1$, also on the fine mesh $\mathcal{T}^h$. For Ω simply connected, we have the *discrete Helmholtz decomposition*

$$\mathbf{Q}_h = \mathbf{curl}_h\mathbf{V}_h \oplus \nabla W_h, \tag{8.29}$$

see, e.g., [5, §2, in particular Remark 2.1] [85, §7.2.1, in particular Lemma 7.4], where $\mathbf{curl}_h$ is the $\mathbf{L}^2$-adjoint of the map $\mathbf{curl} : \mathbf{Q}_h \to \mathbf{V}_h$, and the decomposition is orthogonal both in $\mathbf{L}^2(\Omega)$ and in $\mathbf{H}(\mathrm{curl},\Omega)$.

We define $\mathbf{V}^0_H$ and W^0_H in the same way as $\mathbf{V}_h$ and W_h, but using the coarse mesh $\{\mathcal{T}^H\}$. We also set

$$\mathbf{V}^\ell_h := \mathbf{V}_h \cap \mathbf{H}_0(\mathrm{div},\Omega^\ell) \quad \text{and} \quad W^\ell_h := W_h \cap H^1_0(\Omega^\ell) \quad \text{for } \ell = 1,\dots,N,$$

where fields on Ω^ℓ are identified as fields on Ω by extension by zero. We then have the analogue of the decomposition (8.29):

$$\mathbf{Q}^\ell_h = \mathbf{curl}^\ell_h\mathbf{V}^\ell_h \oplus \nabla W^\ell_h, \tag{8.30}$$

where $\mathbf{curl}^\ell_h$ is the $\mathbf{L}^2$-adjoint of the map $\mathbf{curl} : \mathbf{Q}^\ell_h \to \mathbf{V}^\ell_h$.

For fields in $\mathbf{curl}^\ell_h\mathbf{V}^\ell_h$ we have the following Poincaré–Friedrichs inequality from [60].

Lemma 8.9. (Poincaré–Friedrichs type-inequality [60, Lemma 4.1]) *If Ω_ℓ is polyhedral,*

$$\|\mathbf{q}\|_{\mathbf{L}^2(\Omega_\ell)} \lesssim H_{\mathrm{sub}}\,\|\nabla\times\mathbf{q}\|_{\mathbf{L}^2(\Omega_\ell)} \quad \textit{for all } \mathbf{q} \in \mathbf{curl}^\ell_h\mathbf{V}^\ell_h, \tag{8.31}$$

where the omitted constant is independent of h and H_{sub}.

Finally, we recall the following result from, e.g., [60, Equation 4.10] (with a similar result in [5, Lemma 5.2]).

Lemma 8.10. *If Ω is* either *a convex polyhedron* or *$C^{1,1}$ then, given $\mathbf{q}_h \in \mathbf{curl}_h\mathbf{V}_h$, there exists a unique field in $\mathbf{H}_0(\mathrm{curl},\Omega)$, which we denote by $\mathbf{Sq}_h$, such that*

$$\nabla\times(\mathbf{Sq}_h) = \nabla\times\mathbf{q}_h \quad \textit{and} \quad \nabla\cdot\mathbf{Sq}_h = 0.$$

Furthermore,

$$\|\mathbf{q}_h - \mathbf{Sq}_h\|_{\mathbf{L}^2(\Omega)} \lesssim h\,\|\nabla\times\mathbf{q}_h\|_{\mathbf{L}^2(\Omega)}\,. \tag{8.32}$$

The key point from Lemma 8.10 is that although $\mathbf{q}_h$ is not divergence-free, $\mathbf{Sq}_h$ provides an approximation to $\mathbf{q}_h$ that is divergence-free and has the same curl.

8.4 Theory of two-level Additive Schwarz methods

The following theory establishes rigorous convergence estimates for the preconditioner (8.26) applied to A_ξ. While Sections 8.4.1, 8.4.2, and 8.4.4 are very similar to the Helmholtz theory in [62], Section 8.4.3 is very different, in that we need to use and adapt the arguments of [60] (see the discussion at the beginning of Section 8.4.3).

8.4.1 Stable splitting and associated results

The first lemma is a "stable splitting" property in the $\tilde{\omega}$-weighted $\mathbf{H}_0(\mathrm{curl},\Omega)$ norm.

Lemma 8.11 (Stable splitting in the curl, $\tilde{\omega}$-norm)**.** *For all $\mathbf{v}_h \in \mathbf{Q}_h$, there exist $\mathbf{v}^\ell \in \mathbf{Q}^\ell$ for each $\ell = 0,\dots,N$ such that*

$$\mathbf{v}_h = \sum_{\ell=0}^{N} \mathbf{v}^\ell \quad \textit{and} \quad \sum_{\ell=0}^{N} \left\|\mathbf{v}^\ell\right\|_{\mathrm{curl},\tilde{\omega}}^2 \lesssim \left(1+\left(\frac{H_{\mathrm{cs}}}{\delta}\right)^2\right) \|\mathbf{v}_h\|_{\mathrm{curl},\tilde{\omega}}^2. \tag{8.33}$$

The next lemma is a kind of converse to Lemma 8.11. Here the energy of a sum of components is estimated above by the sum of the energies.

Lemma 8.12. *For all choices of $\mathbf{v}^\ell \in \mathbf{Q}^\ell$, $\ell = 0,\dots,N$, we have*

$$\left\| \sum_{\ell=0}^{N} \mathbf{v}^\ell \right\|_{\mathrm{curl},\tilde{\omega}}^2 \lesssim \sum_{\ell=0}^{N} \left\|\mathbf{v}^\ell\right\|_{\mathrm{curl},\tilde{\omega}}^2 . \tag{8.34}$$

Remark 8.13 (Impedance boundary conditions)**.** The main obstacle to extending the theory in this work to the BVP (8.8) with the PEC boundary condition replaced by the impedance boundary condition (8.5) on $\partial\Omega$ is that the rigorous Hilbert space for the variational formulation of the impedance problem is not $\mathbf{H}(\mathrm{curl},\Omega)$, but $\mathbf{H}_{\mathrm{imp}}(\mathrm{curl},\Omega) = \{\mathbf{v} \in \mathbf{H}(\mathrm{curl},\Omega), \mathbf{v}\times\mathbf{n} \in \mathbf{L}_t^2(\partial\Omega)\}$ (see, e.g., [85, §3.8]). The first step towards extending the theory to the impedance BVP would be to establish a stable-splitting in the norm on $\mathbf{H}_{\mathrm{imp}}(\mathrm{curl},\Omega)$.

8.4.2 Definition of the projection operators and the path towards the bound on the field of values

Now for each $\ell = 0,\dots,N$, we define linear *projection operators* $\mathbf{T}_\xi^\ell : \mathbf{H}_0(\mathrm{curl},\Omega) \to \mathbf{Q}^\ell$ as follows. For each $\mathbf{v} \in \mathbf{H}_0(\mathrm{curl},\Omega)$, $\mathbf{T}_\xi^\ell\mathbf{v}$ is defined to be the unique solution of the equation

$$a_\xi(\mathbf{T}_\xi^\ell\mathbf{v}, \mathbf{w}_h^\ell) \;=\; a_\xi(\mathbf{v}, \mathbf{w}_h^\ell), \quad \mathbf{w}_h^\ell \in \mathbf{Q}^\ell. \tag{8.35}$$

Recall from the discussion underneath (8.23) that $\mathbf{Q}_h^\ell$ can be considered as a subspace of $\mathbf{H}_0(\mathrm{curl},\Omega)$ *by extension by zero*, and we can therefore consider $\mathbf{T}_\xi^\ell\mathbf{v}$ as an element of $\mathbf{H}_0(\mathrm{curl},\Omega)$ with support on Ω_ℓ. We then define

$$\mathbf{T}_\xi = \sum_{\ell=0}^{N} \mathbf{T}_\xi^\ell.$$

We will show in Theorem 8.22 below that the matrix representation of $\mathbf{T}_\xi$ corresponds to the *action of the preconditioner* (8.26) *on the matrix* A_ξ.

Our goal is to bound from above the norm of the operator $\mathbf{T}_\xi$ and from below its *field of values* (or *numerical range*)

$$\frac{(\mathbf{v}_h, \mathbf{T}_\xi \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}}}{\|\mathbf{v}_h\|^2_{\mathrm{curl},\tilde{\omega}}} \quad \text{for all } \mathbf{v}_h \in \mathbf{Q}_h. \tag{8.36}$$

Note that the field of values is computed with respect to the wavenumber-dependent $(\cdot,\cdot)_{\mathrm{curl},\tilde{\omega}}$ inner product.

The upper bound is given by the following theorem.

Theorem 8.14 (Upper bound on the norm of $\mathbf{T}_\xi$)**.**

$$\|\mathbf{T}_\xi \mathbf{v}_h\|_{\mathrm{curl},\tilde{\omega}} \lesssim \left(\frac{\tilde{\omega}^2}{|\xi|}\right) \|\mathbf{v}_h\|_{\mathrm{curl},\tilde{\omega}} \quad \textit{for all} \quad \mathbf{v}_h \in \mathbf{Q}_h.$$

The next two results are two of the three ingredients we use to obtain a bound on the field of values from below. (The third ingredient is provided in §8.4.3, and the bound is proved in §8.4.4.)

Lemma 8.15.

$$\sum_{\ell=0}^{N} \|\mathbf{T}_\xi^\ell \mathbf{v}_h\|^2_{\mathrm{curl},\tilde{\omega}} \gtrsim \left(1 + \left(\frac{H_{\mathrm{cs}}}{\delta}\right)^2\right)^{-1} \left(\frac{|\xi|}{\tilde{\omega}^2}\right)^2 \|\mathbf{v}_h\|^2_{\mathrm{curl},\tilde{\omega}} \quad \textit{for all } \mathbf{v}_h \in \mathbf{Q}_h. \tag{8.37}$$

The bound (8.37) in Lemma 8.15 means that to bound the field of values (8.36) from below it is sufficient to bound $(\mathbf{v}_h, \mathbf{T}_\xi \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}}$ below by $\sum_{\ell=0}^N \|\mathbf{T}_\xi^\ell \mathbf{v}_h\|^2_{\mathrm{curl},\tilde{\omega}}$. The next lemma expresses $(\mathbf{v}_h, \mathbf{T}_\xi \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}}$ in terms of $\sum_{\ell=0}^N \|\mathbf{T}_\xi^\ell \mathbf{v}_h\|^2_{\mathrm{curl},\tilde{\omega}}$ plus a sum of "remainder" terms $R_\xi^\ell(\mathbf{v}_h)$.

Lemma 8.16. *For $\ell = 0, \ldots, N$, set*

$$R_\xi^\ell(\mathbf{v}_h) := \big((\mathbf{I} - \mathbf{T}_\xi^\ell)\mathbf{v}_h, \mathbf{T}_\xi^\ell \mathbf{v}_h\big)_{\mathrm{curl},\tilde{\omega}}.$$

Then

$$(\mathbf{v}_h, \mathbf{T}_\xi \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}} = \sum_{\ell=0}^{N} \|\mathbf{T}_\xi^\ell \mathbf{v}_h\|^2_{\mathrm{curl},\tilde{\omega}} + \sum_{\ell=0}^{N} R_\xi^\ell(\mathbf{v}_h). \tag{8.38}$$

and

$$R_\xi^\ell(\mathbf{v}_h) = (2\tilde{\omega}^2 + \mathtt{i}\xi)\big((\mathbf{I} - \mathbf{T}_\xi^\ell)\mathbf{v}_h, \mathbf{T}_\xi^\ell \mathbf{v}_h\big)_{\mathbf{L}^2(\Omega_\ell)}. \tag{8.39}$$

8.4.3 The key result about the projection operators adapted from [60]

Lemma 8.16 above shows that we can bound the field of values of $\mathbf{T}_\xi$ from below, provided that we can get good estimates for the remainder terms

$$R_\xi^\ell(\mathbf{v}_h) = (2\tilde{\omega}^2 + \mathtt{i}\xi)\big((\mathbf{I} - \mathbf{T}_\xi^\ell)\mathbf{v}_h, \mathbf{T}_\xi^\ell \mathbf{v}_h\big)_{\mathbf{L}^2(\Omega_\ell)}. \tag{8.40}$$

It is at this point that our Maxwell theory deviates substantially from the Helmholtz theory in [62]. There, the Cauchy-Schwarz inequality was used on (8.40),

(a) the analogue of $\|(\mathbf{I} - \mathbf{T}_\xi^0)\mathbf{v}_h\|_{\mathbf{L}^2(\Omega_\ell)}$ was estimated using a duality argument, and

(b) the analogue of $\|\mathbf{T}_\xi^\ell \mathbf{v}_h\|_{\mathbf{L}^2(\Omega_\ell)}$, $\ell = 1, \ldots, N$, was estimated using the standard scalar Poincaré–Friedrichs inequality.

This approach does not immediately carry over to the Maxwell case because (a) duality arguments for Maxwell's equations require divergence-free right-hand sides (since we want to use the bound (8.21)), and (b) the appropriate analogue of the Poincaré–Friedrichs inequality does not hold for gradient fields (see Lemma 8.9 above).

Nevertheless, one of the main technical results obtained by Gopalakrishnan and Pasciak, [60, Lemma 4.3], involves estimating $((\mathbf{I} - \mathbf{T}_\xi^\ell)\mathbf{v}_h, \mathbf{w}_h^\ell)_{\mathbf{L}^2(\Omega_\ell)}$ for $\mathbf{w}_h^\ell \in \mathbf{Q}^\ell$ and $\ell = 0, \ldots, N$. Lemma 8.18 below is essentially this result, adapted to our situation by (i) making everything explicit in $\tilde{\omega}$ and ξ, and (ii) using coercivity of $a_\xi(\cdot,\cdot)$ instead of an error estimate on the Galerkin solution in the duality argument.

Before stating this key result, we need the following result about approximability of the adjoint problem on the coarse grid.

Lemma 8.17 (Coarse-grid approximability of the adjoint problem)**.** *If Assumption 8.6 holds, then, if* $\mathbf{E}$ *is the solution of the adjoint problem* (8.13) *with* $\mathbf{F} \in \mathbf{L}^2(\Omega)$ *and* $\nabla \cdot \mathbf{F} = 0$*, then*

$$\inf_{\boldsymbol{\phi}_H^0 \in \mathbf{Q}_H^0} \left\| \mathbf{E} - \boldsymbol{\phi}_H^0 \right\|_{\mathrm{curl},\tilde{\omega}} \lesssim H_{\mathrm{cs}} \left(\frac{\tilde{\omega}^2}{|\xi|} \right) \|\mathbf{F}\|_{\mathbf{L}^2(\Omega)} . \tag{8.41}$$

We now state the key result that will allow us to estimate the remainder terms $R_\xi^\ell(\mathbf{v}_h)$.

Lemma 8.18. ($\tilde{\omega}$- and ξ-explicit version of [60, Lemma 4.3])
(i) For any $\mathbf{v}_h \in \mathbf{Q}_h$ *and* $\mathbf{w}_h^\ell \in \mathbf{Q}_h^\ell$*,*

$$\left((\mathbf{I} - \mathbf{T}_\xi^\ell)\mathbf{v}_h, \mathbf{w}_h^\ell\right)_{\mathbf{L}^2(\Omega_\ell)} \lesssim H_{\mathrm{sub}} \|(\mathbf{I} - \mathbf{T}_\xi^\ell)\mathbf{v}_h\|_{\mathbf{L}^2(\Omega_\ell)} \|\nabla \times \mathbf{w}_h^\ell\|_{\mathbf{L}^2(\Omega_\ell)} \tag{8.42}$$

for $\ell = 1, \ldots, N$*,*
(ii) If Assumption 8.6 holds, then, given $\tilde{\omega}_0 > 0$*, and for any* $\mathbf{v}_h \in \mathbf{Q}_h$ *and* $\mathbf{w}_H^0 \in \mathbf{Q}_H^0$*,*

$$\left((\mathbf{I} - \mathbf{T}_\xi^0)\mathbf{v}_h, \mathbf{w}_H^0\right)_{\mathbf{L}^2(\Omega)} \lesssim H_{\mathrm{cs}} \left(\frac{\tilde{\omega}}{|\xi|} \right) \|(\mathbf{I} - \mathbf{T}_\xi^0)\mathbf{v}_h\|_{\mathrm{curl},\tilde{\omega}} \|\mathbf{w}_H^0\|_{\mathrm{curl},\tilde{\omega}}, \tag{8.43}$$

for all $\tilde{\omega} \geq \tilde{\omega}_0$*.*

8.4.4 Bound on the field of values

We can now give the estimates on the remainder terms $R_\xi^\ell(\mathbf{v}_h)$.

Lemma 8.19 (Bound on $R_\xi^0(\mathbf{v}_h)$)**.** *Under Assumption 8.6, given* $\tilde{\omega}_0 > 0$ *and for any* $\gamma \geq 0$ *and any* $\mathbf{v}_h \in \mathbf{Q}_h$*,*

$$\left|R_\xi^0(\mathbf{v}_h)\right| \lesssim \tilde{\omega} H_{\mathrm{cs}} \left(\frac{\tilde{\omega}^2}{|\xi|} \right) \left[\left(\frac{\tilde{\omega}^2}{|\xi|} \right)^{\gamma} \left\| \mathbf{T}_\xi^0 \mathbf{v}_h \right\|_{\mathrm{curl},\tilde{\omega}}^2 + \left(\frac{\tilde{\omega}^2}{|\xi|} \right)^{-\gamma} \|\mathbf{v}_h\|_{\mathrm{curl},\tilde{\omega}}^2 \right], \tag{8.44}$$

for all $\tilde{\omega} \geq \tilde{\omega}_0$*.*

Lemma 8.20 (Bound on $\sum R_\xi^\ell(\mathbf{v}_h)$)**.** *For any* $\gamma' \geq 0$ *and any* $\mathbf{v}_h \in \mathbf{Q}_h$*,*

$$\sum_{\ell=1}^{N} \left|R_\xi^\ell(\mathbf{v}_h)\right| \lesssim \tilde{\omega} H_{\mathrm{sub}} \left[\left(\frac{\tilde{\omega}^2}{|\xi|} \right)^{\gamma'} \sum_{\ell=1}^{N} \left\| \mathbf{T}_\xi^\ell \mathbf{v}_h \right\|_{\mathrm{curl},\tilde{\omega}}^2 + \left(\frac{\tilde{\omega}^2}{|\xi|} \right)^{-\gamma'} \|\mathbf{v}_h\|_{\mathrm{curl},\tilde{\omega}}^2 \right]. \tag{8.45}$$

Finally, the bound from below on the field of values of $\mathbf{T}_\xi$ is given by the following theorem.

Theorem 8.21 (Lower bound on the field of values of $\mathbf{T}_\xi$)**.** *Under Assumption 8.6 and given $\tilde{\omega}_0 > 0$, there exists a constant $\mathcal{C}_1 > 0$ such that*

$$\frac{|(\mathbf{v}_h, \mathbf{T}_\xi \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}}|}{\|\mathbf{v}_h\|^2_{\mathrm{curl},\tilde{\omega}}} \gtrsim \left(1 + \left(\frac{H_{\mathrm{cs}}}{\delta}\right)^2\right)^{-1} \left(\frac{|\xi|}{\tilde{\omega}^2}\right)^2 \quad \text{for all } \mathbf{v}_h \in \mathbf{Q}_h \tag{8.46}$$

and for all $\tilde{\omega} \geq \tilde{\omega}_0$, when

$$\max\left\{(\tilde{\omega} H_{\mathrm{sub}}),\ (\tilde{\omega} H_{\mathrm{cs}}) \left(\frac{\tilde{\omega}^2}{|\xi|}\right)\right\} \ \leq\ \mathcal{C}_1 \left(1 + \left(\frac{H_{\mathrm{cs}}}{\delta}\right)^2\right)^{-1} \left(\frac{|\xi|}{\tilde{\omega}^2}\right). \tag{8.47}$$

8.5 Matrices and convergence of GMRES

In this section we convert the results of Theorems 8.14 and 8.21 into results about matrices.

8.5.1 From projection operators to matrices

We now interpret the operators $\mathbf{T}^\ell_\xi$ defined in (8.35) in terms of matrices.

Lemma 8.22. *Let $\mathbf{v}_h = \sum_{j \in \mathcal{I}^h} V_j \phi_j \ \in\ \mathbf{Q}_h$. Then*

$$(i) \quad \mathbf{T}^\ell_\xi \mathbf{v}_h = \sum_{j \in \mathcal{I}^h(\Omega_\ell)} \left((R^\ell)^T (A^\ell_\xi)^{-1} R^\ell A_\xi \mathbf{V}\right)_j \phi_j\ , \quad \ell = 1, \ldots, N,$$

$$(ii) \quad \mathbf{T}^0_\xi \mathbf{v}_h = \sum_{p \in \mathcal{I}^H} \left(R_0^T (A^0_\xi)^{-1} R_0 A_\xi \mathbf{V}\right)_p \Phi_p,$$

where A^ℓ_ξ , $\ell = 0, \ldots, N$ is defined in (8.25)*.*

Proof. For (i), let $\ell \in \{1, \ldots, N\}$, let $\mathbf{w}_{h,\ell}$ and $\mathbf{y}_{h,\ell}$ be arbitrary elements of $\mathbf{Q}^\ell_h$, and denote their coefficient vectors $\mathbf{W}$ and $\mathbf{Y}$ (with degrees of freedom on all of $\mathcal{I}^h$). Then $\mathbf{W} = (R^\ell)^T \mathbf{w}$ and $\mathbf{Y} = (R^\ell)^T \mathbf{y}$, where $\mathbf{w}, \mathbf{y}$ have degrees of freedom on $\mathcal{I}^h(\Omega_\ell)$. The definitions of A_ξ and A^ℓ_ξ, (8.14) and (8.25), then imply that $a_\xi(\mathbf{y}_{h,\ell}, \mathbf{w}_{h,\ell}) = \mathbf{W}^* A_\xi \mathbf{Y} = \mathbf{w}^* A^\ell_\xi \mathbf{y}$. So if $\mathbf{y} := (A^\ell_\xi)^{-1} R^\ell A_\xi \mathbf{V}$ for some $\mathbf{V} \in \mathbb{C}^n$, we have

$$a_\xi(\mathbf{y}_{h,\ell}, \mathbf{w}_{h,\ell}) \ =\ \mathbf{w}^* R^\ell A_\xi \mathbf{V} \ =\ ((R^\ell)^T \mathbf{w})^* A_\xi \mathbf{V} = \ \mathbf{W}^* A_\xi \mathbf{V} \ =\ a_\xi(\mathbf{v}_h, \mathbf{w}_{h,\ell}),$$

where $\mathbf{y}_{h,\ell}$ is the finite element function with degrees of freedom $\mathbf{y}$. Thus, by definition of $\mathbf{T}^\ell_\xi$, we have $\mathbf{y}_{h,\ell} = \mathbf{T}^\ell_\xi \mathbf{v}_h$, which implies the result (i). The proof of (ii) is similar. □

The main results of the previous section - Theorems 8.14 and 8.21 - give estimates for the norm and the field of values of the operator $\mathbf{T}_\xi$ on the space $\mathbf{Q}_h$, with respect to the inner product $(\cdot,\cdot)_{\mathrm{curl},\tilde{\omega}}$ and its associated norm. The next lemma shows that, in order to translate these results into norm and field of values estimates for the preconditioned matrix $M^{-1}_{\xi,AS} A_\xi$, we need to work in the weighted *inner product* $\langle \cdot, \cdot \rangle_{D_{\tilde{\omega}}}$ defined such that if $\mathbf{v}_h, \mathbf{w}_h \in \mathbf{Q}_h$ with coefficient vectors $\mathbf{V}, \mathbf{W}$ then

$$(\mathbf{v}_h, \mathbf{w}_h)_{\mathrm{curl},\tilde{\omega}} \ =\ \langle \mathbf{V}, \mathbf{W} \rangle_{D_{\tilde{\omega}}}; \tag{8.48}$$

note that the definition of $(\cdot,\cdot)_{\mathrm{curl},\tilde{\omega}}$ (8.7) means that $\langle \mathbf{V}, \mathbf{W} \rangle_{D_{\tilde{\omega}}}$ depends on the wavenumber $\tilde{\omega}$.

Lemma 8.23. *Let* $\mathbf{v}_h = \sum_{j\in\mathcal{I}^h} V_j\phi_h \in \mathbf{Q}_h$. *Then*

$$\begin{aligned} (i)\quad (\mathbf{v}_h, \mathbf{T}_\xi \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}} &= \langle \mathbf{V}, M_{\xi,AS}^{-1} A_\xi \mathbf{V}\rangle_{D_{\tilde{\omega}}}, \quad \textit{and} \\ (ii)\quad \|\mathbf{T}_\xi \mathbf{v}_h\|_{\mathrm{curl},\tilde{\omega}} &= \|M_{\xi,AS}^{-1} A_\xi \mathbf{V}\|_{D_{\tilde{\omega}}}. \end{aligned}$$

Proof. For arbitrary $\mathbf{w}_h, \mathbf{v}_h \in \mathbf{Q}_h$, with coefficient vectors $\mathbf{W}$ and $\mathbf{V}$, we use Lemma 8.22 and (8.48) to find that

$$(\mathbf{w}_h, \mathbf{T}_\xi^\ell \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}} \;=\; \langle \mathbf{W}, (R^\ell)^T (A_\xi^\ell)^{-1} R^\ell A_\xi \mathbf{V}\rangle_{D_{\tilde{\omega}}}, \quad \ell = 0,\dots,N.$$

Summing these over $\ell = 0,\dots,N$ and using the definition of $M_{\xi,AS}^{-1}$ (8.26), we obtain

$$(\mathbf{w}_h, \mathbf{T}_\xi \mathbf{v}_h)_{\mathrm{curl},\tilde{\omega}} \;=\; \langle \mathbf{W}, M_{\xi,AS}^{-1} A_\xi \mathbf{V}\rangle_{D_{\tilde{\omega}}},$$

from which (i) and (ii) follow immediately. □

8.5.2 Recap of Elman-type estimates for convergence of GMRES

We consider the abstract linear system

$$C\mathbf{x} = \mathbf{d}$$

in $\mathbb{C}^n$, where C is an $n \times n$ nonsingular complex matrix. Given an initial guess $\mathbf{x}^0$, we introduce the residual $\mathbf{r}^0 = \mathbf{d} - C\mathbf{x}^0$ and the usual Krylov spaces:

$$\mathcal{K}^m(C, \mathbf{r}^0) := \mathrm{span}\{C^j \mathbf{r}^0 : j = 0,\dots,m-1\}.$$

Let $\langle\cdot,\cdot\rangle_D$ denote the inner product on $\mathbb{C}^n$ induced by some Hermitian positive definite matrix D, i.e.

$$\langle \mathbf{V}, \mathbf{W}\rangle_D := \mathbf{W}^* D \mathbf{V}$$

with induced norm $\|\cdot\|_D$, where * denotes Hermitian transpose. For $m \geq 1$, define $\mathbf{x}^m$ to be the unique element of $\mathcal{K}^m$ satisfying the minimal residual property:

$$\|\mathbf{r}^m\|_D := \|\mathbf{d} - C\mathbf{x}^m\|_D = \min_{\mathbf{x}\in\mathcal{K}^m(C,\mathbf{r}^0)} \|\mathbf{d} - C\mathbf{x}\|_D.$$

When $D = I$ this is just the usual GMRES algorithm, and we write $\|\cdot\| = \|\cdot\|_I$, but for more general D it is the weighted GMRES method [51] in which case its implementation requires the application of the weighted Arnoldi process [66].

The following theorem is a simple generalization of the GMRES convergence result of Beckermann Goreinov and Tyrtyshnikov [6] to the weighted setting. This result is an improvement of the so-called "Elman estimate", originally due to Elman [49]; see also [47], [108, Theorem 3.2], [46, Corollary 6.2], and the review [105, §6].

Theorem 8.24 (Elman-type estimate for weighted GMRES)**.** *Let* C *be a matrix with* $0 \notin W_D(C)$, *where*

$$W_D(C) := \big\{\langle C\mathbf{v}, \mathbf{v}\rangle_D : \mathbf{v} \in \mathbb{C}^N, \|\mathbf{v}\|_D = 1\big\}$$

is the field of values or numerical range of C *with respect to the inner product* $\langle\cdot,\cdot\rangle_D$. *Let* $\theta \in [0, \pi/2)$ *be defined such that*

$$\cos\theta = \frac{\mathrm{dist}\big(0, W_D(C)\big)}{\|C\|_D},$$

let γ_θ be defined by

$$\gamma_\theta := 2\sin\left(\frac{\theta}{4-2\theta/\pi}\right), \tag{8.49}$$

and let $\mathbf{r}_m$ be defined as above. Then

$$\frac{\|\mathbf{r}_m\|_D}{\|\mathbf{r}_0\|_D} \le \left(2+\frac{2}{\sqrt{3}}\right)(2+\gamma_\theta)\,\gamma_\theta^m. \tag{8.50}$$

When we apply the estimate (8.50) to $M_{\xi,AS}^{-1}$ in §8.5.3 below, we find that $\theta = \pi/2-\epsilon$, where (for fixed δ, H) $\epsilon = \epsilon(\tilde{\omega},\xi)$ is such that $\epsilon \to 0$ as $\tilde{\omega}\to\infty$. It is therefore convenient to specialize the result (8.50) to this particular situation in the following corollary.

Corollary 8.25. *If $\theta = \pi/2-\epsilon$ then, given $0<\epsilon_0<\pi/2$, there exists $\mathcal{C}>0$ (independent of ϵ) such that, for $0<a<1$,*

$$\text{if}\quad m \ge \frac{\mathcal{C}}{\epsilon}\log\left(\frac{12}{a}\right) \qquad \text{then} \qquad \frac{\|\mathbf{r}_m\|_D}{\|\mathbf{r}_0\|_D} \le a. \tag{8.51}$$

for all $0<\epsilon<\epsilon_0$.

8.5.3 The main results

With the classical two-level additive Schwarz preconditioner $M_{\xi,\mathrm{AS}}^{-1}$ in (8.26) for A_ξ and the inner product $\langle\cdot,\cdot\rangle_{D_{\tilde{\omega}}}$ defined by (8.48) above, we have the following results.

Combining the bounds for the operator $\mathbf{T}_\xi$ in Theorems 8.14 and 8.21, with the matrix interpretation in Lemma 8.23, we obtain the following upper bound on the norm of the preconditioned matrix $M_{\xi,\mathrm{AS}}^{-1}A_\xi$ and lower bound on its field of values.

Theorem 8.26 (Bounds for left preconditioning)**.**
(i)

$$\left\|M_{\xi,AS}^{-1}A_\xi\right\|_{D_{\tilde{\omega}}} \lesssim \left(\frac{\tilde{\omega}^2}{|\xi|}\right) \quad \textit{for all } H_{\mathrm{cs}}, H_{\mathrm{sub}}.$$

(ii) If Ω is a convex polyhedron, then, given $\tilde{\omega}_0>0$, there exists a constant $\mathcal{C}_1$ such that if

$$\max\left\{(\tilde{\omega}H_{\mathrm{sub}}),\ (\tilde{\omega}H_{\mathrm{cs}})\left(\frac{\tilde{\omega}^2}{|\xi|}\right)\right\} \le \mathcal{C}_1\left(1+\left(\frac{H_{\mathrm{cs}}}{\delta}\right)^2\right)^{-1}\left(\frac{|\xi|}{\tilde{\omega}^2}\right), \tag{8.52}$$

then

$$\frac{\left|\left\langle \mathbf{V}, M_{\xi,AS}^{-1}A_\xi\mathbf{V}\right\rangle_{D_{\tilde{\omega}}}\right|}{\|\mathbf{V}\|_{D_{\tilde{\omega}}}^2} \gtrsim \left(1+\left(\frac{H_{\mathrm{cs}}}{\delta}\right)^2\right)^{-1}\left(\frac{|\xi|}{\tilde{\omega}^2}\right)^2,$$

for all $\mathbf{V}\in\mathbb{C}^n$ and for all $\tilde{\omega}\ge\tilde{\omega}_0$.

Observe that, just as in the Helmholtz theory in [62], the condition on the coarse mesh diameter H_{cs} in (8.52) is more stringent than the condition on the subdomain diameter H_{sub}; one finds similar criteria in domain decomposition theory for coercive elliptic PDEs (see, e.g., [61]).

Combining Theorem 8.26 with the result about GMRES convergence in Corollary 8.25, we obtain the final convergence estimate.

Theorem 8.27 (GMRES convergence for left preconditioning)**.** *Let Ω be a convex polyhedron. Consider the weighted GMRES method applied to $M_{\xi,AS}^{-1}A_\xi$, where the residual is minimized in the norm induced by $D_{\tilde\omega}$ (as described in §8.5.2).*

Given $\tilde\omega_0 > 0$, there exists $\mathcal{C} > 0$, independent of all parameters such that, if (i) $\tilde\omega \geq \tilde\omega_0$, (ii) condition (8.52) *holds, and (iii)*

$$m \geq \mathcal{C} \left(\frac{\tilde\omega^2}{|\xi|}\right)^3 \left(1 + \left(\frac{H_{\rm cs}}{\delta}\right)^2\right) \log\left(\frac{12}{a}\right), \tag{8.53}$$

then

$$\frac{\|\mathbf{r}_m\|_{D_{\tilde\omega}}}{\|\mathbf{r}_0\|_{D_{\tilde\omega}}} \leq a$$

for any $0 < a < 1$.

Two *particular cases* of Theorem 8.27 are

1. When $|\xi| \sim \tilde\omega^2$ (maximum absorption) and $\delta \sim H_{\rm cs}$ (generous overlap), Condition (8.52) is satisfied with $H_{\rm sub} \sim H_{\rm cs} \sim \tilde\omega^{-1}$, and then the bound (8.53) implies GMRES will converge with the number of iterations *independent of the wavenumber.*

2. When $|\xi| \sim \tilde\omega$ and $\delta \sim H_{\rm cs}$, Condition (8.52) is satisfied with $H_{\rm sub} \sim \tilde\omega^{-2}$, $H_{\rm cs} \sim \tilde\omega^{-3}$, and then the bound (8.53) implies GMRES will converge with the number of iterations growing at most like $\tilde\omega^3$ as $\tilde\omega \to \infty$.

Using coercivity of the adjoint form in Lemma 8.4, we can obtain the following result about right preconditioning, however in the inner product induced by $D_{\tilde\omega}^{-1}$. From this, the analogue of Theorem 8.27, with $D_{\tilde\omega}$ replaced by $D_{\tilde\omega}^{-1}$, follows.

Theorem 8.28 (Bounds for right preconditioning)**.**
(i)

$$\|A_\xi M_{\xi,AS}^{-1}\|_{D_{\tilde\omega}^{-1}} \lesssim \left(\frac{\tilde\omega^2}{|\xi|}\right) \quad \textit{for all } H_{\rm cs}, H_{\rm sub}.$$

(ii) With the same assumptions as Part (ii) of Theorem 8.26, given $\tilde\omega_0 > 0$ and provided Condition (8.52) *holds,*

$$\frac{\left|\left\langle \mathbf{V}, A_\xi M_{\xi,AS}^{-1}\mathbf{V}\right\rangle_{D_{\tilde\omega}^{-1}}\right|}{\|\mathbf{V}\|_{D_{\tilde\omega}^{-1}}^2} \gtrsim \left(1 + \left(\frac{H_{\rm cs}}{\delta}\right)^2\right)^{-1} \left(\frac{|\xi|}{\tilde\omega^2}\right)^2,$$

for all $\mathbf{V} \in \mathbb{C}^n$ and for all $\tilde\omega \geq \tilde\omega_0$.

Theorem 8.29 (GMRES convergence for right preconditioning)**.** *The result of Theorem 8.27 holds when left preconditioning is replaced by right preconditioning ($M_{\xi,AS}^{-1}A_\xi$ is replaced by $A_\xi M_{\xi,AS}^{-1}$ in the statement of the theorem).*

8.6 Numerical experiments

We solve equation (8.3) (with $\xi \neq 0$ or $\xi = 0$) in the unit cube $\Omega = (0,1)^3$, with PEC boundary conditions (8.4) or with impedance boundary conditions (8.5) on its boundary $\partial\Omega$. The right-hand side function $\mathbf{F}$ in (8.3) is given by the point source

$$\mathbf{F} = [f, f, f], \quad \text{where} \quad f = -\exp(-400((x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2)).$$

The discretization is by curl-conforming finite elements (see Chapters 3 and 4) on a regular mesh of tetrahedra. We will give results for both degree 1 and degree 2 elements. The fine mesh diameter is either chosen as $h \sim \tilde{\omega}^{-3/2}$ (the discretization level generally believed to remove the pollution effect, by analogy with Helmholtz problems), or with a fixed number g of grid-points per wavelength, i.e. $h \sim 2\pi/(g\tilde{\omega})$, when degree 2 elements are used.

The resulting linear system (see (8.14)) is solved using GMRES without restarts with right preconditioning. For each solve we use a *random initial guess*, aiming to ensure that all frequencies are present in the error. The stopping criterion is based on a reduction of the relative residual by 10^{-6} and the maximum number of iterations allowed is 200. All the computations are done in FreeFem++. The code is parallelized and run on the supercomputers Curie (at TGCC-CEA, `http://www-hpc.cea.fr/en/complexe/tgcc-curie.htm`) and OCCIGEN (at CINES, `https://www.cines.fr/en/occigen-the-new-supercomputeur/`). Note that we do not use the library HPDDM, because the coarse space based on a coarse grid is not available yet there, thus the computing times shown in the following tables could be further optimized.

We consider a regular decomposition into overlapping subdomains (cubes). Even if the theoretical result is optimal for "generous overlap" ($\delta \sim H_{\text{sub}}$), this would require substantial communication between subdomains, therefore, apart from the first experiments, we take an overlap of size $\mathcal{O}(2h)$ in all directions (which we call "minimal overlap"). The examined two-level domain decomposition preconditioners are defined at the end of Section 8.3; their one-level version is considered if "1-level" is specified in the tables. To apply the preconditioner, the local problems in each subdomain and the coarse space problem (where used) are each solved with a direct solver (in this case MUMPS [2]) on a single processor. When describing the properties of the preconditioners, it is useful to introduce the parameters:

- N_{sub}, the number of subdomains (previously denoted just by N),
- n, the total number of degrees of freedom in the (fine grid) problem,
- n_{sub}, the number of degrees of freedom per subdomain without considering the overlap contribution (which could be substantial, especially for generous overlap),
- n_{cs}, the number of degrees of freedom in the coarse grid problem.

In the following we will consider experiments on solving (8.3) where the given problem may (or may not) have absorption and where the preconditioner may be built from an absorptive problem. Without loss of generality we assume that the absorption parameter ξ in (8.3) is non-negative and so we will have two parameters

$$0 \leq \xi_{\text{prob}} \leq \xi_{\text{prec}}$$

which define absorption in the problem being solved and in the preconditioner. In the following tables we use # to denote the number of iterations for any given methods (e.g. #HRAS for the HRAS method). "Time" denotes the total time (in seconds) for the solution of the problem.

Throughout this section we will be interested in the effect of various scaling of the subdomain diameter and the coarse grid diameter with respect to the wavenumber $\tilde{\omega}$, thus we shall refer to parameters (α, α') such that

$$H_{\text{sub}} \sim \tilde{\omega}^{-\alpha} \quad \text{and} \quad H_{\text{cs}} \sim \tilde{\omega}^{-\alpha'}.$$

For a given $\tilde{\omega}$, the smaller is α the more (and the smaller) are the subdomains; the smaller is α', the coarser is the coarse grid.

	$\alpha = 1 = \alpha'$			$\alpha = 0.9 = \alpha'$		
$\tilde{\omega}$	#AS	#RAS	#HRAS	#AS	#RAS	#HRAS
10	53(57)	26(37)	12	41(42)	23(25)	13
15	59(64)	28(42)	12	46(48)	26(32)	13
20	76(105)	29(57)	17	74(84)	34(43)	19

	$\alpha = 0.8 = \alpha'$		
$\tilde{\omega}$	#AS	#RAS	#HRAS
10	37(38)	21(21)	13
15	37(38)	22(22)	14
20	62(63)	31(31)	20

Table 8.1 – Iteration numbers for the two-level preconditioners (one-level preconditioners) with $\delta \sim H_{\text{sub}}$, $\xi_{\text{prob}} = \xi_{\text{prec}} = \tilde{\omega}^2$ and PEC boundary condition on $\partial\Omega$.

	$\alpha = 1 = \alpha'$				$\alpha = 0.9 = \alpha'$			
$\tilde{\omega}$	n	N_{sub}	n_{sub}	n_{cs}	n	N_{sub}	n_{sub}	n_{cs}
10	4.6×10^5	1000	4.6×10^2	7.9×10^3	3.1×10^5	343	9.1×10^2	2.9×10^3
15	1.5×10^6	3375	4.6×10^2	2.6×10^4	2.1×10^6	1331	1.5×10^3	1.0×10^4
20	1.2×10^7	8000	1.5×10^3	6.0×10^4	9.9×10^6	2744	3.6×10^3	2.1×10^4

	$\alpha = 0.8 = \alpha'$			
$\tilde{\omega}$	n	N_{sub}	n_{sub}	n_{cs}
10	3.4×10^5	216	1.6×10^3	1.9×10^3
15	1.9×10^6	512	3.7×10^3	4.2×10^3
20	7.1×10^6	1000	7.1×10^3	7.9×10^3

Table 8.2 – Sizes of problems and number of subdomains for methods in Table 8.1.

8.6.1 Illustrations of the theory for conductive media

In this subsection we illustrate the theory in §8.5 by solving problem (8.3) with PEC boundary condition on $\partial\Omega$. We take $\xi_{\text{prob}} = \tilde{\omega}^2$ and $h \sim \tilde{\omega}^{-3/2}$.

In the first experiment (Table 8.1) we set $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}, H_{\text{cs}} \sim \tilde{\omega}^{-\alpha'}$ for various $\alpha = \alpha'$ and $\xi_{\text{prob}} = \tilde{\omega}^2 = \xi_{\text{prec}}$ and use generous overlap ($\delta \sim H_{\text{sub}}$). We compare the performance of AS, RAS and HRAS. The results in Table 8.1 give the iteration numbers for the two-level methods, with the one-level methods in brackets. The sizes of the fine grid problem and the subdomain and coarse grid problems, solved in the preconditioners in Table 8.1, are given in Table 8.2

In Table 8.1 we see that RAS is always superior to AS as expected, and the hybrid version (HRAS) is superior to both additive methods. When $\alpha = \alpha' = 1$ (which corresponds to the optimal case described underneath Theorem 8.27) the coarse grid solve is bringing down the iteration count noticeably. When $\alpha = \alpha' = 0.8$ it is having little effect. For a given $\tilde{\omega}$, the number of iterations of the additive versions seems to improve a little as α and α' are reduced, but we should note (Table 8.2) that the problems being solved sometimes get a little smaller as α, α' are reduced. Indeed, the size of the global matrix, which should depend just on the wavenumber $\tilde{\omega}$, varies also for different α: this is due to the fact that we modify the number of points in each direction to be an exact multiple of the number of subdomains in that direction, so as to have smooth cubes as subdomains. In Table 8.1 we see that RAS appears close to being stable with respect to $\tilde{\omega}$ but that AS

	$\alpha = 0.8$, $\alpha' = 1$					
$\tilde{\omega}$	n	N_{sub}	n_{cs}	#AS	#RAS	#HRAS
10	3.4×10^5	216	7.9×10^3	37	20	11
20	7.1×10^6	1000	6.0×10^4	57	24	11
30	4.1×10^7	3375	2.0×10^5	53	32	16
40	2.0×10^8	6859	4.6×10^5	53	33	16
	$\alpha = 0.6$, $\alpha' = 1$					
$\tilde{\omega}$	n	N_{sub}	n_{cs}	#AS	#RAS	#HRAS
10	2.6×10^5	27	7.9×10^3	27	15	10
20	6.3×10^6	216	6.0×10^4	50	19	11
30	3.3×10^7	343	2.0×10^5	40	20	12

Table 8.3 – $\delta \sim H_{\text{sub}}$, $\xi_{\text{prob}} = \xi_{\text{prec}} = \tilde{\omega}^2$, PEC boundary condition on $\partial\Omega$, $\alpha' = 1$.

$\tilde{\omega}$	#AS	#RAS
10	52 (58)	34 (39)
15	57 (65)	39 (47)
20	61 (71)	43 (51)

Table 8.4 – $\delta \sim 2h$, $\xi_{\text{prob}} = \xi_{\text{prec}} = \tilde{\omega}^2$, PEC boundary condition on $\partial\Omega$, $(\alpha, \alpha') = (0.8, 0.8)$.

has not yet reached stability. Indeed, the theory in Theorems 8.27 and 8.29 covers only the AS case and proves stability asymptotically as $\tilde{\omega} \to \infty$.

Because our parallel code solves each subdomain problem on an individual processor and $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}$, a too large number of processors is required when α is close to 1 and $\tilde{\omega}$ increases (for $\alpha = 1$, we have $N_{\text{sub}} = \tilde{\omega}^3$). Thus relatively small values of $\tilde{\omega}$ are presented in Table 8.1. In Table 8.3 we present more results for this example with $\alpha' = 1$ and $\alpha = 0.8, 0.6$, so that there are fewer subdomains in this case. However note that $\alpha = 0.6$ for $k = 40$ gives too large subdomain problems considering the generous overlap and the memory requirement for this test was too high. Again, RAS is superior to AS, and the hybrid version HRAS is superior to both additive methods. Comparing the results for $\alpha = 0.8$ of Table 8.3 with the ones of Table 8.1 (the common lines are for $\tilde{\omega} = 10, 20$), we see that $\alpha' = 1$, which corresponds to a finer coarse grid, gives better iteration counts than $\alpha' = 0.8$.

In the experiment in Table 8.4 we extend the one in Table 8.1, exploring the effect of reducing the overlap to minimal ($\delta \sim 2h$). We choose the best case in Table 8.1, namely $(\alpha, \alpha') = (0.8, 0.8)$. The number of iterations with minimal overlap is worse than the one with generous overlap, but the method still performs well: the number of iterations grows slightly with $\tilde{\omega}$. We see that this time, with respect to the same case in Table 8.1, the coarse grid solve has a larger effect on the iteration count (in brackets we report the iteration count of the one-level method).

Finally, keeping the problem with $\xi_{\text{prob}} = \tilde{\omega}^2$ and PEC boundary conditions on $\partial\Omega$, we test the use of impedance transmission conditions on $\partial\Omega_\ell \setminus \partial\Omega$ in the local solves. We take an overlap $\delta \sim 2h$ and $(\alpha, \alpha') = (0.8, 0.8)$. The results in Table 8.5 show that impedance transmission conditions improve the iteration counts. Moreover the hybrid versions (HRAS and ImpHRAS) are superior to the additive versions (RAS and ImpRAS), and to the one-level preconditioners.

				Classical local solves		
$\tilde{\omega}$	n	N_{sub}	n_{cs}	#RAS	#HRAS	#1-level
10	3.4×10^5	216	1.8×10^3	34	23	39
20	7.1×10^6	1000	7.9×10^3	43	31	51
30	4.1×10^7	3375	2.5×10^4	47	34	59
40	1.3×10^8	6859	5.1×10^4	49	36	62
				Impedance local solves		
$\tilde{\omega}$	n	N_{sub}	n_{cs}	#ImpRAS	#ImpHRAS	#1-level
10	3.4×10^5	216	1.8×10^3	27	20	32
20	7.1×10^6	1000	7.9×10^3	35	28	37
30	4.1×10^7	3375	2.5×10^4	39	32	42
40	1.3×10^8	6859	5.1×10^4	42	35	43

Table 8.5 – $\delta \sim 2h$, $\xi_{\text{prob}} = \xi_{\text{prec}} = \tilde{\omega}^2$, PEC boundary condition on $\partial\Omega$, $(\alpha, \alpha') = (0.8, 0.8)$.

				$\xi_{\text{prec}} = \xi_{\text{prob}} = \tilde{\omega}$		
$\tilde{\omega}$	n	N_{sub}	n_{cs}	#ImpRAS	#ImpHRAS	#1-level
10	3.4×10^5	216	1.8×10^3	40	29	58
20	7.1×10^6	1000	7.9×10^3	83	60	125
30	4.1×10^7	3375	2.5×10^4	123	90	>200
40	1.3×10^8	6859	5.1×10^4	>200	154	>200
				$\xi_{\text{prec}} = \tilde{\omega}^2$		
$\tilde{\omega}$	n	N_{sub}	n_{cs}	#ImpRAS	#ImpHRAS	#1-level
10	3.4×10^5	216	1.8×10^3	55	37	65
20	7.1×10^6	1000	7.9×10^3	104	70	130
30	4.1×10^7	3375	2.5×10^4	147	101	>200
40	1.3×10^8	6859	5.1×10^4	193	137	>200

Table 8.6 – $\delta \sim 2h$, $\xi_{\text{prob}} = \tilde{\omega}$, impedance boundary condition on $\partial\Omega$, $(\alpha, \alpha') = (0.8, 0.8)$.

8.6.2 Lower absorption with impedance boundary conditions

In this subsection we solve problem (8.3) with impedance boundary conditions on $\partial\Omega$ and also as transmission conditions on $\partial\Omega_\ell \setminus \partial\Omega$ in the local solves. We consider absorption $\xi_{\text{prob}} = \tilde{\omega}$. We take $h \sim \tilde{\omega}^{-3/2}$ and overlap $\delta \sim 2h$.

We set $H_{\text{sub}} \sim \tilde{\omega}^{-\alpha}, H_{\text{cs}} \sim \tilde{\omega}^{-\alpha'}$ with $(\alpha, \alpha') = (0.8, 0.8)$, and examine preconditioners built with $\xi_{\text{prec}} = \xi_{\text{prob}} = \tilde{\omega}$ or with higher absorption $\xi_{\text{prec}} = \tilde{\omega}^2$. We report the results in Table 8.6. We see that when $\tilde{\omega}$ increases the iteration count of the one-level method deteriorates much faster than the two-level methods. Moreover the hybrid version (ImpHRAS) is always superior to the additive version (ImpRAS). The iteration counts with the same level of absorption in the problem and in the preconditioner (above) are generally better than the ones with higher absorption in the preconditioner (below). The iteration numbers may be improved by separating the coarse grid diameter from the subdomain diameter, making the coarse grid finer, for instance taking $\alpha' = 1$.

			$\alpha = 0.6$, $\alpha' = 0.9$				
k	n	N_{sub}	2-level	n_{cs}	Time	1-level	Time
10	2.6×10^5	27	20	2.9×10^3	16.2(1.6)	37	13.7(2.6)
15	1.5×10^6	125	26	1.0×10^4	25.5(4.0)	70	26.1(8.9)
20	5.2×10^6	216	29	2.1×10^4	52.0(9.1)	94	60.6(25.6)
25	1.4×10^7	216	33	4.4×10^4	145.5(29.5)	105	191.2(88.1)
30	3.3×10^7	343	38	6.9×10^4	380.4(128.4)	132	673.5(443.1)
			$\alpha = 0.7$, $\alpha' = 0.8$				
k	n	N_{sub}	2-level	n_{cs}	Time	1-level	Time
10	3.1×10^5	125	28	1.9×10^3	8.2(1.2)	58	7.7(2.0)
15	1.5×10^6	216	39	4.2×10^3	19.0(3.7)	82	20.1(7.2)
20	6.3×10^6	512	58	7.9×10^3	42.4(9.9)	123	49.7(20.8)
25	1.4×10^7	729	60	1.7×10^4	80.6(17.8)	148	94.1(42.2)
30	3.5×10^7	1000	80	2.6×10^4	251.9(95.2)	179	328.0(190.8)

Table 8.7 – ImpHRAS and the corresponding one-level method for $\delta \sim 2h$, $\xi_{\text{prob}} = 0$, impedance boundary condition on $\partial\Omega$, degree 1 elements and $h \sim \tilde{\omega}^{-3/2}$, $\xi_{\text{prec}} = \tilde{\omega}$ (we report also the execution time for GMRES in parentheses).

8.6.3 Maxwell's equations in non-conductive media

In this subsection we solve the pure Maxwell problem without absorption, i.e. $\xi_{\text{prob}} = 0$, and with impedance boundary conditions on $\partial\Omega$. We take an overlap $\delta \sim 2h$.

Results for degree 1 curl-conforming finite elements with $h \sim \tilde{\omega}^{-3/2}$ and with $\xi_{\text{prec}} = \tilde{\omega}$ are given in Table 8.7, for two choices of (α, α'), with $\alpha + \alpha' = 3/2$. For this relation between α and α' the methods are close to being *load balanced* in the sense that the coarse grid and subdomain problem sizes are very similar (see §7.4.3 in the Helmholtz Chapter). Out of the methods tested, the 2-level method (ImpHRAS) with $(\alpha, \alpha') = (0.6, 0.9)$ gives the best iteration count, but is more expensive. The method $(\alpha, \alpha') = (0.7, 0.8)$ is faster in time but its iteration count grows more quickly, so its advantage will diminish as $\tilde{\omega}$ increases further.

In Table 8.8 we repeat the same experiment but switching off the absorption also in the preconditioner, i.e. putting $\xi_{\text{prec}} = 0$. We observe that the resulting iteration counts are almost identical to the ones with absorption $\xi_{\text{prec}} = \tilde{\omega}$.

Finally, in Table 8.9 we repeat the experiment with $\xi_{\text{prec}} = \tilde{\omega}$ for degree 2 curl-conforming finite elements, considering a fixed number $g = 20$ of grid-points per wavelength, i.e. $h \sim 2\pi/(g\tilde{\omega})$. Here we take $H_{\text{sub}} \sim (g\tilde{\omega}/(2\pi))^{-\alpha}$, $H_{\text{cs}} \sim (g\tilde{\omega}/(2\pi))^{-\alpha'}$ and the methods are close to being load balanced when $\alpha + \alpha' = 1$. We obtain quite good iteration counts for the two-level preconditioner.

Note that in the current implementation a sequential direct solver on one processor is used to factorize the coarse problem matrix, which is clearly a limiting factor for the scalability of the algorithm. The timings could be significantly improved by using a distributed direct solver, or by adding a further level of domain decomposition for the coarse problem solve.

			$\alpha = 0.6$, $\alpha' = 0.9$				
k	n	N_{sub}	2-level	n_{cs}	Time	1-level	Time
10	2.6×10^5	27	20	2.9×10^3	17.1(1.8)	37	13.7(2.6)
15	1.5×10^6	125	26	1.0×10^4	25.4(3.9)	71	26.5(9.1)
20	5.2×10^6	216	29	2.1×10^4	53.0(9.0)	95	60.8(25.9)
25	1.4×10^7	216	33	4.4×10^4	145.0(29.6)	107	189.3(90.5)
30	3.3×10^7	343	39	6.9×10^4	387.9(132.7)	134	669.4(444.7)
			$\alpha = 0.7$, $\alpha' = 0.8$				
k	n	N_{sub}	2-level	n_{cs}	Time	1-level	Time
10	3.1×10^5	125	28	1.9×10^3	8.2(1.2)	58	7.7(2.0)
15	1.5×10^6	216	39	4.2×10^3	19.3(3.7)	82	19.7(7.2)
20	6.3×10^6	512	58	7.9×10^3	42.6(10.0)	124	49.1(21.0)
25	1.4×10^7	729	60	1.7×10^4	75.1(17.8)	150	98.2(43.0)
30	3.5×10^7	1000	81	2.6×10^4	223.5(86.7)	181	320.7(199.0)

Table 8.8 – ImpHRAS and the corresponding one-level method for $\delta \sim 2h$, $\xi_{\text{prob}} = 0$, impedance boundary condition on $\partial\Omega$, degree 1 elements and $h \sim \tilde{\omega}^{-3/2}$, $\xi_{\text{prec}} = 0$ (we report also the execution time for GMRES in parentheses).

			$g = 20$, $\alpha = 0.6$, $\alpha' = 0.6$		
$\tilde{\omega}$	n	N_{sub}	2-level	n_{cs}	1-level
10	1.7×10^6	343	30	1.5×10^4	85
20	1.4×10^7	1728	33	7.0×10^4	158
30	4.4×10^7	3375	34	1.4×10^5	>200
			$g = 20$, $\alpha = 0.5$, $\alpha' = 0.5$		
$\tilde{\omega}$	n	N_{sub}	2-level	n_{cs}	1-level
10	1.7×10^6	125	28	5.5×10^3	66
20	9.6×10^6	343	32	1.5×10^4	99
30	3.7×10^7	729	40	3.0×10^4	135

Table 8.9 – ImpHRAS and the corresponding one-level method for $\delta \sim 2h$, $\xi_{\text{prob}} = 0$, $\xi_{\text{prec}} = \tilde{\omega}$, impedance boundary condition on $\partial\Omega$, degree 2 elements and $h \sim 2\pi/(g\tilde{\omega})$, $H_{\text{sub}} \sim (g\tilde{\omega}/(2\pi))^{-\alpha}$, $H_{\text{cs}} \sim (g\tilde{\omega}/(2\pi))^{-\alpha'}$.

Chapter 9

Perspectives

Quite extensive numerical experiments about the two-level preconditioners for Maxwell's equations have been carried out in Chapter 8, but still we need to test their performance when applied to more realistic configurations. Not only we should consider more general geometries, starting for instance with waveguides, but also problems with heterogeneous coefficients. A particularly interesting test case would be the MEDIMAX problem with a model of the human head inside the imaging chamber: the tissues of the brain are indeed highly heterogenous. The absorption in this problem is given naturally by the complex-valued electric permittivity of these tissues: the imaginary part is typically of the same order as the real part.

For the Helmholtz equation we have compared two different definitions of coarse space: the DtN coarse space and the grid coarse space. Also this comparison should be extended to heterogeneous test cases. Moreover, both for the Helmholtz problem in three dimensions and the Maxwell problem we could try to rise further the frequency to test robustness for higher frequencies. The problem is that the constraint $h \sim \tilde{\omega}^{-3/2}$ results in very large problems: in the considered experiments, $\tilde{\omega} = 40$ gives for Helmholtz around 20 million complex-valued unknowns, for Maxwell 130 millions. Thus we should perform new tests relaxing the constraint on the mesh size h with respect to $\tilde{\omega}$.

For what concerns the convergence analysis, note that a complete theory for the DtN coarse space is missing. For the grid coarse space for the Maxwell problem, we could try to extend the theory in Chapter 8 to the following cases: impedance boundary conditions on $\partial\Omega$ (see Remark 8.13); impedance transmission conditions in the local solves, following the very recent work [31] about the Helmholtz problem; problem with heterogeneous coefficients. Currently we are trying to cast the analysis in [62] and our Maxwell analysis in a common abstract framework, in order to state a more general fictitious space lemma (see the references at the end of §7.2) that is suited to wave propagation problems. In this way we could design a new robust coarse space by adapting what is done for positive definite problems in [41, Chapter 7].

Appendix A

FreeFem++ scripts

We report here the sequential codes written to obtain the numerical results of Section 5.4.2. These scripts contain the methodological developments illustrated in the first chapters of these thesis and are partly based on routines developed during the last years by several collaborators.

The main program `main.edp` first calls the file `dataWaveguide.edp`, which contains the physical and numerical parameters, builds the mesh of the domain, defines the variational formulation of the problem to be solved and of the local subproblems appearing in the domain decomposition preconditioner. Note that the data script contains many macros, whose general syntax is

```
macro <identifier>(<parameter list>) <replacement token list> //
```

where `<parameter list>` is optional; this will make it possible to replace every subsequent occurrence of `<identifier>()` with `<replacement token list>`, by using the passed arguments if `<parameter list>` is present in the macro definition. This use of macros permits to use the same scripts for different (two or three dimensional) problems, by changing only the data script.

Then, the main program needs the routines (of `decomp.idp` and `createPartition.idp`) to create a decomposition of the domain and to build the restriction and partition of unity matrices. Finally, it builds the local problems matrices that constitute the preconditioner for the (complex) GMRES method called to solve the problem; in particular the `GMRES.idp` routine requires the matrix-vector product with the problem matrix and with the preconditioner.

main.edp

```
// Call in terminal with
// FreeFem++ main.edp -ns -partitioner 0 -my_schwarz_method oras -medit

include "getARGV.idp"
load "medit"
load "metis"
load "scotch"
load "thresholdings"
load "MUMPS_seq"

string prec = getARGV("-my_schwarz_method", "oras");
int overlap = getARGV("-overlap", 1); // number of overlap layers /2
int partitioner = getARGV("-partitioner", 1);
// 0: simple (regular), 1: metis, 2: scotch
```

```
int bmedit = usedARGV("-medit") > -1 ? 1 : 0; // to activate medit plots

include "dataWaveguide.edp"

fespace Vh(Th,Pk); // in data Pk = Edge03d...
int Ndof = Vh.ndof;
cout << "ndof = " << Ndof << endl;
fespace Ph(Th,P0); // for part in decomp.idp

Varf(vaglobal,Th,Ph)
Varfrhs(vaglobalrhs,Th,Ph)
matrix<complex> Aglobal;
Vh<complex> def(rhsglobal);
// data: macro def(u)[u,u#y,u#z]//
Aglobal = vaglobal(Vh,Vh); // global matrix
rhsglobal[] = vaglobalrhs(0,Vh); // global rhs

Ph part; // piecewise constant function giving the decomposition
int[int] lpart(Ph.ndof);
include "decomp.idp"
if (bmedit)
  medit("partition",Th,part);

include "createPartition.idp" // defines SubdomainsPartitionUnity function

// Domain decomposition data structures
meshN[int] aTh(npart);      // sequence of ovr. meshes
matrix[int] Rihreal(npart); // restriction matrices
matrix[int] Dihreal(npart); // partition of unity matrices
matrix<complex>[int] Rih(npart); // restriction matrices
matrix<complex>[int] Dih(npart); // partition of unity matrices
int[int] Ndeg(npart);       // number of dofs for each mesh
real[int] VolumeThi(npart); // area of each subdomain
matrix<complex>[int] aR(npart);  // local matrices

SubdomainsPartitionUnity(Th,part[],overlap,aTh,Rihreal,Dihreal,Ndeg,VolumeThi);
for (int i=0; i<npart; i++) {
  Rih[i] = Rihreal[i];
  Dih[i] = Dihreal[i];
}

for(int i = 0;i<npart;++i)
{
  cout << " Domain :" << i << "/" << npart << endl;
  meshN Thi = aTh[i];
  fespace Vhi(Thi,Pk);

  if (prec == "oras" || prec == "oas") {
    VarfOpt(RobinInt,Thi,PhOpt)
    aR[i] = RobinInt(Vhi,Vhi,solver=GMRES);
    set(aR[i],solver = sparsesolver);
  }
  else if (prec == "ras") {
    matrix<complex> aT = Aglobal*Rih[i]';
    aR[i] =  Rih[i]*aT;
    set(aR[i],solver = sparsesolver);
  }
  else if (prec != "none")
    assert(0);
}
```

```
include "GMRES.idp"
//(data: macro minit(u)[u,u,u]// EOM)
Vh<complex> def(un), def(sol) = minit(0); // initial guess, final solution

un[] = 0; // initial guess
// random initial guess:
for(int j=0; j<Vh.ndof; j++)
{
  un[][j] = randreal3()+ 1i*randreal3();
}
un[] /= un[].l2;

// GMRES solver
sol[] =  GMRES(un[], tol, maxit);
if (bmedit)
  medit("Final_solution",Th,[real(sol),real(soly),real(solz)]);
```

dataWaveguide.edp

```
/* Domain decomposition parameters */
int nn = 1, mm = 1; // number of sudomains in each direction
int ll = getARGV("-ll", 2);
int npart = nn*mm*ll;

/* Iterative method parameters */
real tol = 1e-6; // tolerance for the iterative method
int maxit = 200; // maximum number of iterations

/* The equation parameters */
real sigma = getARGV("-sigma",0.15);
real mu = 1.26e-6;
real epsilon = 8.85e-12;
real vel = 1/sqrt(epsilon*mu);
real beta = 32e9/vel;
real a = 0.00254*4, b = 0.00127*4, c = 0.0502*2; // waveguide dimensions
int m = 1, n = 0;
real wtilde = sqrt(beta^2+(m*pi/a)^2+(n*pi/b)^2); // wavenumber
real w = wtilde*vel; // angular frequency
complex mukappa = mu*1i*w*sigma - wtilde^2;

/* Mesh for beta */
real hfreq = sqrt(1/beta^3); // relation h^2*beta^3=1
include "cube.idp"
int mx, my, mz;
mx = a/hfreq+1;
my = b/hfreq+1;
mz = c/hfreq+1;
int[int] NN = [mx, my, mz];  // the number of seg in the 3 directions
int guide = 1, in = 2, out = 3;             // labels for the waveguide
real [int,int] BB = [[0,a],[0,b],[0,c]];    // bounding box
int [int,int] L = [[guide,guide],[guide,guide],[in,out]];
// labels of the 6 parallelipiped faces
mesh3 Th = Cube(NN,BB,L);                   // build the mesh
//medit("mesh", Th);                        // plot the mesh

//load "mymsh3" // not needed in new FF versions
load "Element_Mixte3d" // for Edge13d, Edge23d
// and for FF spaces Edge03ds0, Edge13ds0, Edge23ds0 used to build the
// partition of unity for resp Edge03d, Edge13d, Edge23d:
/* Edge03d: edge finite elements of degree 1
   Edge13d: edge finite elements of degree 2
   Edge23d: edge finite elements of degree 3 */
```

```
macro mtrunc trunc// EOM  //macro mtrunc truncvord// EOM
macro def(u)[u,u#y,u#z]// EOM
macro minit(u)[u,u,u]// EOM
func Pk = Edge13d;
func PkP0 = Edge13ds0;
macro defpart(u)u// EOM
macro initpart(u)u// EOM

macro meshN()mesh3// EOM
macro intN()int3d// EOM
macro intbN()int2d// EOM
macro measureN()volume// EOM
macro K complex // EOM

// used in decomp.idp for regular decomposition case:
macro simple(PhGlobal, part, comm)
{
  if (nn*mm*ll != npart)
    cout << "PB SIMPLE PARTITIONING : nn*mm*ll != npart" << endl;
  assert (nn*mm*ll == npart);
  PhGlobal xx=x,yy=y, zz=z;
  part=int(xx/a*nn)*mm*ll + int(zz/c*ll)*mm + int(yy/b*mm);
}
// EOM

searchMethod = 1;

/* Exact solution if sigma=0 and functions for the impedance conditions */
complex Cc = 1i*w*mu/(wtilde^2-beta^2);
func expbz = exp(-1i*beta*z);
func ExTE = Cc*(n*pi)/b*cos(m*pi*x/a)*sin(n*pi*y/b)*expbz;
func EyTE = -Cc*(m*pi)/a*sin(m*pi*x/a)*cos(n*pi*y/b)*expbz;
func EzTE = 0;
// the parameter p in the impedance boundary conditions
// (curl E)xn + 1ip* nx(Exn) = G
real impParam = beta;
// For the impedance condition at the waveguide entrance:
func Gix = 1i*(beta+impParam)*ExTE;
func Giy = 1i*(beta+impParam)*EyTE;
func Giz = 0;
// For the impedance condition at the waveguide exit
// (it is 0 with impParam = beta):
func Gox = 1i*(-beta+impParam)*ExTE;
func Goy = 1i*(-beta+impParam)*EyTE;
func Goz = 0;
real transmImpParam = wtilde; // in the impedance transmission condition

// Macros: Curl and cross product by the normal
macro Curl(ux,uy,uz) [dy(uz)-dz(uy),dz(ux)-dx(uz),dx(uy)-dy(ux)] // EOM
macro CrossN(ux,uy,uz) [uy*N.z-uz*N.y,uz*N.x-ux*N.z,ux*N.y-uy*N.x] // EOM
macro Curlabs(ux,uy,uz) [abs(dy(uz)-dz(uy)),abs(dz(ux)-dx(uz)),
                                              abs(dx(uy)-dy(ux))] // EOM

// Variational formulation for the pb matrix
macro Varf(varfName, meshName, PhName)
  varf varfName([Ex,Ey,Ez],[vx,vy,vz]) =
    intN(meshName)(Curl(vx,vy,vz)'*Curl(Ex,Ey,Ez))
    + intN(meshName)(mukappa*[vx,vy,vz]'*[Ex,Ey,Ez])
    + intbN(meshName,in,out)(1i*impParam*CrossN(vx,vy,vz)'*CrossN(Ex,Ey,Ez))
    + on(guide, Ex=0, Ey=0, Ez=0);
```

```
// EOM

// Variational formulation for the local matrices of the preconditioner
macro VarfOpt(varfName, meshName, PhName)
  varf varfName([Ex,Ey,Ez],[vx,vy,vz]) =
    intN(meshName)(Curl(vx,vy,vz)'*Curl(Ex,Ey,Ez))
    + intN(meshName)(mukappa*[vx,vy,vz]'*[Ex,Ey,Ez])
    + intbN(meshName,in,out)(1i*impParam*CrossN(vx,vy,vz)'*CrossN(Ex,Ey,Ez))
    + intbN(meshName,10)(1i*transmImpParam*CrossN(vx,vy,vz)'*CrossN(Ex,Ey,Ez))
    + on(guide, Ex=0, Ey=0, Ez=0);
// EOM

// Variational formulation for the pb right-hand side
macro Varfrhs(varfName, meshName, PhName)
  varf varfName([Ex,Ey,Ez],[vx,vy,vz]) =
    intbN(meshName,in)([vx,vy,vz]'*[Gix,Giy,Giz])
    + intbN(meshName,out)([vx,vy,vz]'*[Gox,Goy,Goz])
    + on(guide, Ex=0, Ey=0, Ez=0);
// EOM
```

decomp.idp

```
if (npart == 1) {
  part[] = 0.;
}
else if(partitioner != 0) {
  if(partitioner == 2) {
    scotch(lpart,Th,npart);
    for(int i=0;i<lpart.n;++i)
      part[][i]=lpart[i];
   }
  else {
    metisdual(lpart,Th,npart);
    for(int i=0;i<lpart.n;++i)
      part[][i]=lpart[i];
  }
}
else {
  simple(Ph, part, comm)
}
```

createPartition.idp

```
func bool SubdomainsPartitionUnity(meshN & Th, real[int] & partdof,
  int sizeoverlaps, meshN[int] & aTh, matrix[int] & Rih, matrix[int] & Dih,
  int[int] & Ndeg, real[int] & VolumeThi)
{
  int   npart=partdof.max+1;
  meshN Thi=Th;        // freefem's trick, formal definition
  fespace Vhi(Thi,Pk); // freefem's trick, formal definition
  fespace Whpart(Thi,PkP0);

  if (npart == 1) {
    aTh[0]=Th;
    real[int] one(Vhi.ndof);
    one=1.;
    Rih[0]=one;
    Ndeg[0] = Vh.ndof;
    VolumeThi[0] = int2d(Th)(1.);
    Dih[0]=one;
  }
  else {
```

```
    for(int ii=0;ii<npart;++ii)
22  {
      int[int] arrayIntersection;
      int[int][int] restrictionIntersection(0);
      real[int] D;
      int numberIntersection = 0;
27
      meshN overlapName=Th;
      fespace VhGlobal(overlapName, P1);
      fespace PhGlobal(overlapName, P0);
      PhGlobal part;
32    part[]=partdof;

      PhGlobal supp = abs(part - ii) < 0.1;
      VhGlobal suppSmooth;
      AddLayers(overlapName, supp[], sizeoverlaps * 2, suppSmooth[]);
37    {
        meshN neighbors = mtrunc(overlapName, suppSmooth > 0.001
                                              && (suppSmooth < 0.999));
        fespace Oh(neighbors, P0);
        Oh partOverlap = part;
42      Unique(partOverlap[], arrayIntersection);
      }
      fespace Vhl(Thi, P1);
      Vhl[int] partitionIntersection(arrayIntersection.n),
           partitionIntersectionbase(arrayIntersection.n);
47
      overlapName = mtrunc(overlapName, suppSmooth > 0.001);
      supp = supp;
      suppSmooth = suppSmooth;
      Thi = mtrunc(overlapName, suppSmooth> 0.501, label = 10);
52
      Vhl khi = max(suppSmooth*2 - 1.0, 0.) ;
      if(usedARGV("-steep") != -1)
        khi = khi > 0.001 ? 1.0 : 0.0;

57    else if (usedARGV("-raspart") != -1) {
                VhGlobal phir;
                PhGlobal suppP0 = abs(ii - part) < 0.1;
        varf vSuppi(u,v) = intN(overlapName,qforder=1)(suppP0*v);
        phir[] = vSuppi(0,VhGlobal);
62      phir = phir > 0.;
        khi = phir;
      }

      Vhl sum = khi;
67    VhGlobal phi = 0, phibase = 0;;
      real eps=int2d(overlapName)(1.);
      for(int i = 0; i < arrayIntersection.n; ++i) {
        PhGlobal suppPartition = abs(arrayIntersection[i] - part) < 0.1;

72      PhGlobal suppP0;
        suppP0[] = suppPartition[];

        AddLayers(overlapName, suppPartition[], sizeoverlaps, phi[]);
        phibase[] = phi[];
77
        if(usedARGV("-steep") != -1)
          phi = phi > 0.001 ? 1.0 : 0.0;
```

```
      else if (usedARGV("-raspart") != -1) {
        VhGlobal phir;
        varf vSuppi(u,v) = intN(overlapName,qforder=1)(suppP0*v);
        phir[] = vSuppi(0,VhGlobal);
        phir = phir > 0.;
        phi = phir;
      }

      real intersection=intN(overlapName)(phibase)/eps;
      if( intersection>1e-6)
      {
        partitionIntersection[numberIntersection] = phi;
        partitionIntersectionbase[numberIntersection] = phibase;
        sum[] += partitionIntersection[numberIntersection][];
        arrayIntersection[numberIntersection++] = arrayIntersection[i];
      }
    }

    khi[] = khi[] ./= sum[];
    Whpart defpart(func2vec) = initpart(khi); // partition of unity

    aTh[ii]=Thi;
    Dih[ii]=func2vec[];
    Dih[ii].thresholding(1e-10);
    Rih[ii]=interpolate(Vhi,Vh);

    {
      int[int] I(1),J(1);
      real[int] Kc(1);
      [I,J,Kc] = Rih[ii];
      for (int i=0;i<Kc.n;i++)
        if (Kc[i] > 0.99)
          Kc[i] = 1.;
      Rih[ii] = [I,J,Kc];
    }

    Rih[ii].thresholding(1e-10);
    Ndeg[ii] = Vhi.ndof;
    VolumeThi[ii] = intN(Thi)(1.);
  }
 }
 return true;
}
```

GMRES.idp

```
func K[int] A(K[int] &x)
{
  // Matrix vector product with the problem matrix
  K[int] Ax(x.n);
  Ax = 0;
  Ax = Aglobal*x;
  return Ax;
}

func complex[int] PREC(complex[int] &l)
{
  // Application of the preconditioner
  K[int] s(l.n);
  s = 0;
  if (prec == "none") {
    s = l;
```

```
17    }
      else {
        for(int i=0; i<npart; ++i)  {
          complex[int] bi = Rih[i]*l; // restriction
          complex[int] ui = aR[i] ^-1 * bi; // local solves
22        if(prec == "oas")
            bi = ui;
          else
            bi = Dih[i]*ui; // partition of unity
          s += Rih[i]'*bi; // prolongation
27      }
      }
      return s;
    }

32  func complex[int] GMRES(complex[int] x0, real eps, int nbiter)
    {
      ofstream filei(foldername+"Convprec.m");
      Vh<complex> def(r), def(z), def(v), def(w), def(ver), def(un);
      Vh<complex>[int] def(V)(nbiter+1);          // orthonormal basis
37    complex[int,int] Hn(nbiter+2,nbiter+1);     // Hessenberg matrix
      Hn = 0.;
      complex[int,int] rot(2,nbiter+2);
      rot = 0.;
      complex[int] g(nbiter+1),g1(nbiter+1);
42    g = 0.;  g1 = 0.;
      r[] = PREC(rhsglobal[]);
      real normb = r[].l2;
      if (normb < 1.e-20) normb = 1.;
      r[] = A(x0);
47    r[] -= rhsglobal[];
      r[] *= -1.0;
      z[] = PREC(r[]);             // z= M^{-1}(b-A*x0)
      g[0] = z[].l2;      // initial residual norm
      filei << "relres("+1+")=" << g[0] << ";" << endl;
52    V[0][]=1/g[0]*z[];           // first basis vector
      for(int it=0; it<nbiter; it++){
        v[] = A(V[it][]);
        w[] = PREC(v[]);   // w = M^{-1}A*V_it

57      for(int i=0; i<it+1; i++) {
          Hn(i,it) = w[]'*V[i][];
          w[] -=  conj(Hn(i,it))*V[i][];
        }
        Hn(it+1,it) = w[].l2;
62      complex aux = Hn(it+1,it);
        V[it+1][]=1/aux*w[];

        for(int i=0; i<it; i++){     // QR decomposition of Hn
          complex aa = conj(rot(0,i))*Hn(i,it)+conj(rot(1,i))*Hn(i+1,it);
67        complex bb = -rot(1,i)*Hn(i,it)+rot(0,i)*Hn(i+1,it);
          Hn(i,it) = aa;
          Hn(i+1,it) = bb;
        }
        complex sq = sqrt( conj(Hn(it,it))*Hn(it,it) + Hn(it+1,it)*Hn(it+1,it) );
72      rot(0,it) = Hn(it,it)/sq;
        rot(1,it) = Hn(it+1,it)/sq;

        Hn(it,it) = conj(rot(0,it))*Hn(it,it)+conj(rot(1,it))*Hn(it+1,it);
        Hn(it+1,it) =  0.;
77      g[it+1] = -rot(1,it)*g[it];
```

```
      g[it] = conj(rot(0,it))*g[it];
      complex[int] y(it+1);       // Reconstruct the solution
      for(int i=it; i>=0; i--) {
        g1[i] = g[i];
        for(int j=i+1; j<it+1; j++){
          g1[i] = g1[i]-Hn(i,j)*y[j];
        }
        y[i]=g1[i]/Hn(i,i);
      }
      un[] = x0;
      for(int i=0;i<it+1;i++){
        un[]= un[]+ conj(y[i])*V[i][];
      }
      real relerr=0;
      if (bdirect){
        ver[] = un[] - uglob[];
        relerr = ver[].l2/uglob[].l2;
      }
      real relres = abs(g[it+1]);

      if (bdirect)
        cout << "It: "<< it+1 << " Residual = " << relres << " Rel res = " <<
         relres/normb <<  " Relative L2 Error =  "<<  relerr << endl;
      else cout << "It: "<< it+1 << " Residual = " << relres << " Rel res = "
         << relres/normb << endl;

      int j = it+2;
      filei << "relres("+j+")=" << relres << ";" << endl;
      if(relres/normb < eps) {
        cout << "GMRES has converged in " + (it+1) + " iterations " << endl;
        cout << "The relative residual is " +  relres/normb << endl;
        break;    }
    }
    return un[];
}
```

Bibliography

[1] M. Ainsworth and J. Coyle. "Hierarchic finite element bases on unstructured tetrahedral meshes". In: *Internat. J. Numer. Methods Engrg.* 58.14 (2003), pp. 2103–2130.

[2] P. Amestoy, I. Duff, J. L'Excellent, and J. Koster. "A fully asynchronous multifrontal solver using distributed dynamic scheduling". In: *SIAM Journal on Matrix Analysis and Applications* 23.1 (2001), pp. 15–41.

[3] D. N. Arnold, R. S. Falk, and R. Winther. "Differential complexes and stability of finite element methods. I. The de Rham complex". In: *Compatible spatial discretizations*. Vol. 142. IMA Vol. Math. Appl. Springer, New York, 2006, pp. 24–46.

[4] D. N. Arnold, R. S. Falk, and R. Winther. "Finite element exterior calculus, homological techniques, and applications". In: *Acta Numer.* 15 (2006), pp. 1–155.

[5] D. N. Arnold, R. S. Falk, and R. Winther. "Multigrid in $H(\mathrm{div})$ and $H(\mathrm{curl})$". In: *Numer. Math.* 85.2 (2000), pp. 197–217.

[6] B. Beckermann, S. A. Goreinov, and E. E. Tyrtyshnikov. "Some remarks on the Elman estimate for GMRES". In: *SIAM J. Matrix Anal. Appl.* 27.3 (2005), pp. 772–778.

[7] I. Ben Gharbia, M. Bonazzoli, X. Claeys, P. Marchand, and P.-H. Tournier. "Fast solution of boundary integral equations for elasticity around a crack network: a comparative study". Proceedings of CEMRACS 2016, submitted.

[8] P. Bochev, C. J. Garasi, J. Hu, A. Robinson, and R. Tuminaro. "An improved algebraic multigrid method for solving Maxwell's equations". In: *SIAM J. Sci. Comput.* 25.2 (2003), pp. 623–642.

[9] D. Boffi. "A note on the de Rham complex and a discrete compactness property". In: *Appl. Math. Lett.* 14.1 (2001), pp. 33–38.

[10] D. Boffi, P. Fernandes, L. Gastaldi, and I. Perugia. "Computational models of electromagnetic resonators: analysis of edge element approximation". In: *SIAM J. Numer. Anal.* 36.4 (1999), pp. 1264–1290.

[11] D. Boffi, F. Brezzi, and M. Fortin. *Mixed finite element methods and applications*. Vol. 44. Springer Series in Computational Mathematics. Springer, Heidelberg, 2013, pp. xiv+685.

[12] M. Bonazzoli, V. Dolean, F. Rapetti, and P.-H. Tournier. "Parallel preconditioners for high-order discretizations arising from full system modeling for brain microwave imaging". In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* (2017). e2229 jnm.2229, e2229–n/a.

[13] M. Bonazzoli, E. Gaburro, V. Dolean, and F. Rapetti. "High order edge finite element approximations for the time-harmonic Maxwell's equations". In: *2014 IEEE Conference on Antenna Measurements Applications (CAMA) Proceedings*. Nov. 2014.

[14] M. Bonazzoli, F. Rapetti, P.-H. Tournier, and C. Venturini. "High order edge elements for electromagnetic waves: remarks on numerical dispersion". Proceedings of ICOSAHOM 2016 International Conference on Spectral and High Order Methods, submitted. 2017.

[15] M. Bonazzoli, V. Dolean, I. G. Graham, E. A. Spence, and P.-H. Tournier. "A two-level domain-decomposition preconditioner for the time-harmonic Maxwell's equations". hal-01525438 preprint. 2017.

[16] M. Bonazzoli, V. Dolean, I. G. Graham, E. A. Spence, and P.-H. Tournier. "Domain Decomposition preconditioning for the high-frequency time-harmonic Maxwell equations with absorption". In preparation. 2017.

[17] M. Bonazzoli, V. Dolean, I. G. Graham, E. A. Spence, and P.-H. Tournier. "Two-level preconditioners for the Helmholtz equation". hal-01525424 preprint. 2017.

[18] M. Bonazzoli, V. Dolean, F. Hecht, and F. Rapetti. "Explicit implementation strategy of high order edge finite elements and Schwarz preconditioning for the time-harmonic Maxwell's equations". hal-01298938 preprint. 2017.

[19] M. Bonazzoli, V. Dolean, R. Pasquetti, and F. Rapetti. "Schwarz preconditioning for high order edge element discretizations of the time-harmonic Maxwell's equations". In: *Domain Decomposition Methods in Science and Engineering XXIII*. Vol. 116. Lecture Notes in Computational Science and Engineering. Springer, Cham, 2017, pp. 117–124.

[20] M. Bonazzoli and F. Rapetti. "High-order finite elements in numerical electromagnetism: degrees of freedom and generators in duality". In: *Numerical Algorithms* 74.1 (2017), pp. 111–136.

[21] M. Bonazzoli, F. Rapetti, and C. Venturini. "Dispersion analysis of triangle-based Whitney element methods for electromagnetic wave propagation". In: *Applied Mathematics and Computation* Proceedings of ESCO 2016, 5th European Seminar on Computing (2017, in press).

[22] A. Bossavit. "A rationale for 'edge-elements' in 3-D fields computations". In: *Magnetics, IEEE Transactions on* 24.1 (Jan. 1988), pp. 74–79.

[23] A. Bossavit. "Generating Whitney forms of polynomial degree one and higher". In: *IEEE Transactions on Magnetics* 38.2 (Mar. 2002), pp. 341–344.

[24] A. Bossavit. "Mixed finite elements and the complex of Whitney forms". In: *The mathematics of finite elements and applications, VI (Uxbridge, 1987)*. Academic Press, London, 1988, pp. 137–144.

[25] A. Bossavit. "Solving Maxwell equations in a closed cavity, and the question of 'spurious modes'". In: *Magnetics, IEEE Transactions on* 26.2 (Mar. 1990), pp. 702–705.

[26] A. Bossavit and F. Rapetti. "A prolongation/restriction operator for Whitney elements on simplicial meshes". In: *SIAM J. Numer. Anal.* 43.5 (2005), pp. 2077–2097.

[27] A. Bossavit. *Computational electromagnetism*. Electromagnetism. Variational formulations, complementarity, edge elements. Academic Press, Inc., San Diego, CA, 1998, pp. xx+352.

[28] A. Bossavit and F. Rapetti. "Whitney forms, from manifolds to fields". In: *Spectral and high order methods for partial differential equations—ICOSAHOM 2012*. Vol. 95. Lect. Notes Comput. Sci. Eng. Springer, Cham, 2014, pp. 179–189.

[29] A. Brandt and I. Livshits. "Wave-ray multigrid method for standing wave equations". In: *Electron. Trans. Numer. Anal.* 6.Dec. (1997). Special issue on multilevel methods (Copper Mountain, CO, 1997), pp. 162–181.

[30] X.-C. Cai and M. Sarkis. "A restricted additive Schwarz preconditioner for general sparse linear systems". In: *SIAM Journal on Scientific Computing* 21 (1999), pp. 239–247.

[31] E. T. Chung, I. G. Graham, E. A. Spence, and J. Zou. "Domain Decompositon with local Impedance conditions for the Helmholtz equation". In preparation. 2017.

[32] P. G. Ciarlet. *The finite element method for elliptic problems*. Studies in Mathematics and its Applications, Vol. 4. North-Holland Publishing Co., Amsterdam-New York-Oxford, 1978, pp. xix+530.

[33] P. Collino, G. Delbue, P. Joly, and A. Piacentini. "A new interface condition in the non-overlapping domain decomposition method for the Maxwell equations". In: *Comput. Methods Appl. Mech. Engrg.* 148.1-2 (1997), pp. 195–207.

[34] L. Conen, V. Dolean, R. Krause, and F. Nataf. "A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator". In: *J. Comput. Appl. Math.* 271 (2014), pp. 83–99.

[35] A. St-Cyr, M. J. Gander, and S. J. Thomas. "Optimized multiplicative, additive, and restricted additive Schwarz preconditioning". In: *SIAM Journal on Scientific Computing* 29.6 (2007), pp. 2402–2425.

[36] T. Davis. "Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method". In: *ACM Trans. Math. Software* 30.2 (2004), pp. 196–199.

[37] B. Deprés. "Méthodes de décomposition de domains pour les problèms de propagation d'ondes en régime harmonique". PhD thesis. Université Paris IX Dauphine, 1991.

[38] B. Després, P. Joly, and J. E. Roberts. "A domain decomposition method for the harmonic Maxwell equations". In: *Iterative methods in linear algebra (Brussels, 1991)*. Amsterdam: North-Holland, 1992, pp. 475–484.

[39] V. Dolean, M. J. Gander, and L. Gerardo-Giorda. "Optimized Schwarz methods for Maxwell's equations". In: *SIAM J. Sci. Comput.* 31.3 (2009), pp. 2193–2213.

[40] V. Dolean, M. J. Gander, S. Lanteri, J.-F. Lee, and Z. Peng. "Effective transmission conditions for domain decomposition methods applied to the time-harmonic curl-curl Maxwell's equations". In: *J. Comput. Phys.* 280 (2015), pp. 232–247.

[41] V. Dolean, P. Jolivet, and F. Nataf. *An Introduction to Domain Decomposition Methods: algorithms, theory and parallel implementation*. SIAM, 2015.

[42] V. Dolean, F. Nataf, R. Scheichl, and N. Spillane. "Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps". In: *Comput. Methods Appl. Math.* 12.4 (2012), pp. 391–414.

[43] J. Douglas Jr. and J. E. Roberts. "Mixed finite element methods for second order elliptic problems". In: *Mat. Apl. Comput.* 1.1 (1982), pp. 91–103.

[44] M. Dryja and O. B. Widlund. "Some domain decomposition algorithms for elliptic problems". In: *Iterative methods for large linear systems (Austin, TX, 1988)*. Academic Press, Boston, MA, 1990, pp. 273–291.

[45] Y. Efendiev, J. Galvis, R. Lazarov, and J. Willems. "Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms". In: *ESAIM Math. Model. Numer. Anal.* 46.5 (2012), pp. 1175–1199.

[46] M. Eiermann and O. G. Ernst. "Geometric aspects of the theory of Krylov subspace methods". In: *Acta Numer.* 10 (2001), pp. 251–312.

[47] S. C. Eisenstat, H. C. Elman, and M. H. Schultz. "Variational iterative methods for nonsymmetric systems of linear equations". In: *SIAM J. Numer. Anal.* 20.2 (1983), pp. 345–357.

[48] M. El Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri. "Optimized Schwarz methods for the time-harmonic Maxwell equations with dampimg". In: *SIAM J. Scient. Comp.* 34.4 (2012), pp. 2048–2071.

[49] H. C. Elman. "Iterative Methods for Sparse Nonsymmetric Systems of Linear Equations". PhD thesis. Yale University, 1982.

[50] O. G. Ernst and M. J. Gander. "Why it is difficult to solve Helmholtz problems with classical iterative methods". In: *Numerical analysis of multiscale problems*. Vol. 83. Lect. Notes Comput. Sci. Eng. Springer, Heidelberg, 2012, pp. 325–363.

[51] A. Essai. "Weighted FOM and GMRES for solving nonsymmetric linear systems". In: *Numer. Algorithms* 18.3-4 (1998), pp. 277–292.

[52] C. Farhat, P. Avery, R. Tezaur, and J. Li. "FETI-DPH: a dual-primal domain decomposition method for acoustic scattering". In: *J. Comput. Acoust.* 13.3 (2005), pp. 499–524.

[53] C. Farhat, A. Macedo, and M. Lesoinne. "A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems". In: *Numer. Math.* 85.2 (2000), pp. 283–308.

[54] J. Fish and Y. Qu. "Global-basis two-level method for indefinite systems. I. Convergence studies". In: *Internat. J. Numer. Methods Engrg.* 49.3 (2000), pp. 439–460.

[55] F. Fuentes, B. Keith, L. Demkowicz, and S. Nagaraj. "Orientation embedded high order shape functions for the exact sequence elements of all shapes". In: *Comput. Math. Appl.* 70.4 (2015), pp. 353–458.

[56] M. J. Gander. "Schwarz methods over the course of time". In: *Electron. Trans. Numer. Anal.* 31 (2008), pp. 228–255.

[57] M. J. Gander, I. G. Graham, and E. A. Spence. "Applying GMRES to the Helmholtz equation with shifted Laplacian preconditioning: what is the largest shift for which wavenumber-independent convergence is guaranteed?" In: *Numer. Math.* 131.3 (2015), pp. 567–614.

[58] M. J. Gander, F. Magoulès, and F. Nataf. "Optimized Schwarz methods without overlap for the Helmholtz equation". In: *SIAM J. Sci. Comput.* 24.1 (2002), pp. 38–60.

[59] J. Gopalakrishnan, L. E. García-Castillo, and L. F. Demkowicz. "Nédélec spaces in affine coordinates". In: *Comput. Math. Appl.* 49.7-8 (2005), pp. 1285–1294.

[60] J. Gopalakrishnan and J. E. Pasciak. "Overlapping Schwarz preconditioners for indefinite time harmonic Maxwell equations". In: *Math. Comp.* 72.241 (2003), pp. 1–15.

[61] I. G. Graham, P. O. Lechner, and R. Scheichl. "Domain decomposition for multiscale PDEs". In: *Numer. Math.* 106.4 (2007), pp. 589–626.

[62] I. G. Graham, E. A. Spence, and E. Vainikko. "Domain Decomposition preconditioning for high-frequency Helmholtz problems with absorption". In: *Math. Comp.* 86 (2017), pp. 2089–2127.

[63] I. G. Graham, E. A. Spence, and E. Vainikko. "Recent Results on Domain Decomposition Preconditioning for the High-Frequency Helmholtz Equation Using Absorption". In: *Modern Solvers for Helmholtz Problems*. Ed. by D. Lahaye, J. Tang, and K. Vuik. Geosystems Mathematics. Springer, 2017, pp. 3–26.

[64] M. Griebel and P. Oswald. "On the abstract theory of additive and multiplicative Schwarz algorithms". In: *Numer. Math.* 70.2 (1995), pp. 163–180.

[65] J.-L. Guermond and A. Ern. "Finite element quasi-interpolation and best approximation". In: *ESAIM: Mathematical Modelling and Numerical Analysis* (2016, Accepted for publication).

[66] S. Güttel and J. Pestana. "Some observations on weighted GMRES". In: *Numer. Algorithms* 67.4 (2014), pp. 733–752.

[67] F. Hecht. "New development in FreeFem++". In: *J. Numer. Math.* 20.3-4 (2012), pp. 251–265.

[68] R. Hiptmair. "Canonical construction of finite elements". In: *Math. Comp.* 68.228 (1999), pp. 1325–1346.

[69] R. Hiptmair. "Finite elements in computational electromagnetism". In: *Acta Numer.* 11 (2002), pp. 237–339.

[70] F. Ihlenburg and I. Babuška. "Finite element solution of the Helmholtz equation with high wave number. I. The h-version of the FEM". In: *Comput. Math. Appl.* 30.9 (1995), pp. 9–37.

[71] P. Jolivet, F. Hecht, F. Nataf, and C. Prud'homme. "Scalable domain decomposition preconditioners for heterogeneous elliptic problems". In: *Proc. of the Int. Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE. 2013, pp. 1–11.

[72] P. Jolivet and P. H. Tournier. "Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers". In: *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*. Nov. 2016, pp. 190–203.

[73] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp element methods for CFD*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 1999, pp. xii+390.

[74] G. Karypis and V. Kumar. "A fast and high quality multilevel scheme for partitioning irregular graphs". In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 359–392.

[75] J.-H. Kimn and M. Sarkis. "Restricted overlapping balancing domain decomposition methods and restricted coarse problems for the Helmholtz problem". In: *Comput. Methods Appl. Mech. Engrg.* 196.8 (2007), pp. 1507–1514.

[76] T. Kolev and P. Vassilevski. "Auxiliary space AMG for $H(\mathrm{curl})$ problems". In: *Domain decomposition methods in science and engineering XVII*. Vol. 60. Lect. Notes Comput. Sci. Eng. Springer, Berlin, 2008, pp. 147–154.

[77] J. Lai and L. Olson. "Algebraic multigrid for high-order hierarchical $H(curl)$ finite elements". In: *SIAM J. Sci. Comput.* 33.5 (2011), pp. 2888–2902.

[78] J. Li and X. Tu. "Convergence analysis of a balancing domain decomposition method for solving a class of indefinite linear systems". In: *Numer. Linear Algebra Appl.* 16.9 (2009), pp. 745–773.

[79] J. C. Lin and M. J. Clarke. "Microwave imaging of cerebral edema". In: *Proceedings of the IEEE* 70.5 (May 1982), pp. 523–524.

[80] P.-L. Lions. "On the Schwarz alternating method. I." In: *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*. Ed. by R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux. Philadelphia, PA: SIAM, 1988, pp. 1–42.

[81] J. Mandel. "Balancing domain decomposition". In: *Comm. Numer. Methods Engrg.* 9.3 (1993), pp. 233–241.

[82] N. Marsic, C. Waltz, J. F. Lee, and C. Geuzaine. "Domain Decomposition Methods for Time-Harmonic Electromagnetic Waves With High-Order Whitney Forms". In: *IEEE Transactions on Magnetics* 52.3 (Mar. 2016), pp. 1–4.

[83] J. C. Maxwell. *A Treatise on Electricity and Magnetism*. A Treatise on Electricity and Magnetism v. 1. Clarendon Press, 1873.

[84] A. Moiola and E. A. Spence. "Is the Helmholtz equation really sign-indefinite?" In: *SIAM Rev.* 56.2 (2014), pp. 274–312.

[85] P. Monk. *Finite element methods for Maxwell's equations*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2003, pp. xiv+450.

[86] F. Nataf, H. Xiang, and V. Dolean. "A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps". In: *C. R. Math. Acad. Sci. Paris* 348.21-22 (2010), pp. 1163–1167.

[87] F. Nataf, H. Xiang, V. Dolean, and N. Spillane. "A coarse space construction based on local Dirichlet-to-Neumann maps". In: *SIAM J. Sci. Comput.* 33.4 (2011), pp. 1623–1642.

[88] J.-C. Nédélec. "A new family of mixed finite elements in $\mathbf{R}^3$". In: *Numer. Math.* 50.1 (1986), pp. 57–81.

[89] J.-C. Nédélec. "Mixed finite elements in $\mathbf{R}^3$". In: *Numer. Math.* 35.3 (1980), pp. 315–341.

[90] S. V. Nepomnyaschikh. "Decomposition and fictitious domains methods for elliptic boundary value problems". In: *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations (Norfolk, VA, 1991)*. SIAM, Philadelphia, PA, 1992, pp. 62–72.

[91] S. V. Nepomnyaschikh. "Mesh theorems on traces, normalizations of function traces and their inversion". In: *Soviet J. Numer. Anal. Math. Modelling* 6.3 (1991), pp. 223–242.

[92] R. A. Nicolaides. "Deflation of conjugate gradients with applications to boundary value problems". In: *SIAM J. Numer. Anal.* 24.2 (1987), pp. 355–365.

[93] S. J. Orfanidis. *Electromagnetic Waves and Antennas*. 2016. URL: `http://eceweb1.rutgers.edu/~orfanidi/ewa/`.

[94] F. Pellegrini and J. Roman. "SCOTCH: A Software Package for Static Mapping by Dual Recursive Bipartitioning of Process and Architecture Graphs". In: *High-Performance Computing and Networking*. Springer. 1996, pp. 493–498.

[95] M. Persson, A. Fhager, H. D. Trefná, Y. Yu, T. McKelvey, G. Pegenius, J. E. Karlsson, and M. Elam. "Microwave-Based Stroke Diagnosis Making Global Prehospital Thrombolytic Treatment Possible". In: *IEEE Transactions on Biomedical Engineering* 61.11 (Nov. 2014), pp. 2806–2817.

[96] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Vol. 23. Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 1994, pp. xvi+543.

[97] F. Rapetti. "High order edge elements on simplicial meshes". In: *M2AN Math. Model. Numer. Anal.* 41.6 (2007), pp. 1001–1020.

[98] F. Rapetti and A. Bossavit. "Whitney forms of higher degree". In: *SIAM J. Numer. Anal.* 47.3 (2009), pp. 2369–2386.

[99] S. Reitzinger and J. Schöberl. "An algebraic multigrid method for finite element discretizations with edge elements". In: *Numerical Linear Algebra with Applications* 9.3 (2002), pp. 223–238.

[100] O. Schenk and K. Gärtner. "Solving unsymmetric sparse systems of linear equations with PARDISO". In: *Future Generation Computer Systems* 20.3 (2004), pp. 475–487.

[101] J. Schöberl and S. Zaglmayr. "High order Nédélec elements with local complete sequence properties". In: *COMPEL* 24.2 (2005), pp. 374–384.

[102] H. A. Schwarz. "Über einen Grenzübergang durch alternierendes Verfahren". In: *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich* 15 (1870), pp. 272–286.

[103] S. Y. Semenov and D. R. Corfield. "Microwave Tomography for Brain Imaging: feasibility Assessment for Stroke Detection". In: *International Journal of Antennas and Propagation* (2008).

[104] S. Semenov, B. Seiser, E. Stoegmann, and E. Auff. "Electromagnetic tomography for brain imaging: From virtual to human brain". In: *2014 IEEE Conference on Antenna Measurements Applications (CAMA)*. Nov. 2014, pp. 1–4.

[105] V. Simoncini and D. B. Szyld. "Recent computational developments in Krylov subspace methods for linear systems". In: *Numer. Linear Algebra Appl.* 14.1 (2007), pp. 1–59.

[106] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. "Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps". In: *Numer. Math.* 126.4 (2014), pp. 741–770.

[107] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. "A robust two-level domain decomposition preconditioner for systems of PDEs". In: *C. R. Math. Acad. Sci. Paris* 349.23-24 (2011), pp. 1255–1259.

[108] G. Starke. "Field-of-values analysis of preconditioned iterative methods for nonsymmetric elliptic problems". In: *Numer. Math.* 78.1 (1997), pp. 103–117.

[109] B. Szabó and I. Babuška. *Finite element analysis*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1991, pp. xvi+368.

[110] B. Thierry, A. Vion, S. Tournier, M. El Bouajaji, D. Colignon, N. Marsic, X. Antoine, and C. Geuzaine. "GetDDM: an open framework for testing optimized Schwarz methods for time-harmonic wave problems". In: *Computer Physics Communications* 203 (2016), pp. 309–330.

[111] A. Toselli. "Overlapping Schwarz methods for Maxwell's equations in three dimensions". In: *Numer. Math.* 86.4 (2000), pp. 733–752.

[112] P.-H. Tournier, I. Aliferis, M. Bonazzoli, M. de Buhan, M. Darbas, V. Dolean, F. Hecht, P. Jolivet, I. El Kanfoud, C. Migliaccio, F. Nataf, C. Pichot, and S. Semenov. "Microwave Tomographic Imaging of Cerebrovascular Accidents by Using High-Performance Computing". hal-01343687 preprint. 2016.

[113] P.-H. Tournier, M. Bonazzoli, V. Dolean, F. Rapetti, F. Hecht, F. Nataf, I. Aliferis, I. El Kanfoud, C. Migliaccio, M. de Buhan, M. Darbas, S. Semenov, and C. Pichot. "Numerical Modeling and High Speed Parallel Computing: New Perspectives for Tomographic Microwave Imaging for Brain Stroke Detection and Monitoring". In: *IEEE Antennas and Propagation Magazine, accepted for publication* (2017).

[114] H. Whitney. *Geometric integration theory*. Princeton University Press, Princeton, N. J., 1957, pp. xv+387.

Résumé. Les équations de Maxwell en régime harmonique comportent plusieurs difficultés lorsque la fréquence est élevée. On peut notamment citer le fait que leur formulation variationnelle n'est pas définie positive et l'effet de pollution qui oblige à utiliser des maillages très fins, ce qui rend problématique la construction de solveurs itératifs efficaces. Ici nous proposons une stratégie de solution précise et rapide, qui associe une discrétisation par des éléments finis d'ordre élevé à des préconditionneurs de type décomposition de domaine. Les éléments finis d'ordre élevé permettent, pour une précision donnée, de réduire considérablement le nombre d'inconnues du sytème linéaire à résoudre. Ensuite, des méthodes de décomposition de domaine sont employées comme préconditionneurs du sytème linéaire pour le solveur itératif : le problème défini sur le domaine global est décomposé en des problèmes plus petits sur des sous-domaines, qui peuvent être résolus en parallèle et avec des solveurs directs robustes. Cependant la conception, l'implémentation et l'analyse des deux méthodes sont assez difficiles pour les équations de Maxwell. Les éléments finis adaptés à l'approximation du champ électrique sont les éléments finis H(rot)-conformes ou d'arête. Ici nous revisitons les degrés de liberté classiques définis par Nédélec, afin d'obtenir une expression plus pratique par rapport aux fonctions de base d'ordre élevé choisies. De plus, nous proposons une technique pour restaurer la dualité entre les fonctions de base et les degrés de liberté. Nous décrivons explicitement une stratégie d'implémentation qui a été appliquée dans le langage spécialisé et open source FreeFem++. Dans une deuxième partie, nous nous concentrons sur les techniques de préconditionnement du système linéaire résultant de la discrétisation par éléments finis. Nous commençons par la validation numérique d'un préconditionneur à un niveau, de type Schwarz avec recouvrement, avec des conditions de transmission d'impédance entre les sous-domaines. Ensuite, nous étudions comment des préconditionneurs à deux niveaux, analysés récemment pour l'équation de Helmholtz, se comportent pour les équations de Maxwell, des points de vue théorique et numérique. Nous appliquons ces méthodes à un problème à grande échelle qui découle de la modélisation d'un système d'imagerie micro-onde, pour la détection et le suivi des accidents vasculaires cérébraux. En effet, la précision et la vitesse de calcul sont essentielles dans cette application.

Mots-clés : équations de Maxwell en régime harmonique, éléments finis d'ordre élevé, éléments d'arête, éléments finis H(rot)-conformes, décomposition de domaine, préconditionneurs de Schwarz, préconditionneurs à deux niveaux, grille grossière, équation de Helmholtz, calcul haute performance, FreeFem++, imagerie micro-onde.

Abstract. The time-harmonic formulation of Maxwell's equations presents several difficulties when the frequency is large, such as the sign-indefiniteness of their variational formulation, the pollution effect which entails particularly fine meshes, and the consequent problematic construction of efficient iterative solvers. Here we propose a precise and efficient solution strategy that couples high order finite element discretizations with domain decomposition preconditioners. High order finite elements methods make it possible, for a given precision, to reduce significantly the number of unknowns of the algebraic linear system to be solved. Domain decomposition methods are then used as preconditioners for the iterative solver for the linear system: the problem defined on the global domain is decomposed into smaller problems on subdomains, which can be solved concurrently and using robust direct solvers. Nevertheless, the design, implementation and analysis of both these methods are particularly challenging for Maxwell's equations. Finite elements suited for the approximation of the electric field are the curl-conforming (or edge) finite elements. Here, we revisit the classical degrees of freedom defined by Nédélec, in order to obtain a new more friendly expression in terms of the chosen high order basis functions. Moreover, we propose a general technique to restore duality between degrees of freedom and basis functions. We explicitly describe an implementation strategy, which we embedded in the open source domain specific language FreeFem++. In the second part, we focus on the preconditioning of the system resulting from the finite element discretization, starting with a numerical validation of a one-level overlapping Schwarz preconditioner, with impedance transmission conditions between subdomains. Then we investigate how two-level preconditioners recently analyzed for the Helmholtz equation work in the Maxwell case, both from the theoretical and numerical points of view. We apply these methods to the large scale problem arising from the modeling of a microwave imaging system, for the detection and monitoring of brain strokes. In this application accuracy and computing speed are indeed of paramount importance.

Keywords: time-harmonic Maxwell's equations, high order finite elements, curl-conforming finite elements, edge elements, domain decomposition, Schwarz preconditioners, two-level preconditioners, coarse space, sign-indefinite problems, Helmholtz equation, high performance computing, FreeFem++, microwave imaging.