Académie universitaire Wallonie–Europe
Université de Liège — Faculté des Sciences Appliquées
Collège de doctorat en Électricité, électronique et informatique

Multi-Domain Approaches for the Solution of High-Frequency Time-Harmonic Propagation Problems

Doctoral Dissertation presented by

Alexandre VION

in fullfilment of the requirements for the degree of
Docteur en Sciences de l'Ingénieur

October 2014

Académie universitaire Wallonie–Europe
Université de Liège — Faculté des Sciences Appliquées
Collège de doctorat en Électricité, électronique et informatique

Multi-Domain Approaches for the Solution of High-Frequency Time-Harmonic Propagation Problems

Doctoral Dissertation presented by

Alexandre VION

in fullfilment of the requirements for the degree of
Docteur en Sciences de l'Ingénieur

October 2014

Thesis committee:

Prof. Xavier Antoine (Université de Lorraine)
Prof. Eric Bechet (Université de Liège), President
Prof. Victorita Dolean (University of Strathclyde)
Dr. Patrick Dular (Université de Liège), co-Advisor
Prof. Christophe Geuzaine (Université de Liège), Advisor
Prof. Marian Slodicka (Universiteit Gent)
Prof. Christiaan Stolk (University of Amsterdam)

This version generated on December 15, 2014

Abstract

The numerical solution of high-frequency time-harmonic propagation problems by volumic discretization methods is a challenging task, most notably because of the very large size of the resulting linear systems. We present a framework for a class of iterative methods that distribute the work between several CPUs and exchange information between physical or artificial interfaces. The goal is to define subproblems of manageable sizes, and to exploit the power of parallel supercomputers.

Furthermore, the framework comprises the possibility of adding another component, called a preconditioner, that aims at speeding up the convergence of the algorithms. We present two methods, that reduce the size of the subproblems in two different ways, either by defining them on smaller domains or by making them amenable to an alternative formulation that requires less fine meshes. We analyze their convergence and explore different preconditioning strategies.

Résumé

Résoudre numériquement des problèmes de propagation à haute fréquence et en régime harmonique par des méthodes de discrétisation volumique s'avère difficile, principalement en raison de la très grande taille des systèmes d'équations linéaires qui en résultent. Dans ce travail, nous présentons un cadre pour une classe de méthodes itératives, dont le point commun est de partager le travail entre processeurs et d'échanger des données entre un groupe d'interfaces, qui peuvent être physiques ou artificielles. L'objectif est de définir un ensemble de sous-problèmes de taille abordable, tout en exploitant le parallélisme des supercalculateurs modernes.

Le cadre offre en outre la possibilité d'incorporer un préconditionneur, dans le but d'accélérer la convergence. Nous présentons deux méthodes, qui réduisent le nombre d'inconnues des sous-problèmes soit en les définissant sur des domaines de taille réduite ou en permettant leur solution sur des maillages plus grossiers. Nous étudions la convergence des méthodes et explorons les manières de les préconditionner.

Remerciements

La première personne à remercier est tout naturellement le promoteur de cette thèse, le Prof. Christophe Geuzaine. C'est lui qui, en me faisant confiance il y a quasiment 6 ans (pour travailler sur un projet... d'électronique !), m'a permis de mettre le pied dans la porte et d'entamer ce travail dans le domaine de l'analyse numérique, qui je dois l'avouer, m'intéressait bien plus.
Les conditions dans lesquelles il m'a permis d'améliorer mes connaissances, d'effectuer cette recherche et de la communiquer lors des nombreuses conférences auxquelles j'ai eu l'occasion de participer ont été plus qu'optimales. Je suis conscient que bien peu d'étudiants en thèse ont la chance de bénéficier d'un tel support ; par ailleurs, la liberté qu'il m'a laissée sur le plan scientifique, de même que la pertinence de ses conseils lorsque j'en avais besoin, ont été grandement appréciées et ont constitué des sources de motivation indispensables pour mener à bien ce projet.

Je remercie dans son ensemble mon jury de thèse d'avoir accepté de lire mon travail et de le commenter. Parmi eux, Xavier a joué un rôle important puisque nous avons eu l'occasion d'échanger nos idées à de multiples reprises, ce qui m'aura très certainement permis de progresser. Sa gentillesse et son ouverture d'esprit rendent le travail avec lui d'autant plus agréable.

Depuis que j'ai débarqué dans le service de Christophe, une chose aura été constante : la bonne ambiance dans l'équipe. Elle aura pourtant bien changé depuis mes premiers pas où j'ai été accueilli par Fred et Raoul, alors les seuls "juniors" de l'équipe, qui avec mes premiers collègues chez ACE (Ruth, Patrick, David et Véro) m'ont fait une place. Il faut dire que de la place, à l'époque, il y en avait encore... Depuis, les choses ont bien évolué, le couloir s'est peu à peu rempli et beaucoup d'autres sont venus grossir les rangs. Je salue tous ceux que j'ai côtoyés pendant ces années, spécialement les "imports" venus de France : Bertrand, Simon et Vincent, jamais les derniers pour croquer un boulet avant d'attaquer les spéciales.

Entre autres collègues-et-néanmoins-amis, Axel aura partagé pendant quelques années mon bureau, et ce fut un réel plaisir que de faire peu à peu sa connaissance, avec moultes anecdotes et rigolades, mais aussi discussions sérieuses et intéressantes, si, si !

Parmi eux, Dave m'aura en particulier été d'une aide précieuse de par son aide pour l'utilisation des ressources en calcul, ainsi que sa disponibilité à m'écouter pester contre les machines qui déconnent et les "bad users" qui squattent la file. Je souhaite aussi remercier les techniciens du labo, à commencer par Pierre et Jeannot, qui m'ont épaulé lors de ma première mission dans le service. Ils sont toujours prêts à dépanner dans l'instant, même lorsqu'il s'agit d'extras (genre : "Les gars, vous savez changer un phare de voiture ?").

Un moment particulièrement important dans la genèse de ce travail a été mon séjour au MIT. Je tiens donc tout particulièrement à remercier le Prof. Laurent Demanet d'avoir accepté de faire le nécessaire pour pouvoir m'accueillir là-bas. C'est notamment lui qui m'a mis sur la piste de certaines des idées les plus centrales de ma recherche. C'est encore à Boston que j'ai fait la connaissance de Rosalie et Léo, avec qui j'ai gardé une très chouette relation.

Je remercie tous mes amis pour les indispensables moments de détente, que ce soit pour se payer une bonne tranche de rock'n'roll (le bar du CI fait évidemment partie du rock'n'roll sus-mentionné) ou aller faire une sortie en bateau, mes deux grandes passions.

Je ne puis conclure cette liste sans penser à ma famille ; Laurence a évidemment fait partie de celle-ci et aura, entre autres, contribué à me donner l'envie d'entreprendre cette aventure.

Alexandre Vion,
Décembre 2014.

Contents

Introduction

Computational electromagnetics is the branch of science that aims at developing models for the numerical simulation of a broad range of physical phenomena linked to electromagnetic fields. Starting from Maxwell's equations and constitutive material laws, that provide the full model for the accurate description of the physical phenomena in terms of electric and magnetic fields, it consists in the introduction of simplifications and modelling tools, as well as a discretization of the resulting equations, in view of their solution by means of computers.

It is an exciting field of study: it involves many aspects of (applied) mathematics, physical modelling and the identification of relevant approximations, to finally design efficient algorithms; finding the best methods requires careful analysis of the problem, as well as a great deal of creativity in order to propose new and innovative approaches. In that process, it is crucial to be aware of the existing techniques and current directions of research in order to be able to borrow ideas, possibly from other fields, mix them together and create one's own collection of building blocks, that will allow for the design of new methods and the apparition of new ideas.

Computational electromagnetics has seen considerable progress over the past decades, in particular with the development of the finite element method (FEM) that relies on a mathematically sound framework for the discretization of the problems. The FEM has now become an essential tool for the solution of practical engineering problems. In parallel, the emergence of supercomputers and the spectacular drop in the cost of computational resources has opened new horizons for the development of always more accurate numerical models. Such models are needed to accompany the present and future technological progress, in fields as various as consumer electronics or medical imaging.

Among the most computationally demanding engineering problems is the simulation of high-frequency propagation problems, while the industrial need for such simulations is booming. Indeed, recent years have seen tremendous technological progress and connected devices have become omnipresent, while consumers are always requesting faster communications. Another example of application is high-resolution imaging (e.g. for seismic exploration or medical purposes), that is also very much in demand of efficient computational methods suit-

able for high-frequency problems, since the solution of the inverse problems involved requires the solution of many forward problems.

In the context of numerical modelling, our understanding of "high-frequency" is that we are interested in the situation where the wavelength at the working frequency is small compared to the size of the surrounding objects, or more generally of the computational domain, but where it is not relevant to make the assumption of very small (asymptotically zero) wavelength, for which special methods are available.

Motivation of this work

As working frequencies and/or the size of the computational domains are constantly increasing, the solution of propagation problems has appeared to be a particularly challenging problem. Indeed, the size of the corresponding discrete linear systems and their mathematical properties make them hard to solve using usual methods: they either have reached their limit, requiring excessive computation times and memory usage (like e.g. the family of so-called direct solvers) or are not adapted to this particular case, like most currently available iterative methods that have been developed for other problems.

On the other hand, with the advent of the new computational resources have also appeared new concerns, related to their efficient use and the minimization of their environmental impact: running and maintaining high performance computational facilities still involves considerable amount of expenses, both in terms of financial investment and energy. It is therefore important that the philosophy of computational engineers evolves in accordance with the growth of computational power. Indeed, simply using the newly available resources to perform brute force solution of larger problems makes little sense and is bound to fail; it is well known that the cost of solving problems by conventional methods grows much faster than their size, while it is clear that these methods cannot make an efficient use of the power of the massively parallel architectures of computers of today and tomorrow.

Therefore, research in computational electromagnetics is experiencing a shift from a modelling effort (the comprehensive mathematical description of the physical phenomena, and the different ways of formulating the problems) to the design of new strategies for the solution of these models, that can fully take advantage of modern computers and respect their constraints.

We believe that in order to achieve that goal, it is fundamental to take into account the specificities of each particular class of problem, and to look for inspiration into the underlying physics. This philosophy is at the heart of the methods that we propose in this thesis.

Scope and objectives

With this work, we intend to contribute to the development of efficient solvers dedicated to time-harmonic propagation problems, i.e. we will consider both scalar (Helmholtz equation) and vector (Maxwell's equations) cases, and will always aim at methods that are well suited for the high-frequency regime. Most of the presentation will be done for the scalar case and modifications for the vector case will be given as a complement.

We have identified the following main objectives:

- design suitable methods for the solution of the large linear systems arising from the discretization of the set of partial differential equations that describe propagation problems at high frequency, and that can efficiently take advantage of parallel computers;
- for this last purpose, these algorithms should be based on the solution of subproblems defined on a new set of domains; although different methods could be considered for their solution (provided that they are accurate enough), we will choose to solve them by the finite element method;
- identify existing methods that have a good potential for parallelism. In view of the first goal of this list, we will focus on iterative methods since it is clear that direct methods are not good candidates;
- implement the methods inside a general environment, where all components of the method will be described: the general loop that solves the problem, via the definition of an iteration operator, possibly with a preconditioner; the definition of the subproblems; the post-processing that generates the solution to the full problem;
- identify the limitations of the selected methods, understand their origin, propose directions for improvement and implement them inside our solver;
- benchmark the methods on some model problems that reflect the variety of engineering problems encountered in practice.

Outline

We have organized this thesis into four chapters. In the first one, after introducing our problem of interest we give insights to explain why solving it at high frequencies is difficult. For that purpose, we review the different families of classical approaches (partial differential equations and integral equations, as well as a few asymptotic methods for the very high frequencies), and focus on the finite element method. Then, we explore the different strategies that are classically used for solving other problems and show that they fail when frequency is increased.

In Chapter 2, we provide a common framework for methods that share the principle of being based on some redefinition of the computational domain, in a way that naturally enables the iterative solution of an equivalent problem. We will call them "multi-domain" methods, and will detail two of them: a non-overlapping domain decomposition algorithm, and a method for the solution of multiple scattering problems.

The next two chapters are dedicated to the original contributions of this thesis, where we will bring improvements to the aforementioned methods: Chapter 3 provides a way to improve the convergence of the DDM algorithm by means of a preconditioning technique, and in Chapter 4 we take advantage of an alternative formulation applicable to the problems on our newly defined domains, to speed up their solution and make the algorithm usable in practice.

Finally, we will draw some conclusions and formulate perspectives for future research on the topic.

Original contributions

This work is a contribution to the numerical solution of propagation problems at frequencies high enough so that usual methods are no longer applicable. In this regard, we have proposed the following list of presumably original contributions:

- we have defined a common framework for iterative methods based on a decomposition of the original domain into several new domains. It is based on the reformulation of the (existing) algorithms as matrix-free linear systems: we solve a linear system with a Krylov method, without explicitely forming the corresponding matrix but by giving the application of that matrix to some vector, as an iteration. The resulting algorithms are based on subdomain solves and exchange of information;
- this framework naturally provides the possibility of introducing preconditioners to improve their convergence. These preconditioners can be based on similar subproblem operators as the original iteration operator, or be based on other techniques;

- in the context of optimized Schwarz methods, we have compared various types of transmission conditions, including a non-local one based on perfectly matched layers, on a set of model problems used as test cases; the link with a matrix probing technique to obtain the non-local transmission operator has been made, to accelerate its otherwise excessively costly computation. That idea has been published in [182];

- in order to improve the convergence of Schwarz methods, we have proposed a sweeping preconditioner, that works by improving the transfer of information over distant subdomains; our numerical tests have shown that in some cases the convergence of the unpreconditioned method is so slow that it cannot be used in practice, while the preconditioner makes it possible to obtain a solution. This has led to the journal publication [180];

- since this kind of preconditioner is intrinsically non-parallel, and observing that a global sharing of information is not always required in view of the topology of the problem, we propose a modified version that performs smaller and independent sweeps on groups of subdomains. While this modification slightly degrades the convergence rate, it has the advantage of, at least partially, restoring the parallelism of the method. This extension has been published in [179];

- we propose an algorithm for the solution of multiple scattering problems based on an alternative formulation of the simple scattering subproblems involved in the iteration operator. This reformulation has the advantage of being applicable with much coarser meshes than with the original formulation. This has led to the journal publication [99].

The above methods have been implemented and made available for the community of computational electromagnetics engineers in the open-source software GetDP [95, 175]. We also believe that our methods, by their modularity, will be applicable to other fields of engineering and physics, since they are described in terms of transfer operators, that can be redefined to suit the purposes of other communities.

CHAPTER 1

A challenging problem: the numerical solution of high-frequency time-harmonic wave propagation

In this chapter, we detail the problem under consideration and give some practical examples of application. We briefly describe different existing families of methods for its numerical solution, and discuss their advantages and drawbacks. We then particularly focus on the finite element method and motivate this choice. Since the main difficulty of the method resides in the very large size of the resulting linear systems at high frequencies, we conclude the chapter by introducing some usual methods for their solution and expose their limitations when applied to propagation problems.

1.1 Problem description

Wave propagation phenomena occur in various fields of physics, among which practically important ones are in particular: solid mechanics (e.g. vibration of structures), fluid mechanics (e.g. surface waves), acoustics, electromagnetism or quantum physics. Although the physical variable of interest is different in each of them (displacement, acoustic pressure, electric or magnetic fields, ...), they are all mathematically governed by similar sets of equations. In this section, starting from an equation that fully describes many wave phenomena with their time dependency, we will introduce the time-harmonic equation that holds for the systems in steady state, that is for a harmonic excitation with constant frequency and amplitude and after the transients have vanished.

We focus on linear waves, both in constant and heterogeneous media, for acoustic and electromagnetic applications. Most of the presentation will be given for the scalar (Helmholtz) case, with a brief introduction to the vector (Maxwell) case.

1.1.1 Time-dependent wave equation

The equation classically known as the linear wave equation is a scalar partial differential equation (PDE) that involves second order derivatives in both the time t and spatial $\boldsymbol{x} = (x_1, \ldots, x_d)$ variables, in dimension d. Its solutions $U(\boldsymbol{x}, t)$ satisfy:

$$\left(\Delta - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right)U = 0, \tag{1.1}$$

where $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial_{x_i}^2}$ (in cartesian coordinates) is the Laplacian operator and $c(\boldsymbol{x})$ is the local speed of propagation of the wave in the medium. That equation was first proposed in the 18-th century by French scientist d'Alembert in the 1-dimensional case, who studied the shape of a vibrating string [47], and later by Euler in higher dimensions when he was working on the propagation of sound [80].

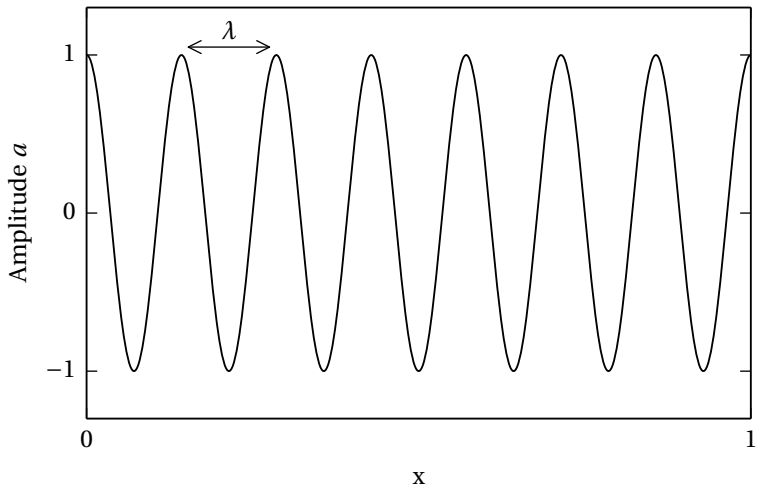


Figure 1.1: Illustration of the real part of a monochromatic wave $u(\boldsymbol{x}) = ae^{\imath k\boldsymbol{x}}$ with amplitude $a = 1$, propagating in a medium $\Omega = [0,1]$ with velocity $c = 1$. The pulsation is set to $\omega = 14\pi$, so the wavenumber is $k = 14\pi$ and the wavelength is $\lambda = 1/7$.

Important concepts for the characterisation of a (monochromatic) wave are the amplitude a, the frequency f and the wavelength λ. The frequency is a time-related quantity: it is the inverse of the period of the excitation, that is typically a periodic function. One also often speaks in terms of the angular frequency or pulsation $\omega = 2\pi f$. The wavelength is the spatial counterpart of the period: it is the distance between two successive crests (or throughs) of a wave, or more generally between two consecutive points of the same phase. The wavenumber k is another frequently used quantity, related to the wavelength as $k = \frac{2\pi}{\lambda}$. In a given medium with speed of propagation c, the wavenumber and angular frequency are related by the relation $k = \frac{\omega}{c}$. This is illustrated on a 1d example on Figure 1.1.

With the above definitions and relations, we see that higher frequencies produce smaller wavelengths. This is an important observation to understand the difficulties that arise in the numerical solution of high-frequency problems: while fine discretizations are required to represent the fast oscillations of the solution in space, the Courant-Friedrichs-Lewy condition [45] states that accordingly small time steps are required to guarantee the stability of some time-integration schemes (e.g. explicit Euler — implicit schemes are unconditionally stable but are more computationally intensive and can be less accurate for large time steps [110]). Since, in many practical cases, one is interested in the steady-state solution, a large number of time steps may be required to reach that state, leading to extremely long simulation times.

An alternative is to directly look for harmonic solutions. That strategy leads to a different equation known as the Helmholtz equation, and is the object of the next Section. We will see that numerically solving that equation is not an easy task: at high-frequency, the wavelength of the solution is small compared to the size of the computational domain, and the number of discretization points required to represent the solution becomes large. Moreover, the properties of the Helmholtz equation cause usual solution techniques to lose their efficiency or even to fail. This thesis is thus devoted to the design of dedicated methods for the solution of such high-frequency problems.

1.1.2 The Helmholtz equation and boundary conditions

Looking for a steady-state solution, hence independent of time, to (1.1), we suppose a harmonic time dependence of the form $U(\boldsymbol{x},t) = \Re e\{u(\boldsymbol{x})e^{-\imath\omega t}\}$, where ω is the angular frequency, and inject that relation in the wave equation (1.1). We obtain a new equation that describes the spatial repartition u of the unknown field at frequency ω:

$$-(\Delta + k^2)u = 0, \tag{1.2}$$

where $k(\boldsymbol{x}) = \frac{\omega}{c(\boldsymbol{x})}$ is the wavenumber introduced above. The wavenumber can either be constant as in free-space problems, or depend on the coordinates as in heterogeneous materials.

Equation (1.2) belongs to the category of elliptic problems (see [140] for a formal definition) and is known as the homogeneous Helmholtz equation. It admits a variety of solutions, depending on the source and boundary conditions, that have in common an oscillatory nature (provided $\Re e\{k\} > 0$; the situation is similar when $\Im m\{k\} > 0$, but it corresponds to a dissipative medium and produces damped waves). We mention in particular plane waves that are solutions of the form $u_{\text{pw}} = e^{\pm\imath\boldsymbol{k}\boldsymbol{x}}$, with the wave vector $\boldsymbol{k} = (k_1, k_2, k_3)$ that determines the direction of propagation of the wave, such that $|\boldsymbol{k}| = k$. Another example is a spherical wave $u_{\text{sw}} = e^{\imath\boldsymbol{k}\boldsymbol{r}}/|\boldsymbol{r}|$ produced by a point source in free space at the origin and $\boldsymbol{r}$ is the radial coordinate in spherical coordinates.

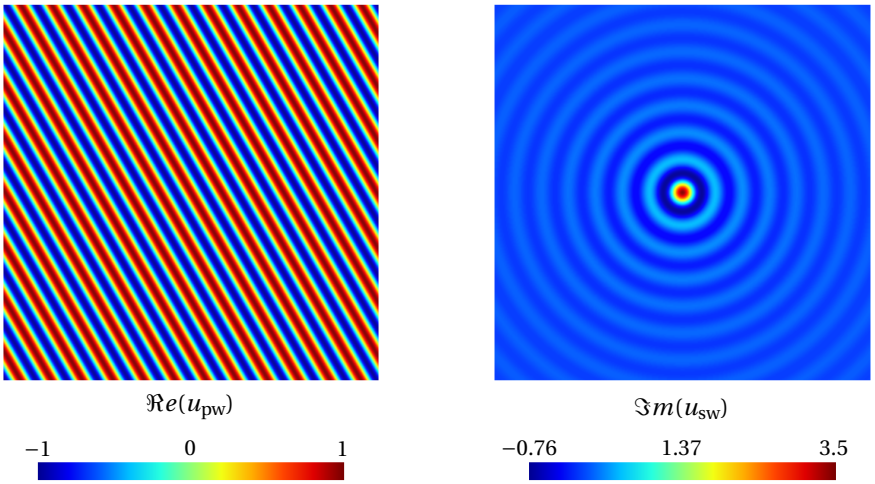


Figure 1.2: Particular solutions to the Helmholtz equation, in the unit square at $k = 7\pi$: a plane wave u_{pw} (left) with wave vector $\boldsymbol{k} = (\cos(\pi/6), \sin(\pi/6), 0)$, and a spherical wave produced by a point source u_{sw} (right); note the attenuation as one moves away from the source.

As for any elliptic PDE problem, an appropriate set of boundary conditions must be specified on the boundary Γ of the computational domain Ω for the solution to be uniquely determined (the definition of Ω varies for each problem; we give some examples in the next section). The governing equation in volume together with the set of boundary conditions define a boundary value problem (BVP). It is common to impose the field to vanish on part of the boundary, which leads to a homogeneous Dirichlet boundary condition if one computes the total field, or a non-homogeneous condition of the form $u = -u_{\text{inc}}$ for the scattered field (see next section for a definition of these notions). One then speaks of a sound-soft obstacle, while sound-hard refers to Neumann conditions on the normal derivative of the field, of the form $\partial_n u = 0$.

An important caveat for the Helmholtz problem with Dirichlet or Neumann boundary conditions only is that the problem can be ill-posed if k^2 corresponds to an eigenvalue of the Laplacian on the domain. Indeed, any multiple of the corresponding eigenfunction, called a resonant mode, solves the problem and the solution is then non-unique. In the following we will assume that the Dirichlet problems are studied at wavenumbers away from the resonant frequencies.

In the particular case of propagation phenomena, since one can define open problems with domains that extend to infinity and since waves have the ability to propagate over very long distances, special care must be taken with boundary conditions imposed at infinity. The Sommerfeld radiation condition is one that ensures that a wave originating from a source at finite distance is outgoing at in-

finity. It therefore prevents the existence of an (unphysical) wave coming from infinity and sinking at the source, which is otherwise also a valid solution to the Helmhotz equation. It reads, in dimension d:

$$\lim_{|\boldsymbol{r}|\to\infty} |\boldsymbol{r}|^{\frac{d-1}{2}} \Big(\frac{\partial}{\partial|\boldsymbol{r}|} - \imath k\Big) u(\boldsymbol{r}) = 0, \tag{1.3}$$

in any radial direction $\hat{\boldsymbol{r}} = \boldsymbol{r}/|\boldsymbol{r}|$. Although the Sommerfeld condition remains exact at finite distance in 1-dimension [102], this is no longer the case in higher dimensions. We will see in Section 1.2.2 that the truncation of the computational domain at finite distance is an essential feature of PDE-based numerical solution of propagation problems in an unbounded domain Ω, and present some appropriate alternatives to (1.3) in Section 1.3.3.

1.1.3 Model problems

We now present a few simple problems that are representative of the variety of practical engineering problems involving propagation. They will be used later in this work as test cases for our methods. Some useful definitions and notations are also given.

Scattering in free space

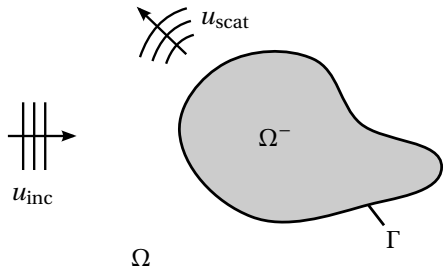


Figure 1.3: Illustration of the scattering of an incident wave by an object. The presence of the object perturbs the wavefronts of the incident field u_{inc}, here depicted as a plane wave, by superimposing the scattered field u_{scat} to it.

When a wave propagates in space and hits one or more objects, the interactions between the wave and the objects create a perturbation of the wavefronts, as part of the wave is reflected back in different directions. A shadow is also created behind the objects. That phenomenon is known as scattering.

A practically important problem is the study of the scattering by an object in free space. That is, the object is considered as immerged in an infinite medium with constant properties, such as vacuum, air or water; an example is the reflection of a radar wave by an aircraft. In many cases, the incident field is supposed to be a plane wave, but any kind of wave could be considered as well.

We will focus on the case of impenetrable objects, noted Ω^- (in case of multiple objects, we will have $\Omega^- = \cup\Omega_i^-$), with boundary $\Gamma = \partial\Omega^-$ (see Figure 1.3). In that case, it is common to impose boundary conditions on the total field on the surface of the objects, that are either of the Dirichlet type: $u_{\text{tot}} = 0$, or of the Neumann type: $\partial_{\boldsymbol{n}} u_{\text{tot}} = 0$, with $\boldsymbol{n}$ the outward normal to the exterior domain, i.e., directed inward to the object(s). By total field, we mean the sum of the incident field and the scattered field: $u_{\text{tot}} = u_{\text{inc}} + u_{\text{scat}}$. In such problems, since the incident field is supposed to be known in the entire space, it is convenient to exploit the linearity of the problem and to compute the solution in terms of the scattered field. In the following, we will use the simplified notation $u = u_{\text{scat}}$. The propagation domain is the exterior of the object: $\Omega = \mathbb{R}^d \setminus \Omega^-$, in dimension $d = 2, 3$. It is also natural to impose that the scattered field is outgoing at infinity, and thus to apply a radiation condition. The problem then writes:

$$\begin{aligned} &\Delta u + k^2 u = 0 && \text{in } \Omega, \quad \text{with } k = k_0 \\ &\qquad u = -u^{\text{inc}} && \text{on } \Gamma, \\ &\text{or} \quad \partial_n u = -\partial_n u^{\text{inc}} && \\ &+ \text{radiation condition} && \text{at } |r| \to \infty. \end{aligned} \tag{1.4}$$

This problem is known to be well posed [135]. However, the propagation domain Ω is unbounded. We will see in Section 1.3.3 that some methods cannot work with such domains, that must therefore be truncated.

Guided waves

At the opposite of waves propagating freely in an open medium, is the case of waves that are confined in a structure with impenetrable walls. These waves are forced to follow the direction given by the structure, therefore called a waveguide, that can be of various shapes. Straight or bent waveguides are commonly encountered; we will use an S-shaped waveguide as test case for our methods.

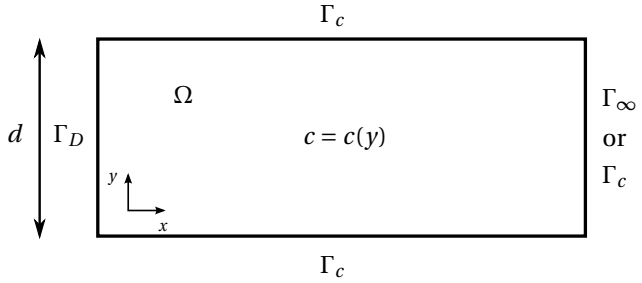


Figure 1.4: Example of a straight waveguide, and definition of the different types of boundaries. The interior medium needs not be homogeneous. Note that the right boundary can be either open or closed (cavity).

An important property of waveguides is that not any wave can propagate inside them: each mode is characterized by a cut-off frequency, below which the mode excited at one end cannot propagate and results in an evanescent wave [113, 153]; losses and internal reflections can also occur. It is therefore usual to study them by exciting one mode at a time, and to compute quantities such as transmission ratios. The medium inside the waveguide can either be homogeneous (e.g. air, or vacuum) or heterogeneous with some velocity profile $c = c(y)$, like in an optical fiber.

As an example, we consider the simple case of a straight, homogeneous 2d semi-infinite waveguide with width d, oriented along the $x-$axis (Figure 1.4). The computational domain Ω is the interior of the waveguide, with boundary $\Gamma = \Gamma_D \cup \Gamma_C \cup \Gamma_\infty$. Suppose that we impose homogeneous Dirichlet boundary conditions on the walls Γ_c, and excite the $m-$th mode at the inlet Γ_D with wavenumber k:

$$u_D^{(m)}(0, y) = \sin(\frac{m\pi}{d} y); \tag{1.5}$$

the solution inside the waveguide is [113, 153]:

$$u_m(x, y) = \sin(\frac{m\pi}{d} y) e^{\imath \beta_m x}, \tag{1.6}$$

with $\beta_m = \sqrt{k^2 - k_c^2}$. The cut-off frequency of mode m is given by $k_c = \frac{m\pi}{d}$; if the wavenumber k is beneath k_c, the propagation constant β_m is complex and the resulting wave does not propagate but its amplitude decreases exponentially in the waveguide instead.

To numerically solve such problems, one must either consider finite waveguides, where the open end can be modeled by means of an absorbing condition, or truncate the continuing waveguide with a similar condition. Another case is to consider a closed waveguide, where a wall condition is also imposed at the end. Such a configuration is called a cavity, and can only be studied at frequencies away from resonances to guarantee the well-posedness of the problem. The definitions are illustrated on Figure 1.4. We solve for the field u, that verifies:

$$\begin{aligned}
\Delta u + k^2(y) u &= 0 && \text{in } \Omega, \\
u &= -u_D^{(m)} && \text{on } \Gamma_D, \\
u &= 0 && \text{on } \Gamma_c, \\
\text{or} \quad \partial_n u &= 0 && \\
+ \text{ absorbing condition} & && \text{on } \Gamma_\infty.
\end{aligned} \tag{1.7}$$

Propagation in non-homogeneous media

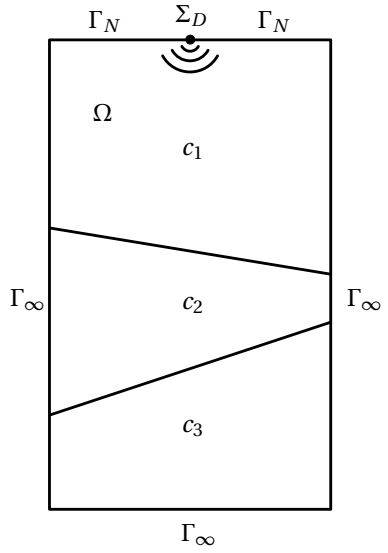


Figure 1.5: Illustration and defintions of the simplified underground propagation of a wave produced by a point source located on surface. The propagation medium is non-homogeneous, here with three different velocities.

Another practically important problem is the propagation of a wave in a strongly heterogeneous medium, such as the top layers of the earth, for underground exploration. This kind of simulation is central in seismic inversion problems. We consider a point source Σ_D with pulsation ω located on the surface, modeled by a Neumann condition and noted Γ_N. The underground sides are modeled by absorbing conditions imposed on the truncation boundary Γ_∞ (Figure 1.5). Thus, we compute the solution in the region Ω enclosed in $\Gamma = \Gamma_N \cup \Gamma_\infty$. This problem writes:

$$\begin{aligned} \Delta u + \frac{\omega^2}{c^2(\boldsymbol{x})} u = 0 \qquad & \text{in } \Omega, \\ u = 1 \qquad & \text{on } \Sigma_D, \\ \partial_n u = 0 \qquad & \text{on } \Gamma_N, \\ + \text{ absorbing condition} \qquad & \text{on } \Gamma_\infty. \end{aligned} \tag{1.8}$$

1.2 Numerical techniques for the Helmholtz equation

Analytical solutions for wave problems are only available for very simple configurations, such as the scattering by a single disc or the propagation in a straight and infinite waveguide. Since real-life engineering problems usually feature complex geometries and possibly highly non-homogeneous media, it becomes nec-

essary to resort to numerical techniques in order to approximate the solution. In this section, we briefly introduce some of the most popular methods and list their advantages and drawbacks. Although this thesis is essentially devoted to the developement of algorithms built on top of the finite element method described in the next section, we will see that some concepts and ideas from the other families of methods described here will be useful and we will emphasize the connections with them whenever relevant.

1.2.1 Boundary integral equations

In the particular case of an homogeneous propagation medium $k(\boldsymbol{x}) = k$, for which fundamental solutions of the Helmholtz equation (1.2) are available, exterior scattering problems can be solved by boundary integral equation methods. In these methods, based on an integral representation of the fields (more details are given in Appendix B; see also [44, 139] for a more complete treatment), one solves an equivalent equation $L\rho = f$, with L an integral operator of the form (other forms are possible):

$$L\rho(\boldsymbol{x}) = \int_\Gamma G(\boldsymbol{x},\boldsymbol{y})\rho(\boldsymbol{y})d\Gamma, \quad \forall \boldsymbol{x} \text{on } \Gamma, \tag{1.9}$$

where the kernel G is the fundamental solution (or Green's function) (B.4) to the Helmholtz equation and ρ is an unknown density function defined on the boundary of the scatterers. Once this function is known, the field can be computed anywhere in the volume by simply evaluating a surface integral.

Such an approach has several advantages:

- it is naturally suitable for unbounded problems: there is no need to introduce domain truncation techniques as for volumic methods;
- reduction of dimensionality: a problem defined in dimension d reduces to a boundary problem, hence of dimension $d-1$;
- far-field calculations are easily done once the surface problem is solved.

Conversely, there are also difficulties:

- the discretized problems lead to dense (sometimes ill-conditioned) linear systems of equations[1];
- their iterative solution requires careful preconditioning, which is still an active field of research;

[1] Algorithms exist however to perform sparse (approximate) matrix-vector products, like the fast multipole method (FMM) [50].

- special care is required in the implementation of these methods in reason of the singularity of the kernel of the integral operator;
- they are only applicable to propagation in homogeneous media, because Green's function must be known everywhere, which is usually not the case for non-homogeneous media.

A typical field of practical application of integral methods is the calculation of the radar cross section (RCS) [14] of conducting objects, like aircrafts; an example of numerical solver based on such methods is BEM++ [164].

1.2.2 PDE-based methods

This family of methods work by locally approximating the differential operators that appear in the equation to be solved on a discrete grid, everywhere in the computional domain, as opposed to the integral methods. For this reason, they are sometimes called respectively volumic and surfacic methods.

The finite difference method (FDM) makes use of a difference quotient to approximate the derivatives on a regular grid with step size h. Examples are the forward or backward difference approximations. In 1d, this gives the simple formulas:

$$\partial_x u(\boldsymbol{x}_i) \approx \frac{u_{i+1} - u_i}{h} \quad \text{or} \quad \partial_x u(\boldsymbol{x}_i) \approx \frac{u_i - u_{i-1}}{h}. \tag{1.10}$$

Second order derivatives are approximated by the central difference formula:

$$\partial_{xx} u(\boldsymbol{x}_i) \approx -\frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2}, \tag{1.11}$$

known as the 3 points stencil. In 2 and 3 dimensions, similar formulas lead to respectively 5 and 9 points stencils. The application of these stencils over the domain and the addition of the boundary conditions yield a linear system of equations, the solution of which is an approximate solution to the PDE problem.

Another important example of such methods is the finite element method (FEM), described in Section 1.3. It is based on a variational formulation of the problem and approximates the solution as a sum of compactly supported, piecewise polynomial functions defined on a discretization of the domain. A major advantage over the FDM is that a structured mesh is not required, allowing to mesh complex-shaped domains and representing fine details, e.g. using triangles for surfaces and tetrahedrons for volumes, without need to also finely mesh regions where a coarser dicretization can be used instead.

Advantages of the methods are:

- ease of implementation (FDM);
- discretization can conform to complex geometries and fine details (FEM);
- suitable for variable coefficients problems, e.g. non-homogeneous media ($k = k(\boldsymbol{x})$);
- local approximation of the operator, resulting in sparse linear systems (optimized storage, fast matrix-vector products).

Associated drawbacks:

- need to truncate the domain for unbounded problems, with special kinds of boundary conditions (which can introduce an error before discretization);
- not well suited for some non-local operators (like some particular boundary operators), therefore requiring local approximations of them;
- very large number of discretization points at high frequency, leading to large linear systems to be solved;
- sign-indefiniteness of the system when applied to propagation problems.

There are many available implementations of these methods, both commercial or open-source. We have used the GetDP [60, 95] finite element solver together with the Gmsh [96, 97] mesh generator and solution viewer.

1.2.3 Asymptotic methods

Different approximations can be made to facilitate the solution of propagation problems, especially by taking the high-frequency (or equivalently the small wavelength) limit, i.e. that the wavenumber asymptotically tends towards infinity. Among these, we choose to briefly describe and discuss two of the most popular techniques, since they will be useful and since interesting connections can be made with other ideas in the remainder of this work.

Geometrical optics (GO)

One of the earliest and most widespread description of the propagation of light is based on the concepts of light rays and wavefronts [25, 166]: in a medium, light rays propagate along curves that are perpendicular to the surfaces of equal phase. This model along with Huygen's principle succesfully predicts practically important phenomena like reflection and refraction in non-homogeneous media.

By searching for a solution to the Helmoltz equation (1.2) of the form:

$$u(\boldsymbol{x}) = a(\boldsymbol{x})e^{\imath k\phi(\boldsymbol{x})}, \tag{1.12}$$

where a and ϕ are slowly varying functions, one obtains after rearranging the terms as powers of $1/k$:

$$\left(1-|\nabla\phi|^2\right)a + \imath\frac{1}{k}(a\Delta\phi + 2\nabla\phi\cdot\nabla a) + \frac{1}{k^2}\Delta a = 0. \tag{1.13}$$

With the high-frequency asymptotic approximation $k \to \infty$, one can neglect the second order term and the above equation turns into the system of geometrical optics:

$$\begin{aligned} |\nabla\phi|^2 &= 1, \\ a\Delta\phi + 2\nabla\phi\cdot\nabla a &= 0. \end{aligned} \tag{1.14}$$

The first equation is known as the eikonal equation. It is a non-linear equation that describes the evolution of the phase ϕ, starting from boundary conditions. The second one is a linear transport equation for the amplitude a.

There are various techniques for the numerical computation of solutions to the eikonal equation, among which we mention the popular and efficient $\mathcal{O}(N\log N)$ fast marching algorithm [163], which is of the expanding wavefront type. Iterative schemes have also been proposed [48, 114, 157].

The geometrical theory of diffraction (GTD) was later developed [118] to take into account the diffraction phenomena that are absent in the GO model [25].

Paraxial approximation

In many situations, waves tend to propagate in a well identified preferential direction, like radio waves in the atmosphere [87, 122]. That direction is called the paraxial direction. Rewriting the 2d helmoltz equation in the (x,z) plane as:

$$\partial_x^2 u + \partial_z^2 u + k^2 u = 0, \tag{1.15}$$

and considering a wave that propagates along the x-axis, we can factor the operator above as:

$$(\partial_x + \imath\mathcal{P}^+)(\partial_x - \imath\mathcal{P}^-)u = 0, \tag{1.16}$$

that can be seen as the decomposition between a forward and backward propagating waves, with the non-local pseudo-differential operators [173] $\mathcal{P}^\pm$ given by:

$$\mathcal{P}^\pm = \sqrt{\partial_z^2 + k^2}. \tag{1.17}$$

A similar idea is used as starting point in the design of the AILU preconditioner for the full discrete problem (Section 1.4.2). The paraxial approximation consists in considering only the forward propagating part of the wave described by the equation $(\partial_x + \imath\mathcal{P}^+)u = 0$, that can be seen as an evolution equation along the x-direction. One must choose a local approximation of the square-root operator (1.17), which can be done by a truncated Taylor expansion:

$$\sqrt{1+X} \approx 1 + \frac{1}{2}X. \tag{1.18}$$

This leads to the parabolic equation:

$$\partial_x u - \imath k u - \frac{\imath}{2k}\partial_z^2 u = 0, \tag{1.19}$$

that can be solved using standard numerical schemes [123]. Other approximations of the square-root can be used to improve the quality of the approximation, especially when the propagation direction deviates from the paraxial direction, like a Padé rationale approximation [42]:

$$\sqrt{1+X} \approx C_0 + \sum_{l=1}^{N_p} \frac{A_j X}{1 + B_j X}, \tag{1.20}$$

where the coefficients are known [12].

Regarding the pros and cons of the asymptotic methods, their main interest resides in their efficiency. They are based on a priori simplifications that succesfully capture the most important features of the solution. However they are, by definition, approximate methods. Therefore, while they are well suited for some particular problems, they do not take into account all the physics of propagation phenomena and will fail when applied to other types of problems.

1.3 The finite element method

In view of its properties, the finite element method introduced in Section 1.2.2 is a good candidate for the solution of propagation problems, especially in the non-homogeneous case. All the model problems presented in Section 1.1.3 are amenabe to it, with so little adjustments regarding the boundary conditions that it is possible to write a general code that is able to solve any of them. To define a new problem, the user needs only provide a geometrical description of the problem and the definition of the different regions and boundaries.

This section introduces in detail the method, as well as the necessary tools to apply it to scalar propagation problems. Details on the vector version will be given in a separate section at the end of the chapter.

1.3.1 Weak formulation

In Section 1.1.2, we have given the governing equations, in the scalar case, of the propagation problems that are the principal subject of this thesis. We now write the corresponding weak formulation, which is the starting point of the finite element method.

We will assume that the domains are truncated at finite distance and hence that at least part of their boundary, denoted by Γ_∞, is an artificial boundary where an approximation of the Sommerfeld condition (1.3) is applied. More details on such conditions can be found later in the section. We give the formulations with the simplest examples of such conditions and will refer to the literature for the formulations with more elaborate techniques.

We start by defining the usual function spaces that will be useful for the description of the method. The space $L^2(\Omega)$ is the space of square integrable scalar fields on the bounded domain Ω:

$$L^2(\Omega) = \{u : \int_\Omega u^2(\boldsymbol{x})\, d\Omega < \infty\}. \tag{1.21}$$

The Sobolev space of scalar fields $H^1(\Omega)$ is defined as:

$$H^1(\Omega) = \{u \in L^2(\Omega) : \partial_x u, \partial_y u, \partial_z u \in L^2(\Omega)\}, \tag{1.22}$$

while the space of 0-trace fields $H^1_0(\Omega)$ is:

$$H^1_0(\Omega) = \{u \in H^1(\Omega) : u = 0 \text{ on } \partial\Omega\}. \tag{1.23}$$

We will also use the trace spaces $H^{1/2}(\Sigma)$ and $H^{-1/2}(\Sigma)$ that are respectively the Sobolev spaces of fields on the boundary of Ω and its dual, see [76] for details.

In the case of homogeneous Dirichlet boundary conditions (supposing that the problem is well-posed), the variational formulation of problem (1.2), modified to incorporate a volumic source term $f \in L^2(\Omega)$ is: find u in H^1_0 s.t.

$$-\int_\Omega \quad (\Delta + k^2) u v \quad d\Omega = \int_\Omega \quad f v \quad d\Omega, \qquad \forall v \in H^1_0(\Omega), \tag{1.24}$$

which, after integration by parts of the second order term in the derivative becomes: find u in H^1_0 s.t.

$$-\int_\Omega \quad (-\nabla u \cdot \nabla v + k^2 u v) \quad d\Omega = \int_\Omega \quad f v \quad d\Omega, \qquad \forall v \in H^1_0(\Omega). \tag{1.25}$$

In practice, more complex sets of boundary conditions are encountered: we will make use of a partition of the boundary $\partial\Omega$, and impose homogeneous Dirichlet or Neumann conditions on parts of the boundary $\Gamma_{D,0}$ or $\Gamma_{N,0}$, non-homogeneous Dirichlet condition on Γ_D and, for instance, the Sommerfeld condition on a

truncation boundary Γ_∞. The definition of the function spaces in that case requires more care; we refer to [76] for details. In addition, we will make use of transmission conditions $\partial_n u + \mathcal{S}u = g$ on some artificial boundaries Σ, that are impedance conditions with data $g \in L^2(\Sigma)$ (see Section 2.4.3). The weak formulation then becomes, with $H^1(\Omega)$ the appropriate Sobolev space: find $u \in H^1(\Omega)$ s.t.

$$\begin{aligned} &\int_\Omega \quad (\nabla u \cdot \nabla v - k^2 uv - fv) \quad d\Omega \\ -&\int_\Sigma \quad (-\mathcal{S}uv + gv) \quad d\Sigma \qquad\qquad \forall v \in H^1(\Omega). \\ -&\int_{\Gamma_\infty} \quad (\imath kuv) \quad d\Gamma \quad = \quad 0, \end{aligned} \tag{1.26}$$

The bilinear form $(u,v) = \int_\Omega (\nabla u \cdot \nabla v - k^2 uv)\, d\Omega$ is known to be sign-indefinite for high values of k. This is an important observation, since we will see that it will cause difficulties. (See [134] for an in-depth discussion on its mathematical properties.)

It is common to encounter distributional source terms for the Helmholtz equation. A typical example in 1d is $f = \delta(\boldsymbol{x})$, the delta function that is 0 everywhere except at 0, with an integral of 1: $\int_{-\infty}^{\boldsymbol{x}} \delta(\boldsymbol{s})\, d\boldsymbol{s} = H(\boldsymbol{x})$, with $H(\boldsymbol{x})$ the unit Heaviside step function. It behaves like a point source producing symmetric waves that propagate in both directions. Such source terms must be integrated by parts, which is easily done in weak formulations. For the delta function we have:

$$\int_\Omega \delta v \, d\Omega = -\int_\Omega H \nabla v \, d\Omega + \int_{\partial\Omega} Hv \, d\Gamma.$$

1.3.2 Nodal finite element methods

Starting from the weak formulation (1.26), we look for an approximate solution u_h of the form:

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} u_i \Phi_i(\boldsymbol{x}), \tag{1.27}$$

where the set $\{\Phi_i\}$ forms a basis of the piecewise polynomial functions over Ω and is a finite dimensional subset of H^1, and the set of u_i are the (unknown) nodal values with the interpolation property $u_h(\boldsymbol{x}_i) = u_i$. The basis functions are commonly chosen as the linear hat functions built on the vertices of the mesh (with coordinates $\boldsymbol{x}_{1\le i\le N}$), which gives the P1-FEM:

$$\Phi_i(\boldsymbol{x}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases} \tag{1.28}$$

They are compactly supported since they are only non-zero in the vicinity of a given vertex. Galerkin's method is to restrict the infinite set of test function v in (1.26) to the set of form functions Φ_i, to obtain the finite-dimensional problem: find u_h such that

$$a(u_h, \Phi_i) = b(f, \Phi_i), \quad \forall \Phi_i(\Omega), \tag{1.29}$$

where $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are respectively the bilinear and linear forms corresponding to the left- and right-hand sides of the problem (1.26).

Since we have discretized the problem by means of the definitions (1.27) to (1.29), this can be rewritten in matrix form as:

$$Au = b, \tag{1.30}$$

with the coefficient matrix $A_{ij} = a(\Phi_i, \Phi_j)$ and the right-hand side $b_i = b(f, \Phi_i)$. The solution to that system is called the finite element solution of the problem. Since the Φ_i functions are compactly supported, most coefficients in A are zero and the system is said to be sparse, which is an important feature of the method.

For vector-valued problems, a similar procedure can be followed, but suitable vector basis functions must be used; we refer to [115, 135] for details. As an example, in Section 1.3.1 we give the weak formulation of the vector high-frequency Maxwell's equations, whose FEM solution is represented by means of edge elements that naturally take into account the continuity of the tangential component of the field, and for which the unknowns are the circulation of the field along the edges of the mesh.

The usual workflow for the solution of a problem, and in particular of a propagation problem, by the FEM follows the steps:

1. Discretize the domain in an appropriate way. The mesh parameters (discretization density) must be carefully chosen to represent the solution with sufficient accuracy. If the domain is infinite, one must decide where to place the truncation boundary;

2. Write the weak form of the problem and select interpolation functions. This includes the definition of appropriate boundary conditions and the choice of a truncation technique for the proper modeling of the external boundary;

3. Generate the linear system. This is usually done in a standard way by FEM codes, that take the previously defined formulation and mesh as inputs.

4. Solve the problem. This is the most challenging part, since propagation problems often result in extremely large, sign-indefinite linear systems that are hard to solve by usual methods, as will be seen in Section 1.4.

1.3.3 Absorbing boundary conditions

This section is dedicated to the techniques used to truncate the infinite domain of a wave propagation problem in view of its discretization by a PDE-based method, since these cannot directly handle unbounded domains. Also, the region of prime interest in the domain is often limited to the area enclosing some objects or extending down to a given "depth". For these reasons, one computes only the restriction of the solution to the full problem inside the area of interest, noted Ω^+, which is enclosed in an artificial truncation boundary Γ_∞: $u^+ = u|_{\Omega^+}$, or an approximation of it; note that if one is interested in the far-field, other techniques based on an integral representation of the field (see Section 1.2.1) or a combination with them, might be better candidates.

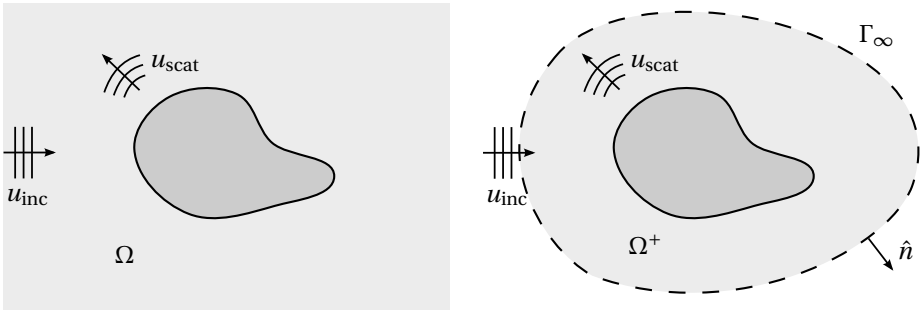


Figure 1.6: Illustration of an open domain Ω (left) and the same domain truncated by an artificial boundary (right), for a scattering problem by an object (darker gray). The computational domain is in lighter gray and extends to infinity in the first case, while being enclosed in an arbitrarily chosen boundary Γ_∞ in the second. Proper boundary conditions must be imposed on Γ_∞ for the restriction in Ω^+ of the solution to the full problem to match the one of the truncated problem with good accuracy.

The boundary conditions to be applied on the truncation boundary Γ_∞ of the truncated domain to simulate a continuing domain have been the subject of much research, see e.g. [81, 100, 101, 117] and references therein. They are often referred to in the literature as absorbing boundary conditions (ABCs), but slightly different names like non-reflecting or transparent conditions are also commonly encountered, though they sometimes refer to slightly different techniques, being approximate [8, 117] or exact [24, 119].

If the computational domain is fully enclosed in a truncation boundary and the source is located outside the truncated domain, which is a frequent situation in scattering problems, it is common to separate the total field into its incident and scattered components: $u_{\text{tot}} = u_{\text{inc}} + u_{\text{scat}}$ (see Figure 1.6). Since the incident field (e.g. a plane wave in free space) is known, one solves for the scattered field only by imposing that it is outgoing through Γ_∞. The source for the

scattered field is imposed via suitable boundary conditions on the surface of the object (e.g. $u_{\text{scat}} = -u_{\text{inc}}$ on sound-soft surfaces).

In other cases the source is located inside the domain and defining a scattered field is irrelevant; the variable of interest is then the total field. Consider for example the modeling of underground wave propagation: in that case the domain is limited to a "cube" of ground (see Figure 1.5); the source can either be located on the surface or underground. In both cases the surface, which is the only physical boundary, is modeled by a Neumann or a Dirichlet condition and Γ_∞ consists of all other boundaries that should let propagating waves escape without (or with little) reflection.

Transparent conditions have recently regained interest in the context of optimized domain decomposition methods, since they are related to the optimal choice for the transmission conditions, as will be explained in Section 2.4.2. Their role in the numerical solution of open propagation problems is thus twofold: besides their original truncation purpose, they are a key ingredient of domain decomposition methods like the optimized Schwarz algorithm (detailed in Section 2.4.2), since they are involved in the transmission of information between the domains. Yet these are two different functions: the ABCs used for domain truncation are modeling tools; the choice of a particular technique for that purpose has an impact on the solution, since the amount of spurious reflected waves will depend on the accuracy of the condition, as well as the location and shape of the truncation boundary. On the other hand, a less accurate transmission condition in a DDM algorithm, provided that it is non-singular, will only slow down convergence but should theoretically not influence the solution after full convergence of the method.

A general expression for absorbing conditions is:

$$(\partial_{\boldsymbol{n}} + \mathcal{S})\, u = 0 \quad \text{on } \Gamma_\infty, \tag{1.31}$$

with $\mathcal{S}$ an operator that should be an approximation of the Dirichlet-to-Neumann (DtN) map, which is a non-local operator, hence difficult to manipulate in a finite element context (see Appendix B.2 for a formal definition in terms of boundary integral operators). Approximations of it by means of local differential operators or simply scalars have been developed in the literature; in the rest of this Section we will present some techniques that are commonly used as absorbing conditions. More advanced techniques that have been developed specifically in the context of domain decomposition methods will be presented later on, in Section 2.4.3.

Basic ABC: the Sommerfeld conditions

A simple approximation of a transparent condition at finite distance directly inspired by the radiation condition (1.3) is given by:

$$(\partial_{\boldsymbol{n}} - \imath k)\, u = 0 \quad \text{on } \Gamma_\infty. \tag{1.32}$$

That approximation is not very accurate when applied on a boundary located too close from the sources or with a strong curvature, and gives rise to non-negligeable spurious reflected waves into the domain. It is only exact for some specific types of waves, e.g. plane waves or spherical waves, under normal incidence.

Higher order ABCs

In an effort to improve the accuracy of the absorbing conditions, more elaborate approximations of the DtN map, involving a second-order surface differential operator have been developed, like in [67] for a straight boundary. Bayliss and Turkel have proposed a condition that accounts for the curvature of a constant radius truncation boundary [15, 17]:

$$(\partial_{\boldsymbol{r}} - (\imath k - \alpha) - \beta \Delta_{\Gamma_\infty})\, u = 0 \quad \text{on } \Gamma_\infty, \tag{1.33}$$

where ∂_r and Δ_{Γ_∞} are rspectively the outward pointing normal derivative and the Laplace-Beltrami operator on Γ_∞, and the coefficients α and β depend on the radius of the truncation boundary. A more general version was proposed in [7], where the local curvature of the boundary is accounted for in the case of a non-constant curvature surface. These conditions can easily be implemented in a finite element code. Many other high-order conditions have been proposed. For a review and a comparison with other techniques such as absorbing layers, see [101, 102].

1.3.4 Perfectly matched layers

The concept of the perfectly matched layer (PML) was originally introduced by Bérenger [22] as an alternative way to truncate the computational domain and guarantee that all waves are outgoing. Its working principle is quite different than the ABCs presented in the previous sections, since it does not rely on boundary conditions imposed on the boundary of the domain.

Instead, the principle is here to modify the properties of the propagation medium in the peripheral region Ω_{PML} of the computational domain Ω_+ (Figure 1.7), in such a way that the waves are damped in that layer. If the attenuation is sufficient, the amplitude of any spurious wave leaving the PML and reentering the domain would be so small that it would lead to a negligible error. This is illustrated on Figure 1.8, where the attenuation coefficient and PML thickness have been purposely set to non-standard values for the sake of clarity; practical PMLs

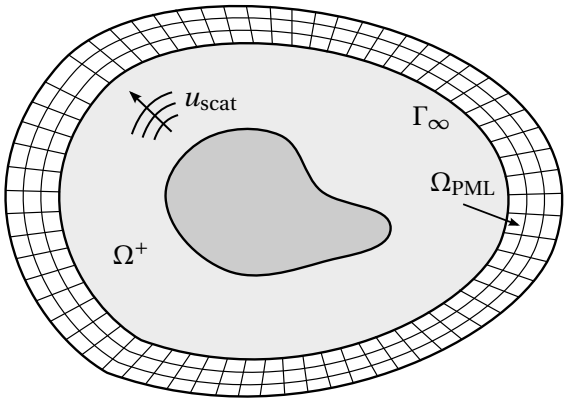


Figure 1.7: Illustration of a perfectly matched layer. The outgoing scattered wave is (almost) entirely absorbed in the layer Ω_{PML} surrounding the computational domain Ω^+ and cannot reenter the domain.

are typically just a few element layers thick, and the amplitude decay in Ω_{PML} is usually exponential.

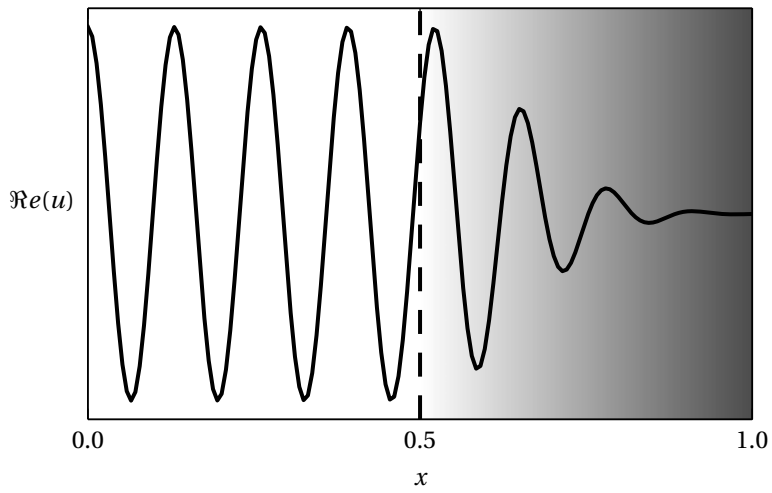


Figure 1.8: Illustration of the attenuation of a wave in a perfectly matched layer, starting at $x_{\text{PML}} = 0.5$. The gradient represents the growth of the absorption coefficient.

This is done by introducing an imaginary part to the wavenumber to simulate a lossy material; inside the PML, the absorption coefficient thus also grows continuously following some design function σ:

$$k \to k\Big(1 + \imath \frac{\sigma(x)}{k}\Big) \quad \text{in } \Omega_{\text{PML}}. \tag{1.34}$$

It is common to choose an absorption function that follows a polynomial law of order p:

$$\sigma(x) = \left(\frac{|x - x_{\text{PML}}|}{d_{\text{PML}}} \right)^p \sigma_0, \tag{1.35}$$

where x_{PML} and d_{PML} are respectively the coordinate of the PML interface and the PML thickness. Higher order p lead to smoother but slower growth of the coefficient. Other types of laws for the growth are possible: it was shown in [23] that functions with an infinite integral in the PML provide the best accuracy:

$$\lim_{t \to d_{\text{PML}}} \int_0^t \sigma(s)\, ds = \infty. \tag{1.36}$$

Of course, adding a PML at the boundary of a domain leads to extra computational overhead since it results in an increased number of unknowns, especially in 3d. The thickness of the PML should thus be kept small to maintain that increase into reasonable limits, while still being efficient. The design of PMLs is an active field of research and abundant literature on the topic is available, see for example [23, 24, 102] and references therein. We conclude this presentation by mentioning that the PML can either be terminated by an homogeneous Dirichlet condition, or alternatively by one of the classical ABCs introduced above to let any undamped wave leave it via its external boundary to improve its accuracy [147].

A comparison of the two approaches presented above (high-order absorbing conditions and PML with various absorption functions) can be found in [152].

1.3.5 Difficulties with FEM at high frequency

Solving the linear systems arising from the discretization with a volumic method of the Helmholtz equation at high frequency is a notoriously difficult problem. The authors of [69] mention two main reasons. First, the size of the linear systems: since the domain must be discretized with a mesh density n_λ sufficient to represent the oscillations of the solution and control the pollution effect (see later in the section), the number of mesh points in each direction is proportional to the wavenumber k. The total size of the system in dimension d is $N \times N = \mathcal{O}(k^d) \times \mathcal{O}(k^d)$, which leads to extremely large systems for large values of k.

Second, the sign-indefiniteness of the matrix: its eigenvalues are complex due to the absorbing condition used to truncate the domain, and located on both positive and negative sides of the real axis, as shown on Figure 1.9. Also, it has a very oscillatory and slowly decaying Green's function, which causes problems with otherwise successful solution strategies for large elliptic problems, as will be discussed in Section 1.4.

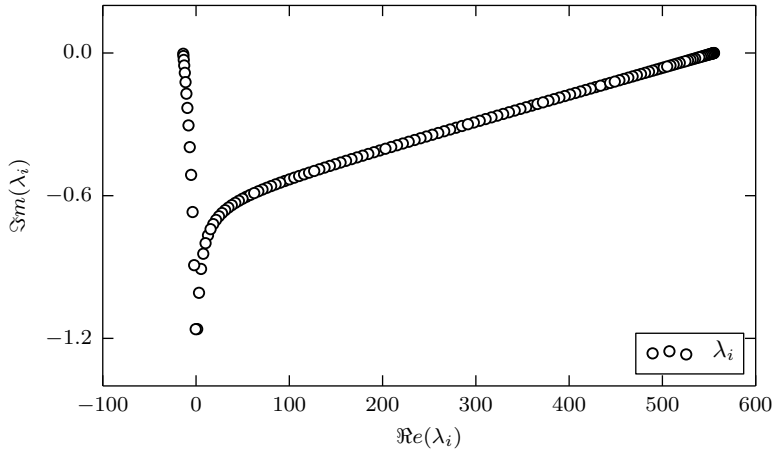


Figure 1.9: Eigenvalues repartition of the FEM matrix of a 1d Helmholtz problem ($\Omega = [0,1]$, $k = 14\pi$, $n_\lambda = 20$, Sommerfeld ABC on both ends). They are complex, with real parts on both positive and negative sides of the real axis.

Another important difficulty with time-harmonic propagation problems solved by PDE-based methods is numerical dispersion[2], commonly known in the abundant literature on the topic as the pollution effect. These names refer to the fact that the numerical solutions require extra-refinement of the discretizations to attain a given accuracy, the problem being that the computed solution differs from the best approximation that can be found on the chosen set of degrees of freedom, as illustrated on Figure 1.10. Moreover, the ratio between the error of the FEM and the error of the best approximation tends to infinity for growing wavenumber [11], unless the discretization step h is decreased inverse-proportionally to k^2 [9], which would quickly lead to impractical discrete system sizes.

In practice, while just a few points (≥ 2) per wavelength are theoretically sufficient to represent an oscillating signal, we observe that the solutions of propagation problems are very inaccurate on such coarse meshes (i.e. they exhibit substantial phase errors (Figure 1.10), and that much more points must be used to satisfy the required accuracy. The pollution effect is intrinsic to the numerical solution of propagation problems and it was proven in [10] that it is hopeless to fully remedy it in dimensions higher than 1. However, some techniques have been developed that enable significant reduction of the phenomenon: let us mention the use of higher-order hp-FEM [112], the definition of generalized FEM [11], or the modification of the FE space to include more suitable basis func-

[2]The term "dispersion" refers to the physical phenomenon that waves of different wavelengths can propagate at different speeds in the same medium, thereby causing distortion of polychromatic signals. A classic example is the refraction of white light by a prism. By analogy, the same term is used to describe a purely numerical artefact.

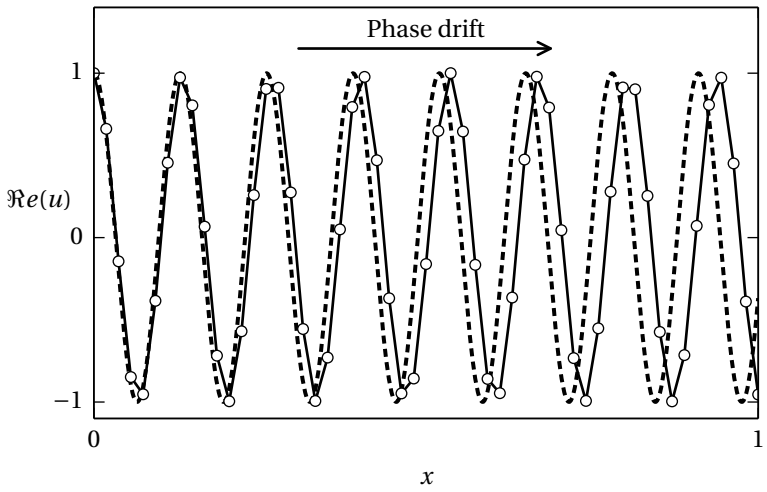


Figure 1.10: Illustration of the numerical dispersion on a 1d problem: the finite element solution u_h (solid), with a discretization density $n_\lambda = 7$, does not match the analytical solution (dashed). The phase error grows with the distance to the source (here at $x = 0$), resulting in a wrong wavenumber in the numerical solution. The phenomenon is particularly visible on coarse meshes like the one used in this example.

tions [38, 109, 111, 146, 174]. Of particular notice is the PR-FEM formulation, described in Chapter 4.

In Appendix D, we derive the dispersion relations in the 1d case for both the FEM and FDM schemes. They are shown on Figure 1.11 in function of the mesh density $n_\lambda = \lambda/h$, for some discretization step h. One can see that low discretization densities produce severe pollution, leading to the rule-of-thumb that at least $10-20$ points per wavelength should be used to guarantee sufficient accuracy. The same rule holds for higher dimensions. A direct consequence of the pollution effect is that the number of unknowns in propagation problems grows extremely fast as one increases the frequency of the problem and the size of the domain is kept constant: to guarantee the accuracy of the solutions, the mesh parameter must evolve at least as $h \sim k^{-1}$, while some authors use $h \sim k^{-3/2}$. This rapidly leads to huge linear systems, the solution of which by usual methods (e.g. LU factorization) requires considerable resources (CPU time and memory).

To give an example of how detrimental the pollution effect can be when designing numerical methods for propagation problems, we will see in Chapter 4 that the pollution effect is the main cause of failure of a solver for multiple scattering problems: when building a subspace of solutions of a multiple scattering problem, discretized by a standard FEM, with solutions of single scattering problems obtained by a method (the PR-FEM) that suffers less pollution than the FEM, we find that our subspace does not contain an accurate approximation of the full

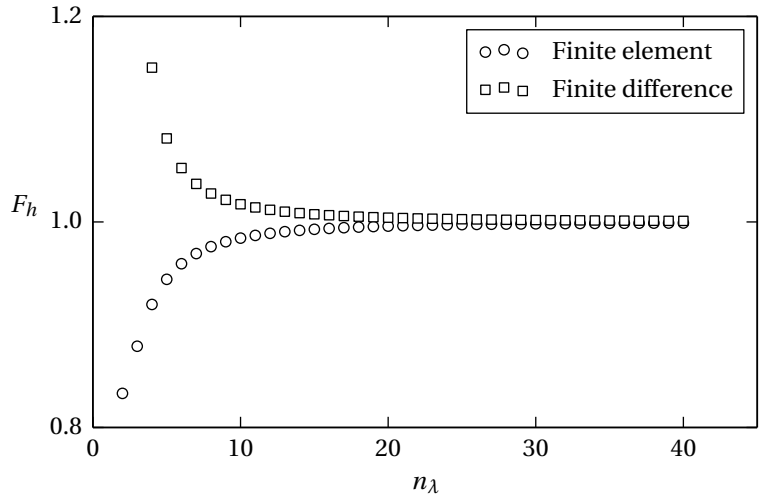


Figure 1.11: Dispersion relations of the FE and FD methods, in function of the discretization density n_λ. The dispersion factor F_h converges to 1 for fine discretizations, but significantly deviates for densities $n_\lambda < 10-15$. Interestingly, the FEM produces a solution with underestimated wavenumber, while the FDM has the opposite behaviour.

problem, because the dispersion relations of the two methods do not match. In other words, our (almost) pollution-free subspace is not included in the space of the standard FEM solutions for a given wavenumber, which stronly limits the accuracy of the method.

Another consequence is the degradation of the accuracy of the artificial boundary conditions (ABCs) used to truncate computational domains that would otherwise extend to infinity (see Section 1.3.3). In our simple 1d example, the Sommerfeld condition (1.32) is theoretically exact, but the computed solution contains a spurious reflected wave in the numerical solution (too small to be observable on the figure). This is because the actual wavenumber of the numerical solution deviates from the one in the ABC; if one replaces k by k_h in the Sommerfeld condition, the reflected wave vanishes. This phenomenon will have practical consequences for the methods developed in this work, since such ABCs (in higher dimensions) are a central component of our algorithms.

1.4 Sparse linear solvers for propagation problems

We have seen that the volumic discretization of the propagation problems leads to linear systems, that, especially when the wavelength becomes small compared to the size of the computational domain, can be extremely large: hundreds of millions of unknowns are not uncommon for scattering problems. The question of their solution therefore becomes prominent. The goal of this section is to show

that usual techniques are unusable, either as a consequence of the size of the system or because of the mathematical properties of the system to be solved.

The most classical strategy is to use a direct solver that performs gaussian elimination to compute a factorization of the matrix [103]. Alternatively, iterative solvers are well adapted for large sparse linear systems, especially when used in conjunction with a carefully designed preconditioner [106, 159]. However, it has become clear that when applied to high-frequency propagation problems, the usual strategies fail [77].

The goal of the present work is to study, implement and improve an alternative approach, that actually combines ingredients of both families of solvers, to bring an answer to that problem. The basic idea is to decompose the domain into other ones, that can be either smaller and for which direct solvers are applicable, or larger but for which fast solvers are available. One can even find a similarity with integral methods since the unknowns will be defined on (actual or artificial) surfaces, which reduces the dimensionality of the problem.

Then, iterations are designed to let the process converge to the solution of the full problem. That idea will be briefly exposed as the conclusion of this Chapter, and further explored as the main topic of this thesis in Chapters 2, 3 and 4.

1.4.1 Direct solvers

The most common way of solving linear problems is to perform some factorization of the system matrix, in a gaussian elimination-like process. Numerically stable algorithms are available for a factorization as $A = LU$ [2, 103, 177], where L and U are lower and upper triangular matrices, respectively (an additional permutation matrix is often required to avoid division-by-0 problems in the elimination process). Then, the problem can easily be solved in a 2-steps process, by solving two triangular systems, which is easily done.

There has been significant progress in the factorization algorithms, with dedicated versions that exploit specific matrix properties (symmetry, positive-definiteness, banded structure, ...). State-of-the-art solvers feature parallel multifrontal algorithms [37] that are well adapted for the solution of FEM-type sparse linear systems on distributed memory architectures [143]. However, their complexity is still typically $\mathcal{O}(N^3)$ (yet some authors report $\mathcal{O}(N^2)$ with multifrontal algorithms [68] in 2d), and the forward and backward substitutions are $\mathcal{O}(N^2)$, which makes the solution of very large systems still extremely resource intensive.

1.4.2 Usual iterative methods and preconditioners

We have seen in the previous section that the size of the linear systems:

$$Au = b \tag{1.37}$$

obtained by discretizing the volumic operators at high frequency with sufficient accuracy makes them hardly amenable to direct solvers, as these require considerable resources. The alternative is to resort to iterative methods. These include the family of Krylov linear solvers, but other iterative schemes, like multigrid and domain decomposition, have been developed more specifically to solve PDE problems. These last methods have been designed as standalone solvers, but can also be seen as preconditioners for (1.37) and can thus be used in conjunction with Krylov solvers [106]. In this section, we present some classical methods that have proved succesful for other problems and explain why they fail when applied to propagation problems. It is partly inspired by the review paper by Ernst and Gander [77].

Unpreconditioned Krylov methods

Krylov methods are a powerful family of linear solvers that are particularly well adapted for the solution of the large sparse linear systems obtained by discretization of elliptic differential operators. We will give more details on these methods in Section 2.2.2, since they play a central role in the framework that we propose for the solution of large propagation problems. For now, we focus on their application as solver for the FEM discretized operators.

It is commonly observed that the convergence of such solvers is slow when directly applied to propagation problems. This is related to the slow decay of the associated Green's functions, in other words the contribution of sources remain significant at very long distance from their location. Consequently, since the solvers propagate information inside the domain in a layer-by-layer fashion starting from the sources, the number of iterations tends to depend on the maximum distance between any source in the domain and the external boundary. Moreover, absorbing conditions such as (1.32) are frequent in propagation problems, and make the coefficient matrix of system (1.37) complex-valued and symmetric, but non-hermitian. This last property prevents the use of memory efficient algorithms like the Conjugate Gradient.

It is clear with these observations that the Krylov methods alone are not satisfactory and that additional techniques are required to make them usable. Preconditioning is a natural way of speeding up the convergence of linear solvers, and many of such techniques have been developed, some of which having proved extremely efficient. In the following paragraphs we examine some of them and try to explain why they do not perform well or fail with propagation problems.

Algebraic preconditioners

The most simple algebraic preconditioners are based on a splitting of the matrix (see Section 2.2.3). It is well known that the damped Jacobi and Gauss-Seidel preconditioners exhibit slow convergence of the lowest spatial frequencies or are even

divergent when applied to the Helmholtz problem, due to the spectral radius of the preconditioned operator being greater than 1. Still, they perform quite well on the high frequencies and are thus frequently used as smoother in the multigrid methods (see below). A modified 2-step version of the Jacobi method has been reported to converge for all modes [107], but at a rate dependent on the grid size and wavelength.

Another class of algebraic preconditioners is based on incomplete decompositions, that are less expensive to compute than a full factorization, but can still be considered as an approximate solver. The idea is to compute an approximate factorization $A \approx LU$, known as ILU, and to apply the forward and backward substitutions with these approximate factors as preconditioner. The generic versions ILU(0) [132] and ILU(tol) [159] are found to perform well on low-frequency Helmholtz, but become inefficient for higher frequencies [77].

More specific versions for the Helmholtz problem have been developed, as the analytic ILU (AILU) [92], which starts from the observation that, upon proper numbering of the unknowns (the lexicographic arrangement), the forward substitution can be interpreted as a time-stepping in one direction, whereas the backward substitution does the same in the reversed direction. We will see that this interpretation of the LU decomposition makes a clear link with the sweeping methods proposed in this work. Making the link with an analytic factorization of the Helmholtz operator similar to that of the paraxial approximation (1.16), that involves non-local operators that are localized in order to obtain sparse approximations, one obtains a new approximate factorization of the discrete operator. It has much better performance than the generic ILU methods, since its design better takes into account the nature of the problem. However, the iteration number still grows with the wavenumber; another limitation is that it is not applicable on general, unstructured, 3d meshes.

Following the same kind of idea, the sweeping preconditioner of Engquist and Ying [68, 69] can be seen as an approximate LDL^T decomposition, in which low rank approximations are used to perform the substitutions efficiently, leading to a method with nearly linear complexity.

Multigrid

A very efficient iterative method for elliptic problems such as the Poisson equation ($\Delta u = f$) is multigrid. It works by alternating the (approximate) solution of problems at different levels of mesh coarsening. Finer grids play the role of smoother, by doing a step of a simple and inexpensive iterative method such as damped Jacobi or Gauss-Seidel, that has the effect of removing the high-frequency component of the error [165, 168]. Conversely and quite intuitively, the coarser grids are

efficient at correcting low frequency (smooth) modes [86], leading to extremely fast convergence, even without Krylov acceleration (though this component is always beneficial). Naturally, medium-frequencies can be better treated if an intermediate level is introduced, resulting in a 3-level multigrid; the same pattern can be followed with even more levels.

Popular multigrid methods follow the symmetric "V" cycle: a call to each level consists in a presmoothing step, followed by a coarse grid correction (that itself repeats this pattern), and completes with a postsmoothing step. A variant is the "W" cycle, where the recursive call to the next level is repeated twice [106, 165]. Many combinations of smoothing steps and coarse corrections are possible, as well as the application of several successive smoothing steps [165]. One can also use more complex smoothers, like a step of a Schwarz method (see Section 2.4).

It is clear that the structure of the method is potentially the source of a lack of robustness: if one of the components fails, the corresponding modes of the error will not be removed, preventing convergence of the method. When applied to indefinite problems such as Helmholtz at high frequency, the ideal behaviour described above is seriously affected, since both the smoother and the coarse grid do not behave as expected: the smoother is quite inefficient since high frequency error modes have small residual and are thus "invisible" on the fine grids [33], while not being resolved on the coarse ones. The smoother can even amplify low-frequency modes associated to negative eigenvalues.

Also, the coarsest level must still be fine enough to properly represent smooth eigenfrequencies of the operator, and the method loses its $\mathcal{O}(N \log N)$ performance [77]. Other problems in relation with eigenvalues close to 0 that change sign when represented on coarser grids have been highlighted [34]. They cause the method to become divergent since the associated coarse grid correction actually has wrong sign.

Using the method as a preconditioner for a Krylov method turns it into a convergent method, but one observes a quick degradation of the performance with increasing wavenumber [77].

Note however that specific multigrid methods or preconditioners have been developed for the Helmoltz equation [72, 79, 126, 170]. Underlying these developments in particular is the use of schemes that will produce the same dispersion error at the different scales.

Shifted Laplace preconditioner

We have seen in the previous sections that the long range interactions in the Helmholtz operator are causing difficulties for its iterative solution. When damping is introduced, the decay of the Green's function is faster, and the problems are more easily solved by iterative methods. Following that observation, the idea has arised to precondition non-dissipative problems by (an approximate inverse of) a modified version with damping [71]. This is done by adding an imaginary part to the wavenumber of the problem: $k \rightarrow k_\varepsilon = k + \imath\varepsilon$.

The preconditioner can then be approximately inverted, e.g. by a multigrid method [72, 141], or an ILU decomposition [142]. Other solvers, e.g. a domain decomposition method, can be considered as well. Of course, the problems with large shifts are easier to solve, but are also more distant from the original problem, resulting in a less efficient preconditioning. Finding the best compromise for the amplitude of the shift has been widely discussed in the literature [73, 94, 178].

1.4.3 Domain decomposition methods

The domain decomposition method as initially introduced by Schwarz (Section 2.4.1) is another example of a method that works well on problems with smooth solutions and that fails when applied to the indefinite Helmholz equation. Again the spectral properties of the operator are at the origin of the problem: a modal analysis reveals that the method converges well for the evanescent modes only, while the propagative ones have a convergence factor equal to one [77, 90]. Therefore, increasing the overlap does not help.

However, a modified version of the method, that makes use of different transmission conditions (see Section 2.4.2), can be shown to converge for all modes. With this modification, Schwarz methods have become amongst the most promising methods for the solution of HF propagation problems. One of its main assets is that it is suitable for use on large parallel computers, and that it naturally couples with efficient direct solvers on subdomains with manageable sizes. This approach will be the starting point of one of the methods presented in this thesis, in Chapter 3.

1.5 Extension to Maxwell's equations

The framework of the next Chapter makes use of the concept of iteration operators, that are defined in terms of transmission operators between boundaries. At this level of representation, it is not necessary to specify the nature of these operators, that can either be scalar or vector valued. Therefore, the methods that are derived for the scalar Helmholtz equation can readily be extended to the vector

case. This Section is a brief presentation of the problem that is solved in that case, and of its discretization.

1.5.1 Time-harmonic Maxwell's equations

We now derive the vector counterpart of the Helmhotz equation, starting from Maxwell's equations and making the time-harmonic assumption. In this presentation, we will use the convention $\boldsymbol{\mathcal{E}} = \mathcal{R}e[\boldsymbol{E}\, e^{-\imath\omega t}]$ for the representation of fields in the harmonic regime. A plane wave propagating in the $\boldsymbol{x}$ direction therefore has expression $\boldsymbol{\mathcal{E}} = \mathcal{R}e[\boldsymbol{E}_0\, e^{\imath(\boldsymbol{k}\boldsymbol{x}-\omega t)}]$. Maxwell's equations in free space (that is, in the absence of charges and currents) write:

$$\begin{aligned} \operatorname{div} \boldsymbol{\mathcal{D}} &= 0; && (1.38)\\ \mathbf{curl}\, \boldsymbol{\mathcal{E}} &= -\partial_t \boldsymbol{\mathcal{B}}; && (1.39)\\ \operatorname{div} \boldsymbol{\mathcal{B}} &= 0; && (1.40)\\ \mathbf{curl}\, \boldsymbol{\mathcal{H}} &= \partial_t \boldsymbol{\mathcal{D}}, && (1.41) \end{aligned}$$

with the constitutive relations:

$$\begin{aligned} \boldsymbol{\mathcal{D}} &= \epsilon_0 \boldsymbol{\mathcal{E}}; && (1.42)\\ \boldsymbol{\mathcal{B}} &= \mu_0 \boldsymbol{\mathcal{H}}. && (1.43) \end{aligned}$$

Injecting our definition in equations (1.39) and (1.41), we get:

$$\begin{aligned} \mathbf{curl}\, \boldsymbol{E} &= \imath\omega\mu_0 \boldsymbol{H}; && (1.44)\\ \mathbf{curl}\, \boldsymbol{H} &= -\imath\omega\epsilon_0 \boldsymbol{E}, && (1.45) \end{aligned}$$

and after elimination of $\boldsymbol{H}$, we obtain the expression of the spatial part of the electric field:

$$\mathbf{curl}\, \mathbf{curl}\, \boldsymbol{E} - k^2 \boldsymbol{E} = 0, \tag{1.46}$$

where k is the wavenumber $k = \omega/c$ and $c = 1/\sqrt{\epsilon_0\mu_0}$ is the speed of light in vacuum. A similar expression can be written for the magnetic field $\boldsymbol{H}$.

Continuity of the tangential component of the electric field on interfaces must be ensured, leading to the condition that $\boldsymbol{n} \times \boldsymbol{E} = 0$ on conducting parts of the domain; we again define $\boldsymbol{n}$ as the outward pointing normal. As in the case of Helmholtz equations, a radiation condition is necessary to ensure unicity of the solution of unbounded domain problems. The equivalent to the Sommerfeld condition (1.32) in the vector case is known as the Silver-Müller condition:

$$\lim_{|\boldsymbol{r}|\to\infty} \boldsymbol{r} \times \mathbf{curl}\, \boldsymbol{E} - \imath k\, \boldsymbol{r} \times \boldsymbol{r} \times \boldsymbol{E} = 0, \tag{1.47}$$

for any radial direction $\boldsymbol{r}$.

Following the same idea as for the Sommerfeld absorbing condition, the radiation condition for Maxwell's equation (1.47) can be adapted to be used as a simple ABC at finite distance. It yields for the electric field:

$$\boldsymbol{n} \times \mathbf{curl}\, \boldsymbol{E} = \imath k\, \boldsymbol{n} \times \boldsymbol{n} \times \boldsymbol{E} \quad \text{on } \Gamma_\infty. \tag{1.48}$$

1.5.2 Weak formulation

To obtain the weak formulation of problem (1.46), we will need the vector Green's formula. Starting from the vector identity:

$$\text{div}\,(\boldsymbol{w} \times \boldsymbol{v}) = \mathbf{curl}\, \boldsymbol{w} \cdot \boldsymbol{v} - \boldsymbol{w} \cdot \text{div}\, \boldsymbol{v}$$

and substituting $\boldsymbol{e}' = \boldsymbol{v}$ and $\mathbf{curl}\, \boldsymbol{e} = \boldsymbol{w}$, we obtain:

$$\mathbf{curl}\, \mathbf{curl}\, \boldsymbol{e} \cdot \boldsymbol{e}' = \mathbf{curl}\, \boldsymbol{e} \cdot \mathbf{curl}\, \boldsymbol{e}' + \text{div}\,(\mathbf{curl}\, \boldsymbol{e} \times \boldsymbol{e}').$$

Then, after integration over the domain Ω and application of the divergence theorem (also using the identity $\boldsymbol{n} \cdot (\boldsymbol{a} \times \boldsymbol{b}) = (\boldsymbol{n} \times \boldsymbol{a}) \cdot \boldsymbol{b}$), Green's formula for integration by parts writes:

$$\int_\Omega \mathbf{curl}\, \mathbf{curl}\, \boldsymbol{e} \cdot \boldsymbol{e}' = \int_\Omega \mathbf{curl}\, \boldsymbol{e} \cdot \mathbf{curl}\, \boldsymbol{e}' + \int_{\partial\Omega} (\boldsymbol{n} \times \mathbf{curl}\, \boldsymbol{e}) \cdot \boldsymbol{e}'. \tag{1.49}$$

Defining the space of square integrable vector fields $\boldsymbol{L}^2(\Omega) = \{\boldsymbol{u} : \int_\Omega ||\boldsymbol{u}||^2 < \infty\}$, we introduce the space $H^{\text{curl}}(\Omega) = \{\boldsymbol{u} \in \boldsymbol{L}^2 : \mathbf{curl}\, \boldsymbol{u} \in \boldsymbol{L}^2\}$. Putting aside the imposition of boundary conditions (see [135] for details), the weak formulation of the problem defined by equation (1.46) is obtained by multiplying it by test functions $\boldsymbol{E}' \in H^{\text{curl}}(\Omega)$ and integrating over domain Ω:

$$\int_\Omega \mathbf{curl}\, \mathbf{curl}\, \boldsymbol{E} \cdot \boldsymbol{E}' - k^2 \boldsymbol{E} \cdot \boldsymbol{E}'\, d\Omega = 0.$$

After integration by parts of the first term with formula (1.49), we obtain:

$$\int_\Omega (\mathbf{curl}\, \boldsymbol{E} \cdot \mathbf{curl}\, \boldsymbol{E}' - k^2 \boldsymbol{E} \cdot \boldsymbol{E}')\, d\Omega + \int_{\partial\Omega} (\boldsymbol{n} \times \mathbf{curl}\, \boldsymbol{E}) \cdot \boldsymbol{E}'\, d\Gamma = 0.$$

Supposing that we have truncated the domain by an external boundary Γ_∞, we incorporate the Silver-Müller absorbing boundary condition (1.47) to obtain the complete version of the weak formulation: find $\boldsymbol{E}$ in $H^{\text{curl}}(\Omega)$ s.t.

$$\begin{array}{rcl} \displaystyle\int_\Omega & (\mathbf{curl}\, \boldsymbol{E} \cdot \mathbf{curl}\, \boldsymbol{E}' - k^2 \boldsymbol{E} \cdot \boldsymbol{E}') & \mathrm{d}\Omega \\ \displaystyle+\int_{\Gamma_\infty} & \imath k (\boldsymbol{n} \times \boldsymbol{n} \times \boldsymbol{E}) \cdot \boldsymbol{E}' \quad d\Gamma & = \quad 0, \end{array} \qquad \forall \boldsymbol{E}' \in H^{\text{curl}}(\Omega). \tag{1.50}$$

Other types of boundary conditions are of course possible and will be used; we refer to [135] for the appropriate definitions of the vector function spaces.

The FEM discretization of the problem is achieved by using Galerkin's procedure with another class of elements, suitable for the discretization of three-dimensional vector fields [26, 115]. For these, the degrees of freedom are not nodal values of the field as in the scalar case, but rather field-related quantities like the circulation along the edges of the elements; of course, such a discretization also suffers the pollution effect and similar discretization density constraints as in the scalar case apply in the vector case.

For the truncation of the domain, an alternative to the Silver-Müller condition is to use a vector version of the PML introduced in Section 1.3.4 for the scalar problem. Higher-order absorbing conditions have also been derived as in the scalar case. A description can be found in [40, 161].

CHAPTER 2

Multi-domain methods and linear systems

Given a computational domain $\Omega \subset \mathbb{R}^d$, we define a set of N other domains Ω_i ($\Omega_i \neq \Omega, \forall i$) such that $\Omega \subseteq \cup\Omega_{1\leq i\leq N}$. Supposing that we can solve our problem on all of the domains Ω_i, e.g. by means of a direct solver or some efficient algorithm, a multi-domain method computes the solution to the problem in the original domain Ω by using only solutions of the subproblems and combining them in an appropriate way.

Obviously, solving the subproblems only once is usually not sufficient, because the information about the structure of the full problem is missing in the subproblems. If we want the solution u_i in every Ω_i to match the restriction of the full solution $u|_{\Omega_i}$ in them once combined, we need to exchange information between the subdomains, and update the current estimation with the additional knowledge about the other domains so obtained.

Repeating that procedure by, at each step, solving the subproblems with new data and exchanging information between them, in a way that it progressively converges towards the desired solution, is the basic idea of the family of multi-domain methods. One can classify them in function of the particular choice of the new domains. Two families are considered: first, conformal decompositions, where the domain is cut into several subdomains, that may or may not overlap. In these, equality holds between the union of the subdomains and the original domain: $\cup\Omega_i = \Omega$; second, non-conformal coverings of domains that contain several impenetrable objects, the interior of which thus being excluded of the domain. The new domains are obtained by "filling the holes" and are actually larger than the original one: $\cup\Omega_i \supset \Omega$.

Note that other decompositions can be chosen, e.g. to handle parametric calculation with moving object [160], model approximation[49, 62, 63], or material changes [61].

The chapter is structured as follows: we will first introduce the general framework of the different methods that share the characteristics of the idea exposed above. Since they can clearly be seen as the engine of the (linear) multi-domain methods, we will then give some important concepts on iterative linear solvers, while the design of suitable preconditioners represents a great potential for improvement for methods relying on linear solvers. We will then present some examples of such methods; for each of them, we detail the mechanism of the transfer of information between the different domains.

2.1 Common framework of multi-domain methods

In practice, since we are looking for equivalent solutions (or combinations of them) inside the domains Ω_i, we will need to find equivalent sources that produce the solutions. Depending on the method, they will be of different natures, but will have in common that the corresponding data are fields defined on a set of interfaces. Apart from volumic sources, not considered here (examples can be found in [62, 63, 160]), the main kinds of sources are basically introduced by means of Dirichlet and Neumann conditions, or some combination of them, possibly involving operators, imposed on part of the boundary of the new problems.

These fields are of course a priori unknown, and therefore become the new unknowns of the modified problem, that is equivalent to the original one. The general framework that we now present tackles the problem under that alternative formulation.

2.1.1 General iterative scheme

We are thus left with a new "interface" problem, of dimension $d-1$, that needs to be solved iteratively by exchanging information between the domains, via the solution of subproblems and an update of the current solution. This defines the general framework of all the multi-domain methods that will be presented in this work. Their workflow is illustrated on Figure 2.1.

As will become clear in the light of Section 2.1.2, it starts with the computation of a right-hand side that contains the contributions of the physical sources of the original problem. Then several steps of the method are performed until some convergence criterion is met, after which the interface problem is considered as solved and the solution of the original problem can be generated by combining the solutions of the subproblems. In the following, we will speak of an application of the iteration operator whenever a step of the method is performed.

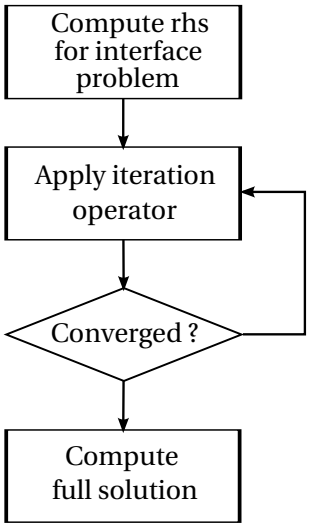


Figure 2.1: All multi-domain methods share a common workflow, based on an iteration operator that involves the solution of subproblems and the sharing of information between disconnected interfaces.

That workflow is simple and, even though the corresponding interface problems are well-posed (their solutions provably exist and are unique if the original problem is itself well-posed), the basic underlying methods may not provide a sufficiently robust solver when used as is: they can converge too slowly, stagnate or even diverge, especially for propagation problems when one is close to a resonant frequency of the problem. However, we will see in the next sections that these methods can be recast as linear systems for which more efficient and robust solvers are available. While it does not fundamentally modify the workflow, the introduction of this new ingredient completes the definition of our framework.

2.1.2 Iteration operators and linear systems

We now more formally transcribe the procedure described above. Let us provisionally define the set of the M (not necessarily equal to N) unknown interface fields as a vector noted $g = [g_1, \ldots, g_M]^T$. Supposing that we have performed k steps of the method, starting with an initial guess $g^{(0)}$, and a non-zero source term b, we denote by $g^{(k)}$ the current approximate solution to the interface problem. The next step of the method will generate a new approximation by solving the subproblems with the current data as sources, collecting and exchanging information between the interfaces. This defines the iteration operator $\mathcal{A}$, and we have the fixed-point iteration:

$$g^{(k+1)} = \mathcal{A} g^{(k)} + b. \tag{2.1}$$

This is a general relation where operator $\mathcal{A}$ contains the definition of the method, and the source vector b is also obtained in a way that depends on each particular method, as will be detailed in Sections 2.4.2 and 2.5.2. If iteration (2.1) is convergent, it defines a series and we can define the solution g as $\lim_{k\to\infty} g^{(k)}$, that verifies:

$$\mathcal{F}g = (\mathcal{I} - \mathcal{A})g = b, \tag{2.2}$$

which is a linear equation; we will see that it can fall into the category of matrix-free problems introduced in Section 2.2.1. That section is dedicated to a simple method for the solution of problems such as (2.2). One can easily verify that iteration (2.3) solves problem (2.2) if the action of matrix A is replaced by that of operator $\mathcal{F} = \mathcal{I} - \mathcal{A}$. Therefore, using a multi-domain method is equivalent to the solution of an unpreconditioned linear system with a simple iteration, and a slight modification of the iteration operator.

This observation is important since it implies that the multi-domain methods are amenable to the theory of linear algebra, and will benefit from its strong theoretical foundation. In particular, Krylov acceleration techniques (see Section 2.2) will guarantee that the methods will converge (though not necessarily fast), as opposed to the fixed-point or Jacobi method (2.1) that can diverge. Improving the convergence properties of the methods is one of the main objectives of the present work, which will be achieved by the technique of preconditioning, later introduced in Section 2.2.3.

At this stage, it is also useful to notice that the decomposition into subdomains, and more specifically the definition of unknown functions on disjoint interfaces, provides a first discretization of the problem, although the definitions of the methods is still made at the continuous level. Therefore, the iteration operators $\mathcal{A}$ or $\mathcal{F}$ can be expressed as finite matrices, the elements of which are the (continuous) transfer operators that describe the transfer of information from an interface to another, and that act on the elements of the vector g defined at the beginning of this section.

These matrices will have different structures in function of the considered method and reflect the connections between the domains, which is in turn a consequence of the kind of partitioning inherent to each method. With the covering of the method of Section 2.5, the matrix will be dense because each domain needs to exchange information with all others. The layered or cyclic decompositions used for the methods of Section 2.4.2 will, in contrast, give raise to sparse matrices. In Chapter 3, we will see how this property can be exploited to compute an approximate inverse of the iteration operator, that will be usable as a preconditioner to speed up the convergence of the method.

2.2 Introduction to iterative linear solvers

This Section briefly presents the concepts of iterative methods for the solution of linear systems and discuss their robustness and convergence. Throughout this work, such systems are encountered either as resulting from the discretization of differential operators, or as the interface problems that underly the multi-domain methods. The nature of the systems in these two cases is very different and even the motivation for using iterative solvers to solve them is different: in the first case they are the only possible way of solving the problems when they become so large that direct solvers are no longer an option, by lack of memory or excessive computation time. In the second case the systems are usually much smaller because they are defined on interfaces rather than in volume. However, they involve an iteration operator, that consists of a routine, while the associated matrix is usually not available, making direct solvers irrelevant for their solution. The important concept of preconditioner is also considered, as it will be useful for both of the aforementioned situations.

First, we present a basic but important algorithm, as we will see that it actually underlies the methods presented in this work, before introducing the more advanced and powerful family of Krylov subspace solvers, thanks to which the methods become usable for practical computations.

2.2.1 Basic iterative scheme

Supposing that we are to solve a linear system $Au = b$; a simple iteration for its solution is:

$$u^{(k+1)} = u^{(k)} + (b - Au^{(k)}), \tag{2.3}$$

where the quantity $r^{(k)} = b - Au^{(k)}$ is called the residual (sometimes also noted $\rho^{(k)}$) of the k-th iteration[1]. That quantity can readily be computed for any approximate solution as it requires only a matrix-vector product. The only operations actually required by the above method are such matrix-vector products and vector sums, which is interesting in cases where the matrix A is sparse, as is the case with matrices obtained from the discretization of differential operators by methods such as the FEM or FDM. Of course, if $u^{(k)}$ is the actual solution, the corresponding residual will be 0. We can see that in the simple iteration above, the residual is used as a correction to the current estimate. Notice that the exact error (which is clearly the optimal correction) can be computed by solving a problem on the residual: $e^{(k)} = u - u^{(k)} = A^{-1}r^{(k)}$. While computing it in this way makes little sense, we will see in Section 2.2.3 that this observation is at the origin of the concept of preconditioning.

[1] A variant is Richardson's method, were a relaxation parameter ω_R multiplies the residual in iteration (2.3).

It is well known that method (2.3) is not guaranteed to converge [106]. Its behaviour is mainly governed by the spectrum of the matrix $I - A$: the algorithm will most likely diverge if any of its eigenvalues is greater than 1. In that case, the component of the initial error e_0 corresponding to the associated eigenvector will expand at each iteration instead of being contracted. To understand this, notice that the successive errors are linked by the relation $e^{(k)} = (I-A)e^{(k-1)} = (I-A)^k e^{(0)}$. Making the link with the power method for the computation of the largest eigenvector of an operator [2], we see that if the largest eigenvalue is smaller than but close to 1, the algorithm will converge very slowly, because the contraction factor of that component is small, while other components may still converge very fast.

Another interest of a solver that requires only matrix-vector products, as already mentioned at the beginning of the section, is when only the application of matrix A to some vector is available, but not the coefficients of the matrix itself. In such a case, it is simply impossible to solve the problem with a direct solver; this situation can arise when the matrix can be seen as the discrete version of an iteration operator such as $\mathcal{F}$ in equation (2.2), representing one step of an iterative method. In that case, it is a routine involving the solution of a set of problems, rather than a matrix. For that reason, such systems are commonly referred to as "matrix-free" problems.

2.2.2 Krylov solvers

We now briefly introduce the principle of the Krylov subspace solvers, that are more modern algorithms than the simple iteration of the previous section and its variants. The motivation for their development is twofold: the need for solvers that are guaranteed to converge is essential, regardless of the properties of the system, and faster converging methods are also desired. For more details on the different solvers we refer to [106, 159].

Given a matrix A with rank R, Krylov solvers work by progressively building a subspace $\mathcal{K}(A,b,r) = \text{span}\{b, Ab, \ldots, A^{r-1}b\}$, into which an approximate solution is seeked at each iteration. One can see that the subspace is augmented with a vector that is the product of the last vector by matrix A, so only matrix-vector products are required for that process. The subspace $\mathcal{K}(A,b,R)$ spans the range of the inverse operator A^{-1} [103], so the solver is guaranteed to find the exact solution (up to machine precision) in at most R iterations. However, one is often satisfied with a sufficiently good approximation of the solution, so the iterations are stopped before full convergence, upon a criterion usually based on the relative decrease of the residual ρ: $||\rho^{(r)}||/||\rho^{(0)}|| < \varepsilon$, for some relative tolerance ε. The next section will introduce the concept of preconditioner, which is very often required to find a good approximation of the solution within a few iterations.

Because the vectors of the Krylov subspace tend to be highly linearly dependent, an orthogonalization of the basis is performed as a first step of every iteration. Indeed, it is required to guarantee the good conditioning of the second step of the iteration which is the computation of the coordinates of the new approximation, that is optimal in some sense. The many different Krylov algorithms essentially differ by the type of orthogonalization procedure and the choice of the optimization criterion for the solution. The algorithms are designed to exploit the properties of the system to be solved so as to be as efficient as possible in terms of operations and memory usage.

In this work we will focus on general systems, for which one of the most efficient algorithms is the Generalized Minimized Residual (GMRES) and will be our solver of choice. It is based on the Arnoldi procedure[103] for the orthogonalization and looks for the solution with minimum residual in the current subspace, which guarantees that the residual decreases at every step. Other popular algorithms are BiCGStab and QMR for general systems, and Conjugate Gradient (CG) for Hermitian systems.

2.2.3 Preconditioning

Even with Krylov acceleration, many classes of problems exhibit very slow convergence, or even stagnation, because a Krylov space of dimension r does not contain a sufficiently good approximation $u^{(r)} \in \text{span}\{b, \ldots, A^{r-1}b\}$ for small values of r. Propagation problems typically belong to that family [77]. To improve that situation, we consider a modification of the iteration (2.3) by introducing an additional component M^{-1} intended to speed up the convergence, by computing a more accurate correction $\tilde{e}^{(k)}$ to the current estimate:

$$\begin{aligned} u^{(k+1)} &= u^{(k)} + M^{-1}(b - Au^{(k)}) \\ &= u^{(k)} + \tilde{e}^{(k)}. \end{aligned} \tag{2.4}$$

The operator M^{-1} is called a preconditioner, because one can easily see that the above iteration is equivalent to (2.3) applied to the system $M^{-1}Au = M^{-1}b$. That modified system has same solution as the original one but will hopefully be better behaved for iterative solution, if $u^{(r)} \in \text{span}\{M^{-1}b, \ldots, (M^{-1}A)^{r-1}b\}$ is a better approximation found in the modified Krylov space.

This naturally leads to the question of the choice of the preconditioner M^{-1}. We mentioned earlier that to obtain the exact correction to an estimate solution, one should solve a problem on the residual. Doing so would make little sense, but if an approximate and less costly solution of the problem is available, it would be a good idea to use it as preconditioner. Another way to understand this is that if the preconditioner is close to the inverse of the operator $M^{-1} \approx A^{-1}$, the preconditioned system should be close to an identity: $M^{-1}A \approx I$, and can be expected to

converge very fast. But again, the cost of computing (in a pre-processing phase) and applying the preconditioner should be small enough for it to be practically interesting.

Therefore, designing an efficient preconditioner for a specific problem is not a trivial task and is often a matter of compromise. There are many classes of preconditioners, among which the incomplete LU factorization (ILU) for general algebraic (non matrix-free) systems and multigrid methods for elliptic problems have proved succesful in many cases and are commonly employed. Let us also mention that the classic Jacobi and Gauss-Seidel solvers are particular cases of (2.4), where M is based on splittings of $A = L + D + U$: it is respectively taken as the diagonal $M_J = D$ and the lower triangular part $M_{GS} = D + L$ of A. Similarly, the reverse Gauss-Seidel is $M_{GSr} = D + U$; they are sometimes called forward and backward sweeps. Other splittings are possible and lead to different methods; we mention the symmetric Gauss-Seidel preconditioner as it will have a clear link with the double sweep methods presented in Chapter 3. The preconditioner expression is:

$$\begin{aligned} M_{SGS} &= (D+L)D^{-1}(D+U) \\ &= M_{GS}M_J M_{GSr}, \end{aligned} \tag{2.5}$$

and alternates forward and backward sweeps, as an iteration with the symmetric Gauss-Seidel amounts to the two half-steps:

$$\begin{aligned} u^{(k+1/2)} &= -(D+L)^{-1}Uu^{(k)} + (D+L)^{-1}b; \\ u^{(k+1)} &= -(D+U)^{-1}Lu^{(k+1/2)} + (D+U)^{-1}b. \end{aligned} \tag{2.6}$$

or in a form similar to (2.4):

$$\begin{aligned} u^{(k+1/2)} &= u^{(k)} + (D+L)^{-1}(b - Au^{(k)}); \\ u^{(k+1)} &= u^{(k+1/2)} + (D+U)^{-1}(b - Au^{(k+1/2)}). \end{aligned} \tag{2.7}$$

Modifying the problem as $M^{-1}Au = M^{-1}b$ is called left- preconditioning. The new system can of course be solved by a Krylov solver as well, with the same benefit on the convergence rate. In that case, each iteration involves the application of A, followed by the application of M^{-1} (the implementations of the solvers take the preconditioner as a separate (optional) argument and applies it separately, to avoid the expensive explicit computation of the preconditioned operator). An alternative preconditioning strategy is to solve the system $AM^{-1}y = b$, with the change of variable $M^{-1}y = u$. This is known as right-preconditioning, and is sometimes prefered because the residual of the right-preconditioned system can be directly compared with the one of the original system, as their right-hand sides are identical. A more general preconditioned system by a matrix $M = M_1M_2$ that combines left- and right-preconditioning can be written as $\tilde{A}\tilde{u} = \tilde{b}$, with $\tilde{A} = M_1^{-1}AM_2^{-1}$, $\tilde{b} = M_1^{-1}b$ and $\tilde{u} = M_2u$.

The workflow for the solution of a multi-domain method with a right preconditioned system is shown on Figure 2.2.

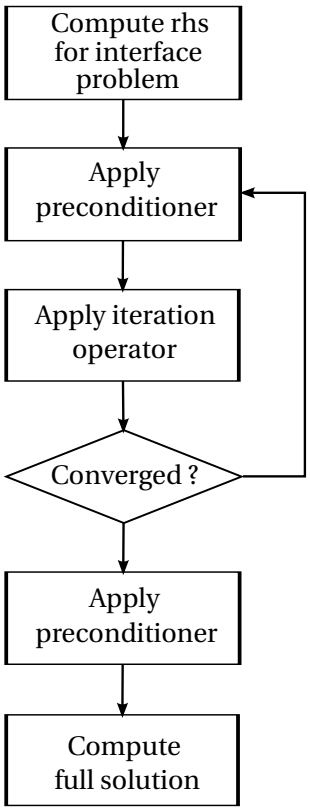


Figure 2.2: Multi-domain workflow with a right preconditioner for the solution of $Au = b$. A subsequent preconditioner application is needed after convergence because right preconditioning solves for an auxiliary variable y, so the original unknown is recovered as $u = M^{-1}y$. In the case of left preconditioning, it would be applied as a preliminary to compute the new right-hand side $M^{-1}b$ of the modified problem.

2.3 Domain partitions

The different algorithms presented in subsequent sections and chapters make use of particular partitionings of a domain. This Section is intended to briefly explain the different partitionings and the reasons that motivate these choices. We will also introduce notations that will be used in the description of the algorithms.

2.3.1 Decompositions

The most simple and natural way of partitioning a domain is to divide it into non-overlapping regions, called subdomains, often with additional constraints such as balanced distribution of the unknowns over the domains (load balancing) and minimized surface of the interfaces. This task can be achieved automatically in the context of finite element-type methods with packages like Metis [116] for complex geometries and irregular grids. A well-known difficulty of the non-overlapping Schwarz methods like the one of Section 2.4.2 is the presence of crosspoints [20, 29, 89], that are points located at the intersection of more than 2 subdomains (Figure 2.3). For that reason, decompositions that avoid such crosspoints are prefered in the present work. Note however that other domain decomposition algorithms such as FETI-DP [84, 120] take advantage of the crosspoints to speed up their convergence.

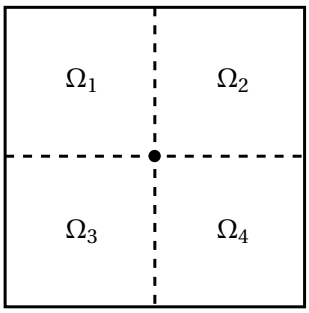


Figure 2.3: Simple decomposition of a square, with a crosspoint located in the center of the domain. Such a crosspoint causes difficulties in optimized Schwarz methods.

Two categories of non-overlapping decompositions without crosspoints are considered: layered decompositions, when the domain is decomposed into slices where the first and last do not share a common boundary as in Figure 2.4, and cyclic decompositions around an object (Figure 2.5).

The common boundary between domains Ω_i and Ω_j is called an artificial interface and is noted $\Sigma_{ij} = \Sigma_{ji}$ with $|i-j| = 1$. It must not necessarily be connected, i.e. the interface may be interrupted by an object that is not part of the computational domain, as in a cyclic decomposition with only 2 subdomains.

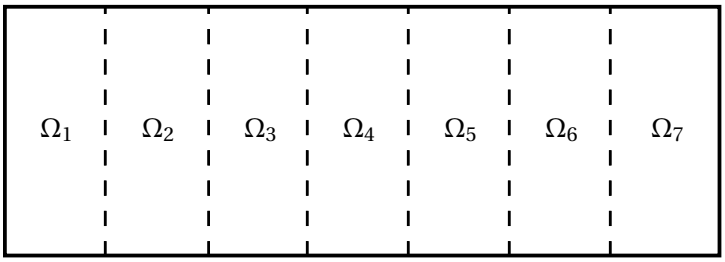


Figure 2.4: Layered decomposition without overlap. This kind of decomposition naturally avoids crosspoints.

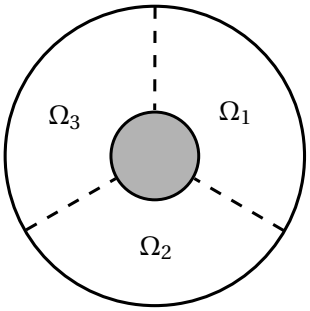


Figure 2.5: Cyclic decomposition around an object (shaded) that is not part of the computational domain.

In the family of Schwarz methods, it is well known that the classical Schwarz methods and its variants presented in Section 2.4.1 stagnate in the absence of overlap. An example of a layered decomposition suitable for these methods, derived from the non-overlapping example is shown on Figure 2.6. It is simply obtained by letting the domains grow into their neighbours. The principle is the same for cyclic decompositions; the case where more than 2 domains overlap in a single patch, as would be obtained by widening the domains of Figure 2.3, is not a problem for these methods. The category of optimized Schwarz methods (Section 2.4.2) are suitable for both overlapping and non-overlapping decompositions.

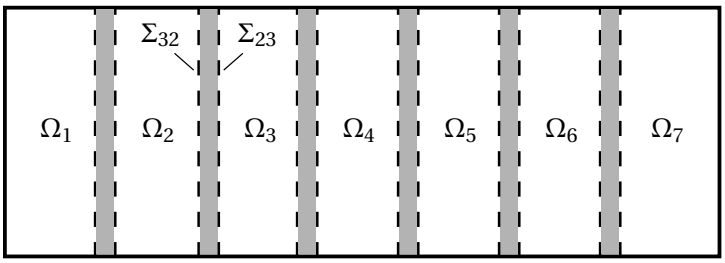


Figure 2.6: Layered decomposition with overlap

With an overlap, the boundaries of adjacent domains no longer coincide: $\Sigma_{ij} \neq \Sigma_{ji}$. We will use the notation Σ_{ij} for the boundary of Ω_i that closes the overlap with Ω_j. Note that for more complex (non-layered) decompositions, the artificial boundaries Σ_{ij} may partly intersect each other.

2.3.2 Covering

This kind of decomposition will be used in the multiple obstacles scattering algorithm, detailed in Section 2.5. That algorithm was designed for multiple scattering problems, where one is interested in the field produced by an incident wave illuminating a collection of $N > 1$ impenetrable obstacles $\Omega^-_{i,1\leq i\leq N}$ with boundaries $\Gamma_{i,1\leq i\leq N}$, in a complementary domain Ω^+. The computational domain is thus a region of the space enclosed in a boundary Γ_∞, with "holes" in it since the incident field is kept outside the objects (Figure 2.7).

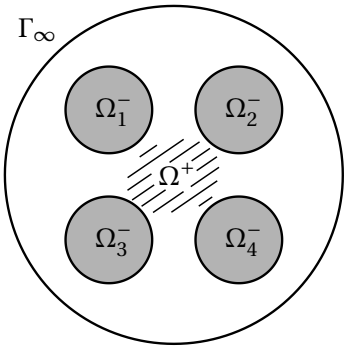


Figure 2.7: Geometry of a multiple scattering problem by 4 impenetrable objects $\Omega^-_{1\leq i\leq 4}$ (shaded on the Figure). The computational domain is Ω^+, with external boundary Γ_∞.

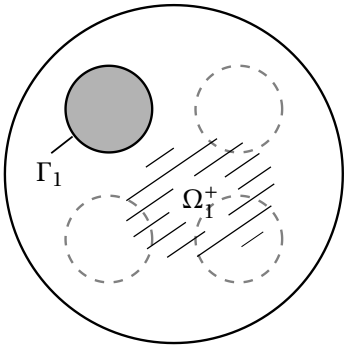


Figure 2.8: Geometry of one of the subproblems involved in the covering of Ω^+.

The method, detailed in Section 2.5, works by solving single scattering problems by each of the objects considered separately in the space truncated by the same boundary Γ_∞, where the holes of the other objects have been "filled". We

denote these domains by $\Omega^+_{i,1\leq i\leq N}$ (Figure 2.8). It is thus clear that each of these domains is larger than the original domain Ω^+. Together, they cover the whole (truncated) space: $\cup_1^N \Omega_i = \Omega^\circ$, hence the name covering. It is less natural than the decompositions presented above, though it can be seen as a limit case where the subdomains totally overlap each other. The method however is not a Schwarz method, although there are similarities that let them fit in the same framework.

The next sections will introduce different methods that fall into the multi-domain framework, and that are suitable for one or more of the partititioning strategies presented above.

2.4 Schwarz methods

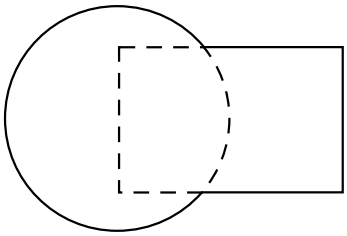


Figure 2.9: The "circle-rectangle" geometry used by Schwarz to expose his method. The dashed portions of the boundaries are the artificial interfaces used to exchange information between the domains and enclose the overlap region.

Schwarz methods are named after the german mathematician Karl H. Schwarz who first proposed his method in 1870 [162], originally to prove the existence of the solution to the Laplace problem in arbitrarily shaped domains. The demonstration, for which the famous circle-rectangle domain of Figure 2.9 was introduced, is based on successive solutions of problems on the circle and the rectangle, for which solutions were known to exist. Later on, the method was rediscovered and used to solve various kinds of elliptic problems on large domains, see [124] and references therein. Since then, many variants of the method have been proposed and studied.

2.4.1 Classical Schwarz

Schwarz methods can be formulated either at the continuous or discrete (or algebraic) levels, even though both approaches eventually lead to a discretization of the problem. This distinction can sometimes be confusing, even more so as these two approaches with apparently similar methods can in some cases be shown to be strictly equivalent (in the sense that the approximation produced after each iteration are identical), while in other cases they can differ substantially.

In this thesis, we will concentrate on the original approach followed by Schwarz, and work on algorithms that are fully described in terms of continuous operators and fields defined in the subdomains, as only these fit in our framework. It is only at the implementation stage that they will be discretized, in contrast with the situation where the problem is first discretized in the full domain, and the domain decomposition consists in manipulating parts of the resulting linear system by means of restriction operators defined on the subdomains [106, 165].

We will refer to these two categories of Schwarz methods respectively as iterative solvers and as preconditioners. To illustrate their similarities and differences, we present some simple examples of such methods.

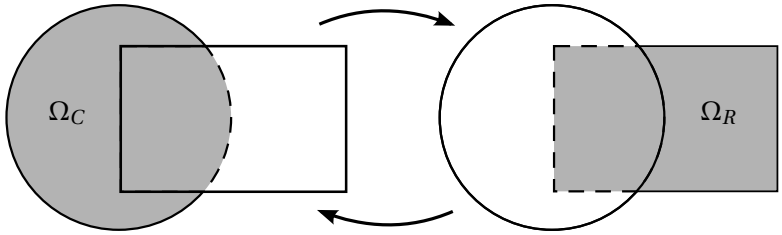


Figure 2.10: Principle of the alternating Schwarz method: the domains Ω_C and Ω_R are repeatedly solved in alternance and the solution in the overlap is updated everytime until some convergence criterion is met.

The original Schwarz method, also known as the "alternating" Schwarz method, uses a decomposition into two domains with overlap. The portion of the boundary of a domain that intersects the interior of the other domain is used to exchange information at each step, by means of Dirichlet conditions imposed on the artificial interfaces Σ_{ij}; it is done by sequentially solving the subproblems, and using the restriction of the solution on the boundary of the other problem as boundary condition for the next problem.

Doing so, the solution inside the overlap is updated several times per iteration (2 in the case of 2 subdomains), as illustrated on Figure 2.10, hence the name "alternating". The previous solution inside the overlap is simply discarded and replaced by the new one; the process continues until the solutions in the overlap match within some prescribed tolerance, in a given norm. The method, summarized in Algorithm 2.1, is described as follows in the case of two subdomains:

$$\begin{aligned} \mathcal{L}u_1^{(k+1)} &= f_1 && \text{in } \Omega_1, \\ u_1^{(k+1)} &= u_2^{(k)} && \text{on } \Sigma_{12}; \end{aligned} \qquad (2.8)$$

$$\begin{aligned} \mathcal{L}u_2^{(k+1)} &= f_2 && \text{in } \Omega_2, \\ u_2^{(k+1)} &= u_1^{(k+1)} && \text{on } \Sigma_{21}. \end{aligned}$$

It makes use of the most recent approximation of the solution to perform the next subdomain solve, and is therefore also called "Gauss-Seidel Schwarz" (GSS). The parallel "Jacobi Schwarz" (JS) is a modification proposed by Lions [124] that works only with data available at the start of the iteration:

$$\begin{aligned} \mathcal{L}u_1^{(k+1)} &= f_1 && \text{in } \Omega_1, \\ u_1^{(k+1)} &= u_2^{(k)} && \text{on } \Sigma_{12}; \end{aligned} \qquad \begin{aligned} \mathcal{L}u_2^{(k+1)} &= f_2 && \text{in } \Omega_2, \\ u_2^{(k+1)} &= u_1^{(k)} && \text{on } \Sigma_{21}. \end{aligned} \tag{2.9}$$

Algorithm 2.1: One step of the alternating Schwarz method (solver)

```
// Solve subproblems sequentially
for i = 1 : N
    Solve Lu_i = f_i, s.t.
      u_i = u_D   on ∂Ω_i ∩ ∂Ω
      u_i = g_ij  on Σ_ij

    // Update solution
    u ← u_i   in Ω_i

    // Exchange information
    for j = 1 : N
        g_ji ← u|_{Σ_ji}
    end
end
```

Analogous to the GSS, but formulated at the algebraic level, is the multiplicative Schwarz (MS) method. It is defined in two half steps as:

$$\begin{aligned} u^{(k+1/2)} &= u^{(k)} + R_1^T A_1^{-1} R_1 (f - A u^{(k)}), \\ u^{(k+1)} &= u^{(k+1/2)} + R_2^T A_2^{-1} R_2 (f - A u^{(k+1/2)}), \end{aligned} \tag{2.10}$$

where the R_j is the discrete restriction operator that selects the unknowns in Ω_i. They can be combined in a single step, where it clearly appears as a preconditioner:

$$
\begin{aligned}
u^{(k+1)} &= u^{(k)} + M_{MS}^{-1}(f - Au^{(k)}),\\
M_{MS}^{-1} &= \left[I - \prod_{j=1}^{N} (I - R_j^T A_j^{-1} R_j A) \right] A^{-1}.
\end{aligned}
\tag{2.11}
$$

Since A_i is exactly the discrete operator on Ω_i (and we have $A_i = R_i A R_i^T$, which is a diagonal block of the full matrix), this method produces exactly the same approximations as its continuous counterpart GSS, after discretization on the same mesh, as shown in [64]. Its additive parallel variant, introduced in [58], is:

$$
M_{AS}^{-1} = \sum_{i=1}^{N} R_j^T A_j^{-1} R_j. \tag{2.12}
$$

This version of the preconditioner solves the subproblems and sums the solutions obtained in all subdomains, including in the overlaps, where one has thus several contributions, leading to an excessive amplitude. Unless coupled to a Krylov solver, this method diverges, since the solution in the overlaps does not converge, as confirmed by a spectral analysis done in [64]. In the same paper, it is shown that the actual equivalent to the discretized JS is the restricted additive Schwarz (RAS) method. It is a modification of AS where a weighted sum is performed in the overlap, via modified restriction restriction matrices $\tilde{R}_j$, such that the solution converges everywhere:

$$
M_{RAS}^{-1} = \sum_{i=1}^{N} \tilde{R}_j^T A_j^{-1} R_j. \tag{2.13}
$$

In the continuous method, this problem does not occur since one keeps two different approximate solutions in the overlap, that eventually match when the solver has converged. In order to obtain a single approximation in the whole domain after any iteration, one needs to combine the different subdomain approximations in some way.

This is somehow similar to the situation of the RAS method. One could either arbitrarily choose one of these contributions, prolonge them to the middle of the overlap, or average them in some way. In all cases, it amounts to weighting the subdomains solutions by partition of unity functions $\mathcal{P}_i$: $u = \sum \mathcal{P}_i u_i$, provided that the subsolutions are extended with 0 outside of their domain of definition.

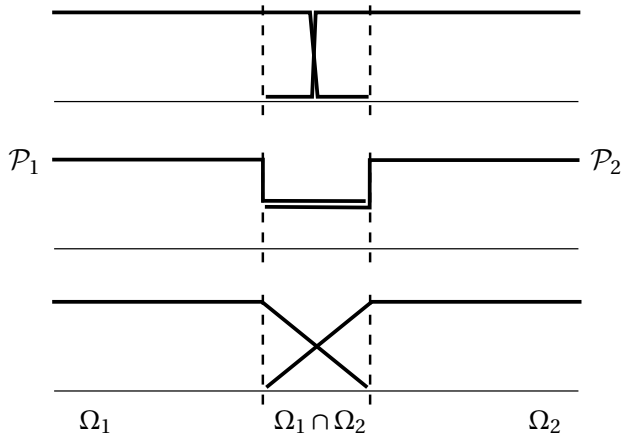


Figure 2.11: Different possible choices of the functions $\mathcal{P}_i$ that define a partition of unity over a 1d domain $\Omega = \Omega_1 \cup \Omega_2$: an arbitrary choice, in all or part of the overlap (top), a simple average (middle) or continuous weighting functions (bottom — smoother functions are also possible). The extension with 0 of $\mathcal{P}_i$ outside Ω_i is not represented.

The weight functions $\mathcal{P}_i$ are defined over the full domain Ω and must satisfy:

$$\begin{aligned} \mathcal{P}_i &= \begin{cases} 1 & \text{in } \Omega_i \setminus \cup_{j \neq i} \Omega_j, \\ 0 & \text{in } \cup_{j \neq i} \Omega_j \setminus \Omega_i, \end{cases} \\ \sum \mathcal{P}_i &= 1 \quad \text{in } \Omega. \end{aligned} \tag{2.14}$$

There are different ways to define a partition of unity over a domain, that differ in the particular choice of weighting function in the overlaps (see Figure 2.11). Discontinuous partitions of unity, like the arbitrary choice and the average, are easy to implement but lead to potentially discontinuous approximations, unless the method is fully converged. Continuous variants give smooth approximations across the interfaces but are harder to implement in complex-shaped overlaps. The method is given at Algorithm 2.2. Note that the choice of the partition of unity does not influence the convergence rate, only the quality of the iterates in the overlaps.

We will now see how the classical Schwarz method in its continuous version can be expressed as a linear system in order to fit in the framework of multi-domain methods. Supposing that we solve a linear problem $\mathcal{L}u = f$ in a domain Ω for some elliptic operator $\mathcal{L}$ and volume source f. We impose a Dirichlet boundary condition on the external boundary: $u = u_D$ on $\partial\Omega$. We first notice that the value of the full solution on the portion of the subproblems boundaries that lays inside the other domain $\Sigma = \cup \Sigma_{ij}$ (the dashed lines on Figure 2.9), together with the boundary conditions of the original problems on the external boundary, is all we need to obtain the restriction of the full solution inside each subdomain $u_i = u|_{\Omega_i}$. Therefore, the problem can be reformulated as a smaller problem with unknowns

Algorithm 2.2: One step of the Jacobi Schwarz (JS) solver, with combination of the subsolutions via a partition of unity

```
// Solve subproblems in parallel
for i = 1 : N
    Solve L u_i = f_i, s.t.
      u_i = u_D    on ∂Ω_i ∩ ∂Ω
      u_i = g_ij   on Σ_ij
end

// Update solution
u ← Σ_{i=1}^N P_i u_i   in Ω

// Exchange information
for i = 1 : N
    for j = 1 : N
        g_ji ← u|_{Σ_ji}
    end
end
```

only on these artificial boundaries. To be consistent with the presentation of the other methods and the general framework introduced above, we will rename these unknowns as $g_{ij} = u|_{\Sigma_{ij}}$.

The next step is to distinguish between the components of the solution produced in each subdomain by the physical sources u_D and f of the full problem and the artificial sources g_{ij} on the internal boundaries. We have $u_i = v_i + w_i$, with the definitions:

$$\begin{aligned} \mathcal{L}w_i &= f & \mathcal{L}v_i &= 0 & &\text{in } \Omega_i; \\ w_i &= 0 & v_i &= u & &\text{on } \Sigma_{ij}; \\ w_i &= u_D & v_i &= 0 & &\text{on } \partial\Omega_i \setminus \Sigma_{ij}. \end{aligned} \tag{2.15}$$

Similarly, we write $u_i^{(k)} = v_i^{(k)} + w_i$. Iterating over the unknown function g_{ij}, we have the update relation:

$$\begin{aligned} g_{ij}^{(k+1)} &= u|_{\Sigma_{ij}}^{(k+1)} \\ &= v_j|_{\Sigma_{ij}}^{(k+1)} + w_j|_{\Sigma_{ij}}, \end{aligned} \tag{2.16}$$

which gives for the full vector of unknowns g the fixed point relation:

$$\begin{aligned} g^{(k+1)} &= \mathcal{A}g^{(k)} + b \\ &= g^{(k)} + (b - \mathcal{F}g^{(k)}). \end{aligned} \tag{2.17}$$

This is the simple iteration relation (see next Section) for the operator $\mathcal{F} = \mathcal{I} - \mathcal{A}$, that solves the linear system $\mathcal{F}g = b$. Operator $\mathcal{A}$ applied to some vector g involves the solution for the variables v_i defined above, with Dirichlet data on Σ_{ij} given by g.

These methods are usually convergent for problems with smooth solutions; more details about them and their convergence can be found in [165]. Roughly speaking, one can say that the methods work well for problems with sufficiently regular solutions. However, they are found to fail for problems with oscillating solutions [77], because the updated values on the boundary can change a lot from one step to the next. This behaviour can cause the methods to oscillate or diverge instead of smoothly converging towards the correct solution.

Many theoretical results exist regarding the convergence of these methods in function of the size of the overlap, see [165] and references therein. Generally speaking, for problems for which the method can be proved to converge, like the Poisson problem, the methods converge faster as the overlap grows in size. In the particular case of non-overlapping subdomains, the methods obviously stagnate because the value on the common boundary is always updated to the same value.

Because they require an overlap and are not suitable for oscillatory functions, the classical Schwarz methods are not satisfactory for the solution of wave problems. The later introduced family of optimized Schwarz methods presented in the next section brings an answer to these issues.

2.4.2 Optimized Schwarz

Optimized Schwarz methods work by making use of more complex transmission conditions than the Dirichlet conditions used in the classical Schwarz methods: they involve both Dirichlet and Neumann data on the artificial interface and generally involve some operator $\mathcal{S}$ linking both. Such boundary conditions are called Robin or Fourier conditions, or sometimes impedance conditions. We will see that the choice of the operator $\mathcal{S}$ is critical for the rate of convergence and that an optimal choice exists, hence the name optimized Schwarz. These methods bring in several improvements over classical Schwarz methods [91], a major difference being that they are also suitable for non-overlapping decompositions. In the following, we will mostly focus on this last option.

A way of understanding the choice of Robin conditions as transmission conditions in the case of non-overlapping decompositions is that we wish to enforce the continuity of both the solution and its first (normal) derivative across the artificial interfaces:

$$\begin{aligned} u_i &= u_j \qquad\qquad \text{on } \Sigma_{ij}. \\ \partial_n u_i &= -\partial_n u_j \end{aligned} \tag{2.18}$$

It is well known that imposing either one of these conditions alone leads to non-convergent methods with non-overlapping decompositions, and that it is not possible to impose both conditions at once. The principle of optimized Schwarz was first introduced by Lions [125] and consists in applying a linear combination of these conditions:

$$\alpha\partial_n u_i + \beta u_i = -\alpha\partial_n u_j + \beta u_j \quad \text{on } \Sigma_{ij}, \tag{2.19}$$

with $\alpha, \beta \neq 0$. Després [51, 53] proposed to use the combination $\alpha = 1$ and $\beta = -\imath k$ when the method is applied to the Helmholtz equation. A more general conception is to consider a combination of linear operators; we will write:

$$\partial_n u_i + \mathcal{S} u_i = -\partial_n u_j + \mathcal{S} u_j \quad \text{on } \Sigma_{ij}, \tag{2.20}$$

for some invertible operator $\mathcal{S}$, that we will call the transmission operator. The algorithm can be described as follows: suppose we wish to solve an elliptic problem with operator $\mathcal{H}$ (for now, we consider the scalar Helmholtz operator $\mathcal{H} = -(\Delta + k^2)$, but the presentation that follows holds for general elliptic operators; we give the vector extension to Maxwell's equations at the end of the chapter), in a domain Ω:

$$\begin{aligned} \mathcal{H} u &= f \quad && \text{in } \Omega; \\ u &= u_D \quad && \text{on } \partial\Omega. \end{aligned} \tag{2.21}$$

We consider a layered decomposition of Ω into N non-overlapping slices $\Omega_{i,1\leq i\leq N}$, with artificial boundaries Σ_{ij} between Ω_i and Ω_j, so that our partitioning contains no loop: $\Omega_i \cap \Omega_j = \emptyset$ if $|i-j| \neq 1$.

The original problem (2.21) can be formulated in the subdomains so as to have $u_i = u|_{\Omega_i}$, by using impedance-matching boundary conditions on the artificial boundaries Σ_{ij}. Introducing the unknown interface data $g = \{g_{ij}, 1 \leq i \neq j \leq N, |i-j| = 1\}$, we look for the solution of:

$$\begin{aligned} -\mathcal{H} u_i &= 0 && && \text{in } \Omega_i, \\ (\partial_n + \mathcal{S}) u_i &= g_{ij} && := (-\partial_n + \mathcal{S}) u_j \quad && \text{on } \Sigma_{ij}, \end{aligned} \tag{2.22}$$

where the operator $\mathcal{S}$ has a twofold role: it must simulate the impedance of the domain that extends beyond the artifial boundary, and ensure that all sources located outside produce an equivalent contribution inside the subdomain. Boundary conditions on $\partial\Omega_i \cap \partial\Omega$ are conserved from the original problem. We will assume in all that follows that the DDM is well-posed, in the sense that each subproblem (2.22) is well-posed, i.e., away from interior resonances. We have defined a pair of unknown functions per interface, and we use the convention that g_{ij} is the impedance data for problem i on the common boundary with subdomain j. To solve for these new unknowns, the Schwarz algorithm works by iteratively solving the subproblems and transferring the updated information to the adjacent

domains via an exchange relation. We present the additive version, where an iteration amounts to solving all subproblems in parallel (as opposed to the faster converging and sequential multiplicative version thereof):

$$\begin{aligned} -\mathcal{H}u_i^{(k+1)} &= 0 \quad \text{in } \Omega_i, \\ (\partial_n + \mathcal{S})u_i^{(k+1)} &= g_{ij}^{(k)} \quad \text{on } \Sigma_{ij}, \end{aligned} \tag{2.23}$$

and then to updating the unknowns:

$$\begin{aligned} g_{ij}^{(k+1)} &= -\partial_n u_j^{(k+1)} + \mathcal{S}u_j^{(k+1)} \quad \text{on } \Sigma_{ij}, \\ &= -g_{ji}^{(k)} + 2\mathcal{S}u_j^{(k+1)}. \end{aligned} \tag{2.24}$$

We still have the choice of operator $\mathcal{S} : H^{1/2}(\Sigma_{ij}) \to H^{-1/2}(\Sigma_{ij})$. (Strictly speaking, we should define 2 operators per artificial interface, but we use a unique notation for convenience.) We have the constraint that its null-space must be equal to $\{0\}$[2]; it is well known that this choice influences the rate of convergence [31, 91], and it was shown in [138] that the optimum is obtained for $\mathcal{S}$ being the exterior Dirichlet-to-Neumann (DtN) map $\mathcal{D}$ of the complement of the subdomain $\Omega \setminus \Omega_i$, defined on a boundary Σ as:

$$\begin{aligned} \mathcal{D}: \quad & H^{1/2}(\Sigma) \to H^{-1/2}(\Sigma), \\ & u_{|\Sigma} \longmapsto \partial_n u_{|\Sigma} = \mathcal{D}u_{|\Sigma}. \end{aligned} \tag{2.25}$$

In practice, most optimized Schwarz algorithms make use of (approximations of) DtN maps that correspond to an open, free-space complement of the subdomain to avoid the very costly computation of the DtN map corresponding to the actual complementary domain. This amounts to defining absorbing boundary conditions on the artificial interfaces. A perfectly matched layer (PML) can also be used for that purpose, as in [169]. We refer to Section 2.4.3 for the description of approximations that will prove useful for numerical applications.

Let us note that by defining a new set of unknowns g, we use the Schwarz procedure as a solver, although DDM algorithms are often regarded as preconditioners [105, 165] as was explained in Section 2.4.1. In the latter case, the iterative solver acts on the full system obtained from the discretization of the original operator in the whole domain, with unknowns u. It was shown in [167] that the multiplicative version of the discretized optimized Schwarz algorithm (2.23)–(2.24) is equivalent to its discrete (preconditioner) equivalent. It is also the case for the additive version presented above, with a condition on the overlap. For the non-overlapping version, the equivalence only holds for an augmented system, where

[2]This is equivalent to requiring that the operator is invertible. It is however more intuitively understood that if a non-zero field belongs to the null-space of the transmission operator, it will not be transmitted and the corresponding mode will not converge.

the unknowns on the artificial interfaces are duplicated.

Similarly to the classical Schwarz methods, we can rewrite this algorithm as a linear system that fits well into our multi-domain framework: by exploiting the linearity of the problem, we can separate the (unknown) solutions of the subproblems into two components: the contribution of the artificial sources on the internal boundaries v_i and the physical sources w_i, such that $u_i = v_i + w_i$. In the course of iterations, we write the current approximation as $u_i^{(k)} = v_i^{(k)} + w_i$, since the physical sources do not vary. We then inject the decomposition of the unknown field in the update of the Schwarz unknowns (2.24):

$$\begin{aligned} g_{ij}^{(k+1)} &= -g_{ji}^{(k)} + 2\mathcal{S} v_j^{(k+1)} + 2\mathcal{S} w_j \qquad \text{on } \Sigma_{ij}, \\ &= -g_{ji}^{(k)} + 2\mathcal{S} v_j^{(k+1)} + b_{ij}. \end{aligned}$$

Considering the full vector of unknowns, we obtain the fixed point iteration:

$$g^{(k+1)} = \mathcal{A} g^{(k)} + b,$$

where the iteration operator $\mathcal{A} : \times_{i,j=1}^{N} L^2(\Sigma_{ij}) \to \times_{i,j=1}^{N} L^2(\Sigma_{ij})$ is one step of the above algorithm with the physical sources set to 0. The vector b contains the local contributions of the external sources, and is computed as the output of the update relation (2.24) applied to w_i. From the iteration above we obtain, at convergence, the linear system:

$$\mathcal{F} g = (\mathcal{I} - \mathcal{A}) g = b. \tag{2.26}$$

The new iteration operator $\mathcal{F}$ is very similar to $\mathcal{A}$, with the difference that the update relation (2.24) is modified to match its definition:

$$g_{ij}^{(k+1)} = g_{ij}^{(k)} + g_{ji}^{(k)} - 2\mathcal{S} u_j^{(k+1)}. \tag{2.27}$$

The application of the iteration operator $\mathcal{F}$ and the construction of the right-hand-side b are summarized in Algorithms 2.3 and 2.4. Once the Schwarz problem has been solved for g, one must compute v_i from (2.22) and finally obtain the original unknown u in each subdomain as $u_i = v_i + w_i$.

The Schwarz problem under form (2.26) can be interpreted as the solution of a linear system of equations, with matrix $\mathcal{F}$, unknowns g and right-hand-side b, ready to be solved by standard linear solvers suitable for non-symmetric systems. In particular, the solution can be accelerated by using a Krylov solver like GMRES [105, 159]. The construction of the Krylov subspace only requires matrix-vector products, which allows for a "matrix-free" implementation of the operator $\mathcal{F}$ — the only operation required being the application of $\mathcal{F}$ to a given vector, which amounts to solving the subproblems and performing the update of the unknowns. We will see in the next section that an explicit expression of this matrix can actually be formed, although doing so is only useful for the purpose of analysis in view of the computational cost of the procedure.

Algorithm 2.3: Application of the iteration operator $g \leftarrow \mathcal{F}g$.

```
// Solve subproblems (in parallel)
```
for $i = 1 : N$
  $u_D \leftarrow 0$ on $\partial\Omega_i \cap \Gamma_D$
  $f_i \leftarrow 0$
  $g_l \leftarrow g_{i,i-1}$
  $g_r \leftarrow g_{i,i+1}$
  Solve $\mathcal{H}_i u_i = f_i$, with impedance data g_l and g_r
end

```
// Update impedance data
```
for $i = 1 : N-1$
  $g_{i,i+1} \leftarrow g_{i,i+1} + g_{i+1,i} - 2\mathcal{S}u_{i+1}|_{\Sigma_{i,i+1}}$
  $g_{i+1,i} \leftarrow g_{i+1,i} + g_{i,i+1} - 2\mathcal{S}u_i|_{\Sigma_{i+1,i}}$
end

Algorithm 2.4: Computation of the right-hand side b.

```
// Solve subproblems
```
for $i = 1 : N$
  $u_D \leftarrow u_D$ on $\partial\Omega_i \cap \Gamma_D$
  $f_i \leftarrow f$
  $g_l \leftarrow 0$
  $g_r \leftarrow 0$
  Solve $\mathcal{H}_i u_i = f_i$, with impedance data g_l and g_r
end

for $i = 1 : N-1$
  $b_{i,i+1} \leftarrow -2\mathcal{S}u_{i+1}|_{\Sigma_{i,i+1}}$
  $b_{i+1,i} \leftarrow -2\mathcal{S}u_i|_{\Sigma_{i+1,i}}$
end

2.4.3 Transmission conditions for optimized Schwarz methods

The optimized Schwarz methods presented in Section 2.4.2 combine Dirichlet and Neumann data, together with an operator $\mathcal{S}$, to build transmission conditions over the artificial interfaces. We mentioned earlier that the optimal choice of the transmission operator $\mathcal{S}$ in the algorithm defined by (2.22) and (2.24) is the DtN map $\mathcal{D}$ defined by (2.25). In that case the convergence of the iterative process is optimal [155, 156]. Beside the simple one-dimensional case where the DtN is trivial (see Section 1.1.2), this however leads to a very expensive procedure in practice, as the DtN is a non-local operator: see e.g. [43].

A great variety of techniques based on local transmission conditions have thus been proposed over the years: these include the class of FETI-H methods [32, 82, 83, 85], the optimized Schwarz approach [90], the evanescent modes damping algorithm [21, 28, 29] and the Padé-localized square-root operator [31]. All these local transmission conditions can be seen as approximations of the exact DtN operator; the better the related impedance operators approximate the exact DtN operator on all the modes of the solution, the better the convergence properties of the resulting DDM.

The DtN map is intimately linked with the artificial boundary conditions that were originally introduced to truncate infinite domains in various fields of application, some of which have been presented in Section 1.3.3. When used in the context of wave propagation problems, they are often called absorbing, non-reflecting or transparent boundary conditions. All these names refer to the fact that such conditions attempt to prevent or minimize the reflection of any outgoing wave that would hit the artificial boundary, hence allowing it to freely leave the domain. This property is interesting when designing a domain decomposition algorithm since a wave propagating in a domain should naturally be able to cross artificial boundaries without distortion or reflection. An efficient transmission operator should therefore fully capture all of the outgoing waves and release them intact at the other side of the boundary.

We will use several local approximations of the DtN operator for the numerical tests presented in the next Section (IBC(χ), OO_2 and GIBC(N_p)), as well as one non-local approximation using perfectly matched layers (PML(n_{PML})).

Local impedance condition: IBC(χ)

A simple local approximation of the DtN is the following Impedance Boundary Condition:

$$\mathcal{S}^{\text{IBC}(\chi)} u = (-\imath k + \chi) u,$$

where χ is a self-adjoint positive operator [28]. When $\chi = 0$, one recovers the classical Sommerfeld radiation condition, used by Deprés in the original non-overlapping Schwarz method for Helmholtz [53]. We only consider here the case where χ is a real-valued positive coefficient, which amounts to approximate the DtN map by a diagonal operator with a constant value on the diagonal. Choosing $\chi \neq 0$ allows to improve the convergence of the DDM for evanescent modes. In Section 3.4.1 we will also use the impedance boundary condition with a modified wavenumber k_h, i.e. $\mathcal{S}^{\text{IBC}_{k_h}(\chi)} u = (-\imath k_h + \chi) u$.

Optimized second order local impedance condition: OO_2

While IBC(χ) is a zero-th order approximation of the DtN, higher order approximations can also be constructed. For a generic transmitting boundary Σ, Gander

et al. [90] proposed to construct a second order local approximation of the DtN in the form:

$$\mathcal{S}^{\mathrm{OO}_2} u = (a - b\Delta_\Sigma)\, u, \tag{2.28}$$

where the complex numbers a and b are obtained by solving a min-max optimization problem on the rate of convergence (hence the name "Optimized Order 2"), and Δ_Σ is the Laplace-Beltrami operator on the interface Σ: $\Delta_\Sigma := \mathrm{div}_\Sigma \nabla_\Sigma$. The optimal transmission coefficients depend on several parameters, chosen to solve the min-max problem on a bounded domain, excluding the Fourier modes close to the cut-off frequency [90]. In all the numerical tests we chose the parameters leading to the optimal convergence rate. (An optimized 0th order approximation can also be constructed in the same way.)

Padé-localized generalized impedance condition: GIBC(N_p)

Instead of a polynomial approximation of the DtN operator, a rational approximation was proposed in [31]:

$$\mathcal{S}^{\mathrm{GIBC}(N_p)} u = C_0 u + \sum_{\ell=1}^{N_p} A_\ell \mathrm{div}_\Sigma(k_\varepsilon^{-2}\nabla_\Sigma)(1 + B_\ell \mathrm{div}_\Sigma(k_\varepsilon^{-2}\nabla_\Sigma))^{-1} u, \tag{2.29}$$

where C_0, A_ℓ and B_ℓ ($\ell = 1, \ldots, N_p$) are the coefficients of a complex Padé expansion of the square root operator (with a rotation of the branch cut of $\pi/4$), and $k_\varepsilon = k + \imath\varepsilon$ is a complexified wavenumber. (In all subsequent tests we used $\varepsilon = k/4$). This condition leads to a DDM with quasi-optimal convergence properties, meaning that the rate of convergence is optimal on the evanescent modes and is improved compared to other local techniques for the remaining modes. To be noted is that, unlike the other techniques presented above, GIBC(N_p) makes use of N_p auxiliary unknown functions. Consequently, the size of the linear system to be solved is augmented. The additional cost is usually very small [31], unless the number of unknowns on the interfaces is large compared to the number of volume unknowns.

Non-local PML condition: PML(n_{PML})

Finally, we also consider a non-local approximation of the DtN operator constructed with the Perfectly Matched Layer (PML) technique introduced in Section 1.3.4, which is an alternative to the construction of the DtN map with an integral formulation as described in Appendix B.2. The procedure was developed in the context of matrix probing of the DtN map [19, 41], that could be applied in our algorithm as suggested in [182]. A fictitious "black-box" domain Ω_{bb} is first created by extruding the mesh of the interface Σ over $n_{\mathrm{PML}} + 3$ layers, using a mesh size identical to the mesh of the subdomain connected to Σ (see Figure 2.12). If the mesh of the interface Σ counts n_Σ internal nodes, the approximate $n_\Sigma \times n_\Sigma$ DtN matrix is built by inspection: column c is obtained by solving a Helmholtz problem with an imposed value of 1 on the c^{th} node and 0 on the other nodes of Σ, and a 1d PML

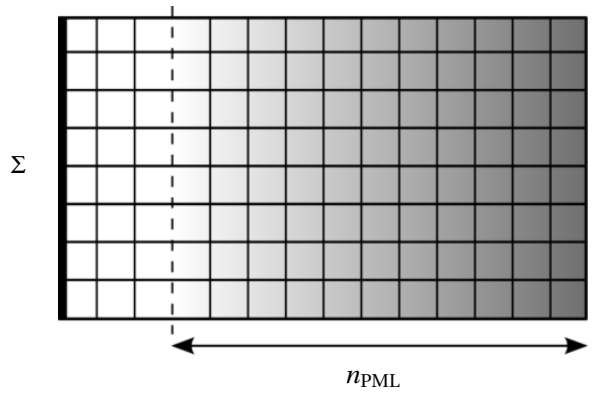


Figure 2.12: Geometry of fictitious "black-box" domain Ω_{bb} used to compute the non-local approximation of the DtN map using a perfectly matched layer. The domain Ω_{bb} is made of a few layers (3 in this case) between the interface of interest Σ and the PML, made of n_{PML} layers. The Dirichlet condition is imposed on Σ, while the boundary conditions on the top and bottom sides of Ω_{bb} are inherited from the ones imposed on the edges of Σ in the original problem. The gradient represents the (1d) growth of the PML absorption factor.

layer in the last n_{PML} cells of Ω_{bb} (the boundary condition on the top and bottom boundary nodes of Ω_{bb} is inherited from the ones on the top and bottom node of Σ; a homogenous Dirichlet condition is imposed on the right). The normal derivative is extracted on Σ using the weak form of the finite element formulation. The number of layers (3 in all our numerical tests) added in front of the PML allow to clearly separate the PML from the elements used in the computation of the normal derivative. More details on the construction of this approximation can be found in Appendices C.1 and C.2.

This method would be too costly to use as-is in practical applications, but it is useful to consider it here for benchmarking the proposed preconditioner, as the quality of the approximate DtN map is directly related to that of the PML, which can be controlled via its thickness (number of layers n_{PML}). We will thus refer to the non-local PML-based approximation as PML(n_{PML}).

Another way of using PMLs in the algorithm would be to proceed like in [169]: instead of extracting the DtN map from the black-box as explained above, a PML is appended on the interfaces of the subdomains. Both approaches are algebraically equivalent, but their implementations differ in several aspects: with the black-box method, most of the computational work is done in preprocessing, and in the frequent practical cases where several interfaces are similar and can use the same DtN map (like in all the test cases of the next Section), the computation of the DtN map must be done only once; the other approach needs no additional preprocessing, but requires a modification of the geometry of all the subdomains and the evaluation of the derivative at the boundary of the PML, with an increased

cost at every iteration. Recalling that a motivation for using a DDM is the size of the linear systems to be inverted, such an addition of the PMLs could make the method less attractive. Details on this procedure are given in Appendix C.3.

2.4.4 Coarse grid and scalability

From the structure of the Schwarz methods and the exchange relations (2.16) and (2.24), it is clear that information is only exchanged locally between adjacent subdomains during one step of the algorithm. Considering layered decompositions (see Figure 2.4) with many subdomains, the number of iterations required for the transfer of information between the most distant domains is large. This seriously impacts the quality of the approximations computed after the first few iterations if a source located in some subdomain has a significant contribution in other subdomains, away from it. This is typically the case for propagation problems, or more generally when the Green's function associated with the problem operator decays slowly [77].

A direct and undesired consequence is that the convergence rate of the methods strongly depends on the number of subdomains [106]. This leads to the concepts of scalability: a method is said to be strongly scalable if it can solve a given problem twice as fast if the number of processors involved is doubled. In a domain decomposition context, if we suppose that each processor is assigned to one subdomain, adding processors means dividing the full domain into smaller subdomains. Achieving strong scalability is illusory, simply because adding processors automatically increases the amount of communication required. Also, it makes little sense to hope that asymptotically increasing the number of processors to infinity would lead to a solution in zero time. Hence, a more useful and achievable concept is that of weak scalability: supposing that we increase the size of the problem and proportionally the number of processors, a weakly scalable algorithm would solve the problems in a fixed amount of time.

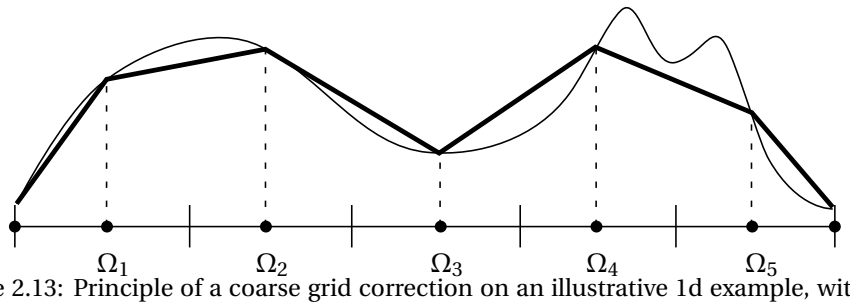


Figure 2.13: Principle of a coarse grid correction on an illustrative 1d example, with one point per subdomain (black dots). The left-hand part of the solution is smooth and quite well represented on the coarse grid, whereas the higher frequency features on the right-hand side are missed by the coarse approximation.

In order to make the methods less sensitive to the decomposition, some mechanism should be employed to share information on a global level between the subdomains, leading to the concept of multi-level methods. We have already come across an example of such a method in Section 1.4.2, where we introduced multigrids. In a DDM context, the first example of such a technique was proposed by [57] for the classical Schwarz methods. It consists in the introduction of an additional component in the method, where the problem is solved on a much coarser discretization, as shown on Figure 2.13. That coarse approximation is then interpolated on the fine mesh, enabling the global exchange of information over the subdomains. This leads to a modification of the additive Schwarz preconditioner in the following way:

$$M_{AS+C}^{-1} = \sum_{i=1}^{N} R_j^T A_j^{-1} R_j + R_C^T A_C^{-1} R_C, \tag{2.30}$$

where R_C is the interpolation operator from the fine to the coarse discretization, and $A_C = R_C A R_C^T$ is the discrete operator at the coarse level.

With this addition, the method is called a two-level algorithm, as it somehow works like a multigrid method: the second level complements very well the standard DDM methods because the coarse approximation helps to remove the low-frequency component of the residual during the iterations, whereas the high-frequencies are resolved by the subproblems solves on the fine mesh. This can clearly be observed when looking at the spectral radius of the method in function of the frequency, with and without coarse grid as in [93].

Even more efficient coarse grids can be designed by carefully choosing the location of the coarse discretization vertices: as the residual is typically 0 inside the subdomains and jumps in the overlap, the coarse grid corrects the error more accurately if its nodes are placed inside the overlap, on both sides of the jump, rather than in the middle of the subdomains [93]. In [59], a coarse grid suitable for optimized Schwarz methods is presented.

This kind of technique proved very succesful for Laplace-like problems. By extension, the name “coarse grid” (or sometimes “coarse space”) became generic for all subsequent techniques with the same purpose, even though their working principle is sometimes very different and does not rely on another discretization, like e.g. deflation-based preconditioners [70, 74, 88].

Unfortunately, the basic principle of the coarse grid fails when employed for problems with oscillating solutions [77], because the coarse discretization cannot represent them and the approximations given by the coarse grid are totally wrong, which makes it inefficient. Using more points in the coarse discretization would make the technique more costly, while numerical dispersion would still

cause phase errors in the correction. Moreover, that technique is implemented as an additional component to a preconditioner (see equation (2.30)); it is not clear how to implement a similar component in a method formulated at the continuous level. Another example of coarse grid is found naturally in the FETI-DP method [84] and similar [20, 29] methods; see [183] for the vector case. We will see in the next chapter how a similar performance improvement can be achieved for the continuous optimized Schwarz algorithm.

2.5 Multiple obstacles scattering algorithm

In scattering problems, one is interested in the field produced by some incident wave u^{inc} propagating in free space and interacting with one or more objects. The presence of the object, or scatterer, influences the propagation of the wave since it causes reflections in different directions depending on its shape and creates a shadow behind itself, as illustrated on Figure 2.14. Similarly, multiple scattering is the effect of a collection of S objects on the propagation of an incident wave. If we denote the portion of space occupied by the objects as $\Omega^- = \cup_{s=1}^{S} \Omega_s^-$, the exterior computational domain is $\Omega^+ = \mathbb{R}^d \setminus \Omega^-$.

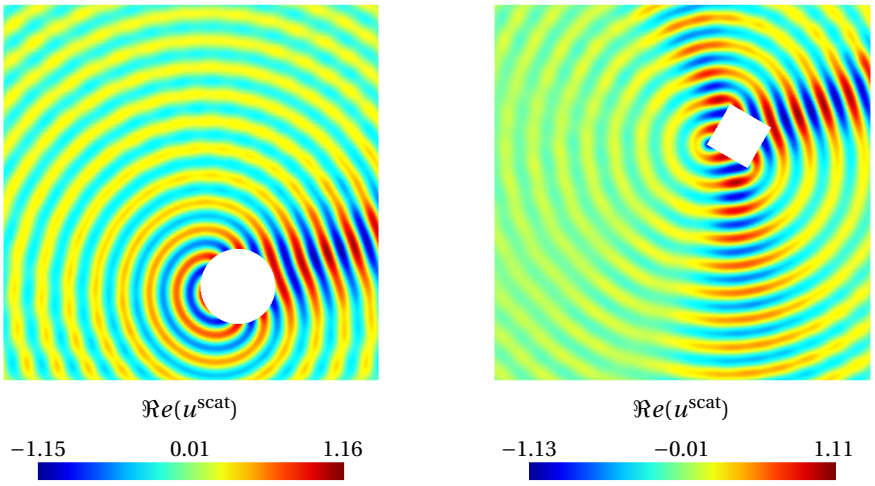


Figure 2.14: Examples of solutions of single scattering problems. Both objects are illuminated by a plane wave under the same incidence, with $k = 7\pi$. The computational domain is the unit square.

While high frequency single scattering problems can be solved by asymptotic methods or even analytically for some particular shapes, multiple scattering problems are much harder to solve because of their interactions (e.g. multiple reflections) with each other (Figure 2.15). Typical examples of multiple scattering problems are the propagation of radio waves in an urban environment, or the scattering of surface waves by a group of islands, that can be observed from an airplane.

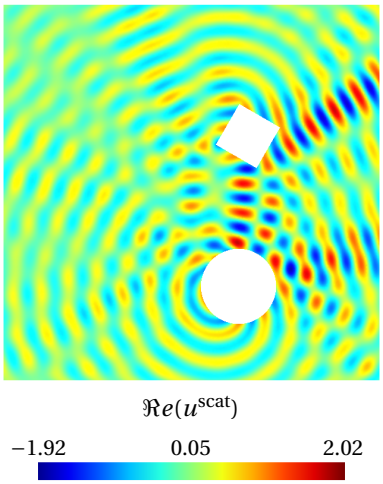


Figure 2.15: Illustration of the scattering by a collection of objects. The interactions between objects in multiple scattering problems create a more complex scattered field than the superposition of the two waves produced by the objects taken separately.

2.5.1 Multiple scattering as coupled problems

Considering sound-soft obstacles, it is common to split the total field into the incoming wave and the scattered field: $u^{\text{tot}} = u^{\text{inc}} + u^{\text{scat}}$ in the computational domain Ω^+, in which case we want to solve the following problem (we use the simplified notation u for u^{scat}):

$$\begin{aligned} \mathcal{H}u &= 0 && \text{in } \Omega^+, \\ u &= -u^{\text{inc}} && \text{on } \Gamma, \\ &+ \text{radiation condition} && \text{at } |r| \to \infty. \end{aligned} \tag{2.31}$$

We refer to Section 1.3.3 for details on the radiation condition. The interior of the objects is denoted by Ω_i^- and is excluded from the original computational domain; see Section 2.3.2 for details on the geometrical setting. In problem (2.31) we imposed a Dirichlet or sound-soft condition; alternatives are the sound-hard or impedance conditions, that correspond respectively to Neumann ($\partial_n u = -\partial_n u^{\text{inc}}$) or Robin conditions ($(\partial_n u + Z)u = -(\partial_n u + Z)u^{\text{inc}}$, with Z some complex-valued scalar or operator). We only consider the sound-soft case in the following, except when the sound-hard case requires special treatment.

Once again, solving this problem with standard discretization methods and a direct solver rapidly becomes difficult as one increases the frequency of the incoming wave. The multiple obstacles scattering algorithm (MOSA) is an iterative method that fits well in the multi-domain framework, as it reformulates the problem in terms of surface unknowns and iterates by solving subproblems and exchanging data between the different scattering surfaces. It is based on the decomposition of the solution as a sum of fictitious waves that are solutions of single

scattering problems, defined in the domains Ω_i^+ that compose the covering of the original domain Ω^+ (see Section 2.3.2):

$$u = \sum_{s=1}^{S} u_s \qquad \text{in } \Omega^+, \tag{2.32}$$

for some particular (and a priori unknown) set of individual sources $\{u_{1\leq s\leq S}\}$ on the boundaries Γ_s that produce an equivalent solution after superposition [13]. In the multi-domain framework presented in Section 2.1, the vector of unknowns g corresponds to precisely this set of Dirichlet data.

These waves are the result of the combination of the incoming wave u^{inc} and the scattered waves by the other objects, and are thus solution to the coupled problems:

$$\begin{aligned} \mathcal{H}u_s &= 0 && \text{in } \Omega_s^+, \\ u_s &= -u^{\text{inc}} - \sum_{q\neq s} u_q && \text{on } \Gamma_s, \\ &+ \text{radiation condition} && \text{at } |r| \to \infty. \end{aligned} \tag{2.33}$$

Under that form, the problem is reformulated as a surface problem, since the knowledge of the u_s on the boundaries Γ_s is sufficient to reconstruct the solution in the whole Ω^+.

2.5.2 Iterative solution of the coupled problem

One can prove [8] that the solution to (2.33) exists and is unique. However, solving the coupled problem for the set of boundary data u_s cannot be done directly. Instead, we can rewrite the interface problem arising from the boundary conditions in (2.33) as a linear system involving transfer operators $\mathcal{G}_{sq}$ from a boundary Γ_s to another Γ_q:

$$\begin{bmatrix} u_1 \\ \vdots \\ u_S \end{bmatrix} = - \begin{bmatrix} 0 & \mathcal{G}_{12} & \cdots & \mathcal{G}_{1S} \\ \mathcal{G}_{21} & 0 & \cdots & \mathcal{G}_{2S} \\ \vdots & & \ddots & \vdots \\ \mathcal{G}_{S1} & \mathcal{G}_{S2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_S \end{bmatrix} - \begin{bmatrix} u_1^{\text{inc}} \\ \vdots \\ u_S^{\text{inc}} \end{bmatrix}, \tag{2.34}$$

that simplifies as $\mathcal{F}u = (\mathcal{I} + \mathcal{G})u = b = -u^{\text{inc}}$. Applying the transfer operator $\mathcal{G}_{sq} : H^{1/2}(\Gamma_q) \to H^{1/2}(\Gamma_s)$ to a function v_q defined on Γ_q implies the solution of a problem in Ω_q^+ with boundary data v_q:

$$\begin{aligned} \mathcal{H}u_q &= 0 && \text{in } \Omega_q^+, \\ u_q &= v_q && \text{on } \Gamma_q, \\ &+ \text{radiation condition} && \text{at } |r| \to \infty. \end{aligned} \tag{2.35}$$

and taking the trace of the solution on Γ_s. Therefore, applying the matrix of system (2.34) involves the solution of S problems (2.35) and summing the $S-1$ corresponding traces for each of them, which provides the exchange of information

between interfaces that is required to define a multi-domain method. The subproblems are independent and can be solved in parallel; Algorithm 2.5 is a routine that performs the application of the MOSA operator. Note that, as opposed to Schwarz methods where the exchange of information is limited to neighbouring interfaces, the MOSA achieves a global exchange, as can be seen with the dense structure of matrix $\mathcal{G}$.

Algorithm 2.5: Application of the MOSA operator: $r \leftarrow Fr$

```
// Solve subproblems in parallel
for i = 1 : N − 1
    u_D ← r_i on Γ_i
    Solve H_i u_i = 0,   s.t. u_i = r_i on Γ_i
end
// Update
for i = 1 : N
    r_i ← Σ_{j=1}^{N} u_j|_{Γ_i}
end
```

Let us already mention that, as will be discussed in Section 4.4.1, the fact that this exchange of information can be performed by solving each problem only once into the whole single scattering domain simplifies the implementation, but doing so also limits the possibilities of designing preconditioners into which selective and less expensive exchange of information would be desirable.

As for the other multi-domain formulations, the linear system with operator $\mathcal{F}$ can be solved by using the techniques presented in Section 2.2. In particular, Jacobi and Gauss-Seidel methods have been compared, as well as an unpreconditioned GMRES in [99] (in Section 4.4.1, we will propose some preconditioning strategies for Krylov methods). Let us also mention that an alternative inspired by the physics of the multiple reflections consists in computing the solution of the original problem as the series [99]:

$$u = \sum_{m=1}^{\infty} \sum_{s=1}^{S} u_s^{(m)} \quad \text{in } \Omega^+, \tag{2.36}$$

with the corrections $u_s^{(m)}$ at each iteration:

$$\begin{aligned} \mathcal{H} u_s^{(m)} &= 0 && \text{in } \Omega_q^+, \\ u_s^{(m)} &= c_s^{(m)} && \text{on } \Gamma_q, \\ &+ \text{radiation condition} && \text{at } |r| \to \infty. \end{aligned} \tag{2.37}$$

and

$$c_s^{(m)} = \begin{cases} -u^{\text{inc}} - \sum_{q=1}^{s-1} u_q^{(m)} & \text{for } m = 1, \\ -\sum_{q=1}^{s-1} u_q^{(m)} - \sum_{q=s+1}^{S} u_q^{(m-1)} & \text{for } m > 1. \end{cases} \tag{2.38}$$

This is somehow like a Gauss-Seidel iteration, where each correction $u_s^{(m)}$ can be interpreted as the contribution introduced by the m-th wave reflection. That method however is not very robust as one can experimentally observe that it diverges for some wavenumbers [99], possibly the ones that are close to a local resonance between any two objects.

An important question, at the heart of the multiple obstacles scattering algorithm, is how the subproblems are solved. Indeed, since each domain in a covering is actually larger than the original domain, it makes little sense in practice to solve the auxiliary problems with the same method on the same discretization. However, these problems have a much simpler structure than the full problem as they are single scattering problems that could be amenable to more efficient solvers. Several techniques, including Fourier series decomposition, integral equations or asymptotic methods, in addition to a PDE based approach, are investigated in [8]. In Chapter 4 we will describe an improvement of the algorithm by giving an example of the use of a fast single scattering solver for the application of the transfer operators $\mathcal{G}_{sq}$ in problem (2.34).

2.6 Extension to Maxwell's equations

We have seen earlier that, in the framework of multi-domain methods, the optimized Schwarz algorithm can be formulated in terms of unknown interface fields and transfer operators. Consequently, by suitably redefining these operators, one can obtain an algorithm for the solution of electromagnetic propagation problems. We present this reformulation in this section, and we also review some local approximations of the magnetic-to-electric (MtE) map, which is the vector counterpart of the DtN map in the scalar case.

2.6.1 Problem setting and non-overlapping optimized Schwarz DDM for Maxwell

Let K be a bounded scatterer in $\mathbb{R}^3$ with smooth closed boundary Γ. The associated unbounded domain of propagation is denoted by $\Omega := \mathbb{R}^3 \backslash \overline{K}$. The exterior electromagnetic scattering problem by a perfectly conducting body K is given by

$$\begin{cases} \textbf{curl curl}\, \boldsymbol{E} - k^2 \boldsymbol{E} = \boldsymbol{0}, & \text{in } \Omega, \\ \gamma_T(\boldsymbol{E}) = -\gamma_T(\boldsymbol{E}^{\text{inc}}), & \text{on } \Gamma, \\ \lim_{r\to\infty} r\left(\boldsymbol{E} - \frac{\imath}{k}\hat{\mathbf{x}} \times \textbf{curl}\, \boldsymbol{E}\right) = \boldsymbol{0}. \end{cases} \tag{2.39}$$

In the above equations, $\boldsymbol{E}$ denotes the scattered electric field. The curl operator is defined by $\mathbf{curl}\,\mathbf{a} := \nabla \times \mathbf{a}$, for a complex-valued vector field $\mathbf{a} \in \mathbb{C}^3$. The nabla operator is $\nabla :=^t (\partial_{x_1}, \partial_{x_2}, \partial_{x_3})$, where $\mathbf{x} =^t (x_1, x_2, x_3) \in \mathbb{R}^3$. The notation $\mathbf{a} \times \mathbf{b}$ designates the cross product and $\mathbf{a} \cdot \overline{\mathbf{b}}$ the inner product between two vectors $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{C}^3$, where $\overline{z}$ is the complex conjugate of $z \in \mathbb{C}$. The associated norm is $||\mathbf{a}|| := \sqrt{\mathbf{a} \cdot \overline{\mathbf{a}}}$. Vector $\mathbf{n}$ is the unit outwardly directed normal to Ω and $\boldsymbol{E}^{\text{inc}}$ defines a given incident electric field. Let us consider a general domain $\mathcal{D}$ with boundary $\partial\mathcal{D}$, $\mathbf{n}$ the outwardly directed unit vector to $\mathcal{D}$, then the tangential traces applications are defined by

$$\gamma_t : \mathbf{v} \mapsto \mathbf{v}_t := \mathbf{n} \times \mathbf{v}_{|\partial\mathcal{D}} \quad \text{and} \quad \gamma_T : \mathbf{v} \mapsto \mathbf{v}_T := \mathbf{n} \times (\mathbf{v}_{|\partial\mathcal{D}} \times \mathbf{n}). \tag{2.40}$$

Let us now write $\mathbf{x} = r\hat{\mathbf{x}} \in \mathbb{R}^3$, where $r := ||\mathbf{x}||$ is the radial distance to the origin and $\hat{\mathbf{x}}$ is the directional vector of the unit sphere $\mathbb{S}_1$. Then, the last equation of system (2.39), which is the so-called Silver-Müller radiation condition at infinity, provides the uniqueness of the solution to the scattering boundary-value problem (2.39).

To numerically solve (2.39) by a volume discretization method, it is standard to truncate the exterior domain of propagation by using a fictitious boundary Γ^∞ surrounding Ω. As a result, we have to solve the following problem in a bounded domain Ω, with boundaries Γ and Γ_∞,

$$\begin{cases} \mathbf{curl}\,\mathbf{curl}\,\boldsymbol{E} - k^2\boldsymbol{E} = \boldsymbol{0}, & \text{in } \Omega, \\ \gamma_T(\boldsymbol{E}) = -\gamma_T(\boldsymbol{E}^{\text{inc}}), & \text{on } \Gamma, \\ \mathcal{B}(\gamma_T(\boldsymbol{E})) - \dfrac{\imath}{k}\gamma_t(\mathbf{curl}\,\boldsymbol{E}) = \boldsymbol{0}, & \text{on } \Gamma^\infty. \end{cases} \tag{2.41}$$

The operator $\mathcal{B}$ can be exact, resulting then in a transparent boundary condition that avoids any spurious unphysical reflection. However, such a boundary condition is global since it is defined by a nonlocal boundary integral operator on Γ_∞ (i.e. the MtE operator $\Lambda : \gamma_T(\boldsymbol{E}) \mapsto \Lambda(\gamma_T(\boldsymbol{E})) = \gamma_t(\mathbf{curl}\,\boldsymbol{E})$). This generates a dense part in the global discretization matrix that must be solved at the end of the computational process. For reducing the cost of computation, a local Absorbing Boundary Condition (ABC) is generally preferred, which means that the operator $\mathcal{B}$ is in fact an approximation of Λ. Since the aim of this paper is not devoted to ABCs, we restrict ourselves to the simplest ABC: $\mathcal{B} = \mathbf{I}$ ($\mathbf{I}$ is the surface identity operator). This corresponds to the well-known Silver-Müller ABC at finite distance.

Let us now focus on the construction of optimized Schwarz Domain Decomposition Methods (DDM) without overlap [1, 31, 52, 54–56, 65, 90, 129, 130, 144, 145, 154] for the approximate boundary-value problem (2.41). The first step of the method [52, 54] consists in splitting Ω into several non-overlapping subdomains Ω_i, $i = 1, \ldots, N_{\text{dom}}$, as described in Section 2.3.1.

In a second step, we solve smaller size problems on each subdomain Ω_i by an iterative process (indexed by (p)) and using transmission boundary conditions

(defined by an operator $\mathcal{S}$ below): we compute $\boldsymbol{E}_i^{(p+1)}$, $1 \le i \le N_{\text{dom}}$, from $\boldsymbol{E}_j^{(p)}$, $1 \le j \ne i \le N_{\text{dom}}$, by

$$\left\{\begin{array}{ll} \textbf{curl curl}\, \boldsymbol{E}_i^{(p+1)} - k^2\, \boldsymbol{E}_i^{(p+1)} = \boldsymbol{0}, & \text{in } \Omega_i, \\ \gamma_T(\boldsymbol{E}_i^{(p+1)}) = -\gamma_T(\boldsymbol{E}_i^{\text{inc}}), & \text{on } \Gamma_i, \\ \gamma_T(\boldsymbol{E}_i^{(p+1)}) - \dfrac{\imath}{k}\gamma_t(\textbf{curl}\, \boldsymbol{E}_i^{(p+1)}) = \boldsymbol{0}, & \text{on } \Gamma_i^{\infty}, \\ \mathcal{S}(\gamma_T(\boldsymbol{E}_i^{(p+1)})) - \dfrac{\imath}{k}\gamma_t(\textbf{curl}\, \boldsymbol{E}_i^{(p+1)}) = \mathcal{S}(\gamma_T(\boldsymbol{E}_j^{(p)})) + \dfrac{\imath}{k}\gamma_t(\textbf{curl}\, \boldsymbol{E}_j^{(p)}) := \boldsymbol{g}_{ij}^{(p)} & \text{on } \Sigma_{ij}, \end{array}\right. \tag{2.42}$$

and then form the quantities $\boldsymbol{g}_{ij}^{(p)}$ through

$$\boldsymbol{g}_{ij}^{(p+1)} = \mathcal{S}(\gamma_T(\boldsymbol{E}_j^{(p+1)})) + \frac{\imath}{k}\gamma_t(\textbf{curl}\, \boldsymbol{E}_j^{(p+1)}) = -\boldsymbol{g}_{ji}^{(p)} + 2\mathcal{S}(\gamma_T(\boldsymbol{E}_j^{(p+1)})) \quad \text{on } \Sigma_{ij}, \tag{2.43}$$

where $\boldsymbol{E}_i = \boldsymbol{E}_{|\Omega_i}$, $\mathbf{n}_i$ (resp. $\mathbf{n}_j$) is the outward unit normal to Ω_i (resp. Ω_j) , $i, j = 1, \ldots, N_{\text{dom}}$, $\Gamma_i = \partial\Omega_i \cap \Gamma$, $\Gamma_i^{\infty} = \partial\Omega_i \cap \Gamma^{\infty}$ and $\mathcal{S}$ is an inversible transmission operator through the interfaces Σ_{ij}. Let us remark that the boundary condition on Γ_i (resp. Γ_i^{∞}) does not take place if the interior of $\partial\Omega_i \cap \Gamma$ (resp. $\partial\Omega_i \cap \Gamma^{\infty}$) is the empty set.

Solving at each step all the local transmission problems through (2.42)-(2.43) may be rewritten as one application of the iteration operator

$$\mathcal{A} : \times_{i,j=1}^{N_{\text{dom}}} (L^2(\Sigma_{ij}))^3 \mapsto \times_{i,j=1}^{N_{\text{dom}}} (L^2(\Sigma_{ij}))^3$$

defined by:

$$\boldsymbol{g}^{(p+1)} = \mathcal{A}\boldsymbol{g}^{(p)} + \mathbf{b}, \tag{2.44}$$

where $\boldsymbol{g}^p$ is the set of boundary data $(\boldsymbol{g}_{ij}^p)_{1 \le i,j \le N_{dom}}$, and $\mathbf{b}$ is given by the incident wave field boundary data. Therefore, (2.42)-(2.43) can be interpreted as an iteration step of the Jacobi fixed point iteration method applied to the linear system

$$(\mathcal{I} - \mathcal{A})\boldsymbol{g} = \mathbf{b}, \tag{2.45}$$

where $\mathcal{I}$ is the identity matrix of size $N_{\text{dom}}^2 \times N_{\text{dom}}^2$. A consequence is that any Krylov subspace iterative solver suitable for general systems could be used for solving this equation. This can significantly improve the convergence rate of the method most particularly if $\mathcal{S}$ is well-chosen.

2.6.2 Optimized transmission boundary conditions

The convergence of the domain decomposition algorithm is fundamentally related to the choice of the operator $\mathcal{S}$. For the time-harmonic Maxwell's equations, the first converging iterative algorithm has been proposed by Després in [54] where a simple impedance boundary operator is proposed

$$\mathcal{S}^0 = \mathbf{I}. \tag{2.46}$$

In the sequel, the corresponding zeroth-order Impedance Boundary Condition (IBC) is designated by IBC(0). A convergence analysis of the DDM method for this boundary condition and for two half-spaces of $\mathbb{R}^3$ has been developed in [55, 65]. The approach, based on Fourier transforms, shows that the algorithm converges only for the propagating modes. For the evanescent modes, the corresponding radius of convergence is equal to 1, which makes the method stagnates or diverges. To improve the convergence factor for these special modes, Alonso *et al.* [1] derive an optimized impedance boundary condition by using a Fourier frequency decomposition. They adapt the technique developed by Gander in [90] for the Helmholz equation to get a zero order optimized impedance boundary condition called here GIBC(α) (GIBC means Generalized Impedance Boundary Condition). For the Maxwell's equation, the GIBC(α) impedance operator writes down

$$\mathcal{S}^{\alpha} = \alpha(\mathbf{I} - \frac{1}{k^2}\mathbf{curl}_{\Sigma_{ij}}\mathrm{curl}_{\Sigma_{ij}}), \tag{2.47}$$

where α is judiciously chosen thanks to an optimization process (see [27]). The same condition is proposed in [55] for the first-order system of Maxwell's equations. In [145], Peng *et al.* show that the DDM converges for a well-chosen complex-valued number α and a decomposition into two half-spaces but by considering both the TE (Transverse Electric) and TM (Transverse Magnetic) modes. The improvement of the rate of convergence for the evanescent modes is obtained at the price of the deterioration of the rate of convergence for the propagative modes. To improve this last transmission boundary condition for the two families of modes, Rawat and Lee [154] introduce the following optimized transmission boundary condition by using two second-order operators

$$\mathcal{S}^{\alpha,\beta} = (\mathbf{I} + \frac{\alpha}{k^2}\nabla_{\Sigma_{ij}}\mathrm{div}_{\Sigma_{ij}})^{-1}(\mathbf{I} - \frac{\beta}{k^2}\mathbf{curl}_{\Sigma_{ij}}\mathrm{curl}_{\Sigma_{ij}}), \tag{2.48}$$

where α and β are chosen so that an optimal convergence rate is obtained for the (TE) and (TM) modes. We denote this boundary condition by GIBC(α,β) in the sequel of the paper. In the half-space case, we refer to [154] for the expression of α and β. Similar boundary conditions are derived in [55] for the first-order Maxwell's equations. Recently, in [56], the authors proved that the convergence rates and the optimization processes for the first- and second-order formulations are finally the same.

When developing optimized DDMs in [31], the authors use highly accurate square-root/Padé-type On-Surface Radiation Conditions (OSRCs) [3–5, 16, 66, 121, 136, 158] as transmission boundary conditions, which are also GIBCs. While being easy-to-use and direct to implement in a finite element environment, these GIBCs lead to the construction of fast converging non-overlapping DDMs, most particularly when computing the solution to high-frequency three-dimensional acoustics scattering problems. In [66], the extension of this high-order OSRC has been developed for the three-dimensional first-order system of Maxwell's equations. When coming back to the second-order formulation, the corresponding

square-root GIBC (that we denote by GIBC(sq, ε)) for the DDM can be written as

$$\begin{aligned} &\mathcal{S}^{\mathrm{sq},\varepsilon} = \boldsymbol{\Lambda}_{1,\varepsilon}^{-1}\boldsymbol{\Lambda}_{2,\varepsilon}, \quad \boldsymbol{\Lambda}_{1,\varepsilon} = (\mathbf{I} + \nabla_{\Sigma_{ij}} \frac{1}{k_\varepsilon^2} \mathrm{div}_{\Sigma_{ij}} - \mathbf{curl}_{\Sigma_{ij}} \frac{1}{k_\varepsilon^2} \mathrm{curl}_{\Sigma_{ij}})^{1/2}, \\ &\boldsymbol{\Lambda}_{2,\varepsilon} = \mathbf{I} - \mathbf{curl}_{\Sigma_{ij}} \frac{1}{k_\varepsilon^2} \mathrm{curl}_{\Sigma_{ij}}, \end{aligned} \tag{2.49}$$

where the complex wavenumber k_ε is defined by: $k_\varepsilon = k + \imath\varepsilon$, with the optimal parameter $\varepsilon = 0.39k^{1/3}\mathcal{H}^{2/3}$. In the previous expression, $\mathcal{H}$ is the local mean curvature at the surface. Finally, $A^{1/2}$ stands for the square-root of the operator A, where the square-root of a complex-valued number z is taken with branch-cut along the negative real axis.

The construction of this GIBC is realized in three steps [66]:

1) the half-space case is considered and the construction of the DtN operator is realized by Fourier analysis,

2) the extension to a sphere $\mathbb{S}_R$ (of radius $R > 0$) is made by considering the local tangent plane approximation of the DtN map to a spherical surface and a regularization procedure of a square-root operator with optimal damping parameter ε for $\mathbb{S}_R$,

3) and finally the approximation (2.49) of the MtE operator for a three-dimensional general convex-shaped smooth surface Γ (:=Σ_{ij} in the DDM context) is obtained by considering the local osculating sphere.

A more adapted form of the square-root GIBC defined by (2.49) is given by

$$\begin{aligned} &\boldsymbol{\Lambda}_{2,\varepsilon}(\gamma_T(\boldsymbol{E}_i^{p+1})) - \frac{\imath}{k}\boldsymbol{\Lambda}_{1,\varepsilon}\gamma_t(\mathbf{curl}\,\boldsymbol{E}_i^{p+1}) \\ &\quad = \boldsymbol{\Lambda}_{2,\varepsilon}(\gamma_T(\boldsymbol{E}_j^{p})) + \frac{\imath}{k}\boldsymbol{\Lambda}_{1,\varepsilon}\gamma_t(\mathbf{curl}\,\boldsymbol{E}_j^{p}) \end{aligned} \quad \text{on } \Gamma = \Sigma_{ij}. \tag{2.50}$$

The IBC (2.46) and the GIBCs (2.47)-(2.48) are defined by local surface operators. In contrast, the GIBC given by (2.49)-(2.50) is nonlocal because of the presence of the square-root operator.

If we set $k_\varepsilon = \alpha^{-1/2}k = -\beta^{-1/2}k$, let us remark that the Rawat-Lee condition GIBC($\alpha, -\alpha$) with the operator (2.48) can be seen as GIBC(sq, ε) where $\boldsymbol{\Lambda}_{1,\varepsilon}$ is approximated by

$$\begin{aligned} \boldsymbol{\Lambda}_{1,\varepsilon} &= (\mathbf{I} + \nabla_{\Sigma_{ij}} \frac{1}{k_\varepsilon^2} \mathrm{div}_{\Sigma_{ij}} - \mathbf{curl}_{\Sigma_{ij}} \frac{1}{k_\varepsilon^2} \mathrm{curl}_{\Sigma_{ij}})^{1/2} \\ &\approx (\mathbf{I} + \nabla_{\Sigma_{ij}} \frac{1}{k_\varepsilon^2} \mathrm{div}_{\Sigma_{ij}})^{1/2} \\ &\approx \mathbf{I} + \nabla_{\Sigma_{ij}} \frac{1}{k_\varepsilon^2} \mathrm{div}_{\Sigma_{ij}}, \end{aligned} \tag{2.51}$$

which corresponds to a first-order Taylor expansion of the square-root operator. We will see in what follows that a high-order complex-valued Padé approximation of the full nonlocal operator (2.50) can be used to get a local representation that is well-suited for a numerical approximation based on finite element methods and leads to quasi-optimal convergence of the DDM.

2.6.3 Localization of the square-root GIBC

The square-root transmission boundary condition, given by (2.49)-(2.50), is nonlocal since it is defined by the pseudodifferential operator $\mathbf{\Lambda}_{1,\varepsilon}$

$$\mathbf{\Lambda}_{1,\varepsilon} := (\mathbf{I}+\mathcal{T})^{1/2}, \tag{2.52}$$

setting

$$\mathcal{T} := \nabla_\Gamma \frac{1}{k_\varepsilon^2}\mathrm{div}_\Gamma - \mathbf{curl}_\Gamma \frac{1}{k_\varepsilon^2}\mathrm{curl}_\Gamma. \tag{2.53}$$

Such an operator is impracticable in a finite element context since it generates a full matrix part associated with the transmitting boundary. A standard way [30, 66, 133] to localize it consists in using rational approximations. Here, we use the approach previously introduced in [66] in the framework of OSRC methods. We introduce the rational Padé approximation of order N_p of the square-root function [133] with a rotation of the branch-cut

$$(1+z)^{1/2} \approx e^{\imath\frac{\theta_p}{2}} R_{N_p}((1+z)e^{-\imath\theta_p}-1) = C_0 + \sum_{\ell=1}^{N_p} \frac{A_\ell z}{1+B_\ell z} = R_0 - \sum_{\ell=1}^{N_p} \frac{A_\ell}{B_\ell(1+B_\ell z)}, \tag{2.54}$$

where R_{N_p} is the standard real-valued Padé approximation of order N_p

$$(1+z)^{1/2} \approx R_{N_p}(z) = 1 + \sum_{\ell=1}^{N_p} \frac{a_\ell z}{1+b_\ell z}, \tag{2.55}$$

and

$$a_\ell = \frac{2}{2N_p+1}\sin^2(\frac{\ell\pi}{2N_p+1}), \quad b_\ell = \cos^2(\frac{\ell\pi}{2N_p+1}).$$

The angle of rotation θ_p is a free parameter that is fixed for the numerical simulations and

$$\begin{aligned} C_0 &= e^{\imath\frac{\theta_p}{2}} R_{N_p}(e^{-\imath\theta_p}-1), & A_\ell &= \frac{e^{-\imath\frac{\theta_p}{2}} a_\ell}{(1+b_\ell(e^{-\imath\theta_p}-1))^2}, \\ B_\ell &= \frac{e^{-\imath\theta_p} b_\ell}{1+b_\ell(e^{-\imath\theta_p}-1)}, & R_0 &= C_0 + \sum_{\ell=1}^{N_p} \frac{A_\ell}{B_\ell}. \end{aligned} \tag{2.56}$$

If one formally considers that $z=\mathcal{T}$, $\mathbf{\Lambda}_{1,\varepsilon}$ can be approximated by

$$\mathbf{\Lambda}_{1,\varepsilon} = (\mathbf{I}+\mathcal{T})^{1/2} \approx \tilde{\mathbf{\Lambda}}_{1,\varepsilon} := (R_0 - \sum_{\ell=1}^{N_p} \frac{A_\ell}{B_\ell}(\mathbf{I}+B_\ell\mathcal{T})^{-1}). \tag{2.57}$$

Now, if we use this approximation, the equation (2.53) and we introduce N_p coupled auxiliary vector fields $\{\boldsymbol{\phi}^\ell\}_{\ell=1,\dots,N_p}$, then we obtain a local and approximate representation of $\boldsymbol{\Lambda}_{1,\varepsilon}\boldsymbol{M}$ through $\tilde{\boldsymbol{\Lambda}}_{1,\varepsilon}\boldsymbol{M}$ defined by

$$\begin{cases} \tilde{\boldsymbol{\Lambda}}_{1,\varepsilon}\boldsymbol{M} = R_0\boldsymbol{M} - \displaystyle\sum_{\ell=1}^{N_p} \frac{A_\ell}{B_\ell}\boldsymbol{\phi}^\ell, \quad \text{on } \Gamma, \\ \boldsymbol{M} - \left(\mathbf{I} + B_\ell\left(\nabla_\Gamma \dfrac{1}{k_\varepsilon^2}\mathrm{div}_\Gamma - \mathbf{curl}_\Gamma \dfrac{1}{k_\varepsilon^2}\mathrm{curl}_\Gamma\right)\right)\boldsymbol{\phi}^\ell = \mathbf{0},\ \ell = 1,\dots,N_p, \quad \text{on } \Gamma. \end{cases} \tag{2.58}$$

More details on this approximation and its properties can be found in [27].

CHAPTER 3

Double sweep preconditioner for Schwarz methods

We mentioned in Section 2.4.4 that Schwarz methods with many subdomains suffer slow convergence, and used an intuitive argument to explain that behavior. We now concentrate on the properties of the iteration operator to find a more mathematically rigorous explanation to this problem. That analysis also highlights that the inverse of the iteration operator can be easily computed in a particular case. Starting from that observation, we then propose a means to speed up convergence and make it independent of the number of subdomains, that takes the form of a preconditioner for the linear system underlying the optimized Schwarz solver. This makes a clear link with the coarse grid techniques introduced in Section 2.4.4. Such a technique is still missing for propagation problems, which makes the material of this chapter a relevant contribution in the field of fast solvers for this kind of problems.

The proposed preconditioner involves sequences of subproblem solves, which makes it intrinsincally non-parallel. While the gain in convergence speed justifies the additional work of the preconditioner in sequential or pipelined (multiple right-hand sides) implementations, we will find that the same strategy can be applied on smaller and independent groups of subdomains, therefore partially restoring the parallelism of the method.

3.1 Matrix representation of the Schwarz operator

As we have done in Section 2.5 for the multiple obstacles scattering algorithm, we now explain how the application of the Schwarz operator $\mathcal{F}$ of system (2.26) on the set of unknown functions g_{ij}, which has been previously described by Algorithm 2.3, can be seen as a matrix-vector product. This representation is interest-

ing because matrices, in the context of PDEs, are usually associated to discretized systems. But so far, the only discretization that we have performed is the decomposition of the domain into subdomains; the problem itself is still written at the continuous level and we have not precised yet how the subproblems are going to be solved. Hence, the entries of the matrix are continuous operators, that we will call transfer operators. They act on (continuous) fields defined on the artificial interfaces and map them to fields defined on other interfaces. Nevertheless, techniques from linear algebra, in particular matrix factorization and inversion, are applicable and will be used.

In the particular case of 1d problems, the transfer operators reduce to scalar values, which enables the spectral study of the iteration operators.

3.1.1 General case

We start with the case of a layered decomposition that does not involve any loop. In the general case, we do not impose any restriction on the actual geometry of the problem other than the decomposition must be topologically equivalent to that of Figure 2.4. We also allow for any velocity distribution in the propagation medium and for the use of imperfect non-reflecting transmission conditions on the artificial interfaces, so that internal reflections may occur inside the subdomains.

We first number the unknowns of the Schwarz algorithm as $g = [g_{12}, g_{21}, g_{23}, \dots]^T$, where an unknown function g_{ij} corresponds to the impedance data of the boundary condition for problem i, on Σ_{ij}. There are two unknowns per artificial interface, for a total of $M = 2(N-1)$ unknowns, which is therefore the size of the (continuous) Schwarz problem.

We introduce the forward and backward transfer operators $\mathcal{B}_i^f$ and $\mathcal{B}_i^b$. They are defined by:

$$\begin{aligned}
&\mathcal{B}_i^f : H^{-1/2}(\Sigma_{i,i-1}) \to H^{-1/2}(\Sigma_{i,i+1}) \\
&\qquad g_{i,i-1} \longmapsto 2\mathcal{S}u_i(g_{i,i-1}, 0)_{|\Sigma_{i,i+1}} = \mathcal{B}_i^f g_{i,i-1}; \\
&\mathcal{B}_i^b : H^{-1/2}(\Sigma_{i,i+1}) \to H^{-1/2}(\Sigma_{i,i-1}) \\
&\qquad g_{i,i+1} \longmapsto 2\mathcal{S}u_i(0, g_{i,i+1})_{|\Sigma_{i,i-1}} = \mathcal{B}_i^b g_{i,i+1},
\end{aligned} \tag{3.1}$$

where $u_i(g_l, g_r)_{|\Sigma}$ refers to the restriction on boundary Σ of the solution to the subproblem $\mathcal{H}_i u_i = f_i$ defined by (2.22), with the simplified notation g_l, g_r respectively corresponding to $g_{i,i-1}, g_{i,i+1}$. These operators involve the solution of subproblems with an impedance source on one side only.

We will also need the self-coupling operators $\mathcal{E}_i^f$ and $\mathcal{E}_i^b$ defined from an interface to itself:

$$\begin{aligned}
&\mathcal{E}_i^f : H^{-1/2}(\Sigma_{i,i-1}) \to H^{-1/2}(\Sigma_{i,i-1}) \\
&\qquad g_{i,i-1} \longmapsto 2\mathcal{S}u_i(g_{i,i-1},0)_{|\Sigma_{i,i-1}} = \mathcal{E}_i^f g_{i,i-1}; \\
&\mathcal{E}_i^b : H^{-1/2}(\Sigma_{i,i+1}) \to H^{-1/2}(\Sigma_{i,i+1}) \\
&\qquad g_{i,i+1} \longmapsto 2\mathcal{S}u_i(0,g_{i,i+1})_{|\Sigma_{i,i+1}} = \mathcal{E}_i^b g_{i,i+1}.
\end{aligned} \tag{3.2}$$

These operators correspond to the contribution on an interface of the part of a wave that travels through the domain and that is reflected back to its interface of origin. With these definitions, the matrix corresponding to the Schwarz operator writes:

$$\mathcal{F}(N) = \left[\begin{array}{cc|cc|cc}
\mathcal{I} & \mathcal{E}_2^f & \mathcal{B}_2^b & & & \\
\mathcal{E}_1^b & \mathcal{I} & 0 & 0 & & \\
\hline
0 & 0 & \mathcal{I} & \mathcal{E}_3^f & \ddots & \\
 & \mathcal{B}_2^f & \mathcal{E}_2^b & \mathcal{I} & & \\
\hline
 & & \ddots & & \ddots & \mathcal{B}_{N-1}^b \\
 & & & & 0 & 0 \\
\hline
 & & & & 0 \quad 0 & \mathcal{I} \quad \mathcal{E}_N^f \\
 & & & & \mathcal{B}_{N-1}^f & \mathcal{E}_{N-1}^b \quad \mathcal{I}
\end{array}\right], \tag{3.3}$$

with $\mathcal{I}$ the identity operator. Horizontal and vertical lines have been added to emphasize the layered structure of the matrix: for each additional domain, two lines and two columns are added, conserving the same pattern. Therefore, that matrix can easily be constructed for an arbitrary number of subdomains; its structure directly follows from the update relation (2.27).

It is a banded matrix, with a bandwidth of 5, although at most 3 elements are non-zero on a given line or column. This structure illustrates well the local couplings between subdomains. Note that for a layered decomposition, the same structure is kept in any dimension, as each subdomain (except the first and last ones) has only 2 neighbors.

If we now consider discretized interface functions g_{ij}, we can still obtain a matrix expression of the iteration operator, with the same structure as (3.3). The major difference is that the transfer and self-coupling operators $\mathcal{B}_i^{f,b}$ and $\mathcal{E}_i^{f,b}$ are replaced by dense numerical blocks $B_i^{f,b}$ and $E_i^{f,b}$, that can be seen as the Green functions for the problems with impedance data, evaluated on the opposite interface. These blocks are very expensive to form, and is never done in practice. But if

the full numerical matrix was available, inverting it would solve the Schwarz problem. (We will see that this inverse can be expected to be quite dense, making its storage, factorization and application in view of using it as a preconditioner rather expensive, however.) The next section explores that idea by noticing that the matrix with transfer operators can be inverted if one neglects the internal reflections, and that its application involves the same kind of subproblem solves as for the iteration operator.

3.1.2 Simplified case

We now consider a simplified case with similar decomposition, but where we suppose that the shape of the domain and the use of ideal non-reflecting conditions guarantee that no internal reflection can take place. In that case, we show in Appendix A that the interfaces do not couple with themselves and that the $\mathcal{E}^{f,b}$ operators vanish. That matrix, denoted by $\mathcal{F}_A$, writes:

$$\mathcal{F}_A(N) = \begin{bmatrix} \mathcal{I} & \mathcal{B}_2^b & & & \\ & \mathcal{I} & & & \\ & & \mathcal{I} & \ddots & \\ & \mathcal{B}_2^f & & \mathcal{I} & \\ & & \ddots & \ddots & \mathcal{B}_{N-1}^b \\ & & & & \mathcal{I} \\ & & & \mathcal{B}_{N-1}^f & \mathcal{I} \end{bmatrix}, \tag{3.4}$$

Compared to (3.3), the matrix has a more simple structure since half of its off-diagonal elements have disappeared. In particular, notice that the first and last columns of that matrix are only made of an identity operator, which will be exploited in the next Section to easily obtain an inverse of it.

3.1.3 Analysis of the 1d case

In 1d, the unknown functions g_{ij} and the transfer operators reduce to scalar values, so the procedure of forming the numerical matrix is fast. The subproblems can be solved analytically and the exact expression of the DtN map is known, so we are in a situation where the optimized Schwarz algorithm is truly optimal. To illustrate this, we first focus on a problem with Sommerfeld radiation conditions on the left and right hand sides of the domain $\Omega = [x_l, x_r]$ and a non-zero volume source inside the domain:

$$\begin{aligned} -(\partial_{xx} + k^2)u &= f \quad \text{in } \Omega, \\ (\partial_n - \imath k)u &= 0 \quad \text{on } \{x_l, x_r\}. \end{aligned} \tag{3.5}$$

The Schwarz iteration operator with ideal transmission conditions and constant medium is in the conditions of the simplified matrix (3.4) and therefore has the general expression for N subdomains:

$$F_A(N) = \begin{bmatrix} 1 & 0 & b_2 & & & & \\ 0 & 1 & 0 & & & & \\ & 0 & 1 & 0 & \ddots & & \\ & b_2 & 0 & 1 & & & \\ & & \ddots & & \ddots & b_{N-1} & \\ & & & & & 0 & \\ & & & & 0 & 1 & 0 \\ & & & & b_{N-1} & 0 & 1 \end{bmatrix}. \tag{3.6}$$

Looking at the spectrum of this matrix, one can easily notice that it has only one eigenvalue $\lambda_{1-M} = 1$, with algebraic multiplicity $M = 2(N-1)$ (the number of unknowns on the artificial interfaces) and geometric multiplicity 2: it has only 2 linearly independent eigenvectors, that are the first and last canonical basis vectors. These eigenvectors correspond to sources located on interfaces on the first and last domains. They produce waves in these domains that leave them via the exact radiation conditions on the external boundaries, hence not creating coupling with any other interface. They are thus the only possible invariants of the system, as sources located elsewhere will generate couplings.

Matrices with the property of not having a full basis of eigenvectors are called defective [171]. Although the inverse of the matrix exists (it has full rank, and we will see in Section 3.2 that the inverse is particularly easy to find), this property is known to cause slow convergence of Krylov solvers [39, 185]. This complements our intuitive understanding that the slow convergence of the algorithm is due to local exchange of information, which can also be inferred from the structure of the matrix with at most two non-zero elements per row, near the diagonal. This is also a counterexample of the widespread idea that a good clustering of the eigenvalues is a sufficient condition for the fast convergence of Krylov solvers: this matrix has perfectly clustered eigenvalues and its condition number is small (see Appendix A), yet its convergence is very slow because it is defective.

We now consider numerical solutions (e.g. by the finite element method) of the PDEs (3.5) in the subdomains. We observe that some entries that used to vanish take a small value, hereunder denoted by ϵ (we use the same notation for all of them, though their values may differ):

$$F_N(N) = \begin{bmatrix} 1 & \epsilon & b_2 & & & & \\ \epsilon & 1 & 0 & & & & \\ & 0 & 1 & \epsilon & \ddots & & \\ & b_2 & \epsilon & 1 & & & \\ & & \ddots & & \ddots & b_{N-1} & \\ & & & & & 0 & \\ & & & & 0 & 1 & \epsilon \\ & & & & b_{N-1} & \epsilon & 1 \end{bmatrix}. \tag{3.7}$$

The parameters b_i are also slightly affected. This is the consequence of the numerical dispersion (also known as the pollution effect) that arises upon discretization of the Helmholtz equation (see Section 1.3.5), as it causes inaccuracy of the absorbing boundary conditions. In other words, our choice of operator $\mathcal{S} = -\imath k$ no longer matches the exact DtN map for the discretized problem, which causes a partial reflection of outgoing waves.

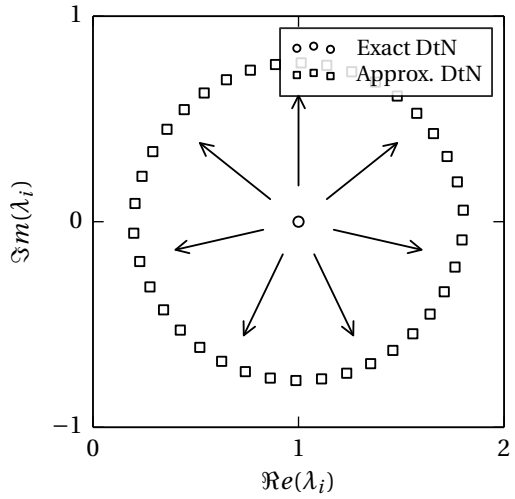


Figure 3.1: The eigenvalues of the iteration operator with exact DtN collapse to 1, as opposed to the case of imperfect impedance conditions where the eigenvalues tend to spread out over the complex plane, in a circle centered in $(1,0)$. Despite the perfect conditioning of the system with the exact DtN, the algorithm does not converge faster, because the corresponding operator proves to be defective.

We observe that the eigenvalues are all distinct, with algebraic multiplicity 1, and tend to spread in the complex plane, in a circular pattern around $(1,0)$ (Figure 3.1). For 3 subdomains and a decomposition in equally sized subdomains, we have $b_i = b, \forall i$, and the eigenvalues are:

$$\begin{aligned} \lambda_{1,2,3,4} &= 1 \pm \sqrt{\epsilon^2 \pm \epsilon b} \\ &\approx 1 \pm \sqrt{\pm \epsilon b_2} \end{aligned}$$

For more domains, the expression becomes too complicated to be reproduced here, yet they can still be seen as perturbations of 1: $\lambda_i = 1 + r_i$, where r_i is a complex number with approximately constant modulus $|r_i| \approx R < 1$ (if we suppose $\epsilon \ll b_i$; small deviations appear for large N) that corresponds to the radius of the circle. This radius tends to increase with the number of subdomains, or when a coarser discretization is used (further degrading the accuracy of the impedance condition). It asymptotically reaches 1 (with the consequence that some eigenvalues are close to 0), which strongly degrades the conditioning of the operator. Note that, as an effect of the cascaded square roots in the expression of the eigenvalues, even very small values of ϵ have a strong effect on the spectral radius.

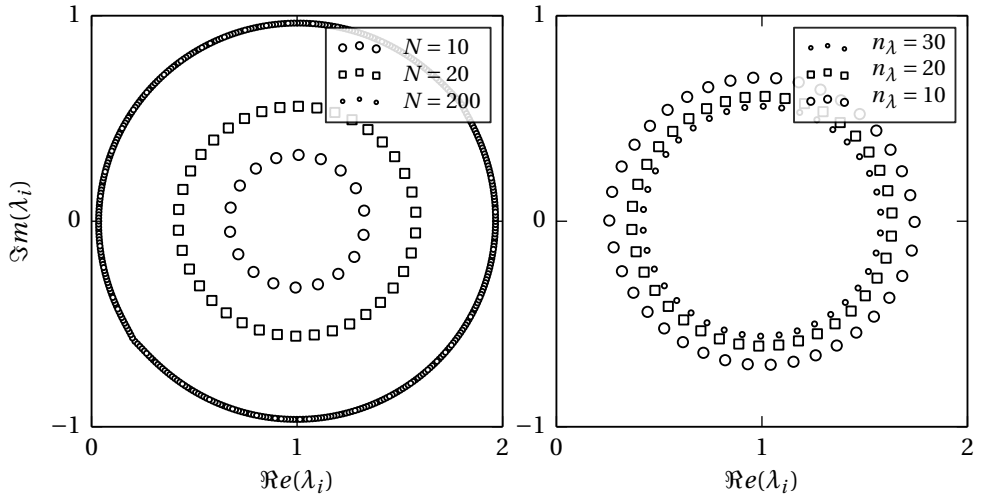


Figure 3.2: Influence of the number of subdomains N and number of discretization points per wavelength n_λ on the eigenvalues distribution of the matrix of the operator in 1d, at $k = 20\pi$. Left: increasing N with same discretization $n_\lambda = 30$; right: coarser discretization (constant $N = 20$). The spectral radius and the condition number increase whenever more subdomains or a coarser grid are used. In extreme cases (very large N), some eigenvalues asymptotically approach 0. There are $M = 2(N-1)$ eigenvalues.

The associated eigenvectors are distinct, but still resemble each other. So the operator $\mathcal{F}_N$ is not strictly defective as it was the case with the exact DtN, but can still be considered as almost defective. Numerical experiments show very similar convergence behaviours in both cases.

3.1.4 Cyclic decompositions

With cyclic decompositions as in Figure 2.5, the matrix of the Schwarz operator is modified compared to the layered decomposition. The first difference is in the size of the vector of unknowns: for N subdomains, there are $2N$ functions defined on N interfaces, instead of $2(N-1)$, because the first and last domains now have a common boundary with 2 associated unknowns. We arrange the unknown vector

as $g = [g_{12}, g_{21}, \ldots, g_{N1}, g_{1N}]$ and again consider a case where no internal reflection can occur, so that the self-coupling operators vanish. If we define the forward direction as a clockwise propagation, the operator matrix now writes:

$$\mathcal{F}_A(N) = \begin{bmatrix} \mathcal{I} & \mathcal{B}_2^b & & & \\ & \mathcal{I} & & & \mathcal{B}_1^f \\ & & \mathcal{I} & \ddots & \\ & \mathcal{B}_2^f & & \mathcal{I} & \\ & & \ddots & & \mathcal{B}_N^b \\ \mathcal{B}_1^b & & & & \mathcal{I} \\ & & & \mathcal{B}_N^f & \mathcal{I} \end{bmatrix}. \tag{3.8}$$

Four new transfer operators have appeared: $\mathcal{B}_1^{\{f,b\}}$ and $\mathcal{B}_N^{\{f,b\}}$ that did not exist in the layered case, and the matrix has become circulant as a consequence of the additional couplings generated by the loop: it has lost its banded structure, with two extra-diagonal elements, and there is no trivial line in the system anymore, since no wave can escape the domain without creating coupling with another interface.

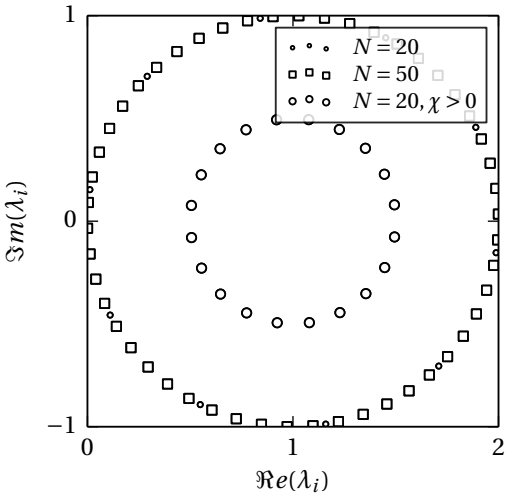


Figure 3.3: A cyclic decomposition of a 1d problem in a non-dissipative medium ($\chi = 0$) causes strong scattering of the eigenvalues with a spectral radius equal to 1, for any number of domains. The spectral radius is smaller when dissipation is introduced ($\chi > 0$).There are N distinct eigenvalues, with algebraic and geometric multiplicities 2.

As can be expected, this modified structure will complicate the inversion of the matrix (see next section), as well as the interpretation that can be given to it in

terms of sequence of subproblem solves. A spectral analysis similar to the one performed for the layered decomposition (see problem definition (3.5), where Ω is the edge of the unit circle) reveals that the eigenvalues scatter around $(1,0)$ in the complex plane with a spectral radius equal to 1: $\lambda_i = 1 + r_i$ with $|r_i| = 1$, even for a few subdomains and ideal transmission conditions. There are N eigenvalues, each with algebraic multiplicity 2, because the forward part is strictly identical to the backward part. Their geometric multiplicity is also 2, so the operator is not defective in this case, but some eigenvalues are very close to 0, which causes the same kind of slow convergence.

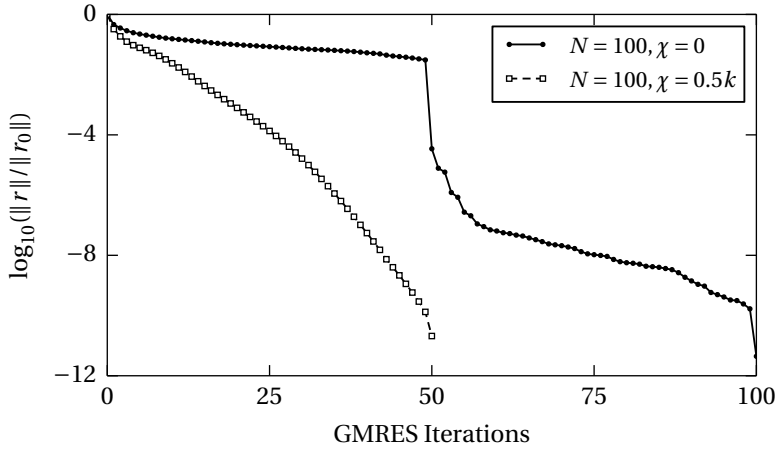


Figure 3.4: Convergence of the GMRES on the 1d circle problem, with and without dissipation in the medium, in the case of many subdomains ($N = 100$). The endless propagation in the undamped case causes strong coupling between all subdomains and leads to slower convergence.

Although this problem seems very simple, it is not surprising that it is hard to solve iteratively, as it is a highly resonant problem: waves are not allowed to leave the domain and propagate indefinitely along the circle, interfering with themselves. The transfer operators therefore have unit modulus: $|\mathcal{B}_i^{\{f,b\}}| = 1$.

Introducing damping in the medium:

$$\partial_{xx} u + (k + \imath\chi)^2 = f \quad \text{in } \Omega, \tag{3.9}$$

with χ a positive real number, yields $|\mathcal{B}_i^{f,b}| < 1$ and leads to a reduced spectral radius, as shown on Figure 3.3. The stronger the dissipation, the smaller the spectral radius and the faster the convergence (Figure 3.4). (It is a well-known property that problems with dissipation are generally easier to solve. It is exploited, for instance, by the shifted Laplace preconditioner, presented in Section 1.4.2.)

Similarly, open 2d problems decomposed in the same fashion but with a non-reflecting condition on part of their boundaries (see e.g. Figure 2.5 for a simple scattering problem) can therefore be expected to converge well because there is less coupling between distant subdomains.

3.2 Inverse operator as preconditioner

We explained in Section 2.2 that an efficient preconditioner should have the following properties: first it should be a good approximation of the inverse of the operator or matrix to be solved, second the costs of preparing the preconditioner and applying it to a vector should be small. In this section, we will describe how a simplified version of the Schwarz operator for a layered decomposition (see Section 3.1.2), that neglects the internal reflections due to inhomogeneities in the medium or interface conditions on physical or artificial boundaries, can be easily inverted and its application to a vector obtained as a simple routine, quite similar to a standard iteration. It is therefore natural to use that inverse of an approximate iteration operator as preconditioner for the original operator.

The procedure for the preconditioner application is identical for any number of subdomains, without any pre-calculation. This already fulfills the first two conditions above. We will see however that this approach, despite its very interesting properties with respect to the number of subdomains and frequency, is quite time-consuming and non-parallel, thereby violating our last requirement. It is also hardly applicable to cyclic decompositions. The next section will bring an answer to that problem, by proposing a modification of the preconditioner routine that partially re-enables the parallelism of the method and make it readily applicable to cyclic decompositions.

3.2.1 Inversion of the simplified Schwarz operator

Looking at the expression of the Schwarz iteration operator (3.4), where we assumed perfectly non-reflecting conditions at the artificial interfaces, two observations can be made: the forward and backward parts are totally independent from each other (it was not the case in the general case (3.3) with self-coupling operators $\mathcal{E}_i^{f,b}$), and both of these parts are triangular, lower and upper respectively. Hence, they can be inverted separately and very easily. Doing so, we obtain the expression for the inverse:

$$\mathcal{F}_A^{-1}(N) = \left[\begin{array}{cc|cc|c|c} I & & -\mathcal{B}_2^b & & \cdots & \mathcal{F}_{1,M-1}^{-1} \\ & I & & & & \\ \hline & & I & & \cdots & \mathcal{F}_{3,M-1}^{-1} \\ & -\mathcal{B}_2^f & & I & & \vdots \\ \hline & \vdots & & \vdots & \ddots & -\mathcal{B}_{N-1}^b \\ \hline & & & & & I \\ \mathcal{F}_{M,2}^{-1} & & \mathcal{F}_{M,4}^{-1} & \cdots & -\mathcal{B}_{N-1}^f & I \end{array}\right], \tag{3.10}$$

with the entries, using the index mappings defined in Appendix A:

$$\mathcal{F}^{-1}_{mn} = \begin{cases} -(-1)^{i(n)+j(m)} \prod_{k=i(n)}^{j(m)} \mathcal{B}^b_k & \text{if } m = 1,3,\ldots \text{ and } m < n; \\ -(-1)^{i(n)+j(m)} \prod_{k=j(m)}^{i(n)} \mathcal{B}^f_k & \text{if } m = 2,4,\ldots \text{ and } m > n; \\ 0 & \text{otherwise.} \end{cases} \tag{3.11}$$

The same transfer operators $\mathcal{B}_i$ appear in the iteration matrix and in its inverse, though the structure of the latter is more complex, with the operators multiplying each other. But we see that, as a consequence of the layered structure of the matrix $\mathcal{F}_A(N)$, a recursion formula can again be found for forming its inverse with increasing number of subdomains.

One might think that forming and applying that inverse matrix will be expensive, but we will see in Section 3.2.3 that the terms of the matrix-vector product with the inverse operator can be rewritten in such a way that the products of operators are fully avoided. This means that, provided that the $\mathcal{B}_i$ operators are available, the matrix-vector products with the iteration operator and with its inverse can be obtained at reasonable cost and without having to form these matrices first. Since the application of $\mathcal{B}_i$ to any vector v amounts to solving the i-th subproblem with v as impedance data on one side and applying operator $\mathcal{S}$ to the restriction of the solution on the other side, we have all the ingredients at hand to build a fully matrix-free algorithm.

3.2.2 Spectrum of the preconditioned operator

In Section 3.1.3, we have identified two sources for the slow convergence of the algorithm: the defectiveness of the operator, and the bad conditioning of the system caused by the small eigenvalues when many subdomains are used, in combination with approximate DtN maps on the artificial interfaces. Figure 3.5 shows the spectrum of the preconditioned iteration operator in the 1d case, i.e. of $F_N F_A^{-1}$. The eigenvalues of the preconditioned system are much more clustered around $(1,0)$ than in the unpreconditioned case. This spectrum resembles the one of the operator with the exact DtN, with the fundamental difference that the eigenvectors are now distinct. So the difficulty with the operator being defective, or close to it, has been removed and the good clustering of the eigenvalues opens the way for fast convergence. The preconditioner produces the same effect with a coarser discretization, yet the eigenvalues are not as well clustered in that case. In relation to this, we will see that the quality of the approximate DtN has a direct impact on the rate of convergence.

3.2.3 Interpretation as the double sweep

Throughout the developments above, we insisted on the recurrence relations in the structure of the iteration operator (3.3) and its approximate inverse (3.10), that

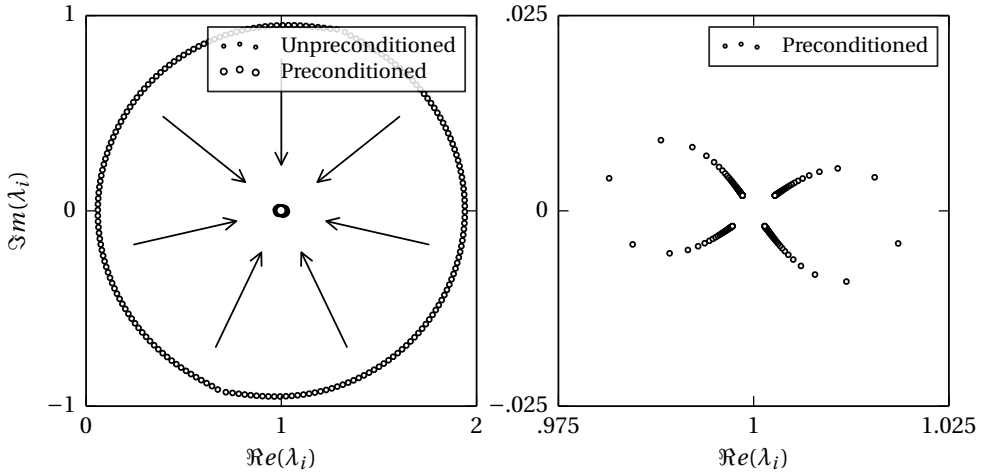


Figure 3.5: The double sweep preconditioner produces an excellent clustering of the eigenvalues of the iteration operator. Left: comparison of the spectrums of the operators ($k = 40\pi, N = 100, n_\lambda = 20$), with and without preconditioner; right: zoom, centered in $(1,0)$, on the eigenvalues of the preconditioned operator. The eigenvalues are again equal or very close to each other, but the preconditioned operator has a full set of distinct eigenvectors, so the convergence is now expected to be fast.

we propose to use as a preconditioner. We now take advantage of that property to write the matrix-vector product with the preconditioner F_A^{-1} in a simplified way and give it an interpretation in terms of a double sweep of subproblems solves. The product $g' = F_A^{-1} r$ (in the context of right-preconditioned Krylov solvers, r denotes the residual), is given component-wise as:

$$\begin{aligned} g'_{i,i-1} &= r_{i,i-1} + \sum_{p=i-1}^{2} (-1)^{i+p} \prod_{q=i-1}^{p} \mathcal{B}_q^f r_{p,p-1}, \quad i = 2,\dots,N; \\ g'_{i,i+1} &= r_{i,i+1} + \sum_{p=i+1}^{N-1} (-1)^{i+p} \prod_{q=i+1}^{p} \mathcal{B}_q^b r_{p,p+1}, \quad i = 1,\dots,N-1. \end{aligned}$$

Factoring these expressions, we rewrite them as:

$$\begin{aligned} g'_{i,i-1} &= r_{i,i-1} - \mathcal{B}_{i-1}^f \Big(r_{i-1,i-2} - \mathcal{B}_{i-2}^f \big(\dots (r_{3,2} - \mathcal{B}_2^f r_{2,1}) \big) \Big), \\ &\qquad i = 2,\dots,N; \\ g'_{i,i+1} &= r_{i,i+1} - \mathcal{B}_{i+1}^b \Big(r_{i+1,i+2} - \mathcal{B}_{i+2}^b \big(\dots (r_{N-2,N-3} - \mathcal{B}_{N-1}^b r_{N-1,N}) \big) \Big), \\ &\qquad i = 1,\dots,N-1, \end{aligned}$$

which finally gives the double recurrence relation:

$$\begin{aligned}
g'_{21} &= r_{21};\\
g'_{i+1,i} &= r_{i+1,i} - \mathcal{B}_i^f g'_{i,i-1}, \quad i = 2,\dots,N-1;\\
g'_{N-1,N} &= r_{N-1,N};\\
g'_{i-1,i} &= r_{i-1,i} - \mathcal{B}_i^b g'_{i,i+1}, \quad i = N-1,\dots,2.
\end{aligned}$$

The first relation describes the forward sequence: we start from the first boundary and propagate the information by solving a problem at each step to move to the next boundary and incorporate the contribution of that boundary. The other relation describes the same procedure in the backward direction; because these sequences are independent of each other, they can be done in parallel. We note that the extreme problems (the first and the N-th) are not solved in any of the sequences, so each of them requires $N-2$ steps.

A sequence of solves over the domain is sometimes called a sweep, hence the name "double sweep" for this procedure. With the recurrence relations above, it becomes natural to implement the preconditioner as a matrix-free operator: each sweep is performed by calling, at every step, the subproblem solve routine and applying the DtN map to its output; the result is then used to start the next step. This gives rise to algorithm 3.1. In practice, the solution of $\mathcal{H}_i u_i = f_i$ is done using the factorization of $\mathcal{H}_i$ that is readily available in the Schwarz algorithm. But one could also envision using an approximate solution if a faster technique is available.

Algorithm 3.1: Application of the double sweep preconditioner $r \leftarrow F_A^{-1} r$

```
// Forward sweep
r_21 ← r_21
for i = 2 : N − 1
    u_D ← 0 on ∂Ω_i ∩ Γ_D
    f_i ← 0
    g_l ← r_{i,i−1}
    g_r ← 0
    Solve H_i u_i = f_i, with data g_l and g_r
    r_{i+1,i} ← r_{i+1,i} + 2S u_i|_{Σ_{i,i+1}}
end
```

```
// Backward sweep
r_{N−1,N} ← r_{N−1,N}
for i = N − 1 : 2
    u_D ← 0 on ∂Ω_i ∩ Γ_D
    f_i ← 0
    g_l ← 0
    g_r ← r_{i,i+1}
    Solve H_i u_i = f_i, with data g_l and g_r
    r_{i−1,i} ← r_{i−1,i} + 2S u_i|_{Σ_{i,i−1}}
end
```

One can further push the idea of the matrix representation of the iteration operator and the preconditioner, by directly considering the product of these matrices as our iteration operator instead of separately applying them. That matrix is computed as a classical matrix-matrix product, and can also be obtained by following a recurrence relation. This is somehow equivalent to defining a new, unpreconditioned algorithm, as it amounts to solve $(F_N F_A^{-1}) g' = b$ with GMRES, with the

matrix-vector product by $(F_N F_A^{-1})$ performed in one single step. In the case of right preconditioning, that operation rewrites well in a matrix-free fashion as a double sequence of subproblems solutions, with the difference that it includes the first and last domains. The advantage of this combined approach is that it is more efficient in terms of number of solves per iteration, by skipping those solves that are redundant in the separate applications of the operators: the integrated version requires $2N-2$ solves, for $3N-4$ solves in the separate version. This results in Algorithm 3.2. (Note that in this case the solution of $\mathcal{H}_i u_i = f_i$ must be computed fully, whereas approximate solves were an option in the separate preconditioner step.)

Computationally, although the sweeps in the algorithm are sequential by nature, parallelism can still be exploited in practice in several ways. Indeed, the most costly step (the factorization) of the direct solutions by sparse LU of all Helmholtz problems is fully parallelizable (the actual solution of each system at each iteration is also parallelizable by itself, but does not scale as well). For problems with multiple excitations (e.g. multiple incidence angle or multiple source types), many (at most $N-2$) sweeps can be pipelined in order to fully utilize each CPU and perfectly balance communications. Finally, partial sweeps can also be envisioned on smaller groups of subdomains, in order to restore some parallel efficiency. This idea will be investigated in Section 3.3.

Algorithm 3.2: Combined application of iteration operator and preconditioner $r \leftarrow FF_A^{-1} r$

```
// Use two auxiliary variables: g^c contains the correction
   to the input data, g^t saves data for use at next
   iteration. Both have same structure as input data r.

// Forward sweep
g^t_{2,1} ← 0
for i = 2 : N
    g_l ← r_{i,i-1} + g^t_{i,i-1}
    g_r ← 0
    Solve H_i u_i = f_i
    g^c_{i-1,i} ← g_l − 2S u_i|_{Σ_{i,i-1}}
    g^t_{i+1,i} ← 2S u_i|_{Σ_{i,i+1}}
end

// Backward sweep
g^t_{N-1,N} ← 0
for i = N − 1 : 1
    g_l ← 0
    g_r ← r_{i,i+1} + g^t_{i,i+1}
    Solve H_i u_i = f_i
    g^c_{i+1,i} ← g_r − 2S u_i|_{Σ_{i,i+1}}
    g^t_{i-1,i} ← 2S u_i|_{Σ_{i,i-1}}
end

// Add correction
r ← r + g^c
```

3.2.4 Inverse operator for cyclic decompositions

In this Section, we attempt to use the same approach as for the layered decomposition in the case of a cyclic decomposition. We will see that, although it remains possible, the expressions of the matrices and the corresponding application routine are more complex. We compute the inverse of matrix (3.8) that is a simplified version with supposedly perfectly non-reflecting transmission operators, in the case of 3 subdomains:

$$
\begin{aligned}
\mathcal{F}_A^{-1} &= \begin{bmatrix}
\frac{1}{1+\mathcal{B}_2^b\mathcal{B}_3^b\mathcal{B}_1^b} & 0 & \frac{-\mathcal{B}_2^b}{1+\mathcal{B}_3^b\mathcal{B}_1^b\mathcal{B}_2^b} & 0 & \frac{\mathcal{B}_2^b\mathcal{B}_3^b}{1+\mathcal{B}_1^b\mathcal{B}_2^b\mathcal{B}_3^b} & 0 \\
0 & \frac{1}{1+\mathcal{B}_1^f\mathcal{B}_3^f\mathcal{B}_2^f} & 0 & \frac{\mathcal{B}_1^f\mathcal{B}_3^f}{1+\mathcal{B}_2^f\mathcal{B}_1^f\mathcal{B}_3^f} & 0 & \frac{-\mathcal{B}_1^f}{1+\mathcal{B}_3^f\mathcal{B}_2^f\mathcal{B}_1^f} \\
\frac{\mathcal{B}_3^b\mathcal{B}_1^b}{1+\mathcal{B}_2^b\mathcal{B}_3^b\mathcal{B}_1^b} & 0 & \frac{1}{1+\mathcal{B}_3^b\mathcal{B}_1^b\mathcal{B}_2^b} & 0 & \frac{-\mathcal{B}_3^b}{1+\mathcal{B}_1^b\mathcal{B}_2^b\mathcal{B}_3^b} & 0 \\
0 & \frac{-\mathcal{B}_2^f}{1+\mathcal{B}_1^f\mathcal{B}_3^f\mathcal{B}_2^f} & 0 & \frac{1}{1+\mathcal{B}_2^f\mathcal{B}_1^f\mathcal{B}_3^f} & 0 & \frac{\mathcal{B}_2^f\mathcal{B}_1^f}{1+\mathcal{B}_3^f\mathcal{B}_2^f\mathcal{B}_1^f} \\
\frac{-\mathcal{B}_1^b}{1+\mathcal{B}_2^b\mathcal{B}_3^b\mathcal{B}_1^b} & 0 & \frac{\mathcal{B}_1^b\mathcal{B}_2^b}{1+\mathcal{B}_3^b\mathcal{B}_1^b\mathcal{B}_2^b} & 0 & \frac{1}{1+\mathcal{B}_1^b\mathcal{B}_2^b\mathcal{B}_3^b} & 0 \\
0 & \frac{\mathcal{B}_3^f\mathcal{B}_2^f}{1+\mathcal{B}_1^f\mathcal{B}_3^f\mathcal{B}_2^f} & 0 & \frac{-\mathcal{B}_3^f}{1+\mathcal{B}_2^f\mathcal{B}_1^f\mathcal{B}_2^f} & 0 & \frac{1}{1+\mathcal{B}_3^f\mathcal{B}_2^f\mathcal{B}_1^f}
\end{bmatrix} \\
&= \begin{bmatrix}
1 & 0 & -\mathcal{B}_2^b & 0 & \mathcal{B}_2^b\mathcal{B}_3^b & 0 \\
0 & 1 & 0 & \mathcal{B}_1^f\mathcal{B}_3^f & 0 & -\mathcal{B}_1^f \\
\mathcal{B}_3^b\mathcal{B}_1^b & 0 & 1 & 0 & -\mathcal{B}_3^b & 0 \\
0 & -\mathcal{B}_2^f & 0 & 1 & 0 & \mathcal{B}_2^f\mathcal{B}_1^f \\
-\mathcal{B}_1^b & 0 & \mathcal{B}_1^b\mathcal{B}_2^b & 0 & 1 & 0 \\
0 & \mathcal{B}_3^f\mathcal{B}_2^f & 0 & -\mathcal{B}_3^f & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\mathcal{D}_1^b & & & & & \\
& \mathcal{D}_2^f & & & & \\
& & \mathcal{D}_2^b & & & \\
& & & \mathcal{D}_3^f & & \\
& & & & \mathcal{D}_3^b & \\
& & & & & \mathcal{D}_1^f
\end{bmatrix}^{-1} \\
&= SD^{-1},
\end{aligned}
\tag{3.12}
$$

with $\mathcal{D}_i^f = 1-(-1)^N\mathcal{B}_k^f\mathcal{B}_j^f\mathcal{B}_i^f$ and $\mathcal{D}_i^b = 1-(-1)^N\mathcal{B}_j^b\mathcal{B}_k^b\mathcal{B}_i^b$, where $\{i,j,k\}$ represents the appropriate permutation of indices. The inverse of the iteration operator is now expressed as the product of two matrices S and D^{-1}, the first of which resembles the double sweep preconditioner of the layered decomposition with some additional entries, while the second is the inverse of a diagonal matrix whose elements can also be regarded as sweeps. Again using F_A^{-1} to right-precondition the system $Fg = b$, we write the preconditioned system as:

$$
\begin{aligned}
FF_A^{-1}g' &= b, \\
F_A^{-1}g' &= g.
\end{aligned}
\tag{3.13}
$$

Using the factored expression of the preconditioner, we get the new preconditioned system:

$$
FSg'' = b, \tag{3.14}
$$

with the change of variable $g' = Dg''$. We can therefore solve system (3.14) by our matrix-free iterative method as usual, and then do an additional step which is the

application of D to g'', before recovering our initial variable g using the second relation of (3.13).

Let us now give an interpretation of the matrix-vector product with matrix S that appears in the factorization of F_A^{-1} in equation (3.12). In the case of the layered decomposition it could be rearranged as two independent sequences of solves thanks to a double recursion formula. It is no longer the case with the cyclic decomposition: we now have to perform as many forward and backward sweeps as there are domains, since each domain is connected to itself via the loop (in 2 possible directions).

It is still possible to reuse the results of the previous solve by passing it as source term for the next domain, but all domains must be solved at each step, whereas only one domain had to be solved (in each direction) in the layered case. However, the total duration of a sweep is unchanged if the domains are solved in parallel, with the difference that no process remains idle during the sweep, hence ruling out the possibility of pipelining different excitations. For that reason also, the forward and backward sweep cannot be performed concurrently.

Algorithm 3.3: Application of the double sweep preconditioner (with loop):
$r \leftarrow Sr$

```
// Forward sweeps
for j = 1 : N − 1
    // Solve all problems in parallel
    for i = 1 : N
        u_D ← 0 on ∂Ω_i ∩ Γ_D
        f_i ← 0
        g_l ← r_{i,i−1}
        g_r ← 0
        Solve H_i u_i = f_i, with data g_l and g_r
    end
    // Transfer to next domain
    for i = 1 : N
        r_{i+1,i} ← r_{i+1,i} + 2S u_i|_{Σ_{i,i+1}}
    end
end
```

```
// Backward sweeps
for j = 1 : N − 1
    // Solve all problems in parallel
    for i = 1 : N
        u_D ← 0 on ∂Ω_i ∩ Γ_D
        f_i ← 0
        g_l ← 0
        g_r ← r_{i,i+1}
        Solve H_i u_i = f_i, with data g_l and g_r
    end
    // Transfer to next domain
    for i = 1 : N
        r_{i−1,i} ← r_{i−1,i} + 2S u_i|_{Σ_{i,i−1}}
    end
end
```

```
// N.B.: Manage indices over- or underflow
```

The recursive structure of the preconditioner is not as obvious as in the layered case, but the matrix elements can still be predicted for any number of domains. An

efficient ($\mathcal{O}(2N)$) matrix-free algorithm can therefore be designed for the application of matrix S to a vector if N processes are available (Algorithm 3.1). Note that the iteration operator itself is $\mathcal{O}(1)$ under the same conditions. If only 1 process is used to run the algorithm, the preconditioned and unpreconditioned versions are respectively $\mathcal{O}(3N)$ and $\mathcal{O}(N)$. Therefore, one should assess the relevance of using the preconditioner in this case, as information is not necessarily massively exchanged between all subdomains, and the regular algorithm could therefore give decent convergence rates, while the preconditioner would bring little improvement compared to the much increased iteration cost.

While matrix D has a much simpler structure than that of matrix S, its application to a vector is quite similar since it also involves a full set of sweeps starting from one boundary of each domain, in both directions. Note that in this case the sweeps cover all domains, while the last one was previously skipped. The forward and backward sweeps are again independent, so the multiple right-hand side strategy can be used here as well to avoid the repetition of the full sequence of solves.

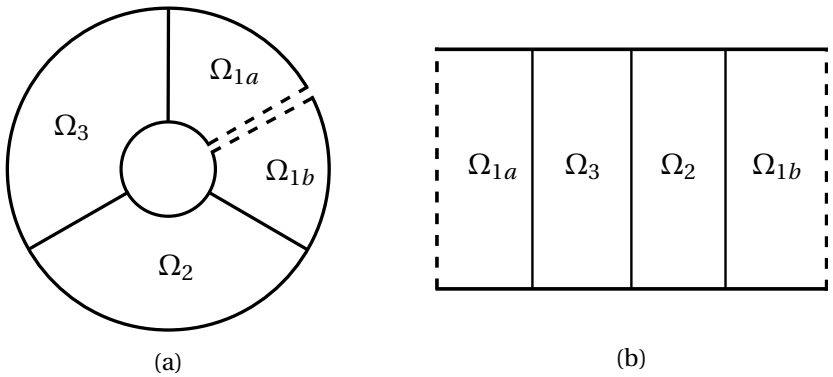
Algorithm 3.4: Application of matrix D: $r' \leftarrow Dr$

```
// Take a copy of r
r' ← r
// Forward sweeps
for j = 1 : N
    // Solve all problems in
       parallel
    for i = 1 : N
        u_D ← 0 on ∂Ω_i ∩ Γ_D
        f_i ← 0
        g_l ← r_{i,i-1}
        g_r ← 0
        Solve H_i u_i = f_i, with data g_l
        and g_r
    end
    // Transfer to next domain
    for i = 1 : N
        r_{i+1,i} ← 2S u_i|_{Σ_{i,i+1}}
    end
end
// Backward sweeps
for j = 1 : N
    // Solve all problems in
       parallel
    for i = 1 : N
        u_D ← 0 on ∂Ω_i ∩ Γ_D
        f_i ← 0
        g_l ← 0
        g_r ← r_{i,i+1}
        Solve H_i u_i = f_i, with data g_l
        and g_r
    end
    // Transfer to next domain
    for i = 1 : N
        r_{i-1,i} ← 2S u_i|_{Σ_{i,i-1}}
    end
end
// Sum with initial value
r' ← r' + r
```

An interesting question is the behaviour of the preconditioner designed for the layered case when applied to a looped decomposition. We need to introduce a cut in the loop, for instance in the middle of domain 1 (Figure 3.6(a)). Perfectly absorbing conditions are imposed on both sides of the cut. This is topologically equivalent to the configuration of the layered decomposition, with one additional domain, as domain 1 is split into $1a$ and $1b$, and the unknowns set remains unchanged. The iteration operator F_{A_c} of the cut problem has the same structure as (3.4). If we use its inverse (3.10) to precondition the looped problem, the preconditioned operator will be:

$$F_{A_l}F_{A_c}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1+\mathcal{B}_1^f\mathcal{B}_3^f\mathcal{B}_2^f & 0 & -\mathcal{B}_1^f\mathcal{B}_3^f & 0 & \mathcal{B}_1^f \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \mathcal{B}_1^b & 0 & -\mathcal{B}_1^b\mathcal{B}_2^b & 0 & 1+\mathcal{B}_1^b\mathcal{B}_2^b\mathcal{B}_3^b & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.15}$$

It has only two lines that are non-trivial (this also holds for more domains). If the coupling between the domains are not too strong (distant domains exchange little information), the product between the $\mathcal{B}_i$ operators quickly become small, and that matrix is close to an identity, making the simplified preconditioner a good approximation of the iteration operator. One can therefore expect good convergence of the algorithm. In the Section 3.3, we will present a modification of the double sweep algorithm that introduces one or more cuts to reduce the length of the sweeps to be performed. This is very similar to the cut introduced here to break the loop, and has a similar effect. The two approaches can easily be combined in the case of a loop involving many domains, which leads to a loop broken in several parts over which the partial sweeps are performed.



(a) (b)

Figure 3.6: Introducing a cut in the middle of a domain (a) makes the configuration topologically equivalent to a layered decomposition (b). Transparent boundary conditions are imposed on the new boundaries (dashed lines).

3.2.5 Relation with incomplete decompositions

We notice that a straightforward factorization can be obtained for F_A: we have $F_A = LU = UL$, with L^{-1} and U^{-1} being immediately related to forward and backward sweeps. Since the sweeps are independent of each other, there is also an additive relation between the factors. The additional property $F_A = (L+U) - I$ indeed shows that applying the sweeps sequentially or in parallel is equivalent, since the following relations hold:

$$\begin{aligned} F_A^{-1} &= L^{-1}U^{-1} = U^{-1}L^{-1} \\ &= (L^{-1} + U^{-1}) - I \end{aligned} \tag{3.16}$$

Going back to the full Schwarz operator, it is now clear that the double sweep can be seen as an incomplete LU decomposition of F_N. This observation establishes an interesting link with the related works on the AILU preconditioner [92] and the sweeping preconditioner [68, 69, 150], since these methods perform an approximate decomposition of the discrete Helmholtz operator in a layer-by-layer fashion.

3.3 Parallelization of the double sweep

The main shortcoming associated with the double sweep preconditioner is its sequential nature, that makes the application of the double sweep a non-parallel and time-consuming step of the algorithm. If the associated performance is excellent in cases where there is a massive transfer of information between all the domains, as is the case in waveguide-like configurations, the situation is different when there is less transfer between domains, like in some scattering problems. In such cases, the performance gain brought by the preconditioner is unsufficient to compensate for the additional overhead.

Therefore, we would like a preconditioner that is able to share information over a sufficient but limited range of domains, unlike the standard double sweep preconditioner that covers all domains, but that has a reduced application cost. To design such a preconditioner, we propose to introduce one or more cuts in the sequence of domains, and to do the sweeps inside these groups of adjacent domains, that we will call portions of the domain in the following. Absorbing conditions are imposed on both sides of the cut. As the groups of domains do not overlap, the partial sweeps can be performed independently and in parallel, therefore reducing the application time of the whole procedure.

The matrix corresponding to these reduced sweeps is derived from the full preconditioner matrix, where the transmission operators corresponding the cut domain is replaced by 0. As a consequence, communication between the two groups of domains is forbidden and the preconditioner matrix has well separated blocks,

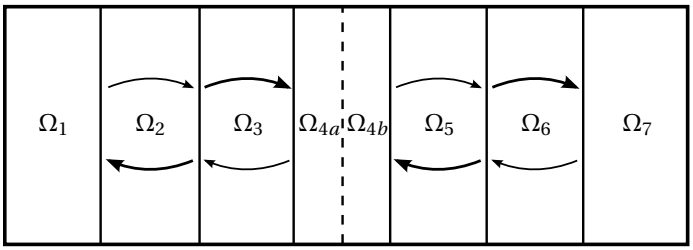


Figure 3.7: Partial sweeps cover non-overlapping groups of domains, separated by the dashed line. The position of the cut inside the domain is not important as the first and last domains are not solved in our sweeps, as shown by the arrows.

that correspond to the independent restricted double sweeps. They can thus be performed in parallel. Each of the block is the inverse of an optimized Schwarz DDM operator F_{A_i} over the i-th portion of the domain consisting of a few adjacent domains, taken apart from the other portions. For the situation of Figure 3.7, where portions 1 and 2 consist of domains $\{1,2,3,4a\}$ and $\{4b,5,6,7\}$ respectively, the matrix is:

$$
\begin{aligned}
F_{A_{\text{cut}}}^{-1} &= \begin{bmatrix}
\begin{bmatrix}
1 & 0 & -\mathcal{B}_2^b & 0 & \mathcal{B}_2^b\mathcal{B}_3^b & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -\mathcal{B}_3^b & 0 \\
0 & -\mathcal{B}_2^f & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & \mathcal{B}_2^f\mathcal{B}_3^f & 0 & -\mathcal{B}_3^f & 0 & 1
\end{bmatrix} & \\
& \begin{bmatrix}
1 & 0 & -\mathcal{B}_5^b & 0 & \mathcal{B}_5^b\mathcal{B}_6^b & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -\mathcal{B}_6^b & 0 \\
0 & -\mathcal{B}_5^f & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & \mathcal{B}_5^f\mathcal{B}_6^f & 0 & -\mathcal{B}_6^f & 0 & 1
\end{bmatrix}
\end{bmatrix} \\
&= \begin{bmatrix} F_{A_1}^{-1} & \\ & F_{A_2}^{-1} \end{bmatrix}.
\end{aligned}
\tag{3.17}
$$

This is very much like a block-Jacobi preconditioner for the simplified Schwarz iteration operator. In the ideal case where the true DtN is used as transmission condition, the preconditioned operator has the following structure:

$$F_A F_{A_{\text{cut}}}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \mathcal{B}_4^b & 0 & -\mathcal{B}_4^b\mathcal{B}_5^b & 0 & \mathcal{B}_4^b\mathcal{B}_5^b\mathcal{B}_6^b & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{B}_2^f\mathcal{B}_3^f\mathcal{B}_4^f & 0 & -\mathcal{B}_3^f\mathcal{B}_4^f & 0 & \mathcal{B}_4^f & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.18}$$

that is relatively close to an identity. Spectral analysis reveals that the eigenvalues are well clustered ($\lambda_{1,\ldots,2N-2} = 1$) but that it is defective, as is the unpreconditioned DDM operator. However, the number of missing eigenvectors is $2N_c$ instead of $2N-4$, if N_c is the number of cuts, which is a strong improvement if $N_c << N$. Numerical experiments with the new version of the preconditioner confirm this analysis, with a convergence rate that degrades as the number of cuts increases for a strongly guided problem, and plateaus appearing on the convergence curve (Figure 3.15). This situation is to be compared with the behaviour of the unpreconditioned DDM algorithm, that corresponds to the limit case where a cut is placed in every other domains.

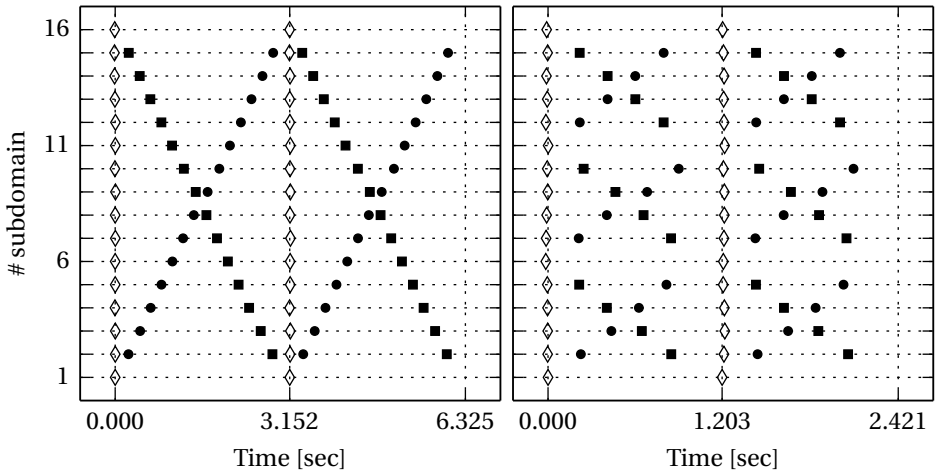


Figure 3.8: Introducing 2 cuts in the double sweep preconditioner (right) enables parallel execution of the partial sweeps, reducing the application time of the preconditioner without cuts (left). The white diamonds indicate solves performed in the iteration operator; the black circles and squares indicate solves in the forward and backward sweeps, respectively. These time lines were obtained for the COBRA test case of Section 3.4.2, with 16 subdomains and cuts in subdomains 6 and 11.

To illustrate the gain in application time and the improved parallelism of the double sweep with cuts, Figure 3.8 presents timelines of the solves performed dur-

ing the application of the Schwarz iteration operator and the double preconditioner, with and without cuts. In Section 3.2.3, we suggested to perform simultaneous application of the Schwarz iteration operator and the double sweep preconditioner, by precomputing the matrix-matrix product of these two steps and inferring an equivalent application routine. This allowed to save redundant solves and reduce the time required to complete an iteration, giving raise to Algorithm 3.2. Of course, the same strategy can be applied to the preconditioner with cuts. Doing so, only the cut domains need to be solved after the subgroups of domains have been sweeped, as illustrated on Figure 3.9.

3.4 Numerical results

We will present numerical results on different test cases of increasing complexity, with the intent of first showing how the preconditioner behaves on simple cases in the scalar case, in homogeneous and non-homogeneous media. This will be the occasion to notice the importance of the accuracy of the transmission conditions, before focusing on more realistic configurations, where the vector case will be treated as well. Following that progression, only the full version of the preconditioner will be studied in a first step. The parallelized version is then introduced to demonstrate its potential on the most challenging cases.

3.4.1 Full sweeps

We first present numerical results in the 1d case with constant parameters, from which the preconditioner was derived in Section 3.2. We then use the same preconditioning strategy on more complex configurations and in the presence of a non-homogeneous medium, with guided and non-guided waves in 2d. In each case we analyse the effect of the accuracy of the DtN approximation on the performance of the preconditioner. In the non-homogeneous cases, we define the pulsation ω and velocity $c(x,y)$, such that $k(x,y) = \omega/c$.

1d with constant parameters

We study the behaviour of the algorithm in an homogeneous 1d medium $\Omega = [0,1]$. This test case reproduces the conditions of the construction of the double sweep preconditioner (Section 3.2): we use absorbing conditions on both sides of the domain, and a volume source modeled by a delta-function located on the left boundary $f = \delta(0)$.

We observe that the convergence is fast compared to the unpreconditioned case (Figure 3.10): no more than a few iterations are required with the double sweep preconditioner. More interestingly, the rate of convergence is independent of the number of subdomains (Table 3.1) and wavenumber (Figure 3.11), under

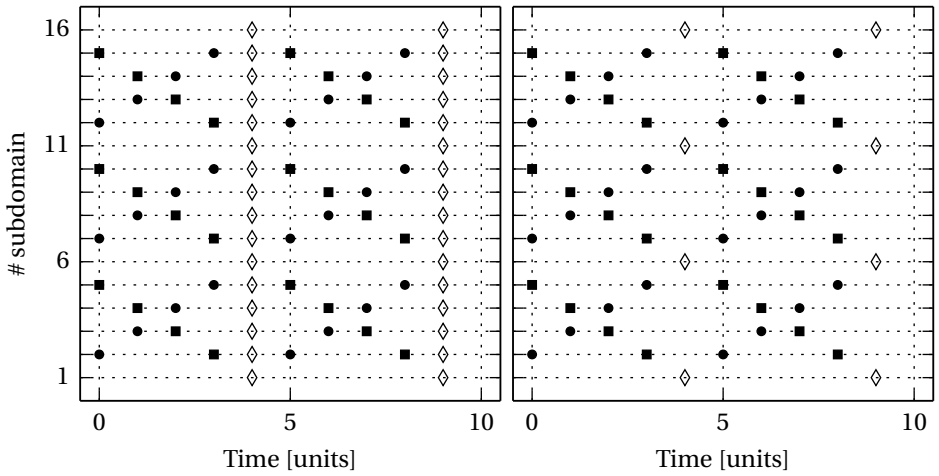


Figure 3.9: Illustration on synthetic timelines of the separate (left) and combined (right) application of the double sweep with cuts and the Schwarz operator. The combined application avoids redundant solves, while not being faster in a fully parallel implementation. The main difference (invisible on the Figure) between both algorithms is in the flow of information during the sweeps.

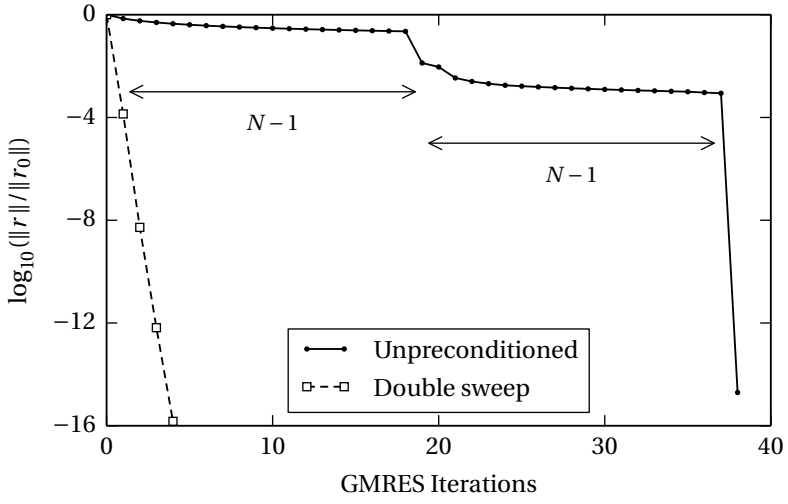


Figure 3.10: A typical convergence history of the optimized Schwarz algorithm with GMRES acceleration applied to a 1d problem, with $N = 20$ subdomains (solid line). The curve exhibits 2 plateaus with length $N-1$, before suddenly converging with full accuracy. The dashed line is obtained for the same problem with the preconditioned algorithm using optimal transmission operators.

the condition that the exact DtN map is used. In 1d, this can be achieved by replacing the wavenumber k in the Sommerfeld condition IBC(0) by the wavenumber k_h accounting for the numerical dispersion of FEM for a discretization step h, as given by formula (D.5).

	$N=5$	25	50	100	150	200
IBC(0), $n_\lambda = 10$	4 (8)	4 (48)	5 (98)	5 (198)	6 (298)	6 (398)
IBC(0), $n_\lambda = 20$	3 (8)	3 (48)	4 (98)	4 (198)	4 (298)	4 (398)
IBC$_{k_h}$(0), $n_\lambda = 10$	3 (8)	3 (48)	3 (98)	3 (198)	3 (298)	3 (398)
IBC$_{k_h}$(0), $n_\lambda = 20$	2 (8)	2 (48)	2 (98)	2 (198)	2 (298)	3 (398)

Table 3.1: The iteration count ($||r||/||r_0|| \leq 10^{-6}$) in the 1d case is very small and steady with the number of domains when the exact DtN is used as transmission operator ($\mathcal{S} = \mathcal{D} = -\imath k_h$); more iterations are required when less accurate approximations of the DtN map are used instead ($\mathcal{S} \neq \mathcal{D} = -\imath k$), with a slight dependence on N. Values between parentheses are for the unpreconditioned algorithm: $N_{it} = M = 2(N-1)$, where M is the size of the Schwarz system, hence the theoretical maximal number of iterations required to solve it by a Krylov method.

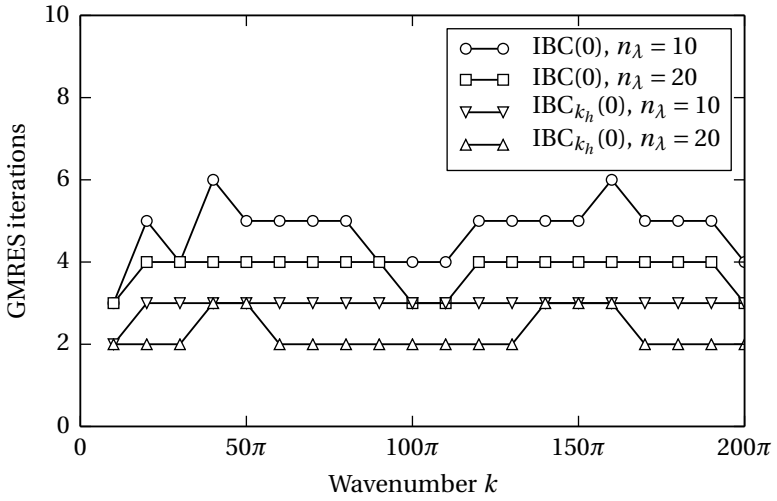


Figure 3.11: 1d problem: the iteration count is also stable with the wavenumber k, for all the tested transmission operators $\mathcal{S}$ ($N = 100$).

In the more practical case when an approximate DtN map is used as transmission operator, the performance of the algorithm is slightly degraded and the number of iterations weakly depends on the number of subdomains. To observe this effect, we repeat the computation with different discretizations (we denote by n_λ the number of grid points per wavelength). As the numerical dispersion is more

pronounced for coarser discretizations (smaller n_λ), the accuracy of the Sommerfeld conditions ($\mathcal{S} = -\imath k$) that we use as transmission operators is also degraded.

This behaviour is easily understood under the light of the analysis of Section 3.2: the double sweep preconditioner was built upon the assumption that no reflection occurs at the artificial boundaries, which can only be verified with an accurate DtN map as transmission operator; in that case, it is an exact inverse of the iteration operator and should converge in 1 iteration in exact arithmetic. With reflections at the artificial boundaries, the double sweep is no longer an exact inverse of the iteration operator and its performance deteriorates with the amplitude of the spurious reflected waves. More intuitively, an effect of the reflections is the transmission of partial information to the neighbouring subdomains. As this information is transmitted multiple times in the course of a sweep, it will be distorted everytime it crosses an artificial boundary. This explains why more iterations are required when more domains are involved if inexact DtN maps are used as transmission operators.

Homogeneous waveguide

The geometry of this test case is a straight waveguide ($\Omega = [0, D] \times [0, d]$) made of an homogeneous medium. Homogeneous Dirichlet conditions are imposed on the upper and lower sides of the guides: $u = 0$ on $y = \{0, d\}$. We excite the second mode on the left hand side: $u(0, y) = \sin(m\pi y/d)$, with $m = 2$, and use an absorbing condition on the right hand side to model a continuing waveguide.

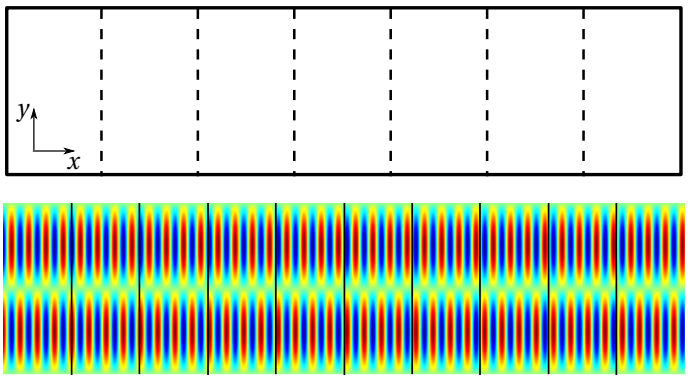


Figure 3.12: Waveguide geometry ($D = 4; d = 1$) and decomposition, and solution for $m = 2$ and $k = 20\pi$.

As there is only one possible propagation direction and no internal reflection for such a problem, the method is expected to behave somewhat like in the 1d case since we are still in the conditions of the preconditioner derivation. Moreover, an analytical solution for this problem is available for each mode, considered separately [148]:

$$u_A(x, y) = \sin\big(\frac{m\pi}{d} y\big) \exp \imath\beta_m x,$$

	$\omega = 20\pi$					$\omega = 40\pi$				
	$N = 5$	10	25	50	100	5	10	25	50	100
IBC(0)	3 (8)	3 (18)	4 (48)	4 (98)	4 (198)	3	3	4	4	4
IBC($k/2$)	8 (8)	8 (18)	23 (50)	56 (120)	88 (326)	8	9	38	49	dnc
OO_2	3 (8)	3 (18)	4 (46)	4 (98)	4 (201)	3	3	3	3	4
GIBC(2)	3 (8)	3 (18)	3 (48)	4 (119)	4 (239)	3	3	4	4	8
GIBC(8)	3 (8)	3 (18)	3 (48)	4 (119)	4 (240)	3	3	4	4	8
PML(5)	4 (8)	4 (18)	5 (48)	6 (96)	6 (196)	4	4	6	8	12
PML(15)	3 (8)	3 (18)	3 (48)	4 (98)	4 (198)	3	3	3	3	4
PML(75)	2 (8)	2 (18)	2 (48)	3 (98)	3 (198)	2	2	2	2	2

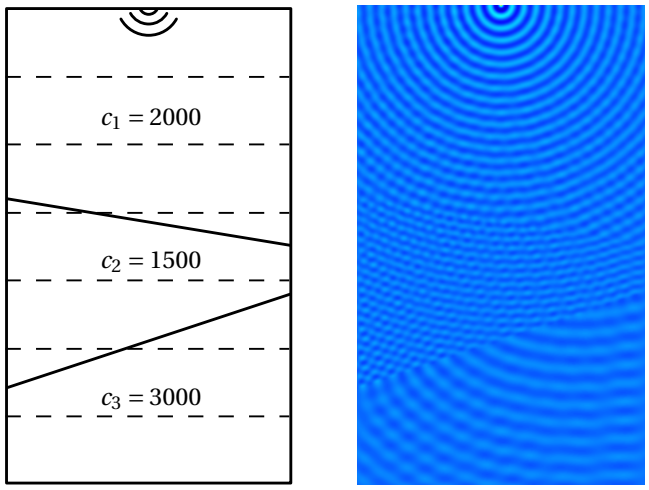
Table 3.2: Homogeneous waveguide: iteration count of the preconditioned GMRES ($||r||/||r_0|| < 10^{-6}$) as a function of N, for different transmission conditions. Values in parentheses are for the unpreconditioned algorithm; "dnc" stands for "did not converge" within the prescribed 500 iterations.

with:

$$\beta_m = \begin{cases} \sqrt{k^2 - (m\pi/d)^2} & \text{if } 1 \le m \le \frac{kd}{\pi} \quad \text{(propagative);} \\ \imath\sqrt{-k^2 + (m\pi/d)^2} & \text{otherwise} \quad \text{(evanescent).} \end{cases}$$

We only present results for the propagative modes. An exact expression of the DtN map can be inferred from the above expression: $\mathcal{D} = -\imath\beta_m$. Modifying the Sommerfeld transmission condition by replacing k by β_m therefore gives an excellent approximation of the DtN map, that can be ever improved by accounting for the numerical dispersion in its definition (β_m^h). However, we do not include these results as they are not sufficiently general (only valid for a single mode.) The results for all the tested approximations of the DtN map are presented in Table 3.2. With the exception of IBC(χ), they all perform well and little penalty is associated with the use of rough approximations of the DtN map; the good performance of the local approximations is therefore not surprising. GIBC($k/2$) fails as it adds a real part to the operator $\mathcal{S}$ while the DtN map in this particular case is purely imaginary.

Simple underground model



(a) (b)

Figure 3.13: (a) Underground propagation test case geometry and decomposition, (b) solution for $\omega = 160\pi$.

We consider a rectangular domain $\Omega = [0,600] \times [0,1000]$ made of an heterogeneous medium with 3 different velocities in regions separated by straight non-parallel boundaries (Figure 3.13(a)). The outside world is modeled by Sommerfeld conditions on the "underground" sides, and a Neumann condition on the top side. A Dirichlet point source is located in the middle of the top line. This test-case was proposed in [75, 149] to test multigrid preconditioners for the Helmholtz equation.

As opposed to the previous test case, we are now in presence of abrupt variations of the wavenumber, that will produce internal reflections in different directions. The Sommerfeld conditions on the external boundaries are also likely to reflect part of the outgoing waves back into the domain. For these reasons, this problem is expected to be more challenging for a Schwarz method, as the multiple reflections will be harder to capture throughout the iterations. This also holds for our preconditioner, since this case is further removed from the hypotheses underlying its construction (recall that we neglected both numerical and physical reflections).

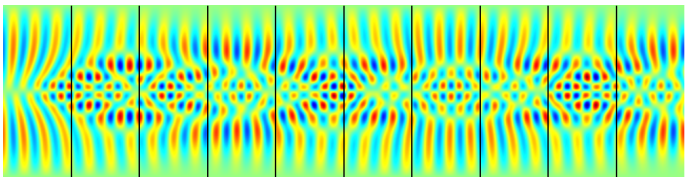
If the number of iterations is indeed slightly increased, the property of independence of the number of iterations with respect to the number of subdomains is actually preserved with the most accurate approximation of the DtN map (PML(75)). Even with less accurate approximations, we find that the number of additional iterations required for convergence grows relatively slowly with N (see Table 3.3). The GIBC(2) condition in particular performs extremely well—at a fraction of the cost of the non-local PML approximation.

	$\omega = 80\pi$					$\omega = 160\pi$				
	$N = 5$	10	25	50	100	5	10	25	50	100
IBC(0)	88 (96)	90 (133)	99 (244)	158 (415)	347 (dnc)	97	98	107	134	265
IBC($k/4$)	62 (70)	63 (110)	68 (231)	92 (404)	178 (dnc)	66	67	73	90	168
OO_2	22 (38)	24 (77)	28 (207)	46 (384)	70 (dnc)	25	27	42	74	186
GIBC(2)	25 (40)	27 (74)	29 (186)	35 (369)	41 (dnc)	25	26	29	36	56
GIBC(8)	18 (36)	19 (73)	22 (186)	32 (369)	39 (dnc)	19	20	24	34	56
PML(5)	15 (38)	16 (75)	17 (195)	23 (368)	29 (dnc)	22	27	43	143	dnc
PML(15)	14 (36)	15 (74)	16 (183)	16 (359)	15 (dnc)	14	15	15	16	79
PML(75)	14 (35)	14 (72)	14 (182)	14 (357)	14 (dnc)	14	14	14	15	15

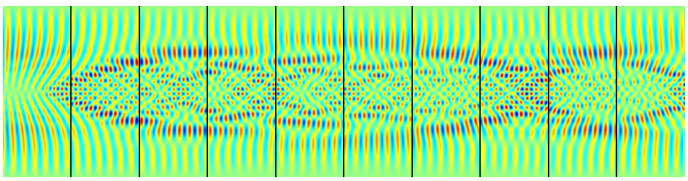
Table 3.3: Results for the underground propagation test case ($||r||/||r_0|| < 10^{-6}$) for different approximations of the DtN map. The convergence rate is independent of the number of domains and wavenumber when a very accurate non-local approximation of the DtN map is used(PML(75)). Less accurate approximations degrade the convergence rate, especially when many subdomains are involved, but the algorithm still appears to be quite robust for the best fast local approximations; "dnc" stands for "did not converge" within the prescribed 500 iterations. Values between parentheses are for the unpreconditioned algorithm.

Gaussian waveguide

The geometry of this test case is comparable to the one presented in Section 3.4.1, but we now consider a non-homogeneous medium, with a velocity profile constant in the propagation direction (along the x-axis) and gaussian in the transverse direction (see Figure 3.12): $c(x,y) = 1.25(1 - .4\exp -32(y-.5)^2)$. This a 2D version of a test case used in [150]. The second mode is excited on the left hand boundary and we model the continuing domain by means of a Sommerfeld condition on the right hand side.



(a)



(b)

Figure 3.14: Solution of the gaussian waveguide problem at (a) $\omega = 20\pi$ and (b) $\omega = 40\pi$. In both examples, the second mode is excited on the left hand side.

	$\omega = 20\pi$					$\omega = 40\pi$				
	$N = 5$	10	25	50	100	5	10	25	50	100
IBC(0)	38 (70)	40 (131)	78 (345)	207 (dnc)	497 (dnc)	51	65	129	217	dnc
IBC($k/2$)	35 (71)	45 (157)	134 (412)	314 (dnc)	dnc (dnc)	56	82	241	495	dnc
OO_2	30 (62)	33 (128)	69 (356)	175 (dnc)	303 (dnc)	41	53	123	202	dnc
GIBC(2)	19 (53)	20 (114)	42 (314)	98 (dnc)	149 (dnc)	27	31	67	103	288
GIBC(8)	19 (53)	20 (114)	42 (314)	98 (dnc)	150 (dnc)	27	31	67	105	283
PML(5)	13 (47)	12 (103)	13 (271)	15 (dnc)	16 (dnc)	16	20	30	52	115
PML(15)	12 (44)	12 (101)	12 (266)	12 (dnc)	12 (dnc)	13	13	13	14	15
PML(75)	11 (44)	11 (99)	11 (264)	11 (dnc)	11 (dnc)	13	13	13	13	13

Table 3.4: Gaussian waveguide: iteration count of the GMRES ($||r||/||r_0|| < 10^{-6}$) as a function of the number of subdomains N.

As opposed to the homogeneous case presented above, the gaussian speed profile will cause curved beams at high frequency that make the problem truly 2D, as can be seen on Figure 3.12. As a consequence, the "oscillations" exhibited in the matrix of the DtN map are more complex (their direction varies with position) and harder to capture by approximate methods. This is verified in practice: Table 3.4 shows that the convergence of all the local approximations degrades significantly for large N, and only the most accurate non-local PML approximations are able to maintain a constant iteration count with N. The interest of the preconditioner is particularly visible on this test case, where the unpreconditioned Schwarz algorithm becomes practically unusable for more than 25 domains.

3.4.2 Sweeps with cuts

We start with an interesting result obtained on the straight 2d waveguide of Figure 3.12. We can observe on Figure 3.15 that the double sweep somehow acts like a compressor on the convergence curve, and that the introduction of cuts progressively reduces that effect. This is probably because the residual at some interfaces does not start to decrease before the information has reached them, which requires more iterations when more cuts are involved.

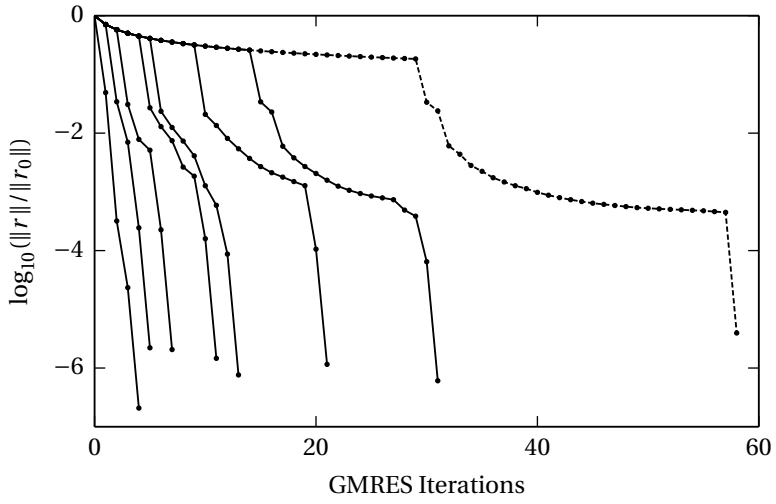


Figure 3.15: Homogeneous waveguide test case with 31 subdomains and increasing number of cuts (solid lines), from left to right: $N_c = 0, 1, 2, 4, 5, 9, 14$. The phase of fast convergence starts with a delay: during the N_c first iterations, the convergence curve follows the unpreconditioned one (dashed line). With many cuts, similar plateaus tend to appear.

We now present results obtained on three different test geometries: a straight 3d (parallelepipedic) waveguide, a 3d S-shaped cavity (the COBRA benchmark defined by the JINA98 workgroup) and the open 2d scattering problem by a circular

object. The first two are solved using a layered decomposition while the third uses a cyclic decomposition. The COBRA is solved for both Helmholtz and Maxwell, while the other two are solved for Helmholtz only. Earlier work ([31, 180, 182]) has shown that without preconditioner, the iteration count for such problems typically grows linearly with the number of domains, and that with the use of the double sweep it becomes almost independent for layered decompositions, provided that the approximation of the DtN map is sufficiently accurate.

Tables 3.5–3.7 summarize the number of iterations required by each algorithm to converge to the prescribed tolerance, together with an estimation of the normalized times required for the completion of the algorithm. Provided that at least 2 CPUs are alloted per group of domains, the time required for the application of the standard Schwarz operator and the double sweep preconditioner with N_d subdomains, N_c cuts and C_{tot} CPUs (assumed evenly distributed between the groups of subdomains) are approximately given, in the case of a layered decomposition by:

$$T_{\text{Sch}} = \left\lceil \frac{N_d}{C_{\text{tot}}} \right\rceil T_p \quad \text{and} \quad T_{\text{sw}}(N_c) = \left\lceil \frac{N_d - N_c - 2}{N_c + 1} \right\rceil T_p, \tag{3.19}$$

with T_p the solution time for one subproblem (supposed identical for all subdomains). Note that T_{sw} would be doubled if only one CPU is available to perform the double sweep per group of domains. Slightly different estimations hold in the case of the cyclic decomposition. The total solution times for the unpreconditioned and double sweep algorithms are then $T_{\text{sol}}^{(np)} = T_{\text{Sch}} N_{it}^{(np)}$ and $T_{\text{sol}}^{(ds)}(N_c) = (T_{\text{Sch}} + T_{\text{sw}}(N_c)) N_{it}^{(ds)}$.

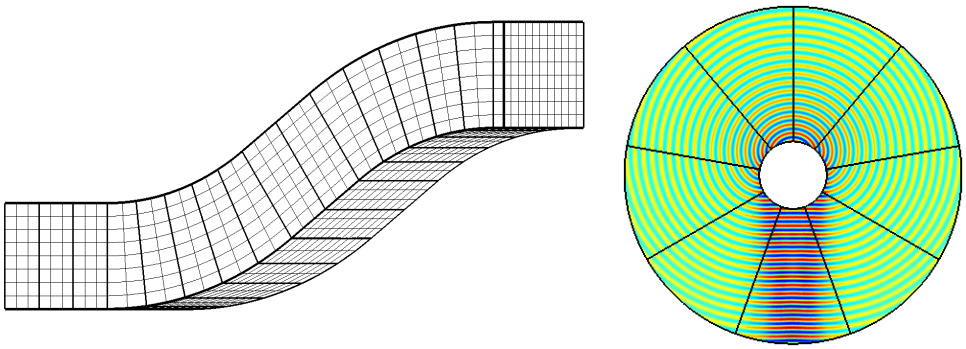


Figure 3.16: Geometry and typical decomposition of the 3d cobra cavity (JINA98) and 2d scattering (unit sound-soft disc with Sommerfeld ABC at radius = $5m$) test cases. They differ by the topology of the decomposition (layered vs. cyclic) and by the type of wave involved (guided vs. free.) The parallelepipedic waveguide (not pictured) has dimensions $0.91m \times 0.084m \times 0.11m$, comparable to the COBRA.

#CPU	2	4	6	8	14	22
$N_{\text{it}}^{(\text{np})}$	62					
$T_{\text{sol}}^{(\text{np})}$	992	496	331	248	142	91
N_c	0	1	2	3	6	10
$N_{\text{it}}^{(\text{ds})}$	5	6	8	10	16	24
$T_{\text{sol}}^{(\text{ds})}$	230	138	128	110	112	96

#CPU	2	4	6	8	14	22
$N_{\text{it}}^{(\text{np})}$	766					
$T_{\text{sol}}^{(\text{np})}$	12256	6128	4086	3064	1751	1115
N_c	0	1	2	3	6	10
$N_{\text{it}}^{(\text{ds})}$	116	153	174	188	241	308
$T_{\text{sol}}^{(\text{ds})}$	5336	3519	2784	2068	1687	1232

Table 3.5: Straight waveguide (top) and COBRA (bottom) cases for Helmholtz with 32 subdomains, $k = 314.16$ (relative residual decrease by 10^{-4}; the transmission operator is GIBC(2); the transmission operator is GIBC(2)).

#CPU	2	4	6	8	14	22
$N_{\text{it}}^{(\text{np})}$	448					
$T_{\text{sol}}^{(\text{np})}$	7168	3584	2390	1792	1024	652
N_c	0	1	2	3	6	10
$N_{\text{it}}^{(\text{ds})}$	21	34	48	62	104	160
$T_{\text{sol}}^{(\text{ds})}$	966	782	768	682	728	640

#CPU	2	4	6	8	14	22
$N_{\text{it}}^{(\text{np})}$	> 1000					
$T_{\text{sol}}^{(\text{np})}$	> 16016	> 8008	> 5339	> 4004	> 2288	> 1456
N_c	0	1	2	3	6	10
$N_{\text{it}}^{(\text{ds})}$	44	74	105	135	230	354
$T_{\text{sol}}^{(\text{ds})}$	2024	1702	1680	1485	1610	1416

Table 3.6: COBRA test case for Maxwell with 32 subdomains, $k = 157.08$ (top) and $k = 314.16$ (bottom) (relative residual decrease by 10^{-4}).

#CPU	2	52	86
$N_{\text{it}}^{(\text{np})}$	55		
$T_{\text{sol}}^{(\text{np})}$	3520	136	82
N_c	1	26	43
$N_{\text{it}}^{(\text{ds})}$	24	27	31
$T_{\text{sol}}^{(\text{ds})}$	4584	189	124

#CPU	2	52	86
$N_{\text{it}}^{(\text{np})}$	85		
$T_{\text{sol}}^{(\text{np})}$	5440	210	127
N_c	1	26	43
$N_{\text{it}}^{(\text{ds})}$	20	29	37
$T_{\text{sol}}^{(\text{ds})}$	3820	203	148

Table 3.7: Scattering test case for Helmholtz with 128 subdomains, k = 6.28 (left) and k = 25.13 (right) (relative residual decrease by 10^{-4}; the transmission operator is GIBC(2)).

Tables 3.5–3.7 show that in all cases the behaviour of the algorithm is similar. The preconditioner strongly reduces the number of iterations, and thus the number of overall linear system solves. Moreover, the parallel version of the preconditioner makes it also an appealing proposition with respect to the overall computational (wall-clock) time when the number of CPUs is smaller than the number of subdomains, especially in the high frequency regime. For example, in the challenging COBRA case for Maxwell, with 32 domains on 8 CPUs (3 cuts), with $k = 100\pi$, the preconditioned version requires $135 \times (32 + 2 \times (32 - 2 - 3)) = 11610$ system solves instead of $> 1000 \times 32$ and runs about 3 times faster than the standard algorithm.

It is also interesting to compare the increase in the number of iterations when cuts are introduced in the two different cases where the waves are guided or propagate in free space. It is clear that in the first case, the coupling between distant domains (e.g. the first and last) is very strong since a source located in one of them will generate waves with significant amplitude in all others. In other words the information exchange between domains is very important in the guided case. Conversely, in problems such as scattering in free space, one can reasonably expect that there will be little exchange of information between diametrically opposed domains.

As a consequence, the introduction of a cut, that somehow breaks the flow of information, will have more impact when the coupling is strong; in Table 3.7, we see that introducing a large number of cuts produces a very limited number of additional iterations, while in Table 3.6 one can observe a fast growth of the iteration count as soon as a cut is added.

The Marmousi model

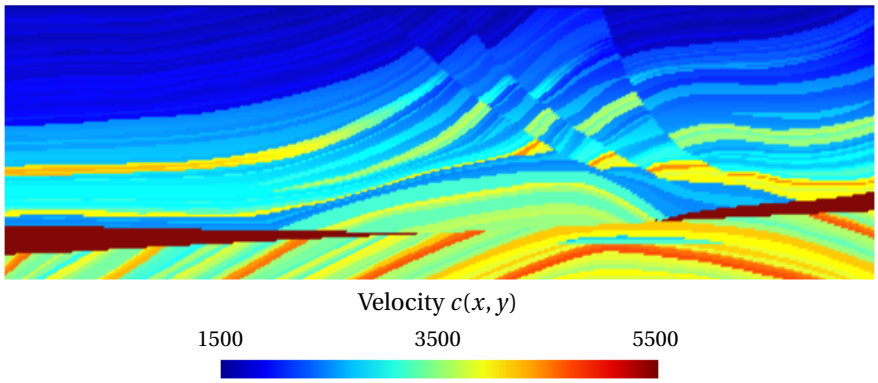
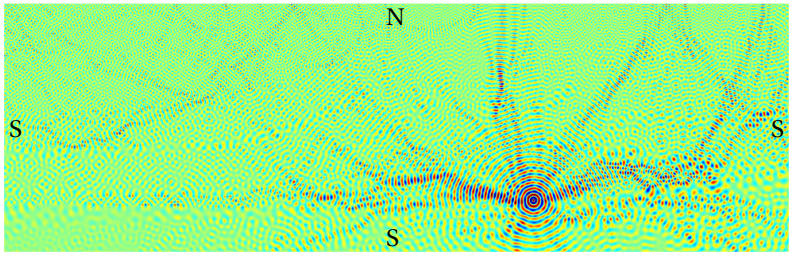


Figure 3.17: Velocity profile of the Marmousi model. Dimensions (in meters) are $[0, 9192] \times [0, -2904]$.

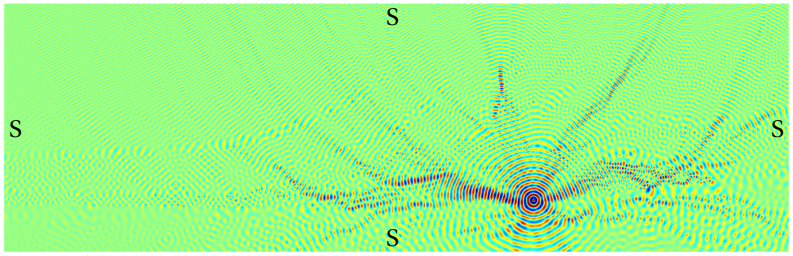
The Marmousi model is a synthetic 2d acoustic model that reproduces the complex velocity profile of a slice of earth. It features a wide range of speeds, from 1500 m/s to 5500 m/s, with many layers an normal faults as depicted on Figure 3.17. It has become a classic test case for benchmarking seismic inversion codes. We will simulate the propagation of the time-harmonic acoustic waves produced by a point source located at coordinates $(6200, -2300)$ as was done in [169], so both methods can be compared.

We solve the Helmholtz equation with Sommerfeld boundary conditions on all sides except the top side where we impose a homogeneous Neumann boundary condition; we will also consider the case of Sommerfeld conditions on all sides, since we assume that this situation was used in [169]. We will also test the methods in a similar domain with an homogeneous medium; Figure 3.18 shows typical solutions in all these configurations.

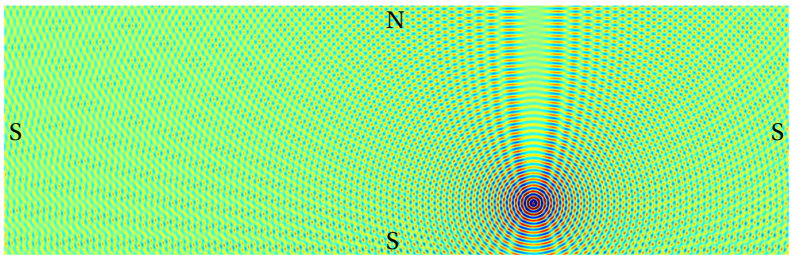
In the following, we report the results of our experiments in tables for each of the configurations, with decompositions into 16, 64 and 256 domains. We have tested two different transmission conditions: the GIBC(2) and the IBC(0), which is the most simple one; each case has been run twice, with decompositions into vertical and horizontal layers. The tables present the number of iterations for each run to converge with a relative residual decrease of 10^{-3}, with a maximum number of iterations of 1000. Values in parentheses are an estimation of the normalized time required to reach convergence when using as many CPUs as there are subdomains. The time unit is the time to solve a single subproblem. Hence, these values cannot be directly compared for different decompositions or frequencies.



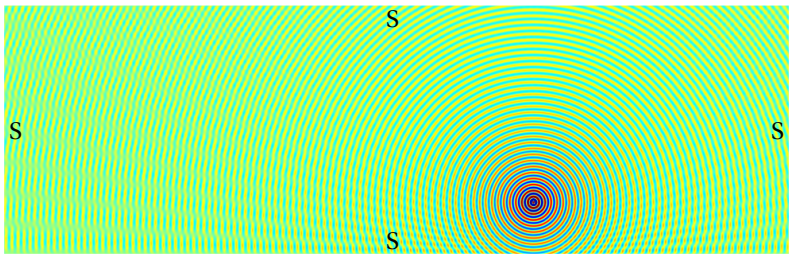
(a)



(b)



(c)



(d)

Figure 3.18: Solutions at $\omega = 100\pi$: (a) marmousi model with Neumann condition on top; (b) marmousi model with absorbing condition on top; (c) homogeneous model ($c(x, y) = 3500$) with Neumann condition on top; (d) homogeneous model with Sommerfeld condition on top. The reflections produced by the Neumann condition are clearly visible.

Marmousi model with Neumann condition on top

This test case is the full physical model, that is also the most difficult to solve with our methods: we will see that using a Sommerfeld condition on the top boundary and considering an homogeneous medium both make the convergence faster. For this particular example, we have tested two different discretizations: a regular grid with step size $h = \lambda_{\min}/20$ that represents the shortest wavelength in the model with 20 points, and a discretization such that $h\frac{\omega}{2\pi} = 200$, as was used in [169]. This second choice leads to a discretization that is about 3 times as coarse as the first one.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	58 (58)	26 (78)	22 (110)	19 (285)	236 (236)	100 (300)	61 (549)	50 (3150)
$\frac{\omega}{2\pi} = 50$	56 (56)	26 (78)	21 (105)	18 (270)	227 (227)	94 (282)	54 (486)	43 (2709)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	> 1000 (> 1000)	444 (1332)	355 (1775)	324 (4860)	364 (92820)
$\frac{\omega}{2\pi} = 50$	> 1000 (> 1000)	538 (1614)	458 (2290)	461 (6915)	564 (143820)

(a) vertical layers, GIBC(2)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	181 (181)	78 (234)	59 (295)	36 (540)	783 (783)	310 (930)	154 (1386)	90 (5670)
$\frac{\omega}{2\pi} = 50$	171 (171)	75 (225)	58 (290)	39 (585)	718 (718)	291 (873)	146 (1314)	93 (5859)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	> 1000 (> 1000)	930 (2790)	591 (2955)	257 (3855)	118 (30090)
$\frac{\omega}{2\pi} = 50$	> 1000 (> 1000)	985 (2955)	660 (3300)	339 (5085)	227 (57885)

(b) horizontal layers, GIBC(2)

Table 3.8: Marmousi model with Neumann condition on top (Figure 3.18(a)). Convergence of the algorithm, with $h = \lambda_{\min}/20$. The preconditioner was used with decreasing number of cuts, with more cuts usually resulting in shorter run times (values in parentheses); n.a. indicates the unpreconditioned algorithm.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	77 (77)	52 (156)	49 (245)	47 (705)	285 (285)	140 (420)	111 (999)	102 (6426)
$\frac{\omega}{2\pi} = 50$	71 (71)	50 (150)	47 (235)	45 (675)	285 (285)	120 (360)	85 (765)	76 (4788)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	986 (986)	742 (2226)	739 (3695)	735 (11025)	727 (185385)
$\frac{\omega}{2\pi} = 50$	> 1000 (> 1000)	596 (1788)	537 (2685)	488 (7320)	466 (118830)

(a) vertical layers, IBC(0)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	234 (234)	103 (309)	83 (415)	65 (975)	868 (868)	437 (1311)	293 (2637)	244 (15372)
$\frac{\omega}{2\pi} = 50$	247 (247)	110 (330)	91 (455)	75 (1125)	929 (929)	411 (1233)	254 (2286)	209 (13167)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	> 1000 (> 1000)	> 1000 (> 3000)	> 1000 (> 5000)	> 1000 (> 15000)	972 (247860)
$\frac{\omega}{2\pi} = 50$	> 1000 (> 1000)	> 1000 (> 3000)	> 1000 (> 5000)	> 1000 (> 15000)	525 (133875)

(b) horizontal layers, IBC(0)

Table 3.9: Marmousi model with Neumann condition on top (Figure 3.18(a)). Convergence of the algorithm, with $h = \lambda_{\min}/20$.

Comparing Tables 3.8(a) and (b), we see that the convergence is faster with vertical layers than with horizontal layers. We will see that it will be the case in all of our subsequent experiments on this test case, even when an homogeneous medium is considered. Two possible explanations for this behaviour are the aspect ratio of the slices (resulting in longer transmission boundaries and more unknowns in the Schwarz problem), and the incidence of the wavefronts that is more tangential in the horizontal case.

The main observation is that the introduction of the cuts efficiently reduces the computation time thanks to the increased parallelization of the preconditioner application (recall that we report timings for the case where $N_{\text{CPU}} = N_{\text{dom}}$), that largely compensates for the increased iteration count. However, we observe that when the (fully parallelized) unpreconditioned algorithm converges, it runs faster

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	57 (57)	26 (78)	21 (105)	18 (270)	239 (239)	100 (300)	60 (540)	48 (3024)
$\frac{\omega}{2\pi} = 50$	56 (56)	26 (78)	21 (105)	18 (270)	229 (229)	92 (276)	52 (468)	41 (2583)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	>1000 (>1000)	446 (1338)	351 (1755)	314 (4710)	353 (90015)
$\frac{\omega}{2\pi} = 50$	>1000 (>1000)	520 (1560)	437 (2185)	443 (6645)	538 (137190)

(a) vertical layers, GIBC(2)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	181 (181)	78 (234)	59 (295)	36 (540)	772 (772)	310 (930)	155 (1395)	92 (5796)
$\frac{\omega}{2\pi} = 50$	167 (167)	74 (222)	57 (285)	37 (555)	706 (706)	289 (867)	145 (1305)	92 (5796)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	>1000 (>1000)	903 (2709)	578 (2890)	253 (3795)	113 (28815)
$\frac{\omega}{2\pi} = 50$	>1000 (>1000)	984 (2952)	658 (3290)	341 (5115)	224 (57120)

(b) horizontal layers, GIBC(2)

Table 3.10: Marmousi model with Neumann condition on top (Figure 3.18(a)). Convergence of the algorithm, with $h\frac{\omega}{2\pi} = 200$.

than the preconditioned one. However, the preconditioner seems to bring some extra reliability, in the sense that it is more likely to converge within a reasonable number of iterations, even when many subdomains are involved.

We again emphasize the importance of the transmission condition to keep that number low; we have seen on other test cases that good approximations of the DtN map gives a convergence independent of the number of domains, even in the non-homogeneous case. This is confirmed by the results obtained in [169] with PMLs, on the same Marmousi test case as we are investigating here. Clearly, the transmission used here are suboptimal, as they are either of too low order (IBC) or have not been developed for non-homogeneous media (GIBC).

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi}=30$	76 (76)	55 (165)	52 (260)	50 (750)	235 (235)	125 (375)	97 (873)	89 (5607)
$\frac{\omega}{2\pi}=50$	74 (74)	53 (159)	50 (250)	48 (720)	237 (237)	117 (351)	87 (783)	78 (4914)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi}=30$	776 (776)	559 (1677)	525 (2625)	505 (7575)	491 (125205)
$\frac{\omega}{2\pi}=50$	831 (831)	478 (1434)	426 (2130)	378 (5670)	362 (92310)

(a) vertical layers, IBC(0)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi}=30$	207 (207)	102 (306)	86 (430)	68 (1020)	699 (699)	363 (1089)	252 (2268)	216 (13608)
$\frac{\omega}{2\pi}=50$	192 (192)	95 (285)	81 (405)	65 (975)	692 (692)	329 (987)	211 (1899)	172 (10836)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi}=30$	>1000 (>1000)	>1000 (>3000)	>1000 (>5000)	>1000 (>15000)	>1000 (>255000)
$\frac{\omega}{2\pi}=50$	>1000 (>1000)	>1000 (>3000)	>1000 (>5000)	>1000 (>15000)	817 (208335)

(b) horizontal layers, IBC(0)

Table 3.11: Marmousi model with Neumann condition on top (Figure 3.18(a)). Convergence of the algorithm, with $h\frac{\omega}{2\pi}=200$.

We have seen in Tables 3.8-3.9 that performing full sweeps is extremely slow in the case where $N_{\text{CPU}} = N_{\text{dom}}$ since most CPUs remain idle during the sweeps. Formulas (3.19) give the timings for the case where less CPUs are used than there are domains. In Table 3.13, we report timing estimations for various number of cuts, in the case where only 2 CPUs per group of subproblems are used, in order to guarantee an optimal use of the resources. This example shows that using the preconditioner in that configuration is always more efficient than using the unpreconditioned algorithm with same number of CPUs.

We notice that the influence of the discretization is extremely limited, since the two different discretizations that we have used produce almost identical convergence results, independently of the transmission condition and decomposition

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	50 (50)	19 (57)	13 (65)	8 (120)	188 (188)	70 (210)	27 (243)	9 (567)
$\frac{\omega}{2\pi} = 50$	49 (49)	19 (57)	13 (65)	7 (105)	184 (184)	67 (201)	26 (234)	8 (504)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	650 (650)	240 (720)	150 (750)	57 (855)	10 (2550)
$\frac{\omega}{2\pi} = 50$	684 (684)	243 (729)	151 (755)	56 (840)	11 (2805)
$\frac{\omega}{2\pi} = 400$	646 (646)	236 (708)	146 (730)	54 (810)	15 (3825)

Table 3.12: Marmousi model with Neumann condition on top (Figure 3.18(a)). Convergence of the algorithm, with $h\frac{\omega}{2\pi} = 200$, with vertical layers and PML transmission condition. The iteration count is stable, even at the much higher frequency $\omega = 800\pi$.

#CPU	2	14	42
$N_{\text{it}}^{(\text{np})}$	227		
$T_{\text{sol}}^{(\text{np})}$	7264	1135	454
N_c	0	6	20
$N_{\text{it}}^{(\text{ds})}$	43	54	94
$T_{\text{sol}}^{(\text{ds})}$	4042	702	376

Table 3.13: Marmousi model with Neumann condition on top (Figure 3.18(a)). Normalized timing estimations in function of the number of CPUs, for the case of Table 3.8(a) with 64 subdomains at $\omega/2\pi = 50$. Even though the method does not perfectly scale, the speed-up obtained by increasing the number of CPUs is still interesting.

direction, as can be seen by comparing tables 3.8 and 3.10, as well as 3.9 and 3.11. In the following, only results with the coarsest discretization will be presented.

Another observation is that, despite a few counter-examples, the GIBC(2) transmission condition generally gives faster convergence than the IBC(0), like in our previous test cases. On Table 3.12, we compare the results obtained with a PML transmission condition. Clearly, this type of transmission condition is more adapted to the non-homogeneous medium and gives faster convergence. We also notice the remarkable stability of the transmission condition at very high frequency.

On Figure 3.19, we present results from the previous tables in a more synthetic way. One can clearly observe that performing longer sweeps reduces the number of iterations.

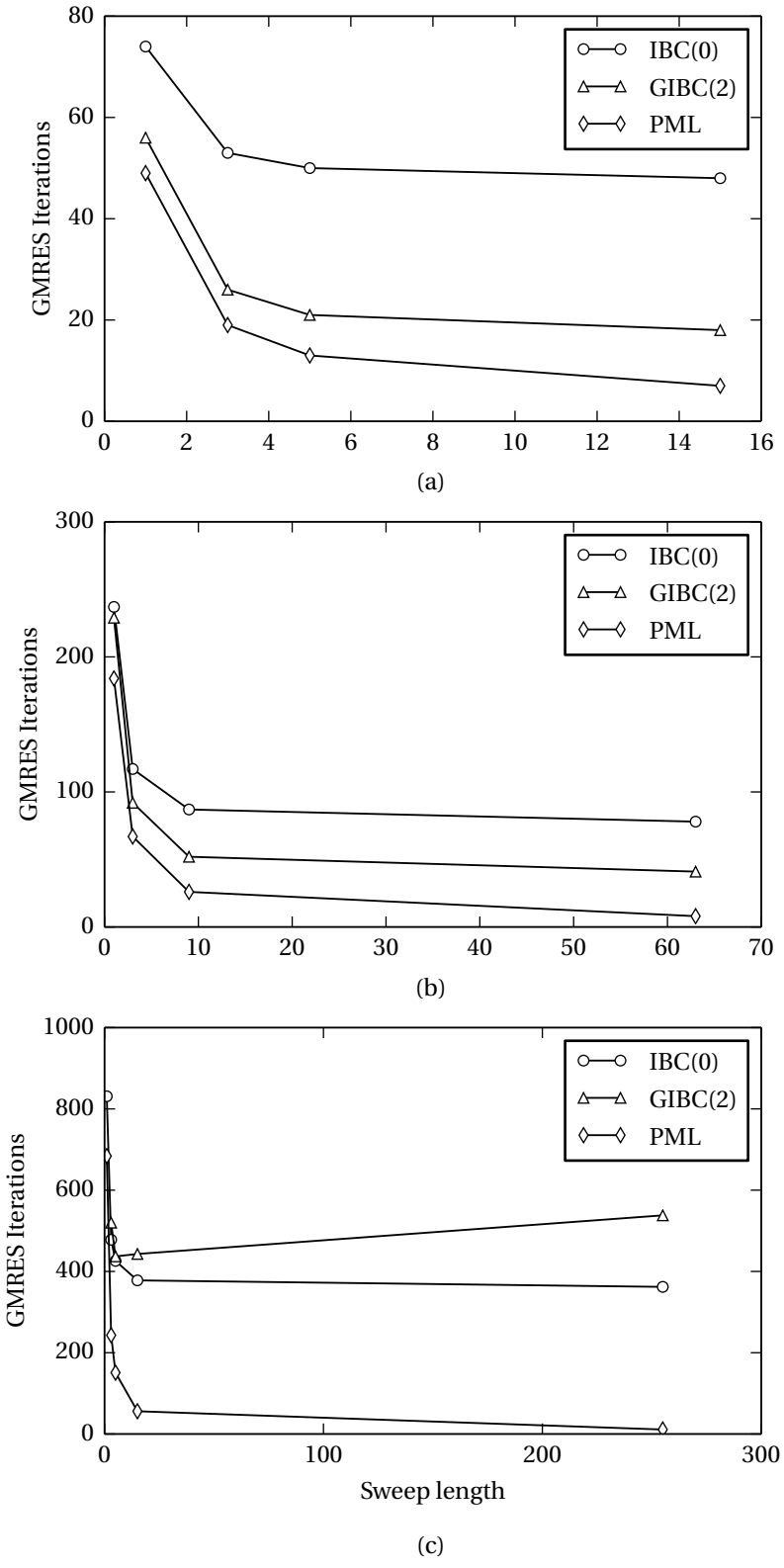


Figure 3.19: Dependence of the length of the sweeps on the iteration count ($\omega = 100\pi$), in terms of number of subdomains: (a) 16 domains, (b) 64 domains, (c) 256 domains. Longer sweeps generally produce faster convergence. The PML is clearly the best transmission condition in this non-homogeneous test case.

Marmousi model with Sommerfeld condition on top

This is the situation that seems to have been simulated in [169]. By comparing these results with the previous ones, one can notice that the convergence is significantly faster in this case. The reflection on the top boundary seem to generate modes that are harder to capture. A possible explanation is to be found in the transmission condition. Table 3.16 shows results with optimized allocation of the CPUs for the preconditioner .

Apart from the faster convergence, the conclusions of the previous section remain mostly unchanged. Let us simply mention the lack of convergence of the preconditioned algorithm with small number of cuts (Table 3.15(b)). This behaviour is unexplained.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	31 (31)	15 (45)	12 (60)	11 (165)	116 (116)	44 (132)	21 (189)	15 (945)
$\frac{\omega}{2\pi} = 50$	32 (32)	15 (45)	13 (65)	12 (180)	115 (115)	44 (132)	21 (189)	15 (945)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	429 (429)	157 (471)	105 (525)	59 (885)	52 (13260)
$\frac{\omega}{2\pi} = 50$	493 (493)	182 (546)	125 (625)	74 (1110)	69 (17595)

(a) vertical layers, GIBC(2)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	65 (65)	38 (114)	35 (175)	27 (405)	277 (277)	124 (372)	74 (666)	58 (3654)
$\frac{\omega}{2\pi} = 50$	67 (67)	40 (120)	40 (200)	30 (450)	273 (273)	134 (402)	86 (774)	72 (4536)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	> 1000 (> 1000)	353 (1059)	235 (1175)	114 (1710)	64 (16320)
$\frac{\omega}{2\pi} = 50$	> 1000 (> 1000)	394 (1182)	275 (1375)	157 (2355)	119 (30345)

(b) horizontal layers, GIBC(2)

Table 3.14: Marmousi model with Sommerfeld condition on top. Convergence of the algorithm in the situation of Figure 3.18(b), with $h\frac{\omega}{2\pi} = 200$.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi}=30$	43 (43)	33 (99)	31 (155)	31 (465)	130 (130)	59 (177)	44 (396)	41 (2583)
$\frac{\omega}{2\pi}=50$	44 (44)	35 (105)	33 (165)	33 (495)	134 (134)	61 (183)	44 (396)	40 (2520)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi}=30$	441 (441)	217 (651)	175 (875)	149 (2235)	141 (35955)
$\frac{\omega}{2\pi}=50$	490 (490)	214 (642)	162 (810)	132 (1980)	126 (32130)

(a) vertical layers, IBC(0)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi}=30$	85 (85)	62 (186)	59 (295)	52 (780)	279 (279)	195 (585)	169 (1521)	161 (10143)
$\frac{\omega}{2\pi}=50$	87 (87)	63 (189)	63 (315)	54 (810)	287 (287)	183 (549)	148 (1332)	140 (8820)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi}=30$	917 (917)	750 (2250)	828 (4140)	>1000 (>15000)	>1000 (>255000)
$\frac{\omega}{2\pi}=50$	981 (981)	826 (2478)	916 (4580)	>1000 (>15000)	>1000 (>255000)

(b) horizontal layers, IBC(0)

Table 3.15: Marmousi model with Sommerfeld condition on top. Convergence of the algorithm in the situation of Figure 3.18(b), with $h\frac{\omega}{2\pi}=200$.

#CPU	2	34	102	170
$N_{\text{it}}^{(\text{np})}$	493			
$T_{\text{sol}}^{(\text{np})}$	63104	3944	1479	986
N_c	0	16	50	84
$N_{\text{it}}^{(\text{ds})}$	69	74	125	182
$T_{\text{sol}}^{(\text{ds})}$	26358	1628	875	728

Table 3.16: Marmousi model with Sommerfeld condition on top. Normalized timing estimations in function of the number of CPUs, for the case of Table 3.8(a) with 256 subdomains at $\omega/2\pi=50$.

Homogeneous model with Neumann condition on top

For comparison purposes, we give the results for the same geometry, but with an homogeneous media. One can notice that the difficulties with the Neumann condition are also present in this case (compare with the next case), despite the fact that the transmission conditions are more appropriate.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	31 (31)	15 (45)	13 (65)	12 (180)	141 (141)	49 (147)	27 (243)	22 (1386)
$\frac{\omega}{2\pi} = 50$	32 (32)	16 (48)	14 (70)	13 (195)	147 (147)	53 (159)	32 (288)	28 (1764)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	503 (503)	184 (552)	120 (600)	76 (1140)	69 (17595)
$\frac{\omega}{2\pi} = 50$	628 (628)	258 (774)	193 (965)	176 (2640)	289 (73695)

(a) vertical layers, GIBC(2)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	94 (94)	33 (99)	28 (140)	24 (360)	391 (391)	159 (477)	95 (855)	104 (6552)
$\frac{\omega}{2\pi} = 50$	95 (95)	36 (108)	30 (150)	27 (405)	516 (516)	210 (630)	201 (1809)	336 (21168)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	>1000 (>1000)	382 (1146)	232 (1160)	104 (1560)	58 (14790)
$\frac{\omega}{2\pi} = 50$	>1000 (>1000)	410 (1230)	270 (1350)	156 (2340)	121 (30855)

(b) horizontal layers, GIBC(2)

Table 3.17: Homogeneous model with Neumann condition on top. Convergence of the algorithm on the homogeneous test case in the situation of Figure 3.18(d), with $h\frac{\omega}{2\pi} = 200$.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	35 (35)	23 (69)	22 (110)	21 (315)	120 (120)	53 (159)	40 (360)	39 (2457)
$\frac{\omega}{2\pi} = 50$	37 (37)	25 (75)	23 (115)	22 (330)	119 (119)	57 (171)	46 (414)	43 (2709)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	384 (384)	236 (708)	207 (1035)	193 (2895)	186 (47430)
$\frac{\omega}{2\pi} = 50$	441 (441)	261 (783)	230 (1150)	223 (3345)	219 (55845)

(a) vertical layers, IBC(0)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	80 (80)	41 (123)	37 (185)	35 (525)	273 (273)	190 (570)	179 (1611)	169 (10647)
$\frac{\omega}{2\pi} = 50$	81 (81)	45 (135)	41 (205)	38 (570)	293 (293)	208 (624)	170 (1530)	164 (10332)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	935 (935)	779 (2337)	874 (4370)	> 1000 (> 15000)	> 1000 (> 255000)
$\frac{\omega}{2\pi} = 50$	976 (976)	807 (2421)	908 (4540)	> 1000 (> 15000)	> 1000 (> 255000)

(b) horizontal layers, IBC(0)

Table 3.18: Homogeneous model with Neumann condition on top. Convergence of the algorithm on the homogeneous test case in the situation of Figure 3.18(d), with $h\frac{\omega}{2\pi} = 200$.

Homogeneous model with Sommerfeld condition on top

This case is close to the propagation of a circular wave in free space. Not surprisingly, it gives the faster convergence. To be noted are still the difficulties with the decomposition in many horizontal layers. However, Table 3.19(b) shows that the preconditioner is quite efficient in this particular case.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	26 (26)	11 (33)	10 (50)	10 (150)	101 (101)	39 (117)	16 (144)	13 (819)
$\frac{\omega}{2\pi} = 50$	26 (26)	12 (36)	10 (50)	10 (150)	102 (102)	38 (114)	17 (153)	14 (882)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	314 (314)	136 (408)	89 (445)	42 (630)	32 (8160)
$\frac{\omega}{2\pi} = 50$	397 (397)	166 (498)	119 (595)	73 (1095)	88 (22440)

(a) vertical layers, GIBC(2)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	74 (74)	33 (99)	28 (140)	23 (345)	256 (256)	121 (363)	81 (729)	76 (4788)
$\frac{\omega}{2\pi} = 50$	71 (71)	36 (108)	31 (155)	26 (390)	359 (359)	172 (516)	183 (1647)	274 (17262)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	795 (795)	251 (753)	152 (760)	76 (1140)	47 (11985)
$\frac{\omega}{2\pi} = 50$	907 (907)	270 (810)	183 (915)	121 (1815)	91 (23205)

(b) horizontal layers, GIBC(2)

Table 3.19: Homogeneous model with Sommerfeld condition on top. Convergence of the algorithm on the homogeneous test case in the situation of Figure 3.18(d), with $h\frac{\omega}{2\pi} = 200$.

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	29 (29)	16 (48)	15 (75)	14 (210)	92 (92)	41 (123)	27 (243)	25 (1575)
$\frac{\omega}{2\pi} = 50$	28 (28)	17 (51)	15 (75)	14 (210)	93 (93)	41 (123)	30 (270)	29 (1827)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	260 (260)	153 (459)	121 (605)	101 (1515)	96 (24480)
$\frac{\omega}{2\pi} = 50$	330 (330)	169 (507)	131 (655)	122 (1830)	118 (30090)

(a) vertical layers, IBC(0)

# doms	16				64			
# cuts	n.a.	4	2	0	n.a.	20	6	0
$\frac{\omega}{2\pi} = 30$	58 (58)	39 (117)	37 (185)	35 (525)	176 (176)	160 (480)	143 (1287)	138 (8694)
$\frac{\omega}{2\pi} = 50$	62 (62)	44 (132)	41 (205)	38 (570)	191 (191)	180 (540)	148 (1332)	138 (8694)

# doms	256				
# cuts	n.a.	84	50	16	0
$\frac{\omega}{2\pi} = 30$	592 (592)	543 (1629)	658 (3290)	> 1000 (> 15000)	> 1000 (> 255000)
$\frac{\omega}{2\pi} = 50$	609 (609)	554 (1662)	644 (3220)	> 1000 (> 15000)	> 1000 (> 255000)

(b) horizontal layers, IBC(0)

Table 3.20: Homogeneous model with Sommerfeld condition on top. Convergence of the algorithm on the homogeneous test case in the situation of Figure 3.18(d), with $h\frac{\omega}{2\pi} = 200$.

CHAPTER 4

Amplitude formulation for the multiple obstacles scattering algorithm

In Section 2.5 we presented the multiple obstacles scattering algorithm for multiple scattering problems, that fits well in our multi-domain framework. The method is based on a covering (Section 2.3.2) of the computational domain, and we concluded the presentation by noticing that, since the subdomains are actually larger than the original domain, the method has little interest if the subproblems are solved using classical finite elements or finite difference formulations. As an alternative, there are different candidate methods for the solution of the subproblems, that include integral representations and asymptotic methods, see [8, 35, 98] for a presentation.

In this Chapter, we focus on the finite element method and propose to take advantage of the nature of these subproblems, that are single scattering problems, to apply a faster solution algorithm to them. Let us mention that it is not the first time that the principle of embedding the computational domain into a larger one to take advantage of an efficient solver is proposed. In [36, 151], the idea was to apply fast Poisson solvers to domains for which the method is not directly applicable. More recently, similar techniques have been proposed for the Helmholtz equation [78, 108]. These methods are known as fictitious domain, domain embedding or capacitance matrix methods.

We will start by presenting an alternative formulation of the Helmholtz problem, applicable to single scattering problems, that allows for a coarser discretization of the subproblems than the standard formulation. Such discretizations will be discussed and the choice of the parameters will be motivated by numerical experiments and efficiency-related arguments. On that basis, we will derive a new implementation of the algorithm that leverages the new formulation to efficiently

produce accurate solutions. After the presentation of some numerical results, we will conclude by the description of some related methods.

4.1 Phase reduction formulation

The basic idea behind the fast solver for time-harmonic single scattering problems was introduced in [6]. It makes use of a representation of the solution $u \in \mathbb{C}(\Omega^+)$ as the product between the oscillatory part and its envelope:

$$u(\boldsymbol{x}) = a(\boldsymbol{x}) e^{\imath k \phi(\boldsymbol{x})}. \tag{4.1}$$

The key point with this representation is that an estimation $\tilde{\phi}$ of the phase ϕ is relatively easy to obtain (Section 4.1.2) in the case of single scattering by a convex obstacle, and that one can solve a problem to find the corresponding amplitude. Also, while $u(\boldsymbol{x})$ is an highly oscillatory function over the domain, requiring fine meshes (typically $n_\lambda = 10 - 20$ points per wavelength, see Section 1.3.5) for an accurate representation, the associated amplitude a and phase ϕ are much smoother functions, especially away from the scatterers boundaries. In expression (4.1), the amplitude function is real-valued only if the phase function corresponds to the exact phase of the field u. If we relax that assumption, that decomposition is no longer unique: suppose that an approximation $\tilde{\phi}$ of the phase is available, it is still possible to decompose the field with a similar expression, using a complex-valued amplitude $\tilde{a}$ that contains the correction to the phase error:

$$\tilde{a}(\boldsymbol{x}) = a(\boldsymbol{x}) e^{\imath k (\phi(\boldsymbol{x}) - \tilde{\phi}(\boldsymbol{x}))}. \tag{4.2}$$

In that case however, the amplitude might oscillate a lot where the phase error (or the gradient thereof) is large, which is of course not desirable, hence the importance of a good phase estimation. Section 4.1.2 is dedicated to a more detailed discussion of this point.

In this Section, we present the Phase Reduction–Finite Element Method (PR-FEM) formulation of the propagation problem; the idea is to solve for the amplitude $\tilde{a}$ of the solution, provided that an estimation of the phase $\tilde{\phi}$ has been obtained in a first step, by means of some (ideally fast) algorithm. If the estimation of the phase is good enough, the corresponding envelope can be expected to oscillate slowly and one can solve for it on a coarse mesh. Once both $\tilde{a}$ and $\tilde{\phi}$ quantities are known, the field of interest u can be reconstructed by plugging them into formula (4.1), interpolating them on an appropriate mesh if need be. Another interest of such a formulation is that it is less subject (even not at all in 1d) to numerical dispersion than the standard formulation [6].

We first introduce what we will call the amplitude formulation of the PR-FEM, before introducing some phase estimation techniques for the single scattering

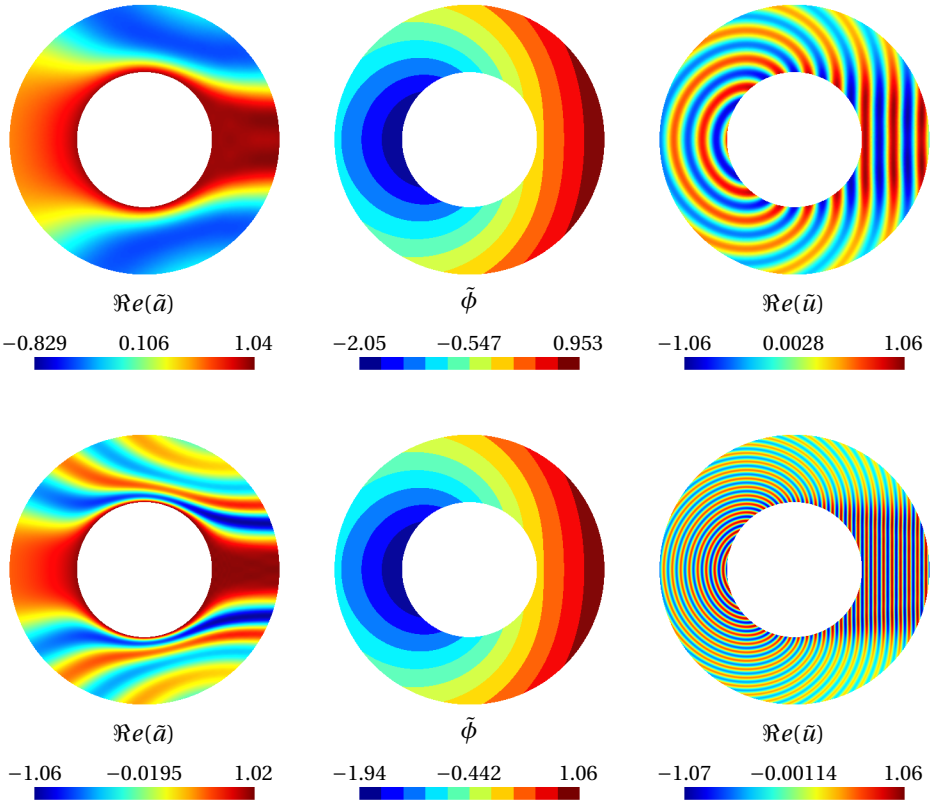


Figure 4.1: From left to right: amplitude, estimated phase and original unknown of the scattered field for a single scattering problem by the unit circle at $k = 15$ (top) and $k = 50$ (bottom). The first two quantities are much less oscillatory than the latter, especially away from the boundary of the object. Some oscillations appear in the amplitude at high frequency, mostly as a consequence of the increased inaccuracy of the low-order phase propagator.

case. We then present a suitable kind of discretization of the problem, that exploits the regularity of the solution in terms of amplitude to significantly reduce the number of unknowns.

4.1.1 Phase reduction formulation

We start by recalling the standard variational formulation of the Helmholtz equation for a scattering problem with an incident field u^{inc}. For the sake of clarity, we consider a simple Sommerfeld approximation of the radiation condition on the truncation boundary Γ_∞. (Higher order conditions like the BGT2 [17] are also applicable, see [6] for a formulation with high order terms.) Supposing we have truncated the domain with the Sommerfeld condition on Γ_∞, the scattered field

verifies:

$$\begin{aligned} -(\Delta + k^2)u &= 0 \qquad \text{in } \Omega; \\ u &= -u^{\text{inc}} \qquad \text{on } \Gamma; \\ (\partial_n - \imath k)u &= 0 \qquad \text{on } \Gamma_\infty. \end{aligned} \tag{4.3}$$

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

$$\begin{aligned} \int_\Omega \quad & \nabla u \cdot \nabla v - k^2 u v \quad d\Omega \\ -\int_{\Gamma_\infty} \quad & \imath k u v \quad d\Gamma \quad = \quad 0, \end{aligned} \qquad \forall v \in H^1(\Omega). \tag{4.4}$$

We reformulate the problem in terms of $\tilde{a} = e^{-\imath k\tilde{\phi}}u$, for some known phase function $\tilde{\phi}$; phase estimation techniques are presented in Section 4.1.2. Injecting that change of unknown in (4.4) and choosing test-functions of the form $\tilde{b} = e^{-\imath k\tilde{\phi}}v$ gives: find $\tilde{a} \in H^1$

$$\begin{aligned} \int_\Omega \quad & \nabla\tilde{a} \cdot \nabla\tilde{b} \\ & - k^2\big(1 - |\nabla\tilde{\phi}|^2\big)\tilde{a}\tilde{b} \\ & + \imath k \nabla\tilde{\phi}\big(\tilde{a}\nabla\tilde{b} - \nabla\tilde{a}\tilde{b}\big) \quad d\Omega \\ -\int_{\Gamma_\infty} \quad & \imath k \tilde{a}\tilde{b} \quad d\Gamma \quad = \quad 0, \end{aligned} \qquad \forall \tilde{b} \in H^1(\Omega). \tag{4.5}$$

This new formulation resembles the classical formulation, with some additional terms; it can still be easily implemented in a standard finite element solver, with some minor modifications. Figure 4.1 shows, for a single scattering example under plane wave illumination, the result of the computation via formulation (4.5) of the envelope $\tilde{a}$ for a low-order phase estimation $\tilde{\phi}$ (see next section), at moderate and high frequencies. The high-frequency example exhibits the typical amplitude oscillations that appear in presence of an erroneous phase estimation, while the output in the low-frequency case is more satisfactory. The next Section proposes more advanced techniques to improve the accuracy of the a priori phase estimation.

Note the similarity with the starting point of geometrical optics presented in Section 1.2.3, with the difference that no high frequency assumption is made to neglect the term in the second derivative in a, and that the phase is supposed to be known.

4.1.2 Phase estimation

We insisted above that a key ingredient for the PR-FEM solution of single scattering problems is the estimation of the phase $\tilde{\phi}$, in the whole computational domain. The authors of [6] propose a starter-propagator approach, where the phase is estimated on the boundary Γ of the scatterer in a first step, then extended in the whole Ω.

The starter in case of a Neumann (sound-hard) problem, approximates the trace of the solution on Γ by applying the On-Surface Radiation Condition (OSRC) technique as approximation of the DtN map, via a Padé rationale approximation of the square root operator [31, 117]. A low-order alternative is to use the Sommerfeld condition to obtain $\tilde{u}_\Gamma$ from $\partial_n u_\Gamma$:

$$\tilde{u}_\Gamma = \frac{1}{\imath k}\partial_n u_\Gamma. \tag{4.6}$$

In case of a Dirichlet (sound-soft) problem, this step is not necessary. The phase on Γ is then extracted via a phase unwrapping technique [6], necessary to avoid the undesired 2π discontinuities resulting from a direct phase calculation by the formula (supposing $u \neq 0$):

$$\tilde{\phi} = \frac{1}{\imath k}\log\left(\frac{u}{\tilde{a}}\right). \tag{4.7}$$

A continuous phase is obtained by differentiating relation $\exp(\imath k\tilde{\phi}) = \frac{u}{\tilde{a}}$ on the (closed) path Γ and taking the real part:

$$\nabla_\Gamma\tilde{\phi} = \Re e\left(\frac{1}{\imath k}\frac{\tilde{a}}{u}\nabla_\Gamma\frac{u}{\tilde{a}}\right) = \mathbf{F}. \tag{4.8}$$

The right-hand side $\mathbf{F}$ of that equation is known. Therefore, after taking the surface divergence of it, one can compute the continuous (unwrapped) ϕ_c by solving a FEM problem with a Dirichlet condition (obtained e.g. via (4.7)) at an arbitrary point $\boldsymbol{x}_0$, via the weak formulation: find $\phi_c \in H^{1/2}(\Gamma)$ s.t.

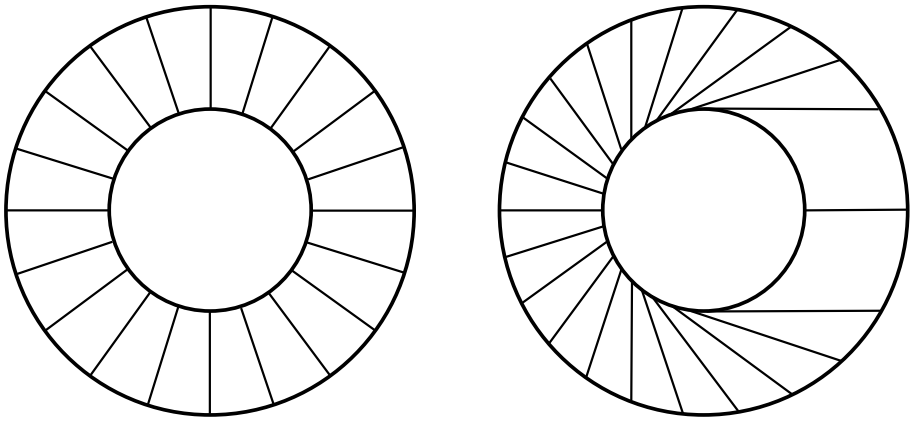
$$\int_\Gamma \nabla_\Gamma\phi_c\cdot\nabla_\Gamma\varphi_c + \nabla_\Gamma\cdot\mathbf{F}\,\varphi_c \;\mathrm{d}\Gamma = 0, \qquad \forall\varphi_c \in H^{1/2}(\Gamma). \tag{4.9}$$

Once the phase is known on Γ, the next step is to extend it in the whole computational domain by means of a propagator, with again different possible techniques. A simple low-order propagator is to assume that the phase is directly proportional to the distance to the obstacle boundary and to approximate it at a point $\boldsymbol{x}$ in Ω away from the boundary by its distance from the boundary:

$$\tilde{\phi}(\boldsymbol{x}) = \tilde{\phi}(\boldsymbol{x}_\pi) + \mathrm{dist}(\boldsymbol{x},\boldsymbol{x}_\pi) \tag{4.10}$$

where $\boldsymbol{x}_\pi$ is the projection of $\boldsymbol{x}$ on the surface, which is uniquely determined for any point of the domain since we have supposed that Γ is convex:

$$\boldsymbol{x}_\pi = \underset{\boldsymbol{x}\in\Gamma}{\arg\min}\,||\boldsymbol{x}-\boldsymbol{x}_\pi||. \tag{4.11}$$



(a) (b)

Figure 4.2: Evolution lines (schematic) of the phase propagator with the simple distance approximation (a) or as solution of the eikonal equation (b), for an incident plane wave coming from the left, scattered by the unit disc. Figure reproduced from [6].

For a more accurate but numerically more costly approximation of the phase, one can solve the eikonal equation as in geometrical optics:

$$|\nabla\tilde{\phi}|^2 = 1 \qquad \text{in } \Omega, \tag{4.12}$$

which is a non-linear partial differential equation. Efficient approximations of the phase field that solves equation (4.12) can be obtained by a fast marching algorithm [163]. Higher order models can also be considered, e.g. the Beam Propagation Method (BPM) [127, 128]. Such methods are beyond the scope of the present work; we again emphasize, though, that with more accurate approximations of the phase $\tilde{\phi}$, less oscillatory envelopes $\tilde{a}$ can be expected to be computed as solution of (4.5). The result is therefore less sensitive to numerical pollution and the overall quality of the final solution u after reconstruction is improved. In the next section, we will also exploit this fact to compute that quantity on a coarser discretization, which is a key ingredient to obtain a computationally efficient method.

4.2 Efficient implementation of the MOSA

The multiple obstacles scattering algorithm is practically unusable with the standard formulation (4.4), because the single scattering subproblems need to be solved on the same grid as the original problem, leading to even larger problems in the covering of Ω by the set of Ω_p defined in Section 2.3.2. In the previous section, we have presented the PR-FEM formulation (4.5) for single scattering problems, that allows for coarser discretizations and makes it possible to solve the high-frequency subproblems efficiently.

While PR-FEM is not suitable for multiple scattering problems, because the phase estimation in that case is difficult, the combination of the MOSA iteration with that formulation leads to a complete and usable multiple scattering FEM solver.

4.2.1 Discretization for PR-FEM and MOSA

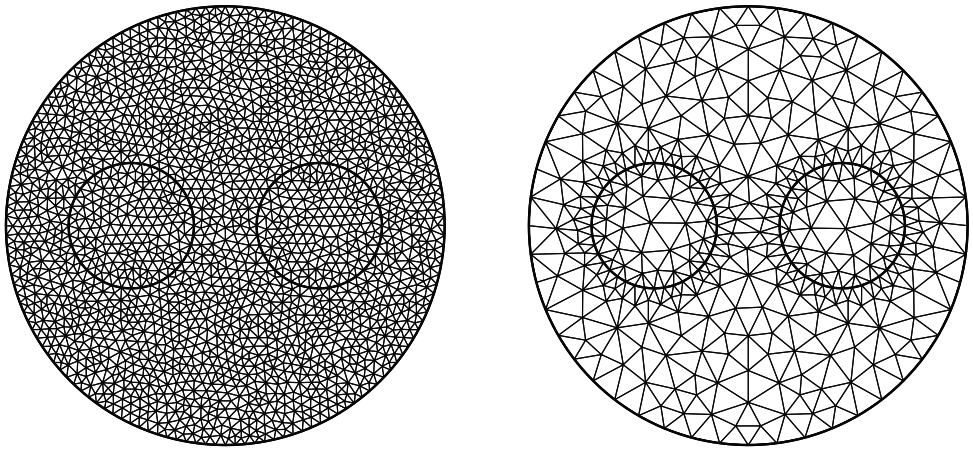


Figure 4.3: Illustration (on a low-frequency example) of the difference between a standard mesh (left) and the mesh used to solve the problem with the MOSA combined with the PR-FEM formulation (right). The grids match on the surface of the scatterers. The standard mesh has 3846 triangles, whereas the PR-FEM mesh has 738 triangles.

To take advantage of the low dispersion properties of the PR-FEM formulation to reduce the computational cost, much coarser meshes than with the standard formulation should be used. However, one must still use sufficiently fine discretizations near the surface of the scattering objects, since an accurate extraction of the phase of the incident wave must be performed in a first step. Since the amplitude and phase are expected to be smooth far off these surfaces, the mesh can be made coarser as one moves away from them. A suitable discretization would therefore be one that progressively coarsens with the distance to the nearest surface, until it reaches a maximum value. The authors of [8] suggest the following mesh density in function of the distance to the nearest boundary:

$$h(\boldsymbol{x}) = \min\left\{\frac{\lambda}{2}, \frac{\lambda}{15} + \operatorname*{dist}_{\Gamma}(\boldsymbol{x})\right\}. \tag{4.13}$$

An example of mesh that follows such a rule is illustrated in Figure 4.3. The actual lower and upper bounds in formula (4.13) should actually result from a compromise between efficiency and accuracy. In our algorithm, the quality of the overall solution is driven by the accuracy of the solution of the coupled problem, defined on the surfaces. Therefore, we prefer to use a finer surface mesh than we would

choose for a single scattering problem, since experiments in Section 4.3.1 show the importance of the surface discretization on the accuracy.

Defining the parameters N_λ and n_λ as respectively the smaller and larger number of discretization points per wavelength (hence leading to the coarsest and finest local mesh densities, resp.), we rewrite rule (4.13) as:

$$h(\boldsymbol{x}) = \min\Big\{\frac{\lambda}{N_\lambda}, \frac{\lambda}{n_\lambda} + \operatorname*{dist}_{\Gamma}(\boldsymbol{x})\Big\}. \tag{4.14}$$

With these notations, we see in Tables 4.1 and 4.2 that the relative increase in the number of degrees of freedom is small when using a finer surface discretization, especially when the domain is large. Therefore, we will use $n_\lambda = 30$.

	$N_\lambda = 30$	15	10	7.5	5	3
$n_\lambda = 15$	–	80300	37150	21900	11300	6200
$n_\lambda = 30$	320500	83350	40300	25500	15000	10000

Table 4.1: Number of degrees of freedom per subproblem. 4 objects, $R_o = 4$, $k = 15$. In comparison, solving the problem on a uniform mesh requires 65500 dofs with $n_\lambda = 15$ and 259355 dofs with $n_\lambda = 30$.

	$N_\lambda = 30$	15	10	7.5	5	3
$n_\lambda = 15$	–	187731	84508	48500	23300	10500
$n_\lambda = 30$	747200	190700	87850	52300	27100	14250

Table 4.2: Number of degrees of freedom per subproblem. 4 objects, $R_o = 6$, $k = 15$. In comparison, solving the problem on a uniform mesh requires 172850 dofs with $n_\lambda = 15$ and about 686000 dofs with $n_\lambda = 30$.

These tables also show that the decrease in number of dofs is fast for small values of the mesh size ratio $r_\lambda = n_\lambda / N_\lambda$, but tends to become approximately linear for larger values. We will see in Section 4.3.2 that the error tends to grow faster for large values of r_λ, so values of $3-5$ are probably a good compromise.

Of course, with a rule such as (4.13) the gain in number of degrees of freedom will depend on the size of the domain. For large domains (i.e. when the distance between the truncation boundary and the objects is large), it will be determined by the largest allowed mesh size (the region with fine mesh being a small subset of the whole domain), whereas for smaller domains this number will be mostly influenced by the smallest mesh size.

To compare the computational cost of the MOSA with PR-FEM and the standard FEM alone, one should take into account the factorization cost of the N subproblems with sizes given in the tables above to the one of the full problems. It is clear that the MOSA becomes especially interesting in large domains, since the

reduction in the number of dofs is more pronounced in such cases. An important feature of the algorithm is that the subproblems are smaller and can be factored in parallel. One should not overlook the cost of the iterations, though this phase is parallelizable as well.

4.2.2 Fast iterations and stabilization

Now suppose that we have designed some method (Jacobi, Gauss-Seidel, a preconditioned GMRES, ...) for solving the coupled interface problem (2.34) and thus have an iteration operator that involves subproblem solves and takes a vector of boundary data as input. An iteration of the algorithm is modified in the following way to incorporate the PR-FEM formulation:

- before each subproblem solve, extract the phase $\tilde{\phi}_c$ using the phase unwrapping method (starter);
- compute the phase in volume with the propagator;
- solve the PR-FEM formulation for the amplitude $\tilde{a}$. This implies assembling and solving a new discrete system, since the phase appears explicitely in the formulation;
- evaluate the solution in terms of the original unknown u on the surface $\Gamma = \cup\Gamma_i$, by injecting $\tilde{a}$ and $\tilde{\phi}$ in relation (4.1).

After convergence, we also compute the solution of the coupled problem in a similar fashion and project on the original mesh to visualize the full solution. As explained in the previous Section, the solutions produced by the two formulations are different. The solutions of the full problem are therefore also (slightly) different, the one obtained with the PR-FEM being supposedly more accurate. Regarding the robustness of the method in function of the solver used, we will see in Section 4.3.2 that its convergence is not much affected by the choice of the solver.

A practical variant of the algorithm above is to conserve the initial phase estimation over the iterations, instead of systematically computing a new one. Indeed, from our observations, the evolution of the phase estimations with a low order propagator becomes quite complicated after a few iterations: fast oscillations appear on the boundary, that propagate in volume. These in turn lead to rapidly changing amplitudes along the same lines, hence missing the goal of being able to capture them on a coarse mesh (Figure 4.4). Eventually, even though the algorithm converges, the reconstructed final solution is totally wrong if the mesh is not able to capture these oscillations. That phenomenon tends to be more pronounced at higher frequencies.

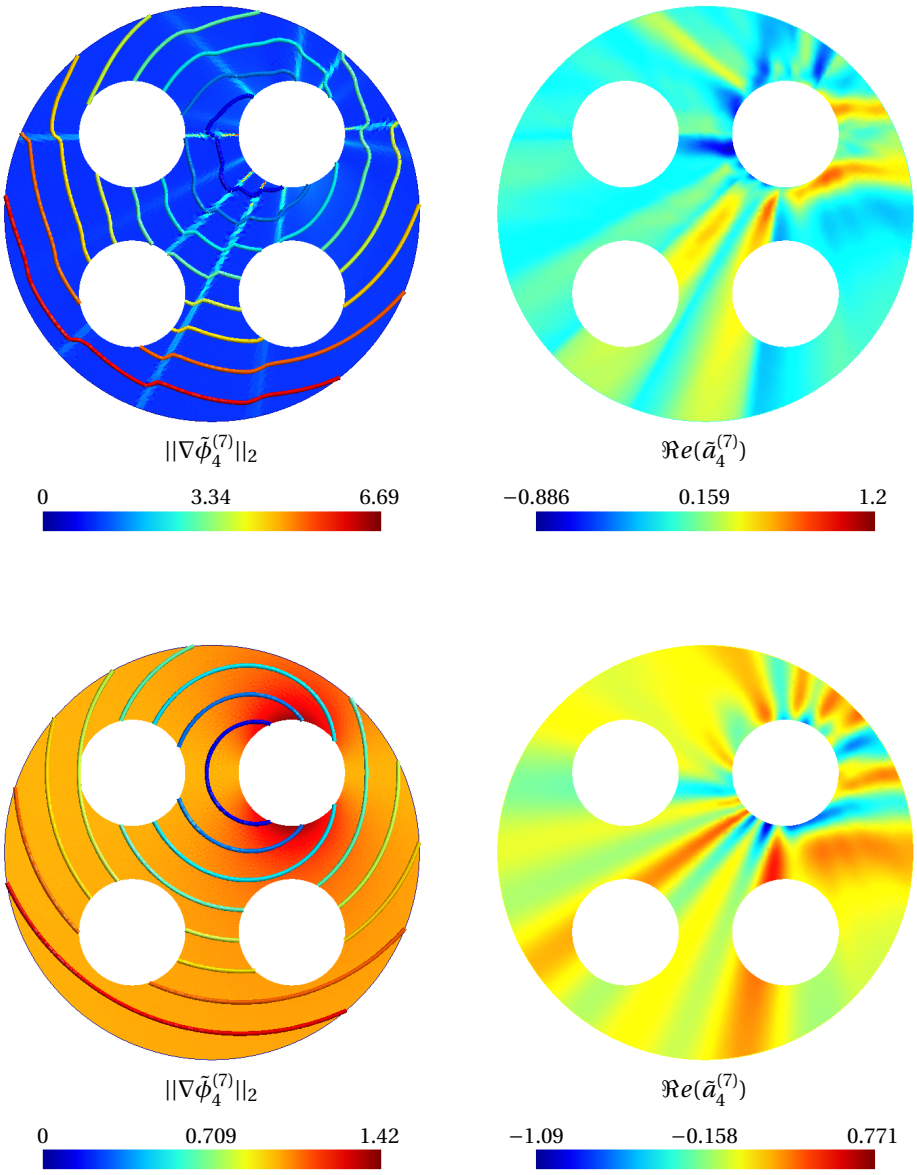


Figure 4.4: The method can be stabilized by not recomputing a phase estimation at each iteration, and keeping the initial estimation instead. Top: iso-values superimposed on the norm of the gradient of the phase and amplitude after 7 iterations with renewed phase estimation. The phase exhibits fast oscillations that are not well represented on the coarsened mesh away from the boundary of the object. They induce aligned amplitude oscillations, resulting in a poor quality reconstructed solution. Bottom: by keeping the initial phase estimation, the amplitude is smoother.

A possible cause of the observed instability is the use of the low-order phase propagator in our computations; we have not verified the behavior of the method with a more accurate propagator such as the one of Figure 4.2(b). Alternatively, we propose to attempt to stabilize the method by skipping the phase-related steps in the iterations above and to keep to the smooth phase estimations $\tilde{\phi}_s(\boldsymbol{x})^{(k)} = \tilde{\phi}_s(\boldsymbol{x})^{(0)}$. Doing so, in addition to saving considerable amounts of work since the amplitude systems to be solved are unchanged through the iterations, results in an amplitude that varies relatively fast in the vicinity of the scatterers boundary but not faster than the original unknown, and becomes smoother away from it. This behaviour is well captured by a discretization such as the one shown in Figure 4.3 (right), provided that the mesh does not coarsen too fast with the distance to the boundary. We found that a modification of rule (4.13) as:

$$h(\boldsymbol{x}) = \min\Big\{\frac{\lambda}{N_\lambda}, \frac{\lambda}{n_\lambda} + \big(\operatorname*{dist}_{\Gamma}(\boldsymbol{x})\big)^c\Big\}, \tag{4.15}$$

with the exponent c between $2-3$ is a good compromise between mesh density and accuracy.

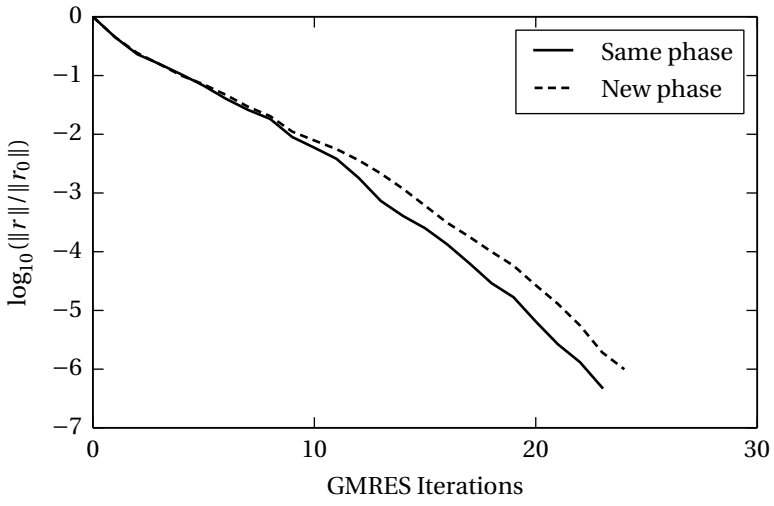


Figure 4.5: Using the initial phase estimation throughout the iterations has several advantages: it leads to a much faster and more stable method, and converges slightly faster.

The method then converges slightly faster (Figure 4.5) and eventually produces the correct solution (Figure 4.6), even at high frequency. In view of the arguments listed above, we will abandon the method with phase estimations at each step and follow our new strategy in all our subsequent numerical experiments.

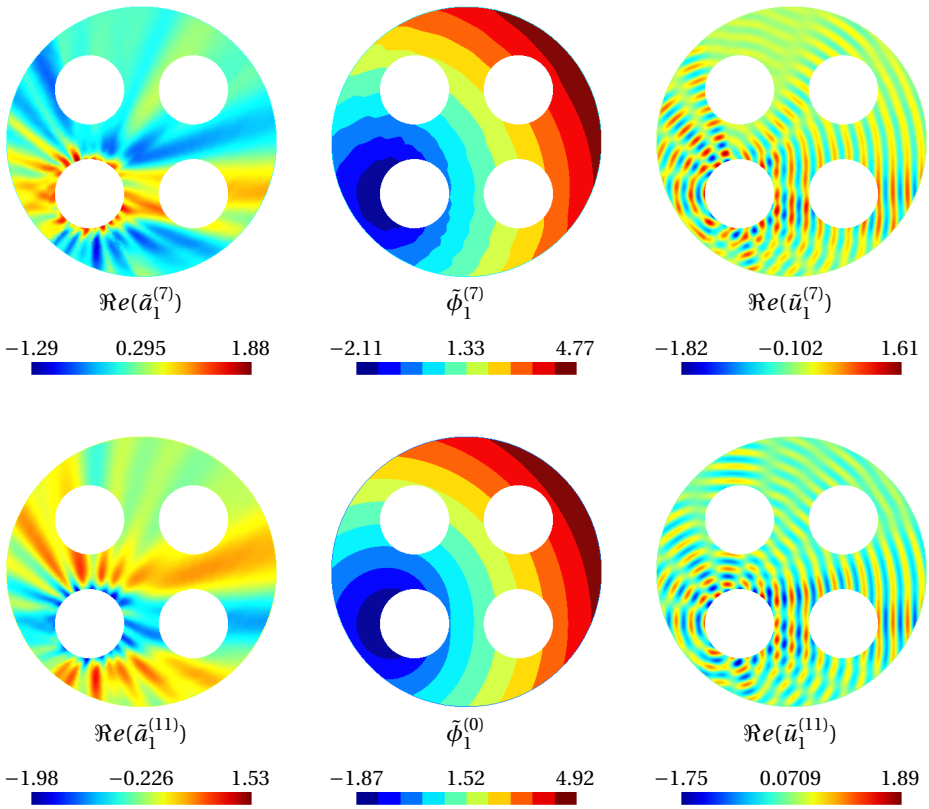


Figure 4.6: Amplitude, phase and original unknown for one of the subproblems after a few iterations ($k = 15$). The $\tilde{a}$ and $\tilde{\phi}$ have been computed by the algorithm on a PR-FEM mesh, while $\tilde{u}$ is reconstructed by interpolating $\tilde{a}$ and $\tilde{\phi}$ on a fine mesh. On the top line, a new phase estimation was computed (by the distance approximation), leading to unstability of the method and wrong solution. On the bottom line, the initial phase estimation was kept throughout the iterations in order to stabilize the method, which produces the correct solution.

4.3 Numerical results

In this section, we present the results of some numerical experiments in relation with the introduction of the PR-FEM formulation in the multiple scattering solver. Some convergence results will be presented, but we will also focus on the evaluation of the error with respect to the coarsening of the mesh, which is a main feature of the proposed method.

To validate our approach, we measure the accuracy of the computed solutions with the new formulation and compare it with a reference solution, which is computed by the classical formulation on a very fine mesh (60 points per wavelength), in order to reduce the effect of numerical dispersion. For the calculations with the PR-FEM formulation, we will make use of the special discretizations introduced

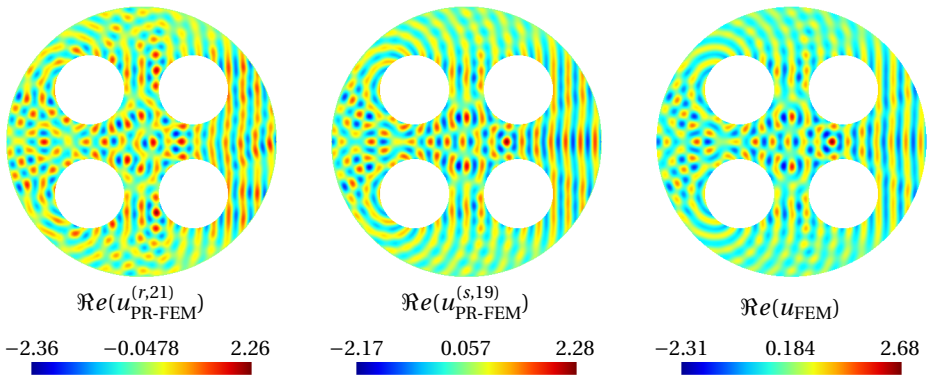


Figure 4.7: Full solution ($k = 15$) with 3 methods. From left to right: converged (tol= 10^{-4}) MOSA solution with PR-FEM and renewed phase estimation $u^{(r,k)}_{\text{PR-FEM}}$, same with conserved phase estimation $u^{(s,k)}_{\text{PR-FEM}}$ and direct FEM solution u_{FEM}. The first two are obtained on a PR-FEM mesh, while the latter was computed on a fine mesh and is used as reference. The solution with renewed phase estimation has required more iterations and is less accurate than the one with conserved phase.

in Section 4.2.1, and compare the evolution of the error for different values of the parameters N_λ and n_λ in rule (4.15) where the exponent has been chosen as $c = 3$, unless otherwise noted.

We will compute relative errors in L^2-norm, on the scattered field u with respect to a reference solution u_{ref}:

$$e = \sqrt{\frac{\int_\Omega |u - u_{\text{ref}}|^2 d\Omega}{\int_\Omega |u_{\text{ref}}|^2 d\Omega}}. \tag{4.16}$$

In a first step, we study the error introduced by the PR-FEM formulation on a single scattering example. We then perform the same kind of analysis on multiple scattering test cases. Finally, we will give some convergence results for the preconditioners discussed in Section 4.4.1.

4.3.1 PR-FEM for single scattering problems

In Tables 4.3 and 4.4, we report the relative error in the case of the scattering by the unit disc, for two different sizes of the computational domain (i.e. we study the same problem and vary the radius R_o of the truncation boundary), since the pollution effect tends to be more pronounced further away from the scattering surfaces. In each case, we use two different mesh densities on the boundary of the object: $n_\lambda = 30$ and $n_\lambda = 15$, and increasing levels of mesh coarsening far from the boundary, following the rule $N_\lambda = n_\lambda / r_\lambda$, where N_λ represents the coarsest discretization density and n_λ remains fixed in the experiment. The case $r_\lambda = 1$ thus

corresponds to a uniform mesh density. (Calculations involving a mesh density smaller than $N_\lambda = 2$ have been skipped.)

	$r_\lambda = 1$	2	3	4	5	10
$n_\lambda = 30$	0.0045	0.0047	0.0066	0.0084	0.0079	0.0273
$n_\lambda = 15$	0.0114	0.0114	0.0105	0.0183	0.0275	–

Table 4.3: Relative errors for different levels of mesh coarsening, at $k = 15$. The computational domain is a disk with unit inner radius and outer radius $R_o = 2$.

	$r_\lambda = 1$	2	3	4	5	10
$n_\lambda = 30$	0.01	0.0095	0.0093	0.0095	0.0097	0.0165
$n_\lambda = 15$	0.0145	0.0132	0.0148	0.0165	0.0178	–

Table 4.4: Relative errors for different levels of mesh coarsening, at $k = 15$. The computational domain is a disk with unit inner radius and outer radius $R_o = 4$.

We observe that the error remains relatively stable when the mesh is coarsened, up to a certain level, confirming that the pollution effect is well reduced and depends only little on the discretization. However, the error is more strongly impacted when the mesh is coarsened on the boundary of the object, which is not surprising since the source field is then less accurately represented.

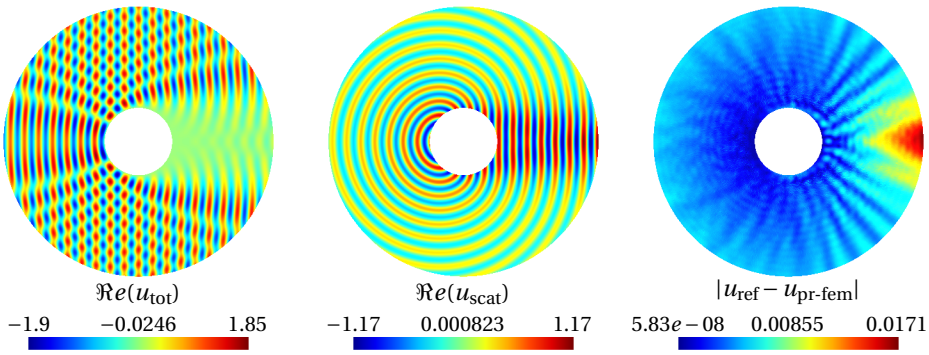


Figure 4.8: Illustration of the error repartition in a domain with size $R_o = 4$. Left: total field u_{tot} with the shadow region clearly visible; middle: scattered field $u_{\text{scat}} = u_{\text{tot}} - u_{\text{inc}}$, with maximal amplitude in the shadow region and decaying amplitude in front of the object; right: the error concentrates in the shadow region, further away from the boundary of the scatterer as an effect of numerical dispersion.

A second observation is that the relative error is larger in the case of the largest computational domain (Table 4.4). This is a consequence of the fact that the error

concentrates in the shadow area, away from the boundary of the object. Indeed, in the case of illumination by a plane wave, the scattered field has maximal amplitude, with no decay, in the shadow behind the object (since it must cancel the incoming field), whereas the reflected wave in front of the object decays with the distance to the boundary. The effect of numerical dispersion, i.e. the discrepancy between the FEM and the PR-FEM solutions, is therefore mostly noticeable in the shadow area, especially away from the object, where the phase error is maximal, as illustrated on Figure 4.8. This effect is less pronounced when the truncation boundary is placed closer to the object.

4.3.2 Multiple scattering

We first investigate the behaviour of the multiple scattering solver with the classical FEM formulation and with the new PR-FEM formulation (on different meshes). We observe on Figure 4.9 that the convergence curves are almost identical during the first iterations, and slightly diverge as the residual decreases. This can be understood as an effect of the coarser discretizations that less accurately represent high frequency features of the solution.

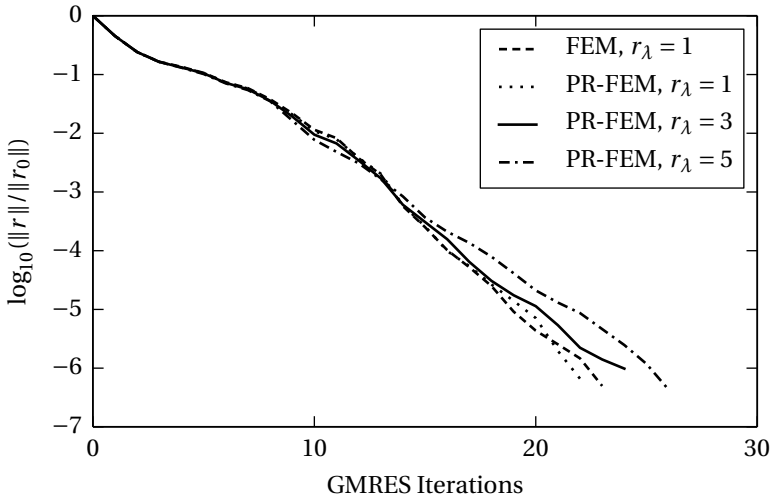


Figure 4.9: Convergence of the multiple scattering algorithm with the classical FEM formulation (on fine mesh), and the PR-FEM formulation on the same grid and on coarser meshes. The convergence is very similar in all cases, demonstrating that the method is quite robust with respect to the solver used for the single scattering problems.

In Table 4.5, we perform the same kind of analysis as in the previous section. Unlike the single scattering case, the error is now much more sensitive to the coarsening of the discretization. The number of iterations for the convergence of the method however remains quite stable. In an attempt to understand the reason for this behaviour, we repeat the experiment, by choosing a much larger value for

	$r_\lambda = 1$	2	3	4	5	10
$n_\lambda = 30$	0.017 (22)	0.0385 (24)	0.0632 (24)	0.0932 (26)	0.1307 (26)	0.2921 (27)
$n_\lambda = 15$	0.0588 (24)	0.1228 (25)	0.1942 (26)	0.2716 (25)	0.3435 (26)	– –

Table 4.5: Relative errors for different levels of mesh coarsening for a multiple scattering problem, at $k = 15$. Values in parentheses are the number of iterations for a residual reduction by 10^6.

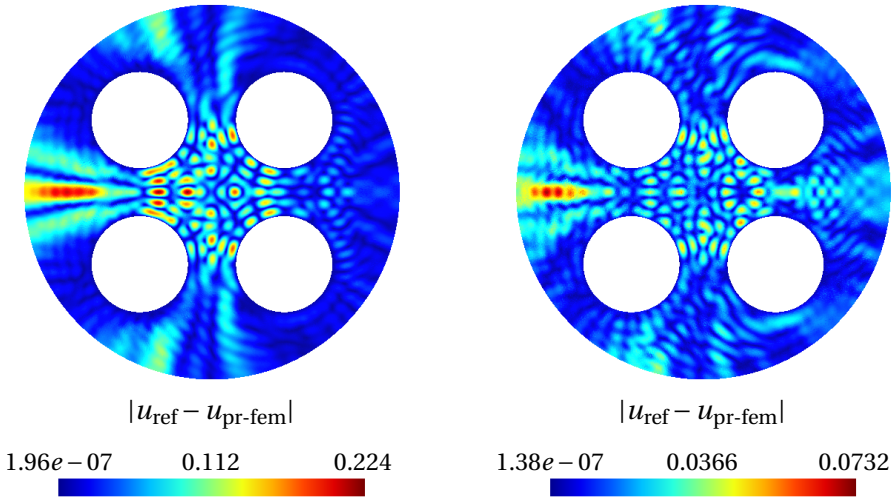


Figure 4.10: Coarse ($c = 3$) versus fine ($c = 15$) discretization in the inner region. The overall amplitude of the error is reduced, especially in the inner region. Both have been obtained with fine mesh density $n_\lambda = 30$ and $r_\lambda = 3$; corresponding errors can be found in Tables 4.5 and 4.6; we note a factor of 3 between the absolute errors in these two examples.

the exponent c in the mesh density formula (4.15), which has the effect of meshing the region between the scatterers more finely.

We now observe that the error is less dependent on the discretization than in the case where we allow a coarse mesh in the inner region, into which most of the exchange of information takes place. It is also at this particular place that resonance phenomena, that are harder to capture numerically, tend to occur. It is therefore natural to maintain a sufficiently fine discretization in that region to guarantee the accuracy of the solution. The impact of this refinement on the convergence rate is very limited.

	$r_\lambda = 1$	2	3	4	5	10
$n_\lambda = 30$	0.017	0.0184	0.0203	0.0239	0.0291	0.0722
	(22)	(22)	(23)	(25)	(25)	(26)

Table 4.6: Relative errors for different levels of mesh coarsening for a multiple scattering problem, at $k = 15$, with a refined mesh in the region between the scatterers ($c = 15$). Values in parentheses are the number of iterations for a residual reduction by 10^6.

	$r_\lambda = 1$	2	3	4	5	10
e_{FEM}	0.049	0.2124	0.4537	0.7195	0.9952	1.2682
#dofs	685725	195603	107358	77266	63575	45867
$e_{\text{PR-FEM}}$	0.0199	0.041	0.0698	0.0993	0.135	0.2749
N_{it}	22	23	25	25	26	27
#dofs ($\approx 4\times$)	746900	222400	129450	98150	83750	65480

Table 4.7: Relative errors on the 4 objects case, at $k = 15$. A larger domain was considered, with external radius $R_o = 6$.

We conclude our analysis by comparing the error of the standard FEM formulation with our algorithm, when applied on similar meshes (Table 4.7). Despite the reduced cost of the standard method, we see that the relative errors are an order of magnitude larger for the standard method, leading to unacceptable values for $r_\lambda > 1$. We see that with a coarsening factor $r_\lambda = 3$, the multiple obstacles scattering algorithm can achieve a comparable error while the 4 problems to be factored are about 5 times smaller, which demonstrates the interest of that method. Furthermore, we have seen that by refining the mesh in the multiple scattering region, the accuracy could ever be improved, at reasonable additional cost.

4.4 Related methods

4.4.1 Preconditioning the MOSA

Since solving a multiple scattering problem with our algorithm involves the solution of the linear system given by (2.34) by some iterative method, it is natural to try and use the preconditioning techniques presented in Section 2.2 to obtain the solution as efficiently as possible. Basic iterative solvers like Jacobi and Gauss-Seidel have been investigated in [99], and their performances have been compared to a Krylov subspace solver (GMRES). We will focus on the latter, because, unlike the basic solvers, it is guaranteed to converge. Indeed, the experiments in [99] have shown that the basic solvers are not stable in the sense that they do not converge at all frequencies. We now investigate some possible techniques for preconditioning the system and discuss their relevance in terms of efficient use of the resources.

We start by recalling the iteration operator of the MOSA:

$$F = \begin{bmatrix} \mathcal{I} & \mathcal{G}_{12} & \cdots & \mathcal{G}_{1S} \\ \mathcal{G}_{21} & \mathcal{I} & \cdots & \mathcal{G}_{2S} \\ \vdots & & \ddots & \vdots \\ \mathcal{G}_{S1} & \mathcal{G}_{S2} & \cdots & \mathcal{I} \end{bmatrix}. \tag{4.17}$$

It is a full matrix and its elements are transfer operators from an interface to another, defined by (2.35). This limits the possibilities of designing a dedicated preconditioner. We will limit our investigation on the basic preconditioners presented in Section 2.2.3: the Gauss-Seidel and symmetrical Gauss-Seidel preconditioners (since the diagonal of the MOSA operator is made exclusively of identity operators, a Jacobi preconditioner would be pointless). Defining the lower and upper triangular parts of F as respectively:

$$F_L = \begin{bmatrix} \mathcal{I} & & & \\ \mathcal{G}_{21} & \mathcal{I} & & \\ \vdots & & \ddots & \\ \mathcal{G}_{S1} & \mathcal{G}_{S2} & \cdots & \mathcal{I} \end{bmatrix}; \qquad F_U = \begin{bmatrix} \mathcal{I} & \mathcal{G}_{12} & \cdots & \mathcal{G}_{1S} \\ & \mathcal{I} & \cdots & \mathcal{G}_{2S} \\ & & \ddots & \vdots \\ & & & \mathcal{I} \end{bmatrix}, \tag{4.18}$$

the Gauss-Seidel preconditioner is given by F_L^{-1}, and the symmetrical version thereof is, in our case, $M_{\text{SGS}}^{-1} = F_U^{-1} F_L^{-1}$. Since they are triangular operators, they can be inverted easily; we will see that we can give the application of these inverse to a vector an interpretation as sequences of solves, as we did for the Schwarz algorithm in Section 3.

We give an example for the case of 3 scatterers. The inverse of F_L is:

$$F_L = \begin{bmatrix} \mathcal{I} & & \\ -\mathcal{G}_{21} & \mathcal{I} & \\ -\mathcal{G}_{31}+\mathcal{G}_{32}\mathcal{G}_{21} & -\mathcal{G}_{32} & \mathcal{I} \end{bmatrix}. \tag{4.19}$$

This example is sufficient to see that our preconditioner for more domains will involve compositions of the transfer operators $\mathcal{G}_{sq}$, while conserving a triangular structure. If we apply it to a vector $r = [r_1, r_2, r_3]^T$, we find the recurrence relation for $r' = F_L^{-1} r$:

$$\begin{aligned} r'_1 &= r_1 & &= r_1 \\ r'_2 &= r_2 - \mathcal{G}_{21} r_1 & &= r_2 - \mathcal{G}_{21} r'_1 \\ r'_3 &= r_3 + (-\mathcal{G}_{31} + \mathcal{G}_{32}\mathcal{G}_{21}) r_1 - \mathcal{G}_{32} r_2 & &= r_3 - \mathcal{G}_{31} r'_1 - \mathcal{G}_{32} r'_2, \end{aligned} \tag{4.20}$$

which is easily extended to the case of more domains:

$$r'_s = r_s - \sum_{q=1}^{s-1} \mathcal{G}_{sq} r'_q. \tag{4.21}$$

The application of F_U^{-1} is very similar. Recalling that the output of $\mathcal{G}_{sq} r'_q$ is obtained for all s at once by solving a single problem (2.35), we can write routines for the application of our preconditioners as Algorithms 4.1 and 4.2, that involve the sequential solution of $N-1$ (resp. $2(N-1)$) problems.

It is clear from the structure of Algorithms 4.1 and 4.2 that they are sequential by nature and cannot be run in parallel, unlike the iteration operator 2.5. Since each sweep involves the solution of $N-1$ subproblems, it is clear that the duration of an iteration of the preconditioned algorithm is doubled (resp. tripled) in a sequential implementation, and multiplied by N in a parallel implementation. Therefore, using such preconditioners seems to have little practical interest since the number of iterations decreases only by a factor of 2 in the case of the Gauss-Seidel preconditioner, and by less than 3 in the case of the symmetric Gauss-Seidel, as will be seen on a numerical experiment in Section 4.4.1. Alternatively, one could envision an hybrid method that makes use of fast approximations (e.g. integral or asymptotic methods, see Sections 1.2.1 and 1.2.3) to speed up the solution of the single scattering subproblems in the preconditioners, e.g. an asymptotic method suitable for high frequencies.

Another option is to adopt a less column-oriented strategy in the matrix-vector products, by splitting the application of the $\mathcal{G}_{\bullet q}$ operators and having dedicated solvers for each transfer operator instead of applying them all at once. This makes sense, since approximately half of the information so obtained is discarded in the

Algorithm 4.1: Gauss-Seidel preconditioner for the MOSA: $r \leftarrow F_L^{-1} r$

for $i = 1 : N-1$
  // Solve subproblem
  $u_D \leftarrow r_i$ on Γ_i
  Solve $\mathcal{H}_i u_i = 0, \quad$ s.t. $u_i = r_i$ on Γ_i
  // Triangular update
  for $j = i+1 : N$
    $r_j \leftarrow r_j - u_i|_{\Gamma_j}$
  end
end

Algorithm 4.2: Symmetric Gauss-Seidel preconditioner for the MOSA: $r \leftarrow F_U^{-1} F_L^{-1} r$

$r_1 \leftarrow r_1$
// Forward sweep
for $i = 1 : N-1$
  $u_D \leftarrow r_i$ on Γ_i
  Solve $\mathcal{H}_i u_i = 0, \quad$ s.t. $u_i = r_i$ on Γ_i
  for $j = i+1 : N$
    $r_j \leftarrow r_j - u_i|_{\Gamma_j}$
  end
end

// Backward sweep
for $i = N : 2$
  $u_D \leftarrow r_i$ on Γ_i
  Solve $\mathcal{H}_i u_i = 0, \quad$ s.t. $u_i = r_i$ on Γ_i
  for $j = i-1 : 1$
    $r_j \leftarrow r_j - u_i|_{\Gamma_j}$
  end
end

triangular update, while the fine discretizations on the surfaces make the overall solves quite costly. Also, it would reintroduce some parallelism in the process. Finally, it would enable the implementation of preconditioners based on the interaction distance between any two obstacles as proposed in [8].

A conclusion of this discussion is that the current implementation with simultaneous transfers by solving a full domain problem (in Ω_q^+ for $\mathcal{G}_{\bullet q}$) is clearly suboptimal in the context of preconditioners, even with the PR-FEM, and is not flexible enough. Though the above ideas or a combination of them could potentially reduce the preconditioners application cost, they are left for further investigation, as well as the design of more advanced preconditioning strategies. A currently unexplored way is the construction of some deflation space by using e.g. plane waves, like has been proposed in the DDM context [137, 172]. The balancing Neumann-Neumann preconditioner [74] could be a good candidate.

We now give some results and compare the convergence of the methods resulting of the implementation of these preconditioners. Since the preconditioned systems have same solution as the initial one, there is no need to study the error here, and only convergence results are presented.

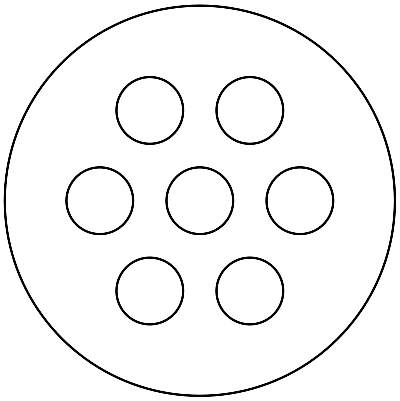


Figure 4.11: Geometry of the scattering problem by 7 objects used to test the method and the preconditioners.

We used the more complex test case of Figure 4.11, with 7 scattering objects. Figure 4.12 shows that in all cases the convergence is smooth, with significantly improved rates when the preconditioners are used, which is the expected result. In Table 4.8, we report iteration counts on the same test case at increasing frequencies.

One can notice that in the unpreconditioned case, the iteration count tends to

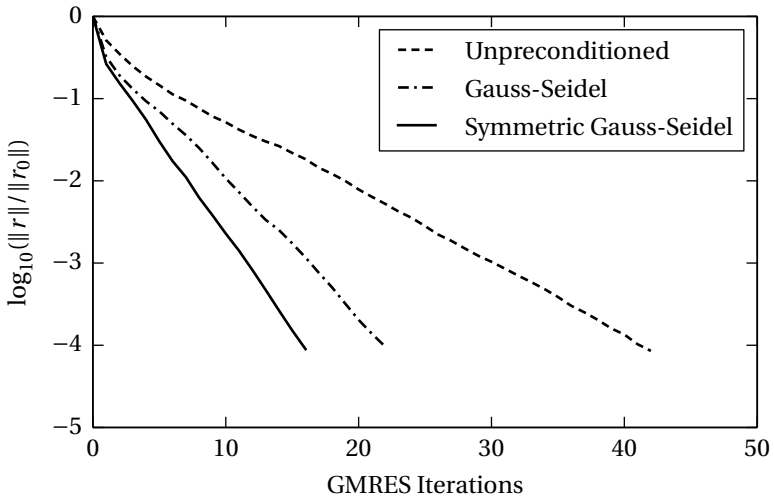


Figure 4.12: Convergence of the MOSA with different preconditioning strategies. All three methods converge smoothly, with improved rates for the preconditioned versions. The cost of the preconditioner applications is however a limiting factor. The test case is the one shown on Figure 4.11 with $k = 45$.

	$k = 15$	30	45	60	75
Unpreconditioned	40	39	42	50	55
Gauss-Seidel	22	20	22	26	29
Symmetric G-S	17	16	16	21	22

Table 4.8: Iteration counts (10^{-4}) of the MOSA with different preconditioners. The algorithm is quite stable with the wavenumber. The test case is the one shown on Figure 4.11. Right preconditioning was used to make the residuals comparable.

grow significantly with the wavenumber, while the preconditioners (especially the symmetric Gauss-Seidel) seem to make that dependency less pronounced, while the iteration count is well reduced in both cases. As discussed above, however, the current implementation does not allow an efficient application of such preconditioners, and their cost is comparable or higher (for the symmetric) to that of the full operator. However the introduction of alternative solvers in the preconditioners, resulting in a hybrid method, could help to make the overall method more computationally attractive.

4.4.2 Macro Basis Functions

The multiple obstacles scattering algorithm, like the other multi-domain methods presented in this work, solves a problem over surface unknowns and then reconstructs the solution in volume. However, at each iteration the subproblems are actually solved in volume and the surface data is extracted, while the rest of the

solution is discarded. From this observation, we attempted in [181] to define an alternative method to take advantage of the available volumic data. The idea is to iterate as in the usual algorithm, with the difference that the subsolutions of the subproblems are progressively collected and orthogonalized to build a subspace, into which the solution to the full problem is sought after completion of each iteration.

This method is inspired from the concept of Macro Basis Functions (MBFs) [46, 104], also referred to as Characteristic Basis Functions (CBF) [184], that are a class of basis functions whose support is the entire computational domain Ω_+. That property contrasts with standard finite element basis functions, the support of which is generally restricted to a few mesh elements (e.g. those touching a given mesh vertex for standard P1 elements). The MBFs are constructed as a linear combination of the standard basis functions, and can thus be seen as higher level basis functions, built on top of the standard ones. As such, if we denote by $W^0(\Omega_+)$ the N-dimensional finite element space in which we are seeking the discrete solution of the problem, a reduced set of P linearly independent MBFs spans a P-dimensional subspace of solutions $W^{\mathrm{MBF}}(\Omega_+) \subset W^0(\Omega_+)$. An approximation $\tilde{u}(\boldsymbol{x})$ of the solution to the original problem can of course be expressed in Ω_+ as a linear combination of the P MBFs:

$$\tilde{u}(\boldsymbol{x}) = \sum_{k=1}^{P} \alpha_k u_k(\boldsymbol{x}). \tag{4.22}$$

In matrix form, we can write

$$\tilde{u} = P_{\mathrm{MBF}}\, \alpha, \tag{4.23}$$

where $P_{\mathrm{MBF}} = [u_1, \ldots, u_P]$ is the MBF basis of $W^{\mathrm{MBF}}(\Omega_h^+)$, and $\alpha = [\alpha_1, \ldots, \alpha_P]$ is the vector of coordinates of $\tilde{u}(\boldsymbol{x})$ in that basis.

Although that subspace does not necessarily contain the standard FE solution, its orthogonal projection onto this subspace is its best approximation in the least-square sense. If the discretized problem to solve reads

$$Au = b, \tag{4.24}$$

and a basis P_{MBF} of dimension P is available, the coordinates α of the projected solution in (4.23) are obtained by solving the normal equation:

$$A_{\mathrm{red}}\, \alpha = P_{\mathrm{MBF}}^T\, b, \tag{4.25}$$

with $A_{\mathrm{red}} = P_{\mathrm{MBF}}^T A P_{\mathrm{MBF}}$ the reduced matrix of the new system. Although this matrix $P \times P$ is dense, it should be much smaller than the original system matrix $N \times N$. Equation (4.25) can thus be straightforwardly solved to find the approximated solution, without solving the original system.

Of course, the success of this approach strongly relies on an appropriate choice of the basis. By noticing that the MOSA produces the terms of decomposition (2.36), one can expect them to form a suitable basis into which a good quality approximation can be found after a few iterations.

While the method seems to work when the standard formulation (4.4) is used to solve the subproblems (which is extremely costly), it fails when the PR-FEM is introduced. This is because the full problem to be solved is defined by the standard formulation; subsolutions obtained with the same formulation verify that equation (in particular, they undergo the same numerical dispersion), and their residual therefore vanishes away from the boundaries. Subsolutions obtained with an alternative solver do not have that property and consequently the subspace they span does not contain the solution to the full problem, thereby limiting the accuracy.

Conclusion

The size of the linear systems arising from the discretization of the PDEs that describe propagation phenomena limits their solution by standard factorization methods based on gaussian elimination, even with state-of-the-art multifrontal solvers: at high frequencies, their solution becomes so demanding in terms of computation time and memory that it becomes practically impossible to handle, at least in an industrial context. Usual iterative methods and preconditioners for the discrete operator also fail to solve such problems.

Therefore, computational engineers are looking for alternative methods, that can better take advantage of the power of massively parallel computer architectures. An idea is to exploit the direct solvers on subsystems of manageable size, while an external loop would try to converge towards the global solution by combining the solutions of these subsystems. The methods investigated in this work belong to that category of "hybrid" direct/iterative solvers.

Multi-domain methods

We have proposed a common framework for a class of methods that reformulates, at the continuous level, the problem in terms of unknown surface sources in a set of new domains, such that they produce an equivalent solution once combined. Since the new problem is formulated as a linear system that can be solved by iterative solvers, we found it natural to look for a preconditioner to speed up its convergence. This is different than directly preconditioning the discretized operator, since we first design a new solver with new unknowns for the problem, and then try to improve it by adding a second level.

Optimized Schwarz method

Our sweeping preconditioner for the non-overlapping optimized Schwarz method is based on an improved exchange of information between distant subdomains; with this observation, we think that it makes sense to make the link with the techniques of "coarse grids" that were previously introduced in the literature and that achieve such global communication.

To our knowledge, preconditioning such a matrix-free iteration operator by means of an approximation of it, also expressed in terms of a combination of subproblem solves as we did for the domain decomposition algorithm, is an original approach. We have presented two of such methods, but any other method that fits into the framework could benefit from a similar kind of preconditioning. Of course, other preconditioning strategies could also be employed.

We have shown with various numerical tests on both scalar (acoustic) and vector (electromagnetic) test cases that the proposed preconditioner allows to substantially reduce the number of iterations, when a good approximation of the transmission operator between the subdomains is available. Such approximations are easy to obtain when the propagation medium is homogeneous; the construction of efficient approximations for non-homogeneous media remains an open problem. In addition to reducing the number of iterations, the preconditioner can also drastically reduce the time-to-solution and exploit parallelism when solving problems with multiple right-hand sides, using pipelining. For single right-hand side problems, we have shown that some level of parallel efficiency can be restored by doing partial sweeps over groups of subdomains.

Multiple obstacles scattering algorithm

Another advantage of the multi-domain approach is that it can enable faster solution techniques for the subproblems that would not be applicable for the full problem, like in the case of the algorithm for multiple scattering problems. Note that this is independent of the use of a preconditioner for the resulting method. In the case of a modified subproblem solver, the final (combined) solution can be different than the solution obtained by a direct solver and the standard formulation, but still constitutes a valid solution that can even be more accurate.

Perspectives for future research

Below is a list of future directions of research or improvements that could be made to our methods:

- we have seen that our preconditioners tend to be sequential, which makes their application quite expensive on parallel machines, unless computations are pipelined or cuts are introduced to make them more parallel. Independently, another way of speeding up their application would be to use fast solvers in the preconditioner stage. The resulting loss of accuracy would decrease the efficiency of the preconditioner and result in an increased iteration count, but this could be compensated for by the gain in application time and result in a shorter time to solution;
- since we iterate over surface data, unknowns in volume are discarded after the solution of the subproblems. Instead, one could eliminate these by com-

puting the (factorization of the) Schur complement of the surface unknowns and solving for these only. This could potentially be implemented efficiently due to the fact that the right-hand sides for all the subproblems are sparse (the subproblems only have surface sources). The solution in volume of the subdomains should only be computed after convergence of the solver, for the construction of the full solution;

- we have seen that propagation problems are more easily solved when the medium is dissipative, and that it is also true for domain decomposition methods. Since the Laplace-shifted preconditioner approximates the operator by a similar one with dissipation, a candidate solver for the shifted problem could be a multi-domain method.

- the efficiency of the domain decomposition algorithm is strongly influenced by the quality of the approximation of the DtN map. While good local approximations are available for homogeneous media, we have found that these are less well-adapted to the non-homogeneous case. Developing such approximations specifically for this situation would greatly improve the efficiency of the method and the double sweep preconditioner, since it is based on the assumption of an ideal transmission condition. Using PMLs for that purpose seems a promising solution, although it currently requires either a more complex implementation or a modification of the geometry to physically append the PMLs. In Appendix C, we give some details on the two different ways of constructing a non-local approximation of the DtN map by means of PMLs, in a finite element context. As both approaches are computationally expensive, efficient implementations would be attractive. An interesting direction of research would be to further investigate the matrix probing technique.

APPENDIX A

Formal construction of the double sweep preconditioner

Here we detail how the matrix of the iteration operator F_A (3.6) was formed in Section 3.1.3, when analytical solutions of the PDEs in the Schwarz algorithm are used. Applying algorithm (2.23–2.24) with Krylov acceleration, we have to solve the subproblems (recall that external sources are cancelled in the definition of operator $\mathcal{F}$ and that we use $\mathcal{S} = \mathcal{D} = -\imath k$, so we apply the same condition on the artificial interfaces than on the external boundaries):

$$\begin{array}{rcll} -(\partial_{xx} + k^2)u_j & = & 0 & \text{in } \Omega_j \\ (\partial_n - \imath k)u_j & = & g_{ji} & \text{on } \Sigma_{ij} \\ (\partial_n - \imath k)u_j & = & 0 & \text{on } \{x_l, x_r\}. \end{array} \tag{A.1}$$

The solutions of these problems are a superposition of a forward and a backward wave:

$$u_j(x, g_l, g_r) = A_{f,j} \exp \imath k x + A_{b,j} \exp -\imath k x, \quad x \in \Omega_j, \quad j = 1, \ldots, N,$$

with respective amplitudes given as functions of the impedance data g_l and g_r on the artificial boundary (we suffix by l and r quantities or coordinates associated with resp. the left and right side of the considered domain):

$$A_{f,j} = -\frac{g_l}{2\imath k} \exp \imath k x_l \quad ; \quad A_{b,j} = -\frac{g_r}{2\imath k} \exp \imath k x_r, \tag{A.2}$$

While the matrix is easy to form numerically, its formal expression is rather technical and requires some definitions. We first number the unknowns of the Schwarz algorithm as $g = [g_{12}, g_{21}, g_{23}, \ldots]^T$, where an unknown g_{ij} corresponds to the

impedance data of the boundary condition for problem i, on Σ_{ij}. There are two unknowns per artificial interface, for a total of $M = 2(N-1)$ unknowns. Then, we classically number the entries of the matrix as:

$$F = \begin{bmatrix} F_{11} & \dots & F_{1M} \\ \vdots & \ddots & \vdots \\ F_{M1} & \dots & F_{MM} \end{bmatrix},$$

where entry F_{mn} refers to the m-th row and n-th column. So we have two different ways of indexing the unknowns: the "matrix index" g_m and the "problem index" g_{ij}. As both are convenient depending on the context, we define the index mappings:

$$m(i,j) = \begin{cases} i+j-1 & \text{if } i > j \\ i+j-2 & \text{if } i < j \end{cases};$$

$$[i,j](m) = \begin{cases} [\frac{m+1}{2}, \frac{m+3}{2}] & \text{if } m = 1,3,\cdots \\ [\frac{m}{2}+1, \frac{m}{2}] & \text{if } m = 2,4,\cdots \end{cases}$$

Each unknown g_{ij} has a companion unknown g_{ji} associated to the same interface Σ_{ij}. In the matrix indexing, we will write g'_m the unknown associated to g_m, with the index relation:

$$m' = m - (-1)^m.$$

The column $F_{\cdot,n}$ is the output of the update relation (2.27) applied to the n-th column of an identity matrix as source vector g^k, so the m-th entry of that column will be:

$$F_{mn} = \delta_{mn} + \delta_{m'n} + 2\imath k u_{j(m)}(x_{\Sigma_{ij(m)}}, \delta_{2(j(m)-1),n}, \delta_{2j(m)-1,n}). \tag{A.3}$$

In the n-th column, at most 3 entries have a non-zero contribution from at least one of its terms: the n-th entry (the one on the diagonal of the matrix) is exactly 1, as only the δ_{nn} term contributes; the n'-th is 0, because the $\delta_{m'n}$ term is exactly canceled by the problem contribution as an effect of the exact transmission condition; and the $(n \pm 2)$-th only has the problem contribution alone. The matrix therefore has the following structure for N subdomains:

$$F_A(N) = \begin{bmatrix} 1 & 0^* & b_2 & & & & \\ 0^* & 1 & 0 & & & & \\ & 0 & 1 & 0^* & \ddots & & \\ & b_2 & 0^* & 1 & & & \\ & & \ddots & & \ddots & b_{N-1} & \\ & & & & 0 & 0 & \\ & & & & 0 & 1 & 0^* \\ & & & & b_{N-1} & 0^* & 1 \end{bmatrix},$$

where parameters b_i depend only on the size of the subdomains $\Delta_i = x_{\Sigma_{i+1}} - x_{\Sigma_i}$ (we introduce the simplified notation $\Sigma_k = \Sigma_{ij}$, with $k = \min(i, j)$): $b_i = \exp \imath k \Delta_i$. The 0^* entries indicate values that are 0 as the result of the cancellation of 2 contributions in expression (A.3). (We will see that in the numerical solution case, they are no longer 0.) The condition number of $F_A(N)$ is small. For example, for the case of 3 subdomains, the matrix is:

$$F_A(3) = \left[\begin{array}{cc|cc} 1 & 0^* & b_2 & \\ 0^* & 1 & 0 & \\ \hline & 0 & 1 & 0^* \\ & b_2 & 0^* & 1 \end{array}\right].$$

One can easily verify that all eigenvalues are equal to 1 ($\lambda_{1-4} = 1$) and that only 2 independent eigenvectors exist: $[1,0,0,0]$ and $[0,0,0,1]$. They are indeed the only possible invariants since the wave they produce leaves the domain via the external boundaries instead of coupling to other subdomains. Since b_2 has unit modulus ($|b_{i,1<i<N}| = 1$), we have for the condition number, in spectral norm:

$$||F_A(3)||_2 = ||F_A^{-1}(3)||_2 = \sqrt{\frac{1}{2}|b_2|^2 + \frac{1}{2}\sqrt{|b_2|^4 + 4|b_2|^2} + 1};$$

$$\begin{aligned} \kappa_2(F_A(3)) &= ||F_A(3)||_2 ||F_A^{-1}(3)||_2 \\ &= \frac{|b_2|^2}{2} + \frac{\sqrt{|b_2|^4 + 4|b_2|^2}}{2} + 1 \\ &= \left(\frac{|b_2|}{2} + \sqrt{\frac{|b_2|^2}{4} + 1}\right)^2 = 2.618. \end{aligned}$$

The condition number grows linearly with the number of subdomains, as shown by Table A.1.

N	2	3	4	5	10	20	50	100
κ_2	1	2.62	4.09	5.41	11.94	24.57	63	126.67

Table A.1: Condition number of the iteration matrix $F_A(N)$ for increasing number of subdomains.

APPENDIX B

Integral representation of the fields (scalar case)

This Section is a very succinct presentation of the potential theory and integral methods, and is not meant to be exhaustive. In particular, questions related to the solvability of the integral equations are not discussed. For a detailed and rigorous presentation, we refer to [44, 139].

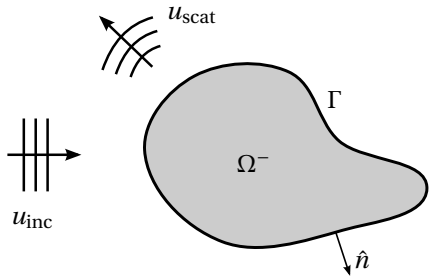


Figure B.1: Scattering by a single object Ω^- with boundary $\Gamma = \partial\Omega^-$. The computational domain of the exterior problem is $\Omega = \mathbf{R}^d \setminus \Omega^-$.

In the practically important case of an homogeneous propagation medium (e.g. air, water or vacuum), the scattering problem by a conducting object defined as (see Figure B.1):

$$\begin{aligned} -(\Delta + k^2)u &= 0 \qquad && \text{in } \Omega^+ = \mathbb{R}^d \setminus \Omega^-; \\ u &= -u_{\text{inc}} && \text{on } \Gamma, \end{aligned} \tag{B.1}$$

together with radiation condition (1.3), admits as exact solution for the scattered

field the integral representation:

$$u(\boldsymbol{x}) = -\int_\Gamma G(\boldsymbol{x},\boldsymbol{y})\ \partial_{n_y} u|_\Gamma(\boldsymbol{y})\ d\Gamma + \int_\Gamma \partial_{n_y} G(\boldsymbol{x},\boldsymbol{y})\ u|_\Gamma(\boldsymbol{y})\ d\Gamma, \quad \forall \boldsymbol{x} \in \Omega. \tag{B.2}$$

This result is known as the representation theorem [139]. It features the Green function (or fundamental solution) of the Helmholtz operator $G(\cdot,\boldsymbol{y})$ centered on $\boldsymbol{y}$ in $\mathbb{R}^d$(1.2), that verifies:

$$\begin{cases} -(\Delta + k^2)G(\cdot,\boldsymbol{y}) = \delta_{\boldsymbol{y}} & \text{in } \mathbb{R}^d, \\ \qquad\qquad G(\cdot,\boldsymbol{y}) & \text{is outgoing.} \end{cases} \tag{B.3}$$

It admits analytical expressions:

$$\forall \boldsymbol{x} \neq \boldsymbol{y}, \qquad G(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \frac{\imath}{4} H_0^1(k\|\boldsymbol{x}-\boldsymbol{y}\|) & \text{if } d = 2, \\ \frac{e^{\imath k\|\boldsymbol{x}-\boldsymbol{y}\|}}{4\pi\|\boldsymbol{x}-\boldsymbol{y}\|} & \text{if } d = 3. \end{cases} \tag{B.4}$$

These functions decay respectively as $\frac{1}{\sqrt{r}}$ and $\frac{1}{r}$ as $r \to \infty$, which is consistent with the radiation condition (1.3) (H_0^1 is zero-th order Hankel's function of the first kind), and exhibit a singularity at their local origin: $G(\boldsymbol{y},\boldsymbol{y}) = \infty$. Since they describe the wave emitted by a point source located in $\boldsymbol{y}$, they correspond to the impulse response of the Helmholtz operator. Note that other fundamental solutions can be used [131].

The integral representation theorem (B.2) states that, provided that the Cauchy data (the field and its normal trace) are known everywhere along the boundary Γ, the solution can be found at any point in Ω by simply evaluating an integral over the surface Γ. This is generally not the case in practice, since the boundary conditions of the problem only specify either the Dirichlet or Neumann data, and computing the one from the other is as costly as solving the problem.

Let $\mathcal{L}$ and $\mathcal{M}$ be respectively the single layer and double layer volume integral operators defined as:

$$\begin{aligned} \mathcal{L}: \quad H^{-1/2}(\Gamma) &\longrightarrow H^1(\mathbb{R}^d \setminus \Gamma) \\ \rho &\longmapsto \mathcal{L}\rho(\boldsymbol{x}) = \int_\Gamma G(\boldsymbol{x},\boldsymbol{y})\ \rho(\boldsymbol{y})\ d\Gamma(\boldsymbol{y}), \qquad \forall \boldsymbol{x} \text{ in } \mathbb{R}^d \setminus \Gamma. \\ \mathcal{M}: \quad H^{1/2}(\Gamma) &\longrightarrow H^1(\mathbb{R}^d \setminus \Gamma) \\ \lambda &\longmapsto \mathcal{M}\lambda(\boldsymbol{x}) = -\int_\Gamma \partial_{n_y} G(\boldsymbol{x},\boldsymbol{y})\ \lambda(\boldsymbol{y})\ d\Gamma(\boldsymbol{y}), \quad \forall \boldsymbol{x} \text{ in } \mathbb{R}^d \setminus \Gamma. \end{aligned} \tag{B.5}$$

applied to surface densities ρ and λ. We speak of the single and double layer potentials $\mathcal{L}\rho$ and $\mathcal{M}\lambda$, that have the properties of verifying the Helmholtz equation together with the radiation condition. Thus these potentials can both be regarded as outgoing waves. The solution field can be written as a sum of such potentials, with an appropriate choice of densities ρ and λ:

$$u(\boldsymbol{x}) = \mathcal{L}\rho(\boldsymbol{x}) + \mathcal{M}\lambda(\boldsymbol{x}), \quad \forall \boldsymbol{x} \text{ in } \Omega. \tag{B.6}$$

Let us remark that this decomposition is not unique: for any incident wave, there are an infinity of (ρ, λ) combinations that produce the scattered field, provided that they verify the boundary conditions. It is important for the following to notice that the same representation also holds for a fictitious field defined inside the object Ω^-.

B.1 Boundary integral operators

The trace and normal trace operators $\gamma_0^\pm$ and $\gamma_1^\pm$ are defined as:

$$\forall \boldsymbol{x} \in \Gamma : \quad \begin{aligned} \gamma_0^\pm g(\boldsymbol{x}) &:= \lim_{z \in \Omega^\pm \to \boldsymbol{x}} g(\boldsymbol{z}), \\ \gamma_1^\pm g(\boldsymbol{x}) &:= \lim_{z \in \Omega^\pm \to \boldsymbol{x}} \partial_{n_z} g(\boldsymbol{z}). \end{aligned} \tag{B.7}$$

We have defined interior $(-)$ and exterior $(+)$ trace operators since we will see that the potentials (B.5) have the property of being discontinuous across the interface Γ. We first introduce the set of boundary integral operators:

$$\begin{aligned}
L: \quad H^{-1/2}(\Gamma) &\longrightarrow H^{1/2}(\Gamma) \\
\rho &\longmapsto L\rho(\boldsymbol{x}) = \int_\Gamma G(\boldsymbol{x},\boldsymbol{y})\ \rho(\boldsymbol{y})\ d\Gamma(\boldsymbol{y}), \qquad \forall \boldsymbol{x} \text{ on } \Gamma; \\
N: \quad H^{-1/2}(\Gamma) &\longrightarrow H^{-1/2}(\Gamma) \\
\rho &\longmapsto N\rho(\boldsymbol{x}) = \partial_{n_x} \int_\Gamma G(\boldsymbol{x},\boldsymbol{y})\ \rho(\boldsymbol{y})\ d\Gamma(\boldsymbol{y}), \qquad \forall \boldsymbol{x} \text{ on } \Gamma; \\
M: \quad H^{1/2}(\Gamma) &\longrightarrow H^{1/2}(\Gamma) \\
\lambda &\longmapsto M\lambda(\boldsymbol{x}) = -\int_\Gamma \partial_{n_y} G(\boldsymbol{x},\boldsymbol{y})\ \lambda(\boldsymbol{y})\ d\Gamma(\boldsymbol{y}), \qquad \forall \boldsymbol{x} \text{ on } \Gamma; \\
D: \quad H^{1/2}(\Gamma) &\longrightarrow H^{-1/2}(\Gamma) \\
\lambda &\longmapsto D\lambda(\boldsymbol{x}) = -\partial_{n_x} \int_\Gamma \partial_{n_y} G(\boldsymbol{x},\boldsymbol{y})\ \lambda(\boldsymbol{y})\ d\Gamma(\boldsymbol{y}), \quad \forall \boldsymbol{x} \text{ on } \Gamma.
\end{aligned} \tag{B.8}$$

Compared to the volume integral operators (B.5), the latters are defined from the boundary Γ to itself, whereas the others were defined in volume. This implies

that the singularity of the Green function is now part of the integrals in (B.8), which requires extra care in their numerical evaluation. This also implies that while the single layer potential is continuous across the boundary Γ, its normal derivative is not. Conversely, the double layer potential is discontinuous, while its normal derivative is continuous. Making use of the trace operators (B.7) on the single and double layer potentials (B.5), one obtains the jump relations summarized as:

$$\begin{aligned} &\gamma_0^{\pm}\mathcal{L}\rho = L\rho; && \gamma_0^{\pm}\mathcal{M}\lambda = \left(\mp\tfrac{I}{2} + M\right)\lambda; \\ &\gamma_1^{\pm}\mathcal{L}\rho = \left(\mp\tfrac{I}{2} + N\right)\rho; && \gamma_1^{\pm}\mathcal{M}\lambda = D\lambda. \end{aligned} \tag{B.9}$$

Using these jump relations, and applying them to the traces of the exterior and (fictitious) interior total fields:

$$\begin{aligned} u_{\text{tot}}(\boldsymbol{x}) &= \mathcal{L}\rho(\boldsymbol{x}) + \mathcal{M}\lambda(\boldsymbol{x}) + u_{\text{inc}} \quad \forall \boldsymbol{x} \text{ in } \Omega^+; \\ u_{\text{tot}}^-(\boldsymbol{x}) &= \mathcal{L}\rho(\boldsymbol{x}) + \mathcal{M}\lambda(\boldsymbol{x}) + u_{\text{inc}} \quad \forall \boldsymbol{x} \text{ in } \Omega^-, \end{aligned} \tag{B.10}$$

one finally finds that the densities ρ and λ are naturally the jumps of the traces of the total fields:

$$\begin{bmatrix} \rho \\ \lambda \end{bmatrix} = \begin{bmatrix} (\partial_n u_{\text{tot}}^- - \partial_n u_{\text{tot}})|_\Gamma \\ (u_{\text{tot}}^- - u_{\text{tot}})|_\Gamma \end{bmatrix}. \tag{B.11}$$

From there, since the interior field is fictitious and can be set to 0, one obtains $\lambda = 0$ (recall the boundary condition $u_{\text{tot}} = 0$ on Γ). Still it must verify some boundary conditions on Γ, the choice of which gives raise to the family of boundary integral equations that are equivalent equations, with unknown ρ, to the scattering problem (B.1):

$$\begin{aligned} L\rho &= -u_{\text{inc}} && \text{on } \Gamma; \\ (\frac{I}{2} + N)\rho &= -\partial_n u_{\text{inc}} && \text{on } \Gamma. \end{aligned} \tag{B.12}$$

The first of these equations is an integral equation of the first kind, sometimes known as the electric field integral equation (EFIE). The second is an integral equation of the second kind, known as the magnetic field integral equation (MFIE). A combination of them is often prefered for its enhanced convergence properties when solved numerically and is then called a combined field integral equation (CFIE).

When these equations are discretized, they result in the boundary element method (BEM), also known as the method of moments (MoM) in the context of electromagnetism. Compared to volumic methods, integral methods have the advantage of reducing the number of unknowns since one solves a problem in dimension $d-1$. This makes them well adapted for applications where one is interested

in the far field, like for the practically important calculation of radar cross-sections (RCS) of metallic objects. However, the resulting matrices are full and can only be solved by iterative solvers. Also, the Green function is only available for homogeneous media, which rules out the application of integral methods for a large category of problems, for which volumic methods are more natural candidates.

B.2 Dirichlet-to-Neumann map

The Dirichlet-to-Neumann (DtN) operator is a practically important mathematical tool. Its definition is rather simple: when applied to a function defined on some surface Γ that is the boundary of a volumic domain Ω, it returns the normal derivative of the prolungation of the function in Ω, for some governing linear operator. The normal derivative is the inner product of the local unit normal to the surface with the gradient of the function: $\partial_{n_x} = (\hat{n}(\boldsymbol{x}) \cdot \nabla)$. In this section we give expressions of the DtN map using the boundary integral operators introduced above.

We can give a formal definition of the DtN map, noted $\mathcal{D}$, by means of the trace operators (B.7) introduced in the previous Section:

$$\begin{array}{rccl} \mathcal{D}: & H^{1/2}(\Gamma) & \longrightarrow & H^{-1/2}(\Gamma) \\ & u & \longmapsto & \gamma_1^+ u(\boldsymbol{x}) = \mathcal{D}\,\gamma_0^+ u(\boldsymbol{x}) = \partial_{n_x} u(\boldsymbol{x}), \quad \forall \boldsymbol{x} \text{ on } \Gamma. \end{array} \tag{B.13}$$

Recalling the representation of the exterior field (B.2) with the Cauchy data and the volume integral operators (B.5), we have:

$$u(\boldsymbol{x}) = -\mathcal{L}(\partial_n u_{|\Gamma})(\boldsymbol{x}) - \mathcal{M}(u_{|\Gamma})(\boldsymbol{x}) \quad , \quad \forall \boldsymbol{x} \text{ in } \Omega^+ \setminus \Gamma. \tag{B.14}$$

Applying the exterior trace operators (B.7) to that expression and using the jump relations (B.9) gives the two relations:

$$\gamma_0^+ u = -L\gamma_1^+ - (-I/2 + M)\gamma_0^+ u; \tag{B.15}$$

$$\gamma_1^+ u = -(-I/2 + N)\gamma_1^+ u - D\gamma_0^+ u. \tag{B.16}$$

Transforming these equations we obtain expressions that can be identified to definition (B.13):

$$\gamma_1^+ u = -L^{-1}(I/2 + M)\gamma_0^+ u; \tag{B.17}$$

$$\gamma_1^+ u = -(I/2 + N)^{-1} D\gamma_0^+ u. \tag{B.18}$$

Thus we have two different expressions of the DtN map in terms of boundary integral operators. They involve the inverse of some of these operators, which, provided that they exist, amounts to solving respectively an EFIE and a MFIE problem

(compare with (B.12)). Thus obtaining the application of an accurate version of the DtN map by these techniques is possible but still computationally demanding.

APPENDIX C

Non-local approximation of the DtN map based on PMLs

C.1 Explicit construction of the DtN map from the black box

Recall the non-overlapping optimized Schwarz algorithm:

$$\begin{aligned} -(\Delta + k^2) u_i^{(m+1)} &= 0 && \text{in } \Omega_i \\ (\partial_n + \mathcal{S}) u_i^{(m+1)} &= (-\partial_n + \mathcal{S}) u_j^{(m)} && \text{on } \Sigma_{ij} \\ &= g_{ij}^{(m)}, \end{aligned} \tag{C.1}$$

with k the wavenumber and the update:

$$\begin{aligned} g_{ij}^{(m+1)} &= -\partial_n u_j^{(m+1)} + \mathcal{S} u_j^{(m+1)} && \text{on } \Sigma_{ij} \\ &= -g_{ji}^{(m)} + 2\mathcal{S} u_j^{(m+1)}. \end{aligned} \tag{C.2}$$

Boundary conditions on $\partial\Omega_i \cap \partial\Omega$ are conserved from the original problem (we assume they are homogeneous Dirichlet in the following).

In order to solve the PDEs in the algorithm by the FEM, we use the weak form (dropping the iteration index):

$$\begin{aligned} 0 &= \int_{\Omega_i} \nabla u_i \nabla v - \int_{\Omega_i} k^2 u_i v - \int_{\partial\Omega_i} \partial_n u_i v && \text{in } \Omega_i, \forall v \in H^1(\Omega_i) \\ &= \int_{\Omega_i} \nabla u_i \nabla v - \int_{\Omega_i} k^2 u_i v - \int_{\partial\Omega_i} (-\mathcal{S} u_i + g_{ij}) v, \end{aligned} \tag{C.3}$$

where the boundary condition was injected to replace the normal derivative in the

boundary term. We note that the term $\int_{\partial\Omega_i} \mathcal{S} u_i v$ appears in the expression above, where $\mathcal{S}$ should be the DtN map. Let's write the weak form of the update (C.2), using the same set of test functions v on Σ_{ij} as above:

$$\int_{\partial\Omega_i} g_{ij}^{(m+1)} v \quad = \quad -\int_{\partial\Omega_i} g_{ji}^{(m)} v + 2\int_{\partial\Omega_i} \mathcal{S} u_j^{(m+1)} v \quad \text{on } \Sigma_{ij}, \forall v \in H_0^1(\Sigma_{ij}). \tag{C.4}$$

The quantity $\int_{\partial\Omega_i} \mathcal{S} u_i v$ is again required to compute the update, with the difference that $u_i^{(m+1)}$ is here a known function. Note that the result of this update directly appears as the right-hand side in the formulation above, at the next iteration. This lets us redefine the unknowns of the Schwarz algorithm as the integral (against the test functions) of the g_{ij} functions used previously:

$$\lambda_{ij}^{(m+1)} = \int_{\partial\Omega_i} g_{ij}^{(m+1)} v \quad = \quad -\lambda_{ji}^{(m)} + 2\int_{\partial\Omega_i} \mathcal{S} u_j^{(m+1)} v \quad \text{on } \Sigma_{ij}. \tag{C.5}$$

Thus, in the weak version of the algorithm, we do not need the explicit version of $\mathcal{S}$, but rather its integral against some test functions. To extract this from the black box, we define a problem in a small domain Ω_{bb}, with Dirichlet boundary condition on the boundary Σ where we would like to obtain the DtN map, a PML on the opposite boundary, and boundary conditions inherited from the reference problem on the other sides.

We proceed in two steps: first we solve (by FE) the problem above for some right hand side, and then we extract the quantity of interest from the solution u_{bb}. Following the approach of [176], we compute it as (assuming homogeneous Dirichlet on all boundaries, except Σ):

$$\begin{aligned} \int_\Sigma \mathcal{S} u_{bb} v = \int_\Sigma \partial_n u_{bb} v \quad &= \quad \int_\Sigma \mathbf{n} \cdot \nabla u_{bb} v \qquad\qquad \text{on } \Sigma, \forall v \in H^1(\Sigma) \\ &= \quad \int_{\Omega_{bb}} \nabla \cdot (\nabla u_{bb} v) \\ &= \quad \int_{\Omega_{bb}} (\nabla u_{bb} \nabla v + \Delta u_{bb} v) \\ &= \quad \int_{\Omega_{bb}} (\nabla u_{bb} \nabla v - k^2 u_{bb} v), \end{aligned} \tag{C.6}$$

using equation (C.1) to obtain the last equality.

In the discrete formulation, where we denote by Λ the vector consisting of the integrals in (C.6), these two terms correspond to parts of the mass and stiffness matrices related to the test functions on Σ and can be obtained easily:

$$\Lambda \quad = \quad A_{\Sigma\Sigma} \mathrm{u}_\Sigma + A_{\Sigma\Omega} \mathrm{u}_\Omega,$$

which is the residual for the unknowns u_Σ of a problem with homogeneous Neumann boundary conditions on Σ and no volume source, if u_Σ and u_Ω are the discrete unknowns on the boundary and in the interior of the domain, and A is the matrix arising from the discretization of this problem.

Finally, to build the discrete integrated DtN map from the black box, we repeat this using each test function as Dirichlet source. The output Λ_i of each of these experiments gives the i-th column of the matrix corresponding to the integral of the DtN map, suitable for the FE formulation of the problem. Note that it is directly summable with the other components of the FE matrix, avoiding significant overhead for the non-local operator integration otherwise required with a non-integrated DtN map.

Back to the DDM solver, the quantities Λ_{ij} are the new (discrete) unknowns of the modified algorithm, and corresponds to the integral of the g_{ij} functions used in the classical version (see equation C.4). It can also be summed directly to the other components of the right-hand side, without requiring extra integration (this is a considerable speed-up of the DDM algorithm, since many right-hand sides must be computed).

This procedure is computationally very expensive. As an alternative, a compressed version of the DtN map can be obtained via matrix probing. We give some insights on this technique in the next section.

C.2 DtN map approximation via probing

We are thus looking for an accurate approximation D to the DtN map $\mathcal{D}$ at some interface Σ. Consider the Helmholtz equation in a PML placed next to Σ. The operator D is viewed as a black box that maps Dirichlet data on Σ to the normal derivative, on Σ as well, of the solution to the Helmholtz equation in the PML: $\partial_n \bar{u} = D\bar{u}$. We first precompute the matrix D offline, then apply it to vectors on the fly as needed.

Matrix probing is used to make the precomputation of D tractable. Suppose that we wish to approximate a matrix $D \in \mathbb{R}^{n\times n}$, but we only have access to a handful of products of D with vectors. We assume D can be written as a linear combination of a small number of basis matrices B_j, $D \approx \sum_{j=1}^{p} c_j B_j$ *fixed ahead of time*. Under various assumptions, notably $p \ll n$ (see [41] for details) we can recover the vector c with great accuracy using only a few black box calls. For illustration, it is often advantageous to consider a single random vector z, so that $Dz \approx \sum_{j=1}^{p} c_j B_j z = \Psi_z c$, where the $B_j z$ are columns of Ψ_z. Solving for c now requires the pseudo-inverse of Ψ_z, which can be quickly obtained since this is an $n \times p$ matrix with $p \ll n$.

Hence we need a relatively small set of basis matrices which can accurately approximate the DtN map D. There are different ways to do this: we can use a geometrical optics approximation with oscillations of the form $e^{i\omega\tau(x,y)}$ times a

parametrized singular amplitude, see [19] for details, or else we can use the relaxed terms of the Padé expansion proposed in [31], obtained from a few 1D PDE solves.

A full description can be found in [18].

C.3 Implicit application of the black blox

The procedure described in Section C.1 features a computationally expensive precomputation phase and is not very practical since it requires a modification of the code to account for the non-local boundary condition. We now present a variant where no precomputation is required, since we rather apply the DtN map in an implicit fashion. The subdomains are meshed with appended PMLs, and the subproblems formulations are modified in the PMLs regions to achieve the non-reflecting boundary condition. After each subproblem solve, our approximation of the DtN map is applied to the Dirichlet data on the interfaces by solving problems in the PMLs only. To obtain the resulting Neumann data, the Dirichlet condition is applied by using Lagrange multipliers $\lambda \in H^1(\Sigma)$, the value of which after solution being naturally the normal derivative at the interface. We give the weak formulation of the black box problem as: find $u \in H^1(\Omega)$, $\lambda \in H^1(\Sigma)$ s.t.

$$\begin{aligned} \int_\Omega \quad & (D_n \nabla u \cdot \nabla v - k_\Sigma^2 c_n u v) \quad d\Omega \qquad && \forall v \in H^1(\Omega), \\ + \int_\Sigma \quad & (\lambda v + u\lambda' - u_D \lambda') \quad d\Sigma \quad = \quad 0, && \forall \lambda' \in H^1(\Sigma), \end{aligned} \tag{C.7}$$

where D_n and c_n are the PML tensor and damping function in the direction normal to the interface. Other boundary conditions are inherited from the main problem. The wavenumber in the PML k_Σ is prolonged normally from its value at the interface. At the next iteration, the source is imposed in the neighbouring domains via a surfacic delta function $\delta(\Sigma)$, in a similar way to how it is done in [169].

The definition of the subproblems geometry requires more care, and the cost of solving the subproblems is increased since unknowns are added in the PMLs, but this version of the method has the advantage of being directly implementable in a finite element code, without requiring any precomputation other than the factorization of the black boxes; since the PMLs are built to be symmetric with respect to the interface, there is one black box problem to be solved per interface. The resulting algorithm might be more efficient if it converges rapidly, especially in the case where there are many unknowns on the interfaces, making the construction of the explicit version overexpensive.

APPENDIX D

Numerical dispersion relation

To derive the numerical dispersion relation in the 1d case, we consider the matrix obtained after discretization of the standard weak formulation of the problem with some wavenumber k, and show that the functions that solve the discrete system have a different wavenumber k_h. The standard P1 FE discretization with uniform step h of the 1d Helmholtz equation:

$$(\partial_{xx} - k^2)\, u = 0 \tag{D.1}$$

gives raise to a system of equations of the form (away from the boundaries):

$$\begin{bmatrix} 0 & \ddots & \ddots & \ddots & 0 & \dots & \\ \dots & 0 & (-\frac{1}{h} - \frac{k^2 h}{6}) & (\frac{2}{h} - \frac{4k^2 h}{6}) & (-\frac{1}{h} - \frac{k^2 h}{6}) & 0 & \dots \\ & \dots & 0 & \ddots & \ddots & \ddots & 0 \end{bmatrix} u = \begin{bmatrix} \vdots \\ 0 \\ \vdots \end{bmatrix}. \tag{D.2}$$

To identify the dispersion relation, that is the relation between the wavenumber k that appears in the weak formulation and the effective wavenumber of the numerical solution, we look for a discrete solution of the form: $u_h(nh) = e^{ik_h(nh)}$ and inject it in the $n-$th equation of the system above:

$$(-\frac{1}{h} - \frac{k^2 h}{6}) e^{ik_h(n-1)h} + (\frac{2}{h} - \frac{4k^2 h}{6}) e^{ik_h nh} + (-\frac{1}{h} - \frac{k^2 h}{6}) e^{ik_h(n+1)h} = 0, \tag{D.3}$$

which becomes after division by $e^{ik_h nh}$ and using the identity $e^{ix} + e^{-ix} = 2\cos(x)$:

$$2\cos(k_h h) = \frac{(2 - \frac{4(kh)^2}{6})}{(1 + \frac{(kh)^2}{6})}, \tag{D.4}$$

which is the desired dispersion relation. Solving that equation for k_h provides the "true" wavenumber of the numerical solution and can be used to account for numerical dispersion and design almost exact absorbing conditions. However, this is only applicable in some particular cases like in waveguides, and in practice more general techniques must be employed.

The discretization density is an important parameter for high frequency simulations. It is defined as $n_\lambda = \lambda/h$; we can obtain a direct relation between the numerical wavenumber k_h and the true wavenumber as a function of n_λ by solving (D.4) for k_h:

$$k_h = k\frac{n_\lambda}{2\pi}\arccos\Big(\frac{1 - \frac{1}{3}\big(\frac{2\pi}{n_\lambda}\big)^2}{1 + \frac{2}{3}\big(\frac{2\pi}{n_\lambda}\big)^2}\Big) = k\, F_h(n_\lambda). \tag{D.5}$$

This relation is linear, with a dispersion factor F_h that depends only on the discretization density.

A similar derivation can be done for the 3-points stencil of a 1d finite difference scheme. We have:

$$\frac{1}{h^2}e^{\imath k_h (n-1)h} - \frac{2}{h^2}e^{\imath k_h nh} + \frac{1}{h^2}e^{\imath k_h (n+1)h} + k^2 e^{\imath k_h nh} = 0, \tag{D.6}$$

leading to:

$$2\cos(k_h h) = 2 - (kh)^2. \tag{D.7}$$

The dispersion relation can be rewritten in terms of the discretization density as:

$$k_h = k\frac{n_\lambda}{2\pi}\arccos\big(1 - \frac{(kh)^2}{2}\big). \tag{D.8}$$

Bibliography

[1] A. Alonso-Rodriguez and L. Gerardo-Giorda. New nonoverlapping domain decomposition methods for the harmonic Maxwell system. *SIAM J. Sci. Comput.*, 28(1):102–122, 2006.

[2] G. Allaire and S. Kaber. *Numerical Linear Algebra*. Texts in Applied Mathematics. Springer, 2008. ISBN 9780387689180.

[3] H. Ammari. Scattering of waves by thin periodic layers at high frequencies using the on-surface radiation condition method. *IMA J. Appl. Math.*, 60: 199–215, 1997.

[4] X. Antoine. Fast approximate computation of a time-harmonic scattered field using the on-surface radiation condition method. *IMA J. Appl. Math.*, 66(1):83–110, 2001. ISSN 0272-4960. doi: 10.1093/imamat/66.1.83. URL `http://dx.doi.org/10.1093/imamat/66.1.83`.

[5] X. Antoine. Advances in the on-surface radiation condition method: Theory, numerics and applications. In *Computational Methods for Acoustics Problems*, pages 169–194. Saxe-Coburg Publications, 2008.

[6] X. Antoine and C. Geuzaine. Phase Reduction Models for Improving the Accuracy of the Finite Element Solution of Time-Harmonic Scattering Problems I: General Approach and Low-Order Models. *Journal of Computational Physics*, 228(8):3114–3136, May 2009. doi: 10.1016/j.jcp.2009.01.008. URL `http://hal.archives-ouvertes.fr/hal-00591420`.

[7] X. Antoine, H. Barucq, and A. Bendali. Bayliss-turkel-like radiation conditions on surfaces of arbitrary shape. *Journal of Mathematical Analysis and Applications*, 229(1):184 – 211, 1999. ISSN 0022-247X. doi: http://dx.doi.org/10.1006/jmaa.1998.6153. URL `http://www.sciencedirect.com/science/article/pii/S0022247X98961534`.

[8] X. Antoine, C. Geuzaine, and K. Ramdani. Computational Methods for Multiple Scattering at High Frequency with Applications to Periodic Structures Calculations. In *Wave Propagation in Periodic Media*, Progress in Computational Physics, Vol. 1, pages 73–107. Bentham, 2010. URL `http://hal.archives-ouvertes.fr/hal-00511341`.

[9] A. Aziz, R. Kellogg, and A. Stephens. A two point boundary value problem with a rapidly oscillating solution. *Numerische Mathematik*, 53(1-2):107–121, 1988. ISSN 0029-599X. doi: 10.1007/BF01395880. URL `http://dx.`

doi.org/10.1007/BF01395880.

[10] I. Babuska and S. Sauter. Is the pollution effect of the fem avoidable for the helmholtz equation considering high wave numbers? *SIAM Review*, 42(3): 451–484, 2000. doi: 10.1137/S0036142994269186. URL http://dx.doi.org/10.1137/S0036142994269186.

[11] I. Babuska, F. Ihlenburg, E. T. Paik, and S. A. Sauter. A generalized finite element method for solving the helmholtz equation in two dimensions with minimal pollution. *Computer Methods in Applied Mechanics and Engineering*, 128(3,Äì4):325 – 359, 1995. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/0045-7825(95)00890-X. URL http://www.sciencedirect.com/science/article/pii/004578259500890X.

[12] G. Baker and P. Graves-Morris. *Padé Approximants*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2010. ISBN 9780521135092.

[13] M. Balabane. Boundary decomposition for helmholtz and maxwell equations I: Disjoint sub-scatterers. *Asymptotic Analysis*, 38(1):1–10, 2004.

[14] C. A. Balanis. *Advanced engineering electromagnetics*, volume 20. Wiley New York, 1989.

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

[16] A. Bayliss and E. Turkel. Radiation boundary conditions for wave-like equations. *Comm. Pure Appl. Math.*, 33(6):707–725, 1980. ISSN 0010-3640. doi: 10.1002/cpa.3160330603. URL http://dx.doi.org/10.1002/cpa.3160330603.

[17] A. Bayliss, M. Gunzburger, and E. Turkel. Boundary conditions for the numerical solution of elliptic equations in exterior regions. *SIAM Journal of Applied Mathematics*, 42:430–451, 1982.

[18] R. Bélanger-Rioux. *Compressed Absorbing Boundary Conditions for the Helmholtz Equation*. PhD thesis, Massachussets Institute of Technology, Aug. 2014.

[19] R. Bélanger-Rioux and L. Demanet. Compressed absorbing boundary conditions via matrix probing. *ArXiv e-prints*, , Jan. 2014.

[20] A. Bendali and Y. Boubendir. Non-overlapping domain decomposition method for a nodal finite element method. *Numerische Mathematik*, 103(4): 515–537, 2006. ISSN 0029-599X. doi: 10.1007/s00211-006-0010-9. URL http://dx.doi.org/10.1007/s00211-006-0010-9.

[21] A. Bendali, Y. Boubendir, and M. Fares. A FETI-like domain decomposition method for coupling FEM and BEM in large-size problems of acoustic scattering. *Computer & Structures*, 85:526–535, 2007.

[22] J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114:185–200, 1994.

[23] A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. *J. Comput. Phys.*, 223(2):469–488, 2007. ISSN 0021-9991. doi: 10.1016/j.jcp.2006.09.018. URL `http://dx.doi.org/10.1016/j.jcp.2006.09.018`.

[24] A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An exact bounded perfectly matched layer for time-harmonic scattering problems. *SIAM J. Sci. Comput.*, 30(1):312–338, Dec. 2007. ISSN 1064-8275. doi: 10.1137/060670912. URL `http://dx.doi.org/10.1137/060670912`.

[25] M. Born, E. Wolf, and A. Bhatia. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 1999. ISBN 9780521642224.

[26] A. Bossavit. Whitney forms: a class of finite elements for three-dimensional computations in electromagnetism. *Physical Science, Measurement and Instrumentation, Management and Education - Reviews, IEE Proceedings A*, 135(8):493–500, November 1988. ISSN 0143-702X. doi: 10.1049/ip-a-1:19880077.

[27] M. E. Bouajaji, B. Thierry, X. Antoine, and C. Geuzaine. A quasi-optimal domain decomposition algorithm for the time-harmonic Maxwell's equations. *in preparation*, 2014.

[28] Y. Boubendir. An analysis of the BEM-FEM non-overlapping domain decomposition method for a scattering problem. *Journal of Computational and Applied Mathematics*, 204(2):282–291, 2007.

[29] Y. Boubendir, A. Bendali, and M. B. Fares. Coupling of a non-overlapping domain decomposition method for a nodal finite element method with a boundary element method. *International Journal for Numerical Methods in Engineering*, 73(11):1624–1650, 2008. ISSN 1097-0207. doi: 10.1002/nme.2136. URL `http://dx.doi.org/10.1002/nme.2136`.

[30] Y. Boubendir, X. Antoine, and C. Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. *Journal of Computational Physics*, 2(231):262–280, 2012.

[31] Y. Boubendir, X. Antoine, and C. Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. *Journal of Computational Physics*, 231(2), 2012.

[32] A. D. L. Bourdonnaye, C. Farhat, A. Macedo, F. Magoulès, and F. Roux. A non overlapping domain decomposition method for the exterior Helmholtz problem. *Contemporary Mathematics*, 218(2):42–66, 1998.

[33] A. Brandt and I. Livshits. Wave-ray multigrid method for standing wave equations. *Electron. Trans. Numer. Anal.*, 6(Dec.):162–181 (electronic), 1997.

ISSN 1068-9613. Special issue on multilevel methods (Copper Mountain, CO, 1997).

[34] A. Brandt and S. Ta'asan. Multigrid method for nearly singular and slightly indefinite problems. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods II*, volume 1228 of *Lecture Notes in Mathematics*, pages 99–121. Springer Berlin Heidelberg, 1986. ISBN 978-3-540-17198-0. doi: 10.1007/BFb0072643. URL http://dx.doi.org/10.1007/BFb0072643.

[35] O. Bruno, C. Geuzaine, J. Monro, Jr., and F. Reitich. Prescribed error tolerances within fixed computational times for scattering problems of arbitrarily high frequency: the convex case. *Philosophical Transactions of the Royal Society (Series A: Mathematical, Physical and Engineering Sciences)*, 362(1816):629–645, 2004.

[36] B. L. Buzbee, F. W. Dorr, J. A. George, and G. H. Golub. The direct solution of the discrete poisson equation on irregular regions. *SIAM Journal on Numerical Analysis*, 8(4):pp. 722–736, 1971. ISSN 00361429. URL http://www.jstor.org/stable/2949602.

[37] V. M. Calo, N. O. Collier, D. Pardo, and M. R. Paszynski. Computational complexity and memory usage for multi-frontal direct solvers used in p finite element analysis. *Procedia Computer Science*, 4(0):1854 – 1861, 2011. ISSN 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2011.04.201. URL http://www.sciencedirect.com/science/article/pii/S1877050911002596. Proceedings of the International Conference on Computational Science, {ICCS} 2011.

[38] O. Cessenat and B. Despres. Application of an ultra weak variational formulation of elliptic pdes to the two-dimensional helmholtz problem. *SIAM journal on numerical analysis*, 35(1):255–299, 1998.

[39] N. Chen. Inverse iteration on defective matrices. *Mathematics of Computation*, 31(139):726–732, 1977.

[40] W. C. Chew, J. M. Jin, and E. Michielssen. Complex coordinate stretching as a generalized absorbing boundary condition. *Microwave and Optical Technology Letters*, 15(6):363–369, 1997. ISSN 1098-2760. URL http://dx.doi.org/10.1002/(SICI)1098-2760(19970820)15:6<363::AID-MOP8>3.0.CO;2-C.

[41] J. Chiu and L. Demanet. Matrix probing and its conditioning. *SIAM J. Numer. Anal.*, 50:171–193, 2012.

[42] J. F. Claerbout. *Fundamentals of geophysical data processing: with applications to petroleum prospecting*. McGraw-Hill Inc., New York, Mar. 1976. ISBN 0070111170.

[43] F. Collino, S. Ghanemi, and P. Joly. Domain decomposition method for harmonic wave propagation: a general presentation. *Comput. Methods Appl. Mech. Engrg.*, 184(2–4):171–211, 2000.

[44] D. Colton and R. Kress. *Integral equation methods in scattering theory*. Pure and applied mathematics. Wiley, 1983. ISBN 9780471864202.

[45] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM J. Res. Dev.*, 11(2):215–234, Mar. 1967. ISSN 0018-8646.

[46] C. Craeye. On the connection between multiple-scattering based macro basis functions and krylov subspace methods. In *Electromagnetics in Advanced Applications, 2009. ICEAA '09. International Conference on*, pages 938–941, Sept 2009. doi: 10.1109/ICEAA.2009.5297330.

[47] J.-B. d'Alembert. Recherches sur la courbe que forme une corde tendue mise en vibration. In *Histoire de l'académie royale des sciences et belles lettres*, volume 3, pages 214–219. Haude and Spener, Berlin, 1747.

[48] F. Dang and N. Emad. Fast iterative method in solving eikonal equations: A multi-level parallel approach. *Procedia Computer Science*, 29(0):1859 – 1869, 2014. ISSN 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2014.05.170. URL `http://www.sciencedirect.com/science/article/pii/S1877050914003470`. 2014 International Conference on Computational Science.

[49] V. Q. Dang, P. Dular, R. V. Sabariego, L. Krahenbuhl, and C. Geuzaine. Subproblem approach for thin shell dual finite element formulations. *Magnetics, IEEE Transactions on*, 48(2):407–410, 2012.

[50] E. Darrigrand. Coupling of fast multipole method and microlocal discretization for the 3-d helmholtz equation. *Journal of Computational Physics*, 181 (1):126 – 154, 2002. ISSN 0021-9991. URL `http://dx.doi.org/10.1006/jcph.2002.7091`.

[51] Després. *Décomposition de Domaine pour les Problèmes de Propagation d'Ondes en Régime Harmonique. Le Théorème de Borg pour l'Equation de Hill Vectorielle*. PhD thesis, Paris VI University, France, 1991.

[52] B. Després. Décomposition de domaine et problème de Helmholtz. *C.R. Acad. Sci. Paris*, 1(6):313–316, 1990.

[53] B. Després. Domain decomposition method and the Helmholtz problem. In G. Cohen, L. Halpern, and P. Joly, editors, *Mathematical and numerical aspects of wave propagation phenomena*, pages 44–52. SIAM, Philadelphia, PA, 1991.

[54] 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)*, pages 475–484, Amsterdam, 1992. North-Holland.

[55] V. Dolean, M. J. Gander, and L. Gerardo-Giorda. Optimized Schwarz methods for Maxwell's equations. *SIAM J. Sci. Comput.*, 31(3):2193–2213, 2009. ISSN 1064-8275. doi: 10.1137/080728536. URL `http://dx.doi.org/10.1137/080728536`.

[56] V. Dolean, J. M. Gander, S. Lanteri, J.-F. Lee, and Z. Peng. Optimized Schwarz methods for curl-curl time-harmonic Maxwell's equations. In *J. Erhel, M. J. Gander, L. Halpern, T. Sassi, and O. Widlund, editors, Proceedings of the 21st international domain decomposition conference. Springer LNCSE*, 2013.

[57] M. Dryja and O. Widlund. Some domain decomposition algorithms for elliptic problems. In L. Hayes and D. Kincaid, editors, *Iterative Methods for Large Linear Systems*, pages 273–291. Academic Press, San Diego, CA, 1989.

[58] M. Dryja and O. B. Widlund. An additive variant of the schwarz alternating method for the case of many subregions. Technical Report 339, Department of Computer Science, Courant Institute, New York, 1987.

[59] O. Dubois, M. J. Gander, S. Loisel, A. St-Cyr, and D. B. Szyld. The optimized Schwarz method with a coarse grid correction. *SIAM J. Scientific Computing*, 34(2), 2012.

[60] P. Dular and C. Geuzaine. GetDP Web page. `http://getdp.info`, 2014. URL `http://getdp.info`.

[61] P. Dular and R. V. Sabariego. A perturbation method for computing field distortions due to conductive regions with-conform magnetodynamic finite element formulations. *Magnetics, IEEE Transactions on*, 43(4):1293–1296, 2007.

[62] P. Dular, R. V. Sabariego, C. Geuzaine, M. F. da Luz, P. Kuo-Peng, and L. Krahenbuhl. Finite element magnetic models via a coupling of subproblems of lower dimensions. *Magnetics, IEEE Transactions on*, 46(8):2827–2830, 2010.

[63] P. Dular, V. Q. Dang, R. V. Sabariego, L. Kahenbuhl, and C. Geuzaine. Correction of thin shell finite element magnetic models via a subproblem method. *Magnetics, IEEE Transactions on*, 47(5):1158–1161, 2011.

[64] E. Efstathiou and M. Gander. Why restricted additive schwarz converges faster than additive schwarz. *BIT Numerical Mathematics*, 43(5):945–959, 2003. ISSN 0006-3835. doi: 10.1023/B:BITN.0000014563.33622.1d. URL `http://dx.doi.org/10.1023/B:BITN.0000014563.33622.1d`.

[65] M. El Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri. Optimized Schwarz methods for the time-harmonic Maxwell equations with damping. *SIAM J. Sci. Comput.*, 34(4):A2048–A2071, 2012. ISSN 1064-8275. doi: 10.1137/110842995. URL `http://dx.doi.org/10.1137/110842995`.

[66] M. El Bouajaji, X. Antoine, and C. Geuzaine. Approximate local magnetic-to-electric surface operators for time-harmonic Maxwell's equations. *Journal of Computational Physics*, Accepted for publication, 2014.

[67] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of Computation*, 23:629–651, 1977.

[68] B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. *Multiscale Model. Simul.*, 9(2):686–710, 2011. ISSN 1540-3459.

[69] B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation. *Communications on Pure and Applied Mathematics*, 64(5):697–735, 2011.

[70] J. Erhel, K. Burrage, and B. Pohl. Restarted GMRES preconditioned by deflation. *Journal of Computation and Applied Math.*, 69, 1996.

[71] Y. Erlangga, C. Vuik, and C. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Applied Numerical Mathematics*, 50(3-4):409 – 425, 2004. ISSN 0168-9274. doi: http://dx.doi.org/10.1016/j.apnum.2004.01.009.

[72] Y. Erlangga, C. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.*, 27: 1471–1492, 2006.

[73] Y. A. Erlangga and R. Nabben. On a multilevel krylov method for the helmholtz equation preconditioned by shifted laplacian. *ETNA. Electronic Transactions on Numerical Analysis*, 31:403–424, 2008.

[74] Y. A. Erlangga and R. Nabben. Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices. *SIAM J. Matrix Anal. Appl.*, 30(2):684–699, 2008. ISSN 0895-4798. doi: 10.1137/060678257. URL `http://dx.doi.org/10.1137/060678257`.

[75] Y. A. Erlangga, C. Vuil, and C. W. Oosterlee. Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Applied Numerical Mathematics*, 56:648–666, 2006.

[76] A. Ern and J. Guermond. *Theory and Practice of Finite Elements*. Number 159 in Applied Mathematical Sciences. Springer, 2004. ISBN 9780387205748.

[77] O. Ernst and M. Gander. Why it is difficult to solve helmholtz problems with classical iterative methods. In I. G. Graham, T. Y. Hou, O. Lakkis, and R. Scheichl, editors, *Numerical Analysis of Multiscale Problems*, volume 83 of *Lecture Notes in Computational Science and Engineering*, pages 325–363. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-22060-9.

[78] O. G. Ernst. A finite-element capacitance matrix method for exterior helmholtz problems. *Numerische Mathematik*, 75(2):175–204, 1996. ISSN 0029-599X. doi: 10.1007/s002110050236. URL `http://dx.doi.org/10.1007/s002110050236`.

[79] O. G. Ernst and M. J. Gander. Multigrid methods for Helmholtz problems: A convergent scheme in 1D using standard components. In *Direct and Inverse Problems in Wave Propagation and Applications*, volume 14 of *Radon Series on Computational and Applied Mathematics*. I. G. Graham, U. Langer, J. M. Melenk, and M. Sini, Berlin/Boston, 2013.

[80] L. Euler. De la propagation du son. In *Histoire de l'académie royale des sciences et belles lettres*, volume 15, pages 209–264. Haude and Spener, Berlin, 1766.

[81] M. Fabrizio and A. Morro. *Electromagnetism of Continuous Media: Mathematical Modelling and Applications*. Oxford University Press, New York, 2003. ISBN 9780198527008.

[82] C. Farhat, A. Macedo, and M. Lesoinne. A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. *Numerisch Mathematik*, 85(2):282–303, 2000.

[83] C. Farhat, A. Macedo, M. Lesoinne, F. Roux, F. Magoulès, and A. D. L. Bourdonnaye. Two-level domain decomposition methods with lagrange multipliers for the fast iterative solution of acoustic scattering problems. *Computer Methods in Applied Mechanics and Engineering*, 184:213–239, 2000.

[84] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method–part I: A faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50(7):1523–1544, 2001. ISSN 1097-0207. doi: 10.1002/nme.76. URL `http://dx.doi.org/10.1002/nme.76`.

[85] C. Farhat, R. Tezaur, and J. Toivanen. A domain decomposition method for discontinuous Galerkin discretizations of Helmholtz problems with plane waves and lagrange multipliers. *International Journal for Numerical Methods in Engineering*, 78:1513–1531, 2009.

[86] R. P. Federenko. A relaxation method for solving elliptic difference equations. *USSR Comput. Math. and Math. Phys.*, 1(4):1092–1096, 1962.

[87] V. Fock. *Electromagnetic Diffraction and Propagation Problems*, volume 1 of *International series of monographs on electromagnetic waves*. Pergamon Press, 1965.

[88] J. Frank and C. Vuik. On the construction of deflation-based preconditioners. *SIAM Journal on Scientific Computing*, 23:442–462, 2001.

[89] M. Gander and F. Kwok. Best robin parameters for optimized schwarz methods at cross points. *SIAM Journal on Scientific Computing*, 34(4):A1849–A1879, 2012. doi: 10.1137/110837218. URL `http://dx.doi.org/10.1137/110837218`.

[90] M. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. of Sci. Comput.*, 24(1):38–60, 2002.

[91] M. J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2): 699–731, 2006.

[92] M. J. Gander and F. Nataf. AILU for Helmholtz problems: a new preconditioner based on the analytic parabolic factorization. *J. Comput. Acoust.*, 9 (4):1499–1506, 2001. ISSN 0218-396X.

[93] M. J. Gander, L. Halpern, and K. Santugini. A new coarse grid correction for RAS/AS. In *Domain Decomposition Methods in Science and Engineering XXI, Lecture Notes in Computational Science and Engineering*. Springer-Verlag,

2012.

[94] M. J. Gander, I. G. Graham, and E. A. Spence. How should one choose the shift for the shifted Laplacian to be a good preconditioner for the Helmholtz equation? *preprint*, 2014.

[95] C. Geuzaine. GetDP: a general finite-element solver for the de Rham complex. *PAMM*, 7(1):1010603–1010604, 2007.

[96] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[97] C. Geuzaine and J.-F. Remacle. Gmsh Web page. `http://gmsh.info`, 2014. URL `http://gmsh.info`.

[98] C. Geuzaine, O. Bruno, and F. Reitich. On the $\mathcal{O}(1)$ solution of multiple-scattering problems. *IEEE Trans. Mag.*, 41(5):1488–1491, 2005.

[99] C. Geuzaine, A. Vion, R. Gaignaire, P. Dular, and R. Sabariego. An amplitude finite element formulation for multiple-scattering by a collection of convex obstacles. *Magnetics, IEEE Transactions on*, 46(8):2963–2966, Aug 2010. ISSN 0018-9464. doi: 10.1109/TMAG.2010.2043419.

[100] D. Givoli. *Numerical methods for problems in infinite domains*, volume 33 of *Studies in Applied Mechanics*. Elsevier Scientific Publishing Co., Amsterdam, 1992. ISBN 0-444-88820-9.

[101] D. Givoli. High-order local non-reflecting boundary conditions: a review. *Wave Motion*, 39(4):319 – 326, 2004. ISSN 0165-2125. doi: http://dx.doi.org/10.1016/j.wavemoti.2003.12.004. URL `http://www.sciencedirect.com/science/article/pii/S0165212503001203`. New computational methods for wave propagation.

[102] D. Givoli. Computational absorbing boundaries. In S. Marburg and B. Nolte, editors, *Computational Acoustics of Noise Propagation in Fluids - Finite and Boundary Element Methods*, pages 145–166. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-77447-1. doi: 10.1007/978-3-540-77448-8_6. URL `http://dx.doi.org/10.1007/978-3-540-77448-8_6`.

[103] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013. ISBN 9781421407944.

[104] D. Gonzalez-Ovejero, E. Acedo, N. Razavi-Ghods, and C. Craeye. Fast mbf based method for large random array characterization. In *Antennas and Propagation Society International Symposium, 2009. APSURSI '09. IEEE*, pages 1–4, June 2009. doi: 10.1109/APS.2009.5171749.

[105] A. Greenbaum. *Iterative methods for solving linear systems*, volume 17 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. ISBN 0-89871-396-X.

[106] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. Frontiers in Applied Mathematics. SIAM, Philadelphia, PA, 1997.

[107] G. R. Hadley. A complex jacobi iterative method for the indefinite helmholtz equation. *Journal of Computational Physics*, 203(1):358 – 370, 2005. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2004.09.015. URL `http://www.sciencedirect.com/science/article/pii/S0021999104004012`.

[108] E. Heikkola, T. Rossi, and J. Toivanen. A parallel fictitious domain method for the three-dimensional helmholtz equation. *SIAM Journal on Scientific Computing*, 24(5):1567–1588, 2003. doi: 10.1137/S1064827500370305. URL `http://dx.doi.org/10.1137/S1064827500370305`.

[109] R. Hiptmair, A. Moiola, and I. Perugia. Plane wave discontinuous galerkin methods for the 2d helmholtz equation: analysis of the p-version. *SIAM Journal on Numerical Analysis*, 49(1):264–284, 2011.

[110] C. Hirsch. Chapter 9 - time integration methods for space-discretized equations. In C. Hirsch, editor, *Numerical Computation of Internal and External Flow*, pages 413 – 489. Butterworth-Heinemann, Oxford, second edition, 2007. ISBN 978-0-7506-6594-0.

[111] T. Huttunen, P. Monk, and J. P. Kaipio. Computational aspects of the ultra-weak variational formulation. *Journal of Computational Physics*, 182(1):27–46, 2002.

[112] F. Ihlenburg and I. Babuska. Finite element solution to the Helmholtz equation with high wavenumber. part II: The h-p version of the FEM. *SIAM J. Numer. Anal.*, 34:315–358, 1997.

[113] J. D. Jackson. *Classical electrodynamics*. Wiley, New York, NY, 3rd edition, 1999. ISBN 9780471309321.

[114] W.-K. Jeong and R. T. Whitaker. A fast iterative method for eikonal equations. *SIAM J. Sci. Comput.*, 30(5):2512–2534, 2008. ISSN 1064-8275. doi: 10.1137/060670298. URL `http://dx.doi.org/10.1137/060670298`.

[115] J. Jin. *The Finite Element Method in Electromagnetics*. Wiley, New York, 2nd edition, 2002. ISBN 9780471438182.

[116] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, Dec. 1998. ISSN 1064-8275. doi: 10.1137/S1064827595287997. URL `http://dx.doi.org/10.1137/S1064827595287997`.

[117] R. Kechroud, X. Antoine, and A. Soulaïmani. Numerical accuracy of a Padé-type non-reflecting boundary condition for the finite element solution of acoustic scattering problems at high-frequency. *Internat. J. Numer. Methods Engrg.*, 64(10):1275–1302, 2005. ISSN 0029-5981. doi: 10.1002/nme.1390. URL `http://dx.doi.org/10.1002/nme.1390`.

[118] J. Keller. Geometrical theory of diffraction. *Journal of the Optical Society of America*, 52(2):116–130, 1962.

[119] J. Keller and D. Givoli. Exact non-reflecting boundary conditions. *J. Comput. Phys.*, 9:172–192, 1989.

[120] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal feti methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM Journal on Numerical Analysis*, 40(1):159–179, 2002. doi: 10.1137/S0036142901388081. URL `http://dx.doi.org/10.1137/S0036142901388081`.

[121] G. A. Kriegsmann, A. Taflove, and K. R. Umashankar. A new formulation of electromagnetic wave scattering using an on-surface radiation boundary condition approach. *IEEE Trans. Antennas and Propagation*, 35(2):153–161, 1987. ISSN 0018-926X. doi: 10.1109/TAP.1987.1144062. URL `http://dx.doi.org/10.1109/TAP.1987.1144062`.

[122] M. Leontovich and V. Fock. Solution of propagation of electromagnetic waves along the Earth's surface by parabolic equations. *Journal of Physics-USSR*, 10:13–23, 1946.

[123] M. Levy. *Parabolic Equation Methods for Electromagnetic Wave Propagation*, volume 45 of *IEE Publication Series*. IEE Press, 2000.

[124] P.-L. Lions. On the Schwarz alternating method I:. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. SIAM, Philadelphia, PA, 1988.

[125] P.-L. Lions. On the Schwarz alternating method III: a variant for non overlapping subdomains. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, held in Houston, Texas, March 20–22, 1989*, pages 202–223. SIAM, Philadelphia, PA, 1990.

[126] I. Livshits. A scalable multigrid method for solving indefinite helmholtz equations with constant wave numbers. *Numerical Linear Algebra with Applications*, 21(2):177–193, 2014. ISSN 1099-1506. doi: 10.1002/nla.1926. URL `http://dx.doi.org/10.1002/nla.1926`.

[127] Y. Y. Lu. Some techniques for computing wave propagation in optical waveguides. *Communications in Computational Physics*, 1(6):1056–1075, 2006.

[128] Y. Y. Lu and S. H. Wei. A new iterative bidirectional beam propagation method. *Photonics Technology Letters, IEEE*, 14(11):1533–1535, Nov 2002. ISSN 1041-1135. doi: 10.1109/LPT.2002.803904.

[129] F. Magoulès and F.-X. Roux. Dirichlet-to-Neumann maps for domain decomposition methods: a unified approach. In *Domain decomposition methods: theory and applications*, volume 25 of *GAKUTO Internat. Ser. Math. Sci. Appl.*, pages 123–145. Gakkōtosho, Tokyo, 2006.

[130] F. Magoulès, P. Iványi, and B. H. V. Topping. Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation.

Comput. Methods Appl. Mech. Engrg., 193(45-47):4797–4818, 2004. ISSN 0045-7825. doi: 10.1016/j.cma.2004.05.004. URL http://dx.doi.org/10.1016/j.cma.2004.05.004.

[131] P. Martin. *Multiple Scattering: Interaction of Time-Harmonic Waves with N Obstacles*. Number v. 10 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2006. ISBN 9780521865548.

[132] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31(137):148–162, 1977. ISSN 0025-5718.

[133] F. Milinazzo, C. Zala, and G. Brooke. Rational square-root approximations for parabolic equation algorithms. *Journal of the Acoustical Society of America*, 101(2):760–766, FEB 1997. ISSN 0001-4966. doi: {10.1121/1.418038}.

[134] A. Moiola and E. A. Spence. Is the Helmholtz equation really sign-indefinite? *SIAM Rev.*, 56(2):274–312, 2014. ISSN 0036-1445. doi: 10.1137/120901301. URL http://dx.doi.org/10.1137/120901301.

[135] P. Monk. *Finite Element Methods for Maxwell's Equations*. Numerical Analysis and Scientific Computation Series. Oxford University Press, Oxford, 2003. ISBN 9780198508885.

[136] R. Murch. The on-surface radiation condition applied to three-dimensional convex objects. *IEEE Trans. Antennas and Propagation*, 41(5):651–658, 1993.

[137] R. Nabben and C. Vuik. Domain decomposition methods and deflated krylov subspace iterations, 2006.

[138] F. Nataf. Interface connections in domain decomposition methods. In *Modern methods in scientific computing and applications (Montréal, QC, 2001)*, volume 75 of *NATO Sci. Ser. II Math. Phys. Chem.*, pages 323–364. Kluwer Acad. Publ., Dordrecht, 2002.

[139] J. Nedelec. *Acoustic and Electromagnetic Equations: Integral Representations for Harmonic Problems*. Number v. 144 in Acoustic and electromagnetic equations: integral representations for harmonic problems. Springer, 2001. ISBN 9780387951553.

[140] J. Nečas. *Direct methods in the theory of elliptic equations*. Springer-Verlag, Berlin Heidelberg, 2012. ISBN 978-3-642-10455-8.

[141] C. Oosterlee, C. Vuik, W. Mulder, and R.-E. Plessix. Shifted-laplacian preconditioners for heterogeneous helmholtz problems. In B. Koren and K. Vuik, editors, *Advanced Computational Methods in Science and Engineering*, volume 71 of *Lecture Notes in Computational Science and Engineering*, pages 21–46. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-03343-8. doi: 10.1007/978-3-642-03344-5_2. URL http://dx.doi.org/10.1007/978-3-642-03344-5_2.

[142] D. Osei-Kuffuor and Y. Saad. Preconditioning helmholtz linear systems. *Applied Numerical Mathematics*, 60(4):420 – 431, 2010. ISSN 0168-9274. doi:

http://dx.doi.org/10.1016/j.apnum.2009.09.003. Special Issue: {NUMAN} 2008.

[143] D. Pardo, M. Paszynski, N. Collier, J. Alvarez, L. Dalcin, and V. Calo. A survey on direct solvers for galerkin methods. *SeMA Journal*, 57(1):107–134, 2012. ISSN 1575-9822. doi: 10.1007/BF03322602. URL http://dx.doi.org/10.1007/BF03322602.

[144] Z. Peng and J.-F. Lee. A scalable nonoverlapping and nonconformal domain decomposition method for solving time-harmonic Maxwell equations in $\mathbb{R}^3$. *SIAM J. Sci. Comput.*, 34(3):A1266–A1295, 2012. ISSN 1064-8275. doi: 10.1137/100817978. URL http://dx.doi.org/10.1137/100817978.

[145] Z. Peng, V. Rawat, and J.-F. Lee. One way domain decomposition method with second order transmission conditions for solving electromagnetic wave problems. *J. Comput. Phys.*, 229(4):1181–1197, 2010. ISSN 0021-9991. doi: 10.1016/j.jcp.2009.10.024. URL http://dx.doi.org/10.1016/j.jcp.2009.10.024.

[146] E. Perrey-Debain, O. Laghrouche, P. Bettess, and J. Trevelyan. Plane-wave basis finite elements and boundary elements for three-dimensional wave scattering. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 362(1816):561–577, 2004.

[147] P. G. Petropoulos. On the termination of the perfectly matched layer with local absorbing boundary conditions. *J. Comput. Phys.*, 143(2):665–673, July 1998. ISSN 0021-9991. doi: 10.1006/jcph.1998.5979. URL http://dx.doi.org/10.1006/jcph.1998.5979.

[148] B. Pinçon and K. Ramdani. Selective focusing on small scatterers in acoustic waveguides using time reversal mirrors. *Inverse Problems*, 23(1):1–25, 2007.

[149] R. Plessix and W. Mulder. Separation-of-variables as a preconditioner for an iterative Helmholtz solver. *Applied Numerical Mathematics*, 44:385–400, 2003.

[150] J. Poulson, B. Engquist, S. Li, and L. Ying. A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations. *SIAM J. Sci. Comput.*, 35(3):C194–C212, 2013. ISSN 1064-8275. doi: 10.1137/120871985. URL http://dx.doi.org/10.1137/120871985.

[151] W. Proskurowski. Numerical solution of helmholtz's equation by implicit capacitance matrix methods. *ACM Trans. Math. Softw.*, 5(1):36–49, Mar. 1979. ISSN 0098-3500. doi: 10.1145/355815.355818. URL http://doi.acm.org/10.1145/355815.355818.

[152] D. Rabinovich, D. Givoli, and E. Bécache. Comparison of high-order absorbing boundary conditions and perfectly matched layers in the frequency domain. *International Journal for Numerical Methods in Biomedical Engineering*, 26(10):1351–1369, 2010. ISSN 2040-7947. doi: 10.1002/cnm.1394. URL http://dx.doi.org/10.1002/cnm.1394.

[153] S. Ramo, J. Whinnery, and T. Van Duzer. *Fields and Waves in Communication Electronics, 2nd Ed.* Wiley, 1984. ISBN 0471811033.

[154] V. Rawat and J.-F. Lee. Nonoverlapping domain decomposition with second order transmission condition for the time-harmonic Maxwell's equations. *SIAM J. Scientific Computing*, 32(6):3584–3603, 2010.

[155] F.-X. Roux, F. Magoulès, S. Salmon, and L. Series. Optimization of interface operator based on algebraic approach. *In Domain Decomposition Methods in Sci. Engrg.*, pages 297–304, 2002.

[156] F. X. Roux, F. Magoulès, L. Series, and Y. Boubendir. Approximation of optimal interface boundary conditions for two-Lagrange multiplier FETI method. *Lecture Notes in Computational Science and Engineering*, 40:283–290, 2005.

[157] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29(3):867–884, 1992. ISSN 0036-1429. doi: 10.1137/0729053. URL `http://dx.doi.org/10.1137/0729053`.

[158] R. Roxburgh. Electromagnetic scattering from a right-circular cylinder using a surface radiation condition. *IMA J. Appl. Math.*, 59:221–230, 1997.

[159] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003.

[160] R. V. Sabariego and P. Dular. A perturbation approach for the modeling of eddy current nondestructive testing problems with differential probes. *Magnetics, IEEE Transactions on*, 43(4):1289–1292, 2007.

[161] Z. Sacks, D. Kingsland, R. Lee, and J.-F. Lee. A perfectly matched anisotropic absorber for use as an absorbing boundary condition. *Antennas and Propagation, IEEE Transactions on*, 43(12):1460–1463, Dec 1995. ISSN 0018-926X. doi: 10.1109/8.477075.

[162] H. A. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, 1870.

[163] J. A. Sethian. Fast marching methods. *SIAM Rev.*, 41(2):199–235, 1999. ISSN 0036-1445. doi: 10.1137/S0036144598347059. URL `http://dx.doi.org/10.1137/S0036144598347059`.

[164] W. Śmigaj, S. Arridge, T. Betcke, and M. S. J. Phillips. Solving boundary integral problems with BEM++. *ACM Trans. Math. Software*, to appear.

[165] B. F. Smith, P. E. Bjorstad, and W. D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[166] A. Sommerfeld and J. Runge. Anwendung der Vektorrechnung auf die Grundlagen der geometrischen Optik. *Annalen der Physik*, 35:277–298, 1911.

[167] A. St-Cyr, M. Gander, and S. Thomas. Optimized multiplicative, additive,

and restricted additive schwarz preconditioning. *SIAM Journal on Scientific Computing*, 29(6):2402–2425, 2007. doi: 10.1137/060652610. URL http://dx.doi.org/10.1137/060652610.

[168] E. Stiefel. Über einige Methoden der Relaxationsrechnung. *Z. Angew. Math. Physik*, 3:1–33, 1952. ISSN 0044-2275.

[169] C. C. Stolk. A rapidly converging domain decomposition method for the Helmholtz equation. *Journal of Computational Physics*, 241(0):240 – 252, 2013. ISSN 0021-9991.

[170] C. C. Stolk, M. Ahmed, and S. K. Bhowmik. A multigrid method for the Helmholtz equation with optimized coarse grid corrections. *ArXiv e-prints,* , Apr. 2014.

[171] G. Strang. *Linear algebra and its applications*. Thomson Brooks/Cole learning, third edition, 1988.

[172] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *J. Sci. Comput.*, 39(3):340–370, 2009. ISSN 0885-7474. doi: 10.1007/s10915-009-9272-6. URL http://dx.doi.org/10.1007/s10915-009-9272-6.

[173] M. Taylor. *Pseudodifferential operators*. Princeton Mathematical series, 34, Princeton University Press, 1991.

[174] R. Tezaur and C. Farhat. Three-dimensional discontinuous galerkin elements with plane waves and lagrange multipliers for the solution of mid-frequency helmholtz problems. *International journal for numerical methods in engineering*, 66(5):796–815, 2006.

[175] B. Thierry, A. Vion, S. Tournier, M. E. Bouajaji, D. Colignon, X. Antoine, and C. Geuzaine. GetDDM: an open framework for testing optimized Schwarz methods for time-harmonic wave problems. *in preparation*, 2014.

[176] A. Toselli and O. Widlund. *Domain decomposition methods – algorithms and theory*. Ser. Comput. Math. Springer, 2005.

[177] L. N. Trefethen and D. Bau, III. *Numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. ISBN 0-89871-361-7. doi: 10.1137/1.9780898719574. URL http://dx.doi.org/10.1137/1.9780898719574.

[178] E. Turkel. Boundary conditions and iterative schemes for the helmholtz equation. In *Computational Methods for Acoustics Problems*, pages 127–158. Saxe-Coburg Publications, 2008. ISBN 978-1-874672-30-2.

[179] A. Vion and C. Geuzaine. Parallel double sweep preconditioner for the optimized Schwarz algorithm applied to high frequency Helmholtz and Maxwell equations. In *Proceedings of the 22th International Conference on Domain Decomposition Methods (DD22)*, Lugano, Italy, Sept. 2013.

[180] A. Vion and C. Geuzaine. Double sweep preconditioner for optimized

schwarz methods applied to the helmholtz problem. *J. Comput. Phys.*, 266: 171–190, June 2014. ISSN 0021-9991. doi: 10.1016/j.jcp.2014.02.015. URL http://dx.doi.org/10.1016/j.jcp.2014.02.015.

[181] A. Vion, R. Sabariego, and C. Geuzaine. A model reduction algorithm for solving multiple scattering problems using iterative methods. *Magnetics, IEEE Transactions on*, 47(5):1470–1473, May 2011. ISSN 0018-9464. doi: 10.1109/TMAG.2010.2078800.

[182] A. Vion, R. Bélanger-Rioux, L. Demanet, and C. Geuzaine. A DDM double sweep preconditioner for the Helmholtz equation with matrix probing of the DtN map. In *Mathematical and Numerical Aspects of Wave Propagation WAVES 2013*, June 2013.

[183] M.-F. Xue and J.-M. Jin. A preconditioned dual-primal finite element tearing and interconnecting method for solving three-dimensional time-harmonic maxwell's equations. *Journal of Computational Physics*, 274(0):920 – 935, 2014. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2014.06.040.

[184] J. Yeo, V. V. S. Prakash, and R. Mittra. Efficient analysis of a class of microstrip antennas using the characteristic basis function method (cbfm). *Microwave and Optical Technology Letters*, 39(6):456–464, 2003. ISSN 1098-2760. doi: 10.1002/mop.11247. URL http://dx.doi.org/10.1002/mop.11247.

[185] J. Zhongxiao. The convergence of Krylov subspace methods for large unsymmetric linear systems. *Acta Mathematica Sinica*, 14(4):507–518, 1998.