SCALABLE DOMAIN DECOMPOSITION METHODS FOR TIME-HARMONIC WAVE PROPAGATION PROBLEMS

BY

Alexandros Kyriakis

Supervised by

Victorita Dolean

A thesis submitted to the
Department of Mathematics and Statistics
University of Strathclyde
in partial fulfilment of the require for the degree of

Doctor of Philosophy

Academic Field **Mathematics**

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed:

Date: 19/02/2022

Abstract

The construction of efficient solvers for non self-adjoint problems, like Helmholtz equations is a challenging task. After the discretisation of the PDE by a finite element method, the resulting linear systems are large and because of their spectral properties, difficult to analyse theoretically and to solve by iterative methods. Domain decomposition methods are hybrid methods, as they use an iterative coupling of smaller problems which are solved in turn by direct methods. They rely on dividing the global problem into local subproblems on smaller subdomains. These methods can be used as iterative solvers but also as preconditioners in a Krylov method. Robustness with respect of the number of subdomains is important as this is related to the notion of scalability. We focus here on a configuration where scalability is achieved without the addition of a coarse-space correction. However, convergence can still be improved by modifying the transmission conditions imposed between the subdomains.

In this manuscript, we start by giving an overview of the basic domain decomposition methods and their use as preconditioners. Then we consider these methods from an iterative point of view and we perform a study of convergence analysis of overlapping Schwarz methods with Dirichlet, Robin, zeroth and second order transmission conditions for many subdomains. We also present more sophisticated methods, which implement more effective transmission conditions depending on some optimised parameters. In our analysis, we focus on the Helmholtz problem and the magnetotelluric approximation of Maxwell's equation for stripwise decompositions into many domains. Our theoretical findings are being demonstrated by the appropriate numerical evidence.

Acknowledgments

Firstly, I would like to deeply thank the reviewers of my thesis, Dr Gabriele Ciaramella from Politecnico di Milano and Prof. John Mackenzie from University of Strathclyde for their patience, effort and determination to go through my manuscript which will give me the opportunity to broaden my horizons on my field of research.

I would also like to wholeheartedly express my appreciation towards Prof. Martin Jakob Gander from the University of Geneva for his insight, ideas, remarks, suggestions and the enthusiasm that has showed towards my work. It was a great privilege for me to be able to have fruitful conversations with one of the best specialists of domain decomposition methods.

I would also like to thank my parents and my brother Mario for their support and understanding throughout this journey of life. Also I would like to include James, Ioana, Andrew, Michael, Alistair who where the people that I first met and I had productive conversations with them.

Last but not least, I would like to wholeheartedly express my endless gratitude and appreciation towards my supervisor Victorita Dolean for her understanding, patience, guidance and support of my research. Her advice and remarks helped me in my work and in producing a good manuscript.

Contents

List of Figures

List of Tables

Introduction

Motivation

Computational electromagnetics or electromagnetic modeling is the process of modeling the interaction of electromagnetic fields with physical objects and the environment having precise and identified physical characteristics. A realistic environment is very often heterogeneous, i.e. the physical properties depend on the spatial location which makes the problem even more difficult to solve. The typical mathematical model is Maxwell's equations which is a three dimensional system of PDEs for which we need computationally efficient approximations.

Although the principles of the propagation of electromagnetic waves are generally understood, their application to practical configurations is highly complicated and far beyond analytical calculations in most of the cases. These complications arise from the geometry (of a general shape or presenting singularities) of the medium, its physical properties (heterogeneity, physical dispersion and dissipation) and the characteristics of the sources (e.g. wires) which can condition the type of the solution. As a general rule, whenever the model can be simplified mathematically because of one of these aspects can be neglected in a given practical situation, it will often be.

However, in most of the cases, propagation of electromagnetic waves is three dimensional in nature, the unknowns are time-dependent vector fields (or complex valued in the case of the time-harmonic versions of these equations) and the medium is heterogeneous. Numerical approximation needs to take into account all these aspects. However, the significant advances in computer modelling of electromagnetic interactions that happened over the last decades have been such that nowadays the design of electromagnetic devices heavily relies on computer simulations. Computational electromagnetics has thus taken on great technological importance and, largely due to this, it has become a central discipline in present-day computational science.

Mathematical model

The system of Maxwell's equations modelling the propagation of electromagnetic waves is obtained by combining a few fundamental physical laws (Gauss', Ampère's and Faraday's)

$$\begin{array}{rcl} -\dfrac{\partial \mathbf{D}}{\partial t} + \nabla \times \mathbf{H} & = & \mathbf{J}, \\ \dfrac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} & = & 0, \\ \nabla \cdot \mathbf{D} & = & \rho, \\ \nabla \cdot \mathbf{B} & = & 0, \end{array} \tag{1}$$

where the unknown vector fields involved are

- **E** the *electric field*,
- **D** the *electric induction field*,
- **H** the *magnetic field*,
- **B** the *magnetic induction field*.

The given quantities are ρ, *the free charge density* and **J**, *the free current density*, which are in fact related by the equation of charge conservation. Like the names of these quantities suggest, a few of them are related by what we call constitutive relations. For example, in the case of linear isotropic materials, these relations are given by

$$\mathbf{D} = \varepsilon(\mathbf{x})\,\mathbf{E}, \quad \mathbf{B} = \mu(\mathbf{x})\,\mathbf{H}. \tag{2}$$

where $\mathbf{x} := (x, y, z)$ denotes the vector of spatial coordinates. The coefficient $\varepsilon(\mathbf{x})$ is called the *permittivity*, which measures the ability of a material to be electrically polarized, and $\mu(\mathbf{x})$ is called the *permeability*, which measures the ability of a material to be magnetized. Note that, in the case of anisotropic materials ε and μ are symmetric positive definite matrices. These physical quantities do not depend on time in general unless the propagation takes place in dispersive media like for example in the case of nanophotonics [Dru00] [1].

[1] We can use the Drude model `https://en.wikipedia.org/wiki/Drude_model` stating that the physical properties depend on a given frequency for the Fourier transformed equations

The *law of conservation of the electric charge* does not appear as a separate equation in the classical Maxwell equations, but it is part of one of them. It states that an electric charge can neither be created nor destroyed and can be written in differential form as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0. \tag{3}$$

Using the constitutive relations, we can rewrite Maxwell's equations in their classical form containing only the unknown vector fields **E** and **H**,

$$\begin{aligned} \varepsilon \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{H} &= -\mathbf{J}, && (4)\\ \mu \frac{\partial \mathbf{H}}{\partial t} + \nabla \times \mathbf{E} &= 0, && (5)\\ \nabla \cdot (\varepsilon \mathbf{E}) &= \rho, && (6)\\ \nabla \cdot (\mu \mathbf{H}) &= 0. && (7) \end{aligned}$$

From this set of equations we can remove the two Gauss laws (6) and (7). The reason is, we can easily prove that if these laws are satisfied by the electric and magnetic fields at the initial time, this will be the case for each time t.

The time dependent Maxwell equations (4), (5), (6), and (7) need also initial conditions of the form

$$\mathbf{E}(\mathbf{x}, 0) = \mathbf{E}_0(\mathbf{x}) \quad \text{and} \quad \mathbf{H}(\mathbf{x}, 0) = \mathbf{H}_0(\mathbf{x}), \tag{8}$$

and what we did not observe yet is that Maxwell's equations contain an intrinsic geometric property: by applying the divergence operator to (5), we obtain for μ constant in time

$$\nabla \cdot \left(\mu \frac{\partial \mathbf{H}}{\partial t} + \nabla \times \mathbf{E} \right) = \nabla \cdot \left(\mu \frac{\partial \mathbf{H}}{\partial t} \right) = \frac{\partial}{\partial t} \left(\nabla \cdot (\mu \mathbf{H}) \right) = 0, \tag{9}$$

since the divergence of the curl vanishes, $\nabla \cdot (\nabla \times \mathbf{E}) = 0$. This implies that if Gauss' law (7) for magnetism is verified at the initial time, $\nabla \cdot (\mu \mathbf{H}_0) = 0$, then it is verified for all time t, since this quantity does not change over time because of (3). Similarly, by applying the divergence operator to (4), we obtain for ε constant in time and using the conservation of the charge (3) that

$$\nabla \cdot \left(\varepsilon \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{H} \right) + \nabla \cdot \mathbf{J} = \frac{\partial}{\partial t} \left(\nabla \cdot (\varepsilon \mathbf{E}) \right) - \frac{\partial \rho}{\partial t} = \frac{\partial}{\partial t} \left(\nabla \cdot (\varepsilon \mathbf{E}) - \rho \right) = 0. \tag{10}$$

This implies that if Gauss' law (6) is verified at the initial time, $\nabla \cdot (\varepsilon \mathbf{E}_0) = \rho(0)$, then is is also verified for all time t. So it suffices to consider only the first two sets of Maxwell's equations

$$\varepsilon \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{H} = -\mathbf{J}, \quad \mu \frac{\partial \mathbf{H}}{\partial t} + \nabla \times \mathbf{E} = 0, \tag{11}$$

but with initial conditions satisfying the last two,

$$\nabla \cdot (\varepsilon \mathbf{E}_0) = \rho, \quad \nabla \cdot (\mu \mathbf{H}_0) = 0. \tag{12}$$

A conforming discretisation, by finite elements for example, will need to satisfy these laws for these assumptions to hold (i.e. we need a discrete counterpart of the divergence and the curl which satisfy the same relations as in the continuous case) and in general this is very difficult to achieve. The classical Yee finite difference scheme [Yee66] and Nédélec edge finite elements [Ned01] provide naturally this discrete framework.

There is furthermore another constitutive relation called *Ohm's law*, which given in its modified form

$$\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}_e, \tag{13}$$

where σ is a material dependent parameter called the *conductivity*. Here we consider the presence of a possible further external current source $\mathbf{J}_e$. This leads to the *damped Maxwell's equations*, where the equation (4) is replaced by

$$\varepsilon \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{H} + \sigma \mathbf{E} = -\mathbf{J}_e. \tag{14}$$

Second order Maxwell's equations

It is possible to eliminate half of the unknowns in the Maxwell's equations. For example by applying the curl operator $\nabla\times$ to the equation (5) after dividing by μ and after inserting into the equation (4) after taking a time derivative of this equation, and we obtain the classical *second order form of Maxwell's equations*,

$$\varepsilon \frac{\partial^2 \mathbf{E}}{\partial t^2} + \nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{E} \right) = -\frac{\partial \mathbf{J}}{\partial t}. \tag{15}$$

Similarly, one can also obtain a second order formulation containing only the magnetic field **H**:

$$\mu\frac{\partial^2\mathbf{H}}{\partial t^2}+\nabla\times\left(\frac{1}{\varepsilon}\nabla\times\mathbf{H}\right)=\nabla\times\mathbf{J}. \tag{16}$$

A further simplification is possible if ε and μ are constant: in that case we obtain for example from the second order formulation for the electric field **E** in (15) that

$$\varepsilon\frac{\partial^2\mathbf{E}}{\partial t^2}+\frac{1}{\mu}\left(\nabla\times(\nabla\times\mathbf{E})\right)=-\frac{\partial\mathbf{J}}{\partial t}, \tag{17}$$

and using the identity $\nabla\times(\nabla\times\mathbf{E})=\nabla(\nabla\cdot\mathbf{E})-\Delta\mathbf{E}$, where the Laplacian Δ is applied to each component in **E**, we get

$$\varepsilon\frac{\partial^2\mathbf{E}}{\partial t^2}+\frac{1}{\mu}\left(\nabla(\nabla\cdot\mathbf{E})-\Delta\mathbf{E}\right)=-\frac{\partial\mathbf{J}}{\partial t}. \tag{18}$$

Now if the initial conditions for the first order equations satisfy the Gauss laws, then we know that this Gauss law is satisfied: $\varepsilon\nabla\cdot(\mathbf{E})(\mathbf{x},t)=\rho(\mathbf{x},t)$ for all time hence we can replace the term $\nabla\cdot\mathbf{E}$ in (18) to obtain

$$\frac{\partial^2\mathbf{E}}{\partial t^2}=\frac{1}{\mu\varepsilon}\Delta\mathbf{E}-\frac{1}{\mu\varepsilon}\nabla\left(\frac{\rho}{\varepsilon}\right)-\frac{1}{\varepsilon}\frac{\partial\mathbf{J}}{\partial t}. \tag{19}$$

We thus see that the electric field **E** satisfies component wise a second order wave equation with wave speed $c:=\frac{1}{\sqrt{\mu\varepsilon}}$, and a similar computation shows that also the magnetic field **H** satisfies such a vector valued second order wave equation. Even we cannot solve directly the wave equations above, from here we see that clearly the electric and magnetic field solutions have wave character.

Time-harmonic Maxwell's equations

In many contexts, it is of interest to study the electromagnetic field associated to currents and charges that admit a harmonic dependence in time with a prescribed frequency f, measured in Hz. In this case, according to the limiting amplitude principle [Mor62], the electric and magnetic fields follow with a time harmonic source current $\mathbf{J}:=\hat{\mathbf{J}}e^{-i\omega t}$, where $\omega:=2\pi f$, a time harmonic behavior with the same frequency in the long time limit, regardless of the initial conditions,

$$\mathbf{E}(\mathbf{x},t)=\Re\hat{\mathbf{E}}(\mathbf{x},\omega)e^{-i\omega t},\quad \mathbf{H}(\mathbf{x},t)=\Re\hat{\mathbf{H}}(\mathbf{x},\omega)e^{-i\omega t}.$$

The positive real parameter ω is called the *pulsation* of the time harmonic wave. The quantities $\hat{\mathbf{E}}$ and $\hat{\mathbf{H}}$ are called *complex amplitudes* of $\mathbf{E}$ and $\mathbf{H}$, or time harmonic vector fields. They satisfy the *time-harmonic Maxwell's equations*

$$\begin{aligned} \nabla \times \hat{\mathbf{H}} + i\omega\varepsilon\hat{\mathbf{E}} &= \hat{\mathbf{J}}, \\ \nabla \times \hat{\mathbf{E}} - i\omega\mu\hat{\mathbf{H}} &= \mathbf{0}. \end{aligned} \tag{20}$$

Eliminating $\hat{\mathbf{H}}$ in (20), the system can also be recast as a second order PDE for $\hat{\mathbf{E}}$, which leads to the curl-curl formulation

$$\varepsilon\omega^2\hat{\mathbf{E}} - \nabla \times \left(\frac{1}{\mu}\nabla \times \hat{\mathbf{E}}\right) = -i\omega\hat{\mathbf{J}}. \tag{21}$$

Similarly, one can also obtain a curl-curl formulation for $\hat{\mathbf{H}}$,

$$\mu\omega^2\hat{\mathbf{H}} - \nabla \times \left(\frac{1}{\varepsilon}\nabla \times \hat{\mathbf{H}}\right) = -\nabla \times (\frac{1}{\varepsilon}\hat{\mathbf{J}}). \tag{22}$$

In order to simplify the notation, we drop in the following the hat symbol for the time-harmonic vector fields when there is no confusion with their time-domain counterparts. If we suppose now that Gauss' laws are satisfied we obtain the time-harmonic counterpart of (19)

$$-\omega^2\mathbf{E} - \frac{1}{\mu\varepsilon}\Delta\mathbf{E} + \frac{1}{\mu\varepsilon}\nabla\left(\frac{\rho}{\varepsilon}\right) = i\omega\mathbf{J}, \tag{23}$$

which is a vector valued Helmholtz equation.

In what follows, we will mostly focus on time-harmonic problems in their different variants which can be either Maxwell's or Helmholtz equations. We shall see that there is another simplified model that can be used in certain situations.

The magnetotelluric approximation of Maxwell's equations: a complex-diffusion model

Maxwell's equations can also be used to model the propagation of electromagnetic waves in the subsurface of the Earth. Radar imaging for example is based on the interaction of waves with the subsurface which will further produce measurable responses carrying information about the structure of the underground. This will allow geophysicists, by solving inverse problems, to detect the presence of minerals or oil. Further simplification can be made by supposing there is a certain invariance with respect to one direction parallel to the subsurface. This way the model can be reduced to a two dimensional one and after simplification this leads to the magnetotelluric approximation of Maxwell's equations

$$(\sigma - i\varepsilon)u - \Delta u = -f$$

where f is the source function and u can be either the complex amplitude of the electric or magnetic field depending on the approximation used. The magnetotelluric approximation is a well posed scalar problem, however the operators involved are not self-adjoint and its numerical solution can present a certain number of challenges.

In this thesis we aim at designing solution methods for simplified versions of Maxwell's equations such as Helmholtz and complex diffusion and generalise the methods to Maxwell's equations whenever possible.

Domain decomposition methods

After discretisation of the previous equations, say by a finite element method we obtain linear systems whose underlying matrix is usually large and although symmetric is non-hermitian. In this case we cannot use direct methods as the complexity increases quickly, in the sense that the problems that arise are indefinite and ill-posed and iterative methods lack robustness. Moreover, in the high frequency regime the situation tends to be even worse making the problem even harder [GZ19]. Especially for time harmonic equations, when we are in the high frequency regime the wavenumber is large and this results to an ill-conditioned problem [EG12]. In addition to that, the highly oscillatory nature of the solutions and the pollution effect [BS97] add an extra difficulty to the task. For the previously mentioned reasons, direct solvers are not appropriate for this class of equations because they are mostly applied to well-conditioned equations, have memory limitations and they utterly fail to handle the pollution effect where the wavenumber scales with the mesh size in a particular way.

It is well known that direct methods are usually very robust and provide the exact solution (up to the machine precision) after a finite number of steps but they suffer from memory requirements and have poor parallel properties. Iterative methods, on the other hand, are not exact and provide usually only a sequence of approximations to the solution but their properties are dependent on the properties of the matrix (in the case of a non-normal or non-hermitian matrix, this would be the field of values for which is in general very difficult to find appropriate bounds). In this case, we need a preconditioner.

Domain decomposition methods are hybrid methods since they have both an iterative and direct ingredient. We usually couple iteratively smaller problems which are solved by direct methods. By this kind of technique we usually achieve the best of the two worlds. Domain decomposition methods are based on the simple idea of *divide et impera*: reformulate the global problem into subproblems by decomposing the domain into smaller subdomains on which problems are solved on parallel, communicate the results and update the solutions in an iterative manner.

Among the domain decomposition methods we distinguish two different types: *overlapping* and *non-overlapping*. In the second case the subdomains share only one interface (an artificial common boundary created by the decomposition) whereas in the first subdomains have more in common than just the interface which usually leads to a better convergence at the expense of a double storage of data.

Domain decomposition methods are commonly used as a preconditioners in a Krylov type method, which means that instead of solving the global problem defined by

$$\mathbf{A}\boldsymbol{U} = \boldsymbol{F} \qquad\qquad \text{we solve} \qquad\qquad \mathbf{M}^{-1}\mathbf{A}\boldsymbol{U} = \mathbf{M}^{-1}\boldsymbol{F}.$$

If $\mathbf{M}^{-1}$ is a good approximation of $\mathbf{A}^{-1}$, then the spectral properties of $\mathbf{M}^{-1}\mathbf{A}$ are much better than those of $\mathbf{A}$.

They can also be used as solvers although they are rather slow and for this reason their interest is rather limited. However convergence studies are very revealing even if they are performed first at the iterative level to give a useful overview.

Content and contributions

One of the objectives of this thesis is the analysis of domain decomposition methods for complex problems and the study of their scalability properties. We mainly consider these methods from an iterative point of view (although numerical results with Krylov acceleration are also considered) and a strip-wise decomposition in many subdomains. The analysis of the continuous methods will give us the insight and a more general idea, and the extensive numerical simulations of the discrete methods will provide a quantitative idea. We start by the Helmholtz problem which is notoriously challenging from the iterative methods point of view and we continue with the magneto-telluric approximation to the Maxwell's equations (also called the complex-diffusion problem) by deriving optimised variants of Schwarz methods at the theoretical level and illustrating the theoretical findings by numerical results.

Chapter 2 In this chapter we analyse the iterative counterpart of the Schwarz algorithm in the case of a decomposition into many subdomains. We show that under certain conditions on the parameters, the classical Schwarz method (using Robin transmission conditions) is scalable. The analysis is facilitated by writing the iterative method in matrix form where the iteration matrix is in fact a non-hermitian block-Toeplitz matrix. No results from the literature can be used to characterise its spectrum and our first purpose is to give a close formula for the limiting spectrum when the size of the matrix increases. We can use this result to quantify and analyse the convergence factor of the algorithm in the case of many subdomains for one and two dimensional Helmholtz and Maxwell's equations.

Chapter 3 In this chapter we derive and analyse optimised variants of the Schwarz method for the complex-diffusion equations. We use again the idea of limiting spectrum in order to have an estimate of the convergence factor. Then, we optimise this convergence factor w.r.t. the Robin parameters from the interface transmission conditions by solving numerically a min-max problem. We are able to provide asymptotic formulae of these parameters that can be used in numerical implementations.

Chapter 4 In this chapter we present numerical results for the algorithms introduced in Chapter 3.

Appendix A includes numerical optimisation with Matlab used in Chapter 3.

Appendix B contains FreeFem++ codes used to generate the numerical results in Chapter 4.

The content of the Chapter 2 and 3 gave raise to the following contributions:

- N. Bootland, V. Dolean, A. Kyriakis, J. Pestana, *Analysis of parallel Schwarz algorithms for time-harmonic problems using block Toeplitz matrices*, accepted for publication in ETNA, see the preprint for a preliminary version `https://arxiv.org/abs/2006.08801`

- V. Dolean, M.J. Gander, A. Kyriakis, *Optimizing transmission conditions for multiple subdomains in the Magnetotelluric Approximation of Maxwell's equations*, accepted for publication in the Proceedings of the 26th International Conference on Domain Decomposition Methods, preprint `https://arxiv.org/abs/2103.07879`

Chapter 1

Domain decomposition methods and preconditioners

With the availability of supercomputers, it has become necessary to design robust and efficient algorithms. By robust we mean weakly dependent of the physical properties of the medium which can be the permittivity or permeability for Maxwell's equations or the distribution of wave propagation speed leading to different wave-numbers for the Helmholtz problem. Computational efficiency and robustness (which means the weak dependence on the the number of processors available) can be measured mathematically using convergence rate and condition number estimates. The final purpose is obtaining the solution in an optimal time given the computational resources. Domain decomposition (DD) algorithms are very suitable candidates.

We will give a short introduction to domain decomposition methods. What we commonly call *classical Schwarz method* was introduced for the first time by the German analyst Hermann Amandus Schwarz back in 1870 with the sole purpose of proving the existence and uniqueness of the solution of a Dirichlet Poisson boundary value problem on a domain composed of the union of a rectangle and a circle (as seen in Figure 1.1). The reason is, for those domains Fourier transform techniques for the computation of the solution were not available.

The method consisted in an alternate iteration (1.2) (that will be described in the next section) which was converging towards the solution of the boundary value problem (BVP). Even if the method is relatively old, as it was discovered in the 19th century, it has regained a lot of interest in the 20th century with the advent of the parallel computers. The initial method was not parallel (since it was not designed with parallel computers in mind), a *parallel version* of it was introduced by P.-L. Lions, which represents in fact only a slight modification of the original method. The seminal works of Lions were presented on the occasion of the first domain decomposition conference (as a reminder, the very last one was the 26th one) and since then the literature on the topic covering various aspects of the field has considerably expanded.

Among the reference works, we would like to mention several books and reference monographs. In chronological order, the first one is [SBG96] which presents the methods essentially from an algebraic point of view and by using matrix formulations of problems, illustrating them on different applications. This was probably the most practical presentation of domain decomposition methods. Another reference book by Quarteroni and Valli [QV99] defines and analyses these methods on the continuous versions of BVP and PDE models, being less focused on computational aspects. However, it is more focused on the analysis of simple configurations and less about computational notions as scalability and parallel performance.

Later on, Toselli and Widlund [TW05] discuss in their monograph, domain decomposition methods for finite element discretisations presenting rigorous analysis for a variety of problems and an overview of the properties of these as preconditioners. In the analysis of preconditioners, a very important aspect is the estimate of various condition numbers of preconditioned operators which is a good indicator of scalability of the algorithms.

The most recent book from this series, co-authored by V. Dolean, P. Jolivet and F. Nataf [DJN15], includes also the optimized methods, new advances in coarse spaces and provides implementations in the Freefem++ open-source finite element software.

As it is common practice, domain decomposition methods are used as preconditioners. However, their analysis as iterative methods is very important as it provides a useful insight on the behaviour of these methods. For this reason we consider in this work both of the aspects.

In their use as solvers, we can also design more sophisticated interface transmission conditions also called optimised transmission conditions. This research topic around *Optimised Schwarz methods* has expanded considerably in the past decades, as it is seen as a cheap way to achieve better convergence for the same computational cost as more classical transmission conditions. Its origin can be found in [Lio90], where for the first time more effective conditions at the interfaces between the subdomains than the usual Dirichlet or Neumann boundary conditions were used. Optimised transmission conditions of Robin type can also be used in a non-overlapping framework not only for overlapping methods as initially designed. During the past decades a rich literature developed on this topic, with applications to various equations.

However in practical applications, most of domain decomposition methods are used as preconditioners in a Krylov method. We can cite the most popular ones like Additive Schwarz (AS) which has been extensively analysed in [TW05] for a large class of symmetric positive definite (SPD) problems. For a SPD problem, this preconditioner remains symmetric which makes it very easy to analyse. However there is another variant called Restricted Additive Schwarz (RAS) which was introduced by X.-C. Cai and M. Sarkis in [CS99] and whose convergence properties were proved to be better than those of the AS method, even if no theory is available. This preconditioner is no longer symmetric even for SPD problems but represents the natural version corresponding to the initial Lions algorithm. Also the Optimised Schwarz methods can also be used as preconditioners and preconditioners are called Optimized RAS (ORAS), Optimized MS (OMS) and Optimized AS (OAS) preconditioners. All these variants are very useful especially in the case of indefinite or non-self adjoint problems like Helmholtz or complex diffusion.

1.1 Classical Schwarz and optimised Schwarz methods

In this section we make a concise introduction to our desired method with the proper definitions. The first domain decomposition method was introduced by Hermann Schwarz back in 1870 in order to solve the following elliptic equation on an irregular domain Ω as shown in Figure 1.1

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ \quad u = g & \text{on } \partial\Omega. \end{cases} \tag{1.1}$$

To solve problem (1.1) on the union of the disk (Ω_1) and the rectangle (Ω_2), Schwarz constructed an iterative method which consists in computing successive approximations repeatedly on the local subdomains on which the solution could be computed by using Fourier series and then exchanging the data between neighbouring subdomains. He proved the convergence of the iterative method to a solution meaning that the solution on the whole domain exists.

This method is now known as the *classical Schwarz method* and can be simply described as follows: given an initial guess u_2^0 one solves iteratively by alternating the successive solves on both subdomains

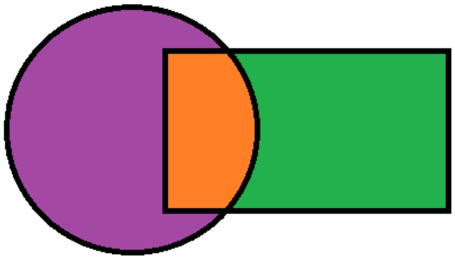

Figure 1.1: The irregular domain of the classical Schwarz algorithm

$$(1.2)\quad \begin{cases} -\Delta u_1^{n+1} &= f & \text{in } \Omega_1 \\ u_1^{n+1} &= 0 & \text{on } \partial\Omega \cap \partial\Omega_1 \\ u_1^{n+1} &= u_2^n & \text{on } \partial\Omega_1 \setminus \partial\Omega \end{cases} \quad \begin{cases} -\Delta u_2^{n+1} &= f & \text{in } \Omega_2 \\ u_2^{n+1} &= 0 & \text{on } \partial\Omega \cap \partial\Omega_2 \\ u_2^{n+1} &= u_1^{n+1} & \text{on } \partial\Omega_2 \setminus \partial\Omega. \end{cases}$$

According to this definition, this algorithm is not parallel and its convergence is very slow. Moreover, in the case of non-overlapping subdomain the algorithm does not converge.

Lions modified the classical Schwarz method in a sequence of two seminal papers presented at the first international domain decomposition conferences [Lio89], [Lio90] (1.2) and proposed a parallel algorithm where the transmission conditions were no longer Dirichlet conditions. This algorithm proved to be convergent for even for non-overlapping domains. This algorithm depends on parameters that appear in the Robin transmission conditions and can be optimised in order to achieve faster convergence. Even if Lions didn't consider the optimisation of interface conditions, his work opened up an avenue of research towards the development of even faster algorithms

$$(1.3)\quad \begin{cases} -\Delta u_1^{n+1} = f \text{ in } \Omega_1 \\ u_1^{n+1} = g \quad \text{on} \quad \partial\Omega_1 \cap \partial\Omega \\ (\partial_{n_1} + p_1)\, u_1^{n+1} = (\partial_{n_1} + p_1)\, u_2^n \quad \text{on} \quad \Omega_2 \cap \partial\Omega_1 \end{cases}$$

$$(1.4) \qquad \begin{cases} -\Delta u_2^{n+1} = f \text{ in } \Omega_2 \\ u_2^{n+1} = g \quad \text{on} \quad \partial\Omega_2 \cap \partial\Omega \\ (\partial_{n_2} + p_2)u_2^{n+1} = (\partial_{n_2} + p_2)u_1^n \quad \text{on} \quad \Omega_1 \cap \partial\Omega_2 \end{cases}$$

where p_1, p_2 are well chosen constants. The constants p_1, p_2 can be computed explicitly, either by analytical or numerical techniques in order to achieve the best convergence possible of the method.

On the other hand, due to its simplicity it can be generalised easily to a well posed boundary value problem defined by the elliptic scalar partial differential operator $\mathcal{L}$

$$\begin{cases} \mathcal{L}u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega. \end{cases}$$

with some further modifications to other categories of problems.

We will study in more detail this kind of algorithms applied to the Helmholtz equation with absorption in Chapter 2 and the magnetotelluric equation in Chapter 3 respectively.

In addition to the one level algorithms, there is a key element that needs to be indicated. Following the work from P. L. Lions [Lio90], the transmission conditions should be taken into account as they have a critical impact on the convergence. Since they can accelerate the convergence of an algorithm at a minimal cost, transmission conditions are an important ingredient of domain decomposition methods and they should be used whenever possible. If information is not exchanged efficiently between neighboring subdomains, the iterative method is not effective and might even not converge. How good transmission conditions should be defined depends on the problem under consideration. Standard Dirichlet transmission conditions, where the values of the local solution on the interface are passed to the neighboring subdomains, work fairly well for the Laplace equation and a simple Robin condition can further improve convergence. For the Helmholtz equation, however, they fail to reduce the error in large parts of the spectrum and Dirichlet conditions are therefore not suitable. Not to mention that in the case of Helmholtz problem, Dirichlet boundary value problems might be ill-posed and depending on the frequency and shape of the domain local problems may become singular. The use of Robin condition is key in the case of Helmholtz problem.

From practical point of view, the general procedure of computing optimised transmission condition relies on involved computations of convergence factors via Fourier analysis in a simplified and simple framework. Very often these computations are done in the case of a decomposition into two unbounded subdomains and the subsequent parameters are then used in the case of many subdomains. In this work we will show a technique where the computation is done for many subdomains at once and in a bounded case. We will establish the link with the results that can be obtained in the case of two-subdomains.

1.2 Scalability of Schwarz methods

Since the main focus of the thesis is scalability of domain decomposition algorithms applied to time harmonic wave propagation problems, we will give below a few important definitions both from a mathematical and from a computational point of view.

Domain decomposition methods are naturally parallel and are therefore perfect candidates for a parallel computing environment. To be more precise, parallel computing is the use of multiple processing elements simultaneously for solving any complex problem. From a parallel computing point of view, scalability is defined as the ability to handle more work in an optimal time as the size of the computational units or of the problem to solve grow. Scalability or scaling is widely used to indicate the ability of hardware and software to deliver great computational power when the amount of resources is increased.

We can distinguish two different aspects of scalability i.e. strong and weak scaling. Strong scaling refers to solving a large but a fixed size problem when a large computational platform is available. In an ideal world a problem would scale in a linear fashion, which means the program would speed up by a factor of N when it runs on a machine having N nodes. From a practical point of view this is not always the case since communication between different processes comes into play and can further slow down the whole process. We can aim for a nearly linear speedup or to be as close as possible to a linear speedup.

As far as the weak scaling is concerned, both the number of processors and the problem size are increased while keeping a constant workload per processor. Weak scaling is mostly used for large memory bound applications where the required memory cannot be satisfied by a single node. For an application that scales perfectly weakly, the work done by each node remains the same as the scale of the machine increases, which means that we are solving progressively larger problems in the same time as it takes to solve smaller ones on a smaller machine.

There is relevant literature on strong scaling ("Amdahl's law and strong scaling")

[Amd67] and weak scaling (”Gustafson’s law and weak scaling”) [Gus88]. Since our work is focused on Schwarz algorithms, we will use compatible definitions from [CCGV18] where scalability results are obtained for Laplacian equation, and [DJN15].

Definition 1 (Strong scalability)**.** *An algorithm is strongly scalable if the acceleration generated by the parallelization scales proportionally with the number of processors that are used.*

Definition 2 (Weak scalability)**.** *A domain decomposition method is weakly scalable if its rate of convergence does not deteriorate when the number of subdomains grows.*

From a mathematical point of view, the notion of weak scaling is more attractive since it can be quantified by condition number or spectral radius estimates, namely we want to achieve algebraic properties which are not varying when the number of degrees of freedom is kept fixed per domain. Also a purely iterative Schwarz method (without a Krylov acceleration) is weakly scalable if the spectral radius of the iteration matrix is bounded above by a strictly positive constant strictly less than one.

This last definition will be extensively used in Chapter 2 and 3 of this manuscript where the Schwarz algorithms are analysed in detail.

We also need to mention the special case of one-dimensional Helmholtz equation without absorption ($\sigma = 0$). In that case the impedance transmission conditions are exact and the method will lead to convergence a number of iterations equal to the number of domains (this situation arises only in the 1d case as in higher dimensions, exact transmission operators are usually non local). According to the definition above, the method is not weakly scalable although the iteration matrix is nilpotent. We will therefore focus on the case of absorptive media ($\sigma > 0$) which is a key ingredient in proving scalability for one level methods.

In the past years, a few authors have studied extensively the scalability of one-level methods for symmetric positive definite elliptic problems [CG18a], [CG18b], [CHS20]. We should note however that the situation is different for the Helmholtz problem (although it is elliptic, it is non self adjoint). Also the study of decomposition into chains of subdomains (or strip-wise decompositions) is justified by the geometry of a simple yet realistic structure which is a waveguide. The work presented in this manuscript provides a deeper understanding of the scalability analysis for Schwarz methods for wave propagation problems.

1.3 Krylov methods and preconditioning

In [DJN15] it is shown that purely iterative Schwarz methods are in fact stationary iterations of block Jacobi type and for this reason their convergence is potentially very slow. On the other side, suppose that after the discretisation the problem (either Helmholtz of complex diffusion), say by a finite element method, we obtain the following linear system

$$A\boldsymbol{U} = \boldsymbol{F},$$

where A is the discretisation matrix on the domain Ω, $\boldsymbol{U}$ is the vector of unknowns and $\boldsymbol{F}$ is the right hand side. If we want to use a Krylov method, the behaviour of this method depends on the mathematical properties of the matrix. To accelerate the performance of such a method applied to this system we will consider two preconditioners inspired by an overlapping domain decomposition which are naturally parallelisable [DJN15, Chapter 3].

For the sake of completeness we reproduce below some of the results presented in the chapter Krylov methods from [DJN15]. Schwarz methods can be written as preconditioned fixed point iterations

$$\mathbf{U}^{n+1} = \mathbf{U}^n + M^{-1}\mathbf{r}^n, \quad \mathbf{r}^n := \mathbf{F} - A\,\mathbf{U}^n$$

where M^{-1} is the method used (RAS or ASM). When convergent the iteration will converge to the solution of the preconditioned system

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}\,.$$

The above system which has the same solution as the original system is called a preconditioned system; here M^{-1} is called the preconditioner. If we denote the error vector $\mathbf{e}^n := \mathbf{U}^n - \mathbf{U}$, then it verifies

$$\mathbf{e}^{n+1} = (I - M^{-1}A)\mathbf{e}^n = (I - M^{-1}A)^{n+1}\mathbf{e}^0.$$

For this reason $I - M^{-1}A$ is called the *iteration matrix* related to the *stationary iteration.* It is known that a fixed point stationary iteration converges for arbitrary initial error $\mathbf{e}^0$ (that is $\mathbf{e}^n \to 0$ as $n \to \infty$) if and only if the spectral radius of the iteration matrix is inferior to 1, that is $\rho(I - M^{-1}A) < 1$.

In order to show that stationary iterative methods are slow, we start by proving that the solution of a fixed point iteration can be written as a series.

Lemma 1.1 (Fixed point solution as a series)**.** *Let*

$$\mathbf{z}^n := M^{-1}\mathbf{r}^n = M^{-1}(\mathbf{b} - A\mathbf{x}^n)$$

be the residual preconditioned by M at the iteration n for the fixed point iteration

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{z}^n = \mathbf{x}^n + M^{-1}(\mathbf{b} - A\mathbf{x}^n). \tag{1.5}$$

Then, the fixed point iteration is equivalent to

$$\mathbf{x}^{n+1} = \mathbf{x}^0 + \sum_{i=0}^{n}(I - M^{-1}A)^i\,\mathbf{z}^0\,. \tag{1.6}$$

Proof. In order to simplify the presentation we introduce the notation $P = M - A$. Note that we have that

$$\mathbf{x}^{n+1} = \mathbf{x}^n + M^{-1}\mathbf{r}^n = \mathbf{x}^n + \mathbf{z}^n \Rightarrow \mathbf{x}^{n+1} = \mathbf{x}^0 + \mathbf{z}^0 + \mathbf{z}^1 + \ldots + \mathbf{z}^n. \tag{1.7}$$

We can also see that the residual vector $\mathbf{r}^n = \mathbf{b} - A\mathbf{x}^n = -A(\mathbf{x}^n - \mathbf{x})$ verifies

$$\begin{aligned}\mathbf{r}^n &= -A\mathbf{e}^n = (P - M)(M^{-1}P)^n\mathbf{e}^0 = (PM^{-1})^nP\mathbf{e}^0 - (PM^{-1})^{n-1}P\mathbf{e}^0\\ &= (PM^{-1})^n(P\mathbf{e}^0 - M\mathbf{e}^0) = (PM^{-1})^n\mathbf{r}^0.\end{aligned} \tag{1.8}$$

From (1.8) we have that

$$\mathbf{z}^n = M^{-1}\mathbf{r}^n = M^{-1}(PM^{-1})^n\mathbf{r}^0 = M^{-1}(PM^{-1})^nM\mathbf{z}^0 = (M^{-1}P)^n\mathbf{z}^0. \tag{1.9}$$

From (1.7) and (1.9) we obtain

$$\mathbf{x}^{n+1} = \mathbf{x}^0 + \mathbf{z}^0 + (M^{-1}P)\mathbf{z}^1 + \ldots + (M^{-1}P)^n\mathbf{z}^n = \mathbf{x}^0 + \sum_{i=0}^{n-1}(M^{-1}P)^i\,\mathbf{z}^0. \tag{1.10}$$

which leads to the conclusion. Thus the error $\mathbf{x}^{n+1} - \mathbf{x}^0$ is a geometric series of common ratio $M^{-1}P$. Note that (1.10) can be also written in terms of the residual vector.

$$\begin{aligned}\mathbf{x}^{n+1} &= \mathbf{x}^0 + M^{-1}\mathbf{r}^0 + (M^{-1}P)M^{-1}\mathbf{r}^1 + \ldots + (M^{-1}P)^nM^{-1}\mathbf{r}^n\\ &= \mathbf{x}^0 + \sum_{i=0}^{n-1}(M^{-1}P)^iM^{-1}\,\mathbf{r}^0.\end{aligned} \tag{1.11}$$

■

In conclusion the solution of a fixed point iteration is generated in a space spanned by powers of the iteration matrix $M^{-1}P = I - M^{-1}A$ applied to a given vector. The main computational cost is thus given by the multiplication by the matrix A and by the application of M^{-1}. At nearly the same cost, we could generate better approximations in the same space (or better polynomials of matrices). These polynomial spaces of matrices are called Krylov spaces and give rise to different families of Kyrlov methods. The same remark on the performance of stationary iterative methods and Krylov subspace methods can be found in [GC21] (Theorem 32 in section 4.1).

For the reasons mentioned above, preconditioned Krylov methods can be considered as accelerations of the stationary iterative methods. A comprehensive description of the topic can be found in [Saa03], [LS13] or [GC21]. We give a short description below.

In 1931 N. M. Krylov introduced the Krylov subspaces [Kry31] in a paper. The Krylov subspace methods can be distinguished into two families. Methods in the first family are based on orthogonalization of the residual with respect to Krylov space:

- CG, the conjugate gradient method for symmetric positive definite matrices, which was the first method of this type, invented independently by David Hestenes and Eduard Stiefel in 1952.

- SymmLQ, for symmetric but indefinite matrices, invented by Chris Paige and Michael Sanders in 1975. This method is based on the Lanczos process and an LQ factorization of the obtained tridiagonal matrix and thus has a short recurrence with low storage requirements similar to CG. The LQ factorization is the analog of the QR factorization, but with a lower triangular matrix L instead of the upper triangular matrix R. For SPD problems CG and SymmLQ give esentially the same results at convergence but CG is computationally more efficient.

- FOM, the Full Orthogonalization Method, which works for arbitrary matrices, and was invented by Yousef Saad in 1981. The method uses Arnoldi, and thus requires substantially more storage like GMRES.

- BiCGStab, the Bi-Conjugate Gradient method with stabilization, which is also a method for general matrices, invented by Henk A. Van Der Vorst in 1992. The method constructs two bi-orthogonal sequences of vectors. The method uses short recurrences requiring therefore less storage than FOM, but it does not fully solve the problem of orthogonalization like in FOM.

Methods in the second family are based on minimisation of the residual norm:

- QMR, the Quasi-Minimum Residual method, also for general matrices, and using a short recurrence based on the unsymmetric Lanczos process with storage requirements similar to CG. This method was invented by Roland W. Freund and M. Nachtigal in 1991, and only approximately solves the minimization problem.

- MINRES, the minimum residual method, for symmetric but possibly indefinite matrices. This method was also invented by Paige and Sanders in 1975, in the same paper as SymmLQ and uses a short recurrence based on Lanczos process with storage requirements similar to CG.

- GMRES, the Generalized Minimum Residual method, for arbitrary matrices, invented by Saad and Schultz in 1986 based on the Arnoldi process. Even though this method needs a lot of storage, it is very popular for testing preconditioners since it really minimizes the residual.

The Krylov method of choice for matrices with no particular property (e.g. symmetric but non-normal) is GMRES since no a priori assumption is required. However the study of the convergence of the GMRES methods is very involved and not always possible from the theoretical point of view. In this work, we will use the GMRES method for numerical simulations but at the theoretical level we will mainly focus on iterative counterparts of Schwarz methods requiring only estimates of the spectral radius. This is still very revealing for the overall ranking of the methods and for their general behaviour.

1.4 Schwarz methods as preconditioners

The definition of these preconditioners relies on a few ingredients:

- $\mathcal{T}_h$ a triangulation of the computational domain and $\{\mathcal{T}_{h,i}\}_{i=1}^N$ be **a non-overlapping partition** of this triangulation. Such a partition can be typically obtained by using a mesh partitioner like METIS [KK98].

- **An overlapping partition** defined as follows. For an integer value $l \geq 0$, we build the decomposition $\{\mathcal{T}_{h,i}^l\}_{i=1}^N$ such that $\mathcal{T}_{h,i}^l$ is a set of all triangles from $\mathcal{T}_{h,i}^{l-1}$ and all triangles from $\mathcal{T}_h \setminus \mathcal{T}_{h,i}^{l-1}$ that have non-empty intersection with $\mathcal{T}_{h,i}^{l-1}$, and $\mathcal{T}_{h,i}^0 = \mathcal{T}_{h,i}$. With this definition the width of the overlap will be of $2l$. Furthermore, if W_h stands for the finite element space associated with $\mathcal{T}_h$, $W_{h,i}^l$ is the local finite element spaces on $\mathcal{T}_{h,i}^l$ that is a triangulation of Ω_i.

- $\mathcal{N}$: the set of indices of **degrees of freedom** (dofs) of the global finite element space W_h and $\mathcal{N}_i^l$ the set of indices of degrees of freedom of the local finite element spaces $W_{h,i}^l$ for $l \geq 0$.
- **Restriction operators** from the global set of dofs to the local one

$$R_i : W_h \to W_{h,i}^l.$$

  At a discrete level this is a rectangular matrix $|\mathcal{N}_i^l| \times |\mathcal{N}|$ such that if $\boldsymbol{V}$ is the vector of degrees of freedom of $v_h \in W_h$, then $R_i\boldsymbol{V}$ is the vector of degrees of freedom of W_h in Ω_i.

- **Extension operators** from $W_{h,i}^l$ to W_h and its associated matrix are both then given by R_i^T.
- **A partition of unity** D_i as a diagonal matrix $|\mathcal{N}_i^l| \times |\mathcal{N}_i^l|$ such that

$$Id = \sum_{i=1}^{N} R_i^T D_i R_i, \tag{1.12}$$

  where $Id \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$ is the identity matrix.

With these ingredients at hand we can now define the main preconditioners used in this work (see also [DJN15, Chapter 1.4] for details)

- **RAS preconditioner** introduced in [CS99] :

$$M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i \left(R_i A R_i^T\right)^{-1} R_i. \tag{1.13}$$

- **Optimized RAS (ORAS) preconditioner** which is based on local boundary value problem with Robin boundary conditions (absorbing boundary conditions). In this case, let B_i be the matrix associated to a discretisation of the corresponding local BVP on the domains Ω_i with Robin boundary conditions on $\partial\Omega_i \cap \partial\Omega_j$:

$$M_{ORAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i {B_i}^{-1} R_i. \tag{1.14}$$

In this work, we will solve the following preconditioned system by a GMRES method

$$M^{-1}A\mathbf{U} = M^{-1}\mathbf{F}$$

where M^{-1} is given by (1.13) or (1.14). Both versions (1.13) or (1.14) of the Schwarz preconditioners are called *one-level preconditioners*. In other words, the whole domain is decomposed into smaller subdomains. Each local subproblem with the discretisation matrix $A_i = R_i A R_i^T$ (in the case of Dirichlet transmission condifions) or B_i (Robin transmission conditions) is solved by a direct method. If local problems are too large, iterative methods or even domain decomposition methods can be used but we haven't considered this situation here.

There are multiple aspects related to the implementation of these methods, especially when we consider the robustness with respect to physical parameters and parallel performance. Especially for time harmonic problems there are various computational challenges and finding the best numerical strategy is of critical importance. In this case, and mainly for Helmholtz problem we refer the reader to the paper [BDJT21] where all these aspects are carefully analysed for two level domain decomposition preconditioners. This is beyond the scope of this thesis, even if we have in mind the design of scalable methods.

Parallel implementation aspects of Schwarz algorithms can also be found in the Chapter 8 book [DJN15]. In the in the Appendix section of the thesis we provide fully commented codes allowing reproducibility of the results. As a reminder the implementation of the methods is achieved by using Freefem++, a high level programming language for solving PDEs by variational discretisation such as the finite element method.

Chapter 2

Schwarz methods for time harmonic problems with many subdomains

The content of this chapter is an enriched and modified version of the preprint `https://arxiv.org/abs/2006.08801` entitled **”Analysis of parallel Schwarz algorithms for time-harmonic problems using block Toeplitz matrices”**

2.1 Motivation

In this chapter we study the convergence properties of the one-level parallel Schwarz method with Robin transmission conditions applied to the one-dimensional and two-dimensional Helmholtz and Maxwell's equations. One-level methods are not scalable in general. However, it has recently been proven that when impedance transmission conditions are used in the case of the algorithm applied to the equations with absorption, under certain assumptions, weak scalability can be achieved for fixed-size subdomains and no coarse space is required. We show here that this result is also true for the iterative version of the method at the continuous level for strip-wise decompositions into subdomains that can typically be encountered when solving wave-guide problems. The convergence proof relies on the particular block Toeplitz structure of the global iteration matrix. Although non-Hermitian, we prove that its limiting spectrum has a near identical form to that of a Hermitian matrix of the same structure. We illustrate our results with numerical experiments.

2.2 Introduction

Time-harmonic wave propagation problems, such as those arising in electromagnetic and seismic applications, are notoriously difficult to solve for several reasons. At the continuous level, the underlying boundary value problems lead to non self-adjoint operators (when impedance boundary conditions are used). The discretisation of these operators by a Galerkin method requires an increasing number of discretisation points as the wave number grows in order to avoid the pollution effect, that is a shift in the numerical wave velocity with respect to the continuous one [BS97]. This leads to increasingly large linear systems with non-Hermitian matrices that are difficult to solve by classical iterative methods [EG12].

In the past two decades, different classes of efficient solvers and preconditioners have been devised; see the review paper [GZ19] and references therein. One important class is based on domain decomposition methods [DJN15], which are a good compromise between direct and iterative methods. Some of these domain decomposition methods rely on improving the transmission conditions, that pass data between subdomains, to give optimised transmission conditions; see the seminal work on Helmholtz equations [GHM07] and its extension to Maxwell's equations [DGL$^+$15, DGG09, DLP08, EDGL12] as well as to elastic waves [BDG19b, MDG19]. For large-scale problems, in order to achieve robustness with respect to the number of subdomains (scalability) and the wave number, two-level domain decomposition solvers have been developed in recent years: they are based on the idea of using the absorptive counterpart of the equations as a preconditioner, which in turn is solved by a domain decomposition method. These methods were successfully applied to Helmholtz and Maxwell's equations, which arise naturally in different applications [BDG$^+$19a, DJTO20, GSV17].

However, an alternative idea emerged in the last few years by observing that, when using Robin or impedance transmission conditions, under certain assumptions involving the physical and numerical parameters of the problem (i.e., absorption, size of the subdomains, etc.) one-level Schwarz algorithms can scale weakly (have a convergence rate that does not deteriorate as the number of subdomains grows) without the addition of a second level [GGS20, GSZ20]. The notion of scalability here applies over a family of problems rather than for a fixed problem. In essence, weak scalability is achieved such that the convergence rate of the domain decomposition method does not deteriorate for harder problems in the family when an appropriate number of subdomains is used. In other words, adding more subdomains allows us to solve harder problems while achieving the same convergence rate.

Achieving scalability without a coarse space in the case of a decomposition into chains of subdomains was first observed for problems arising in computational chemistry; see [CMS13]. However, the first true scalability analysis, based on Fourier techniques, was developed in [CG17] for a classical parallel Schwarz method on a rectangular chain of fixed-size subdomains and provides the first concrete construction of the Schwarz iteration operator in Fourier space. This technique was extended in [CCGV18] to other types of one-level methods. Weak scalability results for the Laplace problem have been proven for more general chain-type geometries using various techniques, such as the maximum principle in [CG18a] and a fully variational analysis in [CG18b]. The most recent work on the topic without restrictive assumptions can be found in [CHS20] where a propagation-tracking analysis based on graph theory and the maximum principle permitted a scalability analysis for very general decompositions. To our knowledge, there is no such analysis on Schwarz methods for time-harmonic wave propagation problems, where previous techniques no longer extend as the nature of the underlying equations is very different.

In our work, we would like to explore this idea of weak scalability at the continuous level (independent of the discretisation) for a strip-wise decomposition into subdomains as it arises naturally in the solution of wave-guide problems. While in [GGS20, GSZ20] the family of problems is parametrised by the wave number k and the focus is on k-robustness, here we focus on the weak scalability aspect for a family consisting of a growing chain of fixed-size subdomains. Nonetheless, we will see that k-robustness in certain scenarios can easily be derived from our theory. The main contributions of the paper are the following:

- We provide analysis of the limiting spectrum, as the number of subdomains grows, for a one-level Schwarz method applied to a strip-wise decomposition. While our analysis is limited to this simple yet realistic configuration (wave propagation in a rectangular wave-guide with Dirichlet conditions on the top and bottom boundaries and Robin condition at its ends), it is valid at the continuous level both for one-dimensional and two-dimensional Helmholtz and Maxwell's equations.
- We build on the formalism of iteration matrices acting on interface data introduced in [CCGV18] (where Schwarz methods using strip-wise decompositions were analysed for Laplace's equation), but here we are able to characterise the entire spectrum of these iteration matrices if the number of subdomains is sufficiently large by using their block Toeplitz structure, even if upper bounds on the iteration matrix norm could have been derived in a similar manner.

- Despite the fact that the block Toeplitz structure is non-Hermitian, and thus results from the standard literature on Toeplitz matrices do not apply in a straightforward manner, we prove that the limiting spectrum of the iteration matrices as their size grows (corresponding to an increasing number of subdomains) tends to the limit predicted by the eigenvalues of the symbol of the block Toeplitz matrix, except perhaps for two additional eigenvalues. This novel approach, utilising the limiting spectrum, is quite general and can be applied to other problems as an analysis tool for domain decomposition methods where such block Toeplitz structure arises naturally.
- We show that the limiting spectrum is descriptive of what is observed in practice numerically, even for a relatively small number of subdomains.
- As a corollary to our theory we show that, in certain scenarios and with k-dependent domain decomposition parameters, the one-level method can be k-robust as the wave number k increases; in the Maxwell case we believe this to be a novel result.

2.3 A non-Hermitian block Toeplitz structure

Consider a non-Hermitian block Toeplitz matrix $\mathcal{T} \in \mathbb{C}^{2m\times 2m}$ of the form

$$\mathcal{T} = \begin{bmatrix} A_0 & A_1 & & & \\ A_{-1} & A_0 & A_1 & & \\ & \ddots & \ddots & \ddots & \\ & & A_{-1} & A_0 & A_1 \\ & & & A_{-1} & A_0 \end{bmatrix}, \tag{2.1a}$$

where

$$A_0 = \begin{bmatrix} 0 & b \\ b & 0 \end{bmatrix}, \ A_1 = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}, \ A_{-1} = \begin{bmatrix} 0 & 0 \\ 0 & a \end{bmatrix}, \tag{2.1b}$$

for some non-zero complex coefficients a and b. We will see in the sections that follow that such non-Hermitian block Toeplitz structures arise naturally for iterative Schwarz algorithms applied to wave propagation problems. We are interested in a characterisation of the complete spectrum of the matrix $\mathcal{T}$ in (2.1) when its dimension becomes large. This will equate to the number of subdomains N in the Schwarz method being large. The coefficients a and b stem from the particular PDE and domain decomposition used; we consider them to be fixed independent of the dimension of $\mathcal{T}$, and thus N, which corresponds to fixed-size subdomains.

The so-called Szegő formula enables the asymptotic spectrum, i.e., the spectrum as $m \to \infty$, of a wide class of Hermitian block Toeplitz matrices to be characterised by the eigenvalues of an associated matrix-valued function called the (block) symbol [Til98]. For non-Hermitian matrices, analogous results do not exist in general [Til98], but do hold when the union of the essential ranges of the eigenvalues of the block symbol has empty interior and does not disconnect the complex plane [DNSC12]. Unfortunately, $\mathcal{T}$ in (2.1) has symbol $F(z) = A_{-1}z + A_0 + A_1 z^{-1}$ and, for relevant values of a and b, the union of essential ranges is a closed curve. Additional characterisations of the asymptotic spectrum of (block) banded Toeplitz matrices are available [Hir67, SS60, Wid74], but these do not provide explicit formulae for the eigenvalues, as we shall in Theorem 2.1. Other formulae for the eigenvalues [SM13] and determinant [Tis87] of block tridiagonal Toeplitz matrices are known, however, they are applicable only when A_1 (or A_{-1}) is nonsingular.

We also remark that the matrix $\mathcal{T}$ will be an iteration matrix in the Schwarz algorithms we later consider. Hence, to prove convergence of these Schwarz methods it would be sufficient to bound the spectral radius of $\mathcal{T}$, for example using a matrix norm. It is straightforward to see that $\|\mathcal{T}\|_\infty = |a| + |b|$, and it is also possible to show, using [SC02, Corollary 3.5], that

$$\|\mathcal{T}\|_2 \le \max\left\{\sqrt{|a|^2 \pm 2\Re(a\bar{b}) + |b|^2}\right\}.$$

However, since a and b are complex neither norm is straightforward to bound above by 1. Additionally, characterising the full spectrum provides more information than the spectral radius alone. Accordingly, in this section we derive the limiting spectrum of $\mathcal{T}$.

In order to establish a result on the spectrum of $\mathcal{T}$, we first show that the characteristic polynomials of (2.1) for increasing m obey a three-term recurrence relation.

Lemma 2.1 (Three-term recurrence and generating function). *Let $p_m(z)$ denote the characteristic polynomial of the block Toeplitz matrix $\mathcal{T} \in \mathbb{C}^{2m\times 2m}$ defined in* (2.1). *Then $p_m(z)$ satisfies the three-term recurrence relation*

$$p_m(z) + B(z)p_{m-1}(z) + A(z)p_{m-2}(z) = 0, \qquad \textit{for } m \ge 2, \tag{2.2}$$

with $A(z) = a^2z^2$ *and* $B(z) = -z^2 + b^2 - a^2$ *and where* $p_0(z) = 1$ *and* $p_1(z) = z^2 - b^2$. *Furthermore, this recurrence relation is encoded in the generating function*

$$\sum_{m=0}^{\infty} p_m(z)t^m = \frac{N(t,z)}{D(t,z)}, \tag{2.3}$$

where

$$D(t,z) = 1 + B(z)t + A(z)t^2, \tag{2.4a}$$

$$N(t,z) = p_0(z) + (p_1(z) + B(z)p_0(z))t. \tag{2.4b}$$

Thus, in our case, $D(t,z) = 1 - (z^2 - b^2 + a^2)t + a^2z^2t^2$ *while* $N(t,z) = 1 - a^2t$.

Proof. We first prove the recurrence relation. Let D_m be the $2m \times 2m$ matrix whose determinant is the characteristic polynomial of $\mathcal{T}$ in the variable z. Note that the first two characteristic polynomials are

$$p_1(z) = \det(D_1) = \begin{vmatrix} -z & b \\ b & -z \end{vmatrix} = z^2 - b^2, \tag{2.5a}$$

$$p_2(z) = \det(D_2) = \begin{vmatrix} -z & b & a & 0 \\ b & -z & 0 & 0 \\ 0 & 0 & -z & b \\ 0 & a & b & -z \end{vmatrix} = (z^2 - b^2)^2 - a^2b^2. \tag{2.5b}$$

To derive a recurrence relation, let us also define the intermediary determinants $r_m(z)$ which arise as the minor of D_m having removed the second row and first column,

$$r_m(z) := \left|\begin{array}{c|cccc} b & a & 0 & 0 & \cdots \\ \hline 0 & & & & \\ a & & D_{m-1} & & \\ 0 & & & & \\ \vdots & & & & \end{array}\right| = \left|\begin{array}{c|cc|cc} b & a & 0 & 0 & \cdots \\ \hline 0 & -z & b & a & 0 \\ a & b & -z & 0 & 0 \\ \hline 0 & 0 & 0 & & \\ \vdots & 0 & a & \multicolumn{2}{c}{D_{m-2}} \end{array}\right| = b\,p_{m-1}(z) + a^2\,r_{m-1}(z),$$

where we use the cofactor expansion of the determinant. Similarly, for $p_m(z)$ we obtain

$$p_m(z) = z^2\,p_{m-1}(z) - b\,r_m(z) = (z^2 - b^2)\,p_{m-1}(z) - a^2b\,r_{m-1}(z).$$

We can then rearrange this relation to give an expression for $r_{m-1}(z)$ in terms of $p_m(z)$ and $p_{m-1}(z)$. Substituting this into the recurrence for $r_m(z)$ above, along with the equivalent expression for $r_m(z)$, yields the desired recurrence relation

$$p_{m+1}(z) = (z^2 - b^2 + a^2)\, p_m(z) - a^2 z^2\, p_{m-1}(z), \tag{2.6}$$

where $A(z) := a^2 z^2$ and $B(z) := -z^2 + b^2 - a^2$. Finally, note that setting $p_0 = 1$ is consistent with this recurrence relation and initial characteristic polynomials (2.5).

To show the equivalence of the generating function, we multiply (2.2) by t^m and sum over $m \geq 2$ before adding relevant terms to isolate $\sum_{m=0}^{\infty} p_m(z) t^m$ as follows

$$\begin{aligned}
&\sum_{m=2}^{\infty} \left[p_m(z) + B(z)p_{m-1}(z) + A(z)p_{m-2}(z)\right] t^m = 0 \\
\iff &\sum_{m=0}^{\infty} \left[1 + B(z)t + A(z)t^2\right] p_m(z)t^m = p_0(z) + \left(p_1(z) + B(z)p_0(z)\right) t \\
\iff &\sum_{m=0}^{\infty} p_m(z)t^m = \frac{p_0(z) + \left(p_1(z) + B(z)p_0(z)\right) t}{1 + B(z)t + A(z)t^2}.
\end{aligned}$$

Substituting in the appropriate values gives $D(t, z) = 1 - (z^2 - b^2 + a^2)t + a^2 z^2 t^2$ and $N(t, z) = 1 - a^2 t$ in our case, as required. ■

Before continuing, we remark on the convergence of the Maclaurin series in t of the generating function. Note that the Maclaurin series of any rational function (without a pole at 0) satisfies a linear recurrence relation, which can be seen by following backwards an analogous argument to that in the above proof. Moreover, the Maclaurin series is convergent (to the rational function) on the open disc centred at 0 with a radius equal to the minimum root of the denominator in absolute value; this can be discerned from a partial fractions decomposition (over $\mathbb{C}$) and noting that it is a (finite) sum of geometric series. As such, in our present case, $p_m(z)$ are precisely the coefficients in the Maclaurin series for any given z since the denominator is such that 0 is never a pole of the generating function and so there is always a non-trivial disc where the series converges.

We now introduce a useful tool that will help us to characterise the spectrum of (2.1): the q-analogue of the discriminant known as the q-discriminant [Tra14]. The q-discriminant of a polynomial $P_n(t)$ of degree n with leading coefficient p is defined as

$$\mathrm{Disc}_t(P_n; q) = p^{2n-2} q^{n(n-1)/2} \prod_{1 \leq i < j \leq n} (q^{-1/2} t_i - q^{1/2} t_j)(q^{1/2} t_i - q^{-1/2} t_j), \tag{2.7}$$

where t_i, $1 \leq i \leq n$, are the roots of $P_n(t)$. A key point is that the q-discriminant is zero if and only if a quotient of roots t_i/t_j equals q. Note that as $q \to 1$ the q-discriminant becomes the standard discriminant of a polynomial.

In particular, we will consider the q-discriminant of the denominator $D(t, z)$ as a quadratic in t. Direct calculation using the quadratic formula yields

$$\operatorname{Disc}_t(D(t, z); q) = q\left(B(z)^2 - (q + q^{-1} + 2)A(z)\right), \tag{2.8}$$

for any $q \neq 0$. If q is a quotient of the two roots in t of $D(t, z)$ then (2.8) is zero and so q must satisfy

$$\frac{B(z)^2}{A(z)} = q + q^{-1} + 2, \tag{2.9}$$

where, in general, q will depend on z. The q-discriminant condition (2.9) for $D(t, z)$ will be crucial in what follows since it will allow us to characterise roots of $p_m(z)$ in terms of the quotient q. We now state our main result on the limiting spectrum of $\mathcal{T}$ as its dimension becomes large in which we adapt some ideas from [Tra14] for finding roots of polynomials verifying a three-term recurrence but now with a different generating function.

Theorem 2.1 (Limiting spectrum). *The limiting spectrum, as $m \to \infty$, of the block Toeplitz matrix $\mathcal{T} \in \mathbb{C}^{2m \times 2m}$, defined in* (2.1)*, lies on the curve defined by*

$$\lambda_{\pm}(\theta) = a\cos(\theta) \pm \sqrt{b^2 - a^2\sin^2(\theta)}, \qquad \theta \in [-\pi, \pi], \tag{2.10}$$

except perhaps for the eigenvalues

$$\lambda = \pm\sqrt{\tfrac{1}{2}b^2 - a^2}, \tag{2.11}$$

which can only occur if $|a^2| > |\frac{1}{2}b^2 - a^2|$.

Proof. Suppose that z_m is a root of the characteristic polynomial $p_m(z)$ for $m \geq 2$. If $z_m = 0$ then we must have that $a^2 = b^2$. To see this, assume for a contradiction that $a^2 \neq b^2$, then $B(0) \neq 0$ while $A(0) = 0$ and $p_m(0) = 0$ and thus the recurrence relation (2.2) gives that $p_{m-1}(0) = 0$. Following this recursion down to $m = 2$ gives that $p_1(0) = 0$, which is false as $b \neq 0$. Further, if $p_m(0) = 0$ then also $p_{m+1}(0) = 0$ by (2.2) since $A(0) = 0$ and so a sequence of zero roots occurs as m increases giving 0 in the limiting spectrum. This case is covered by choosing $\theta = \frac{\pi}{2}$ in (2.10) and noting that

$a^2 = b^2$ must hold. As such, for the remainder of the proof we assume that $z_m \neq 0$.

Now consider the denominator $D(t, z_m)$. Since $A(z_m) \neq 0$ by the assumption that $z_m \neq 0$, the denominator as a quadratic in t has two roots t_1 and t_2. Note that, by Vieta's formula for the product of roots, neither of these two roots can be zero since $t_1 t_2 A(z_m) = 1$. If $t_1 = t_2$ then the (standard) discriminant of $D(t, z_m)$ is zero, giving $B(z_m)^2 - 4A(z_m) = 0$. Solving for z_m given our expressions for $A(z)$ and $B(z)$ yields solutions $z_m = \pm(a \pm b)$ for all choices of signs. These cases are also covered by (2.10) when $\theta = 0$ or $\theta = \pi$.

As such, we now assume that $t_1 \neq t_2$ and so $D(t, z_m) = A(z_m)(t-t_1)(t-t_2)$. Considering the generating function (2.3) we observe that

$$\begin{aligned}\frac{N(t,z_m)}{D(t,z_m)} &= \frac{1-a^2t}{A(z_m)(t-t_1)(t-t_2)} = \frac{1-a^2t}{A(z_m)(t_1-t_2)}\left(\frac{1}{t-t_1} - \frac{1}{t-t_2}\right)\\ &= \frac{1-a^2t}{A(z_m)(t_1-t_2)}\sum_{m=0}^{\infty}\frac{t_1^{m+1}-t_2^{m+1}}{t_1^{m+1}t_2^{m+1}}t^m\\ &= \frac{1}{A(z_m)(t_1-t_2)}\sum_{m=1}^{\infty}\left[\frac{t_1^{m+1}-t_2^{m+1}}{t_1^{m+1}t_2^{m+1}} - a^2\frac{t_1^m-t_2^m}{t_1^m t_2^m}\right]t^m + 1. \qquad (2.12)\end{aligned}$$

The sum introduced in the second line is the Maclaurin series in t and, as the difference of two geometric series, is convergent in the open disc $|t| < \min\{|t_1|, |t_2|\}$. Note that this is non-trivial since neither t_1 or t_2 are zero. In (2.12) we identify that the coefficient of t^m is exactly $p_m(z_m)$. Thus, as z_m is a root of $p_m(z)$, the coefficient of t^m in (2.12) must be zero. Now suppose $t_1 = qt_2$ for some quotient $q \neq 0$ (as neither t_1 nor t_2 is zero), then this condition on the coefficient of t^m translates into

$$\frac{q^{m+1}-1}{q^{m+1}t_2^{m+1}} - a^2\frac{q^m-1}{q^m t_2^m} = 0 \implies q^{m+1} - 1 = a^2 t_2 q(q^m - 1).$$

Since $t_1 t_2 A(z_m) = 1$, we deduce that $t_2 = \pm(A(z_m)q)^{-1/2}$ and thus q must solve

$$\left(q^{m+1}-1\right)^2 = \frac{a^4}{A(z_m)}\left(q^m-1\right)^2 q.$$

Let us define the coefficient, depending on z_m,

$$c_m = \frac{a^4}{A(z_m)} = \frac{a^2}{z_m^2}. \qquad (2.13)$$

Then q must be a root of the $2m+2$ degree polynomial

$$f_m(q) = q^{2m+2} - c_m q^{2m+1} + 2(c_m - 1)q^{m+1} - c_m q + 1. \tag{2.14}$$

In order to characterise the roots of (2.14) we will make use of the following corollary of Rouché's theorem (see, e.g., [Kra99, Section 5.3.2]): for a polynomial f of degree d with coefficients $\{\alpha_j\}_{j=0}^{d}$, if $R > 0$ is such that for an integer $0 \leq k \leq d$ we have

$$|\alpha_0| + \ldots + |\alpha_{k-1}|R^{k-1} + |\alpha_{k+1}|R^{k+1} + \ldots + |\alpha_d|R^d < |\alpha_k|R^k, \tag{2.15}$$

then there are exactly k roots of f, counted with multiplicity, having absolute value less than R. In particular, we will use this result for the polynomial $f_m(q)$ with $k = 0$, $k = 2m+1$ or $k = 2m+2$.

We first point out some facts about (2.14). Note that $q = 0$ is not a root of f_m. Moreover, by symmetry of the coefficients, we have (for $q \neq 0$)

$$f_m(q^{-1}) = q^{-(2m+2)} f_m(q). \tag{2.16}$$

Thus, if q_m is a root of f_m then q_m^{-1} is also a root. Further, since f_m has a unique factorisation in $\mathbb{C}$, applying this both in the variable q^{-1} and q in (2.16) shows that the multiplicities of the roots q_m and q_m^{-1} must be identical. This means that we only need to study roots with $|q_m| \leq 1$, with roots outside the unit disc being precisely the reciprocal values of those inside the unit disc, or vice versa.

We will use (2.15) to determine how many roots of $f_m(q)$ in (2.14) do not approach the unit circle as $m \to \infty$. This information, along with (2.9), will allow us to determine conditions for z_m. A significant challenge is that the coefficient c_m depends on m and so we will need to consider several cases. To proceed, we let $\varepsilon > 0$ be small. We will show that for all $m \geq M$, for a suitable $M(\varepsilon)$, all but potentially two roots of $f_m(q)$ lie in an annulus which shrinks to the unit circle as $\varepsilon \to 0$. The remaining two roots can only persist if $|c_m| > 1$ and, should they exist, consist of a root s_m close to c_m^{-1} and the corresponding reciprocal root outside the unit circle. Given $\varepsilon > 0$, for $m \geq M$ we consider three cases depending on c_m:

1. $|c_m| \leq 1$,
2. $1 \leq |c_m| \leq (1+\varepsilon)^2$,
3. $|c_m| \geq (1+\varepsilon)^2$.

Case 1 To start the analysis we suppose that we are in case 1 so that $|c_m| \leq 1$ and define $R_- = 1 - \varepsilon$. Let M_1 be such that

$$4R_-^{m+1} + R_-^{2m+1} + R_-^{2m+2} < \varepsilon$$

for all $m \geq M_1$. Such an M_1 exists since $|R_-| < 1$. Then, for $m \geq M_1$, we have that

$$\begin{aligned} |c_m|R_- + 2|c_m - 1|R_-^{m+1} + |c_m|R_-^{2m+1} + R_-^{2m+2} &\leq R_- + 4R_-^{m+1} + R_-^{2m+1} + R_-^{2m+2} \\ &< 1. \end{aligned}$$

Thus, for large enough m, by using $k = 0$ in the corollary of Rouché's theorem we deduce that there are no roots of f_m with modulus less than $R_- = 1 - \varepsilon$. In this case, by the reciprocal nature of the roots, for $m \geq M_1$ we conclude that all $2m + 2$ roots q_m of f_m lie in the annulus

$$1 - \varepsilon \leq |q_m| \leq \frac{1}{1 - \varepsilon}. \tag{2.17}$$

Case 2 We now turn to the analysis of case 2 where $1 \leq |c_m| \leq (1 + \varepsilon)^2$. To aid in the next case we first relax this condition to consider $|c_m| \geq 1$ and prove a useful bound for all roots of f_m. Define $R_+ = 1 + \varepsilon$ and let M_2 be such that

$$R_+^{-(2m+1)} + R_+^{-2m} + 4R_+^{-m} < \frac{\varepsilon}{1 + \varepsilon}$$

for all $m \geq M_2$. Now let $R_\top = |c_m|(1 + \varepsilon) = |c_m|R_+$. We will want to show that

$$1 + |c_m|R_\top + 2|c_m - 1|R_\top^{m+1} + |c_m|R_\top^{2m+1} < R_\top^{2m+2}, \tag{2.18}$$

in order to apply the corollary of Rouché's theorem with $k = 2m + 2$. To do so we consider dividing by $R_\top^{2m+2}$, in which case, for $m \geq M_2$, we have

$$\begin{aligned} &R_\top^{-(2m+2)} + |c_m|R_\top^{-(2m+1)} + 2|c_m - 1|R_\top^{-(m+1)} + |c_m|R_\top^{-1} \\ &= |c_m|^{-(2m+2)}R_+^{-(2m+2)} + |c_m|^{-2m}R_+^{-(2m+1)} + \frac{2|c_m - 1|}{|c_m|}|c_m|^{-m}R_+^{-(m+1)} + R_+^{-1} \\ &\leq R_+^{-(2m+2)} + R_+^{-(2m+1)} + 4R_+^{-(m+1)} + R_+^{-1} \\ &< \frac{\varepsilon}{(1 + \varepsilon)^2} + \frac{1}{1 + \varepsilon} < 1. \end{aligned}$$

Thus we have the required inequality and deduce from the corollary of Rouché's theorem that all $2m+2$ roots q_m lie in the disc given by $|q_m| < |c_m|(1+\varepsilon)$. This will prove useful later in case 3. For now we turn back to case 2 where $1 \le |c_m| \le (1+\varepsilon)^2$. Using this upper bound on $|c_m|$ and the reciprocal nature of the roots, we conclude that, for $m \ge M_2$, all $2m+2$ roots q_m of f_m lie in the annulus

$$\frac{1}{(1+\varepsilon)^3} < |q_m| < (1+\varepsilon)^3. \tag{2.19}$$

Case 3 Finally, consider case 3 where $|c_m| \ge (1+\varepsilon)^2$. Let $R_+ = 1+\varepsilon$ and M_2 be as defined in case 2. We will want to show that

$$1 + |c_m|R_+ + 2|c_m - 1|R_+^{m+1} + R_+^{2m+2} < |c_m|R_+^{2m+1}, \tag{2.20}$$

in order to apply the corollary of Rouché's theorem with $k = 2m+1$. To do so we consider dividing by $|c_m|R_{\top}^{2m+1}$, in which case, for $m \ge M_2$, we have

$$\begin{aligned}
&|c_m|^{-1}R_+^{-(2m+1)} + R_+^{-2m} + \frac{2|c_m-1|}{|c_m|}R_+^{-m} + |c_m|^{-1}R_+ \\
&\le R_+^{-(2m+1)} + R_+^{-2m} + 4R_+^{-m} + (1+\varepsilon)^{-2}R_+ \\
&< \frac{\varepsilon}{1+\varepsilon} + \frac{1}{1+\varepsilon} = 1.
\end{aligned}$$

Thus we have the required inequality and deduce from the corollary of Rouché's theorem that $2m+1$ roots q_m lie in the disc given by $|q_m| < 1+\varepsilon$. In this case, by the reciprocal nature of the roots, for $m \ge M_2$ we conclude that $2m$ roots q_m of f_m lie in the annulus

$$\frac{1}{1+\varepsilon} < |q_m| < 1+\varepsilon. \tag{2.21}$$

We pause to note at this stage that, combining all three cases, we have just shown that all but potentially two roots of f_m lie in a small annulus around the unit circle for $m \ge M = \max\{M_1, M_2\}$, independently of the value of c_m. In particular, this will be the largest annulus of the three cases which, for small $\varepsilon > 0$, is that in (2.19). Letting $\varepsilon \to 0$ we deduce that all but potentially two roots of f_m must tend to the unit circle as $m \to \infty$.

The remaining question is what happens to the other two roots, which only appear in case 3. We know from the bound in (2.18) that, for large enough m, all roots satisfy $|q_m| < |c_m|(1+\varepsilon)$ while all but one satisfy $|q_m| < (1+\varepsilon)$. We now show that the remaining root in case 3 satisfies $|q_m| \geq |c_m|(1-\varepsilon)$ for large enough m. To do so let $R_\perp = |c_m|(1-\varepsilon) = |c_m|R_-$ and note that, assuming ε is small enough ($\varepsilon < \frac{1}{3}$ suffices), then $R_\perp > 1$ since

$$R_\perp = |c_m|(1-\varepsilon) \geq (1+\varepsilon)^2(1-\varepsilon) \geq 1 + \frac{\varepsilon}{2}.$$

Now let M_3 be such that

$$\left(1+\frac{\varepsilon}{2}\right)^{-(2m+1)} + \left(1+\frac{\varepsilon}{2}\right)^{-2m} + 4\left(1+\frac{\varepsilon}{2}\right)^{-m} < \varepsilon$$

for all $m \geq M_3$. We will want to show an identical bound to (2.20) holds but now for $R_\perp$ in order to again use the corollary of Rouché's theorem with $k = 2m+1$. We proceed in a similar manner and consider dividing by $|c_m|R_\perp^{2m+1}$, so that for $m \geq M_3$ we have

$$\begin{aligned}
&|c_m|^{-1}R_\perp^{-(2m+1)} + R_\perp^{-2m} + \frac{2|c_m-1|}{|c_m|}R_\perp^{-m} + |c_m|^{-1}R_\perp \\
&\leq R_\perp^{-(2m+1)} + R_\perp^{-2m} + 4R_\perp^{-m} + R_- \\
&\leq \left(1+\frac{\varepsilon}{2}\right)^{-(2m+1)} + \left(1+\frac{\varepsilon}{2}\right)^{-2m} + 4\left(1+\frac{\varepsilon}{2}\right)^{-m} + R_- \\
&< 1.
\end{aligned}$$

Thus we have the required inequality and deduce from the corollary of Rouché's theorem that $2m+1$ roots q_m lie in the disc given by $|q_m| < |c_m|(1-\varepsilon)$. Thus, for large enough m, we conclude that the single remaining root lies in the annulus $|c_m|(1-\varepsilon) \leq |q_m| < |c_m|(1+\varepsilon)$.

This result makes it clear that roots which do not tend to the unit circle persist only when we have $|c_m|$ values which stay bounded away from 1 as $m \to \infty$, and their size is dictated by c_m. That is, for such roots to persist there must exist an infinite subsequence with $|c_m| > c > 1$ for some fixed c and so we now assume this condition. We further focus on the reciprocal root which is inside the unit circle and show that it approximates c_m^{-1} for large m. Define this single root to be s_m and note, through the reciprocal nature of roots, we have just shown that it satisfies the bound $|s_m| \leq |c_m^{-1}|\frac{1}{1-\varepsilon}$, which in turn gives that $|c_m s_m| \leq \frac{1}{1-\varepsilon}$. Moreover, $|c_m| > c$ yields the bound $|s_m| \leq c^{-1}\frac{1}{1-\varepsilon}$, where $c > 1$ is fixed, and thus choosing $\varepsilon > 0$ small enough we have $|s_m| < r < 1$ for a fixed r. This provides the ingredients for the following limit:

$$|s_m^{2m+2} - c_m s_m^{2m+1} + 2(c_m-1)s_m^{m+1}|$$

$$\leq |s_m|^{2m+2} + \frac{1}{1-\varepsilon}|s_m|^{2m} + 2|s_m|^{m+1} + \frac{2}{1-\varepsilon}|s_m|^m \to 0$$

as $m \to \infty$, since $|s_m| < r < 1$. Now, by definition of s_m as a root of f_m, we have that $f_m(s_m) = 0$ and hence we must have that $1 - c_m s_m \to 0$ and thus $s_m - c_m^{-1} \to 0$ as $m \to \infty$, due to $|c_m^{-1}|$ being bounded above by $c^{-1} < 1$. This says that the root which stays inside the unit circle approximates c_m^{-1} for large m while the root which stays outside the unit circle must approximate c_m by reciprocal.

We would now like to interpret what this shows for the potential corresponding root z_m in the limit $m \to \infty$ using the q-discriminant condition (2.9). For this we use the definition of the coefficient $c_m = a^2/z_m^2$ from (2.13) and denote $\delta_m = c_m s_m - 1$ where $\delta_m \to 0$ as $m \to \infty$. Then, with $q = s_m = c_m^{-1}(1 + \delta_m)$, (2.9) becomes

$$\begin{aligned} \frac{B(z_m)^2}{A(z_m)} &= c_m^{-1}(1+\delta_m) + c_m(1+\delta_m)^{-1} + 2 \\ \Longrightarrow \frac{(-z_m^2 + b^2 - a^2)^2}{a^2 z_m^2} &= \frac{z_m^2}{a^2}(1+\delta_m) + \frac{a^2}{z_m^2}(1+\delta_m)^{-1} + 2 \\ \Longrightarrow b^4 - 2a^2b^2 - 2b^2 z_m^2 &= \delta_m(z_m^4 - a^4) + \mathcal{O}(\delta_m^2), \end{aligned} \tag{2.22}$$

where we have used the binomial expansion $(1+\delta_m)^{-1} = 1 - \delta_m + \mathcal{O}(\delta_m^2)$, which is valid for large m since $\delta_m \to 0$. Recall that, given we are in case 3, $|c_m|$ is bounded below away from zero and so $|z_m|$ is bounded above for all m. Now note that (2.22) is a singular perturbation [BO99, Section 7.2] and as $\delta_m \to 0$ all possible solutions for z_m go to infinity except for those which satisfy the left-hand side being zero. As such, the only possibility for any z_m being a true root of the characteristic polynomial is that they tend to one of the limiting roots

$$z = \pm\sqrt{\tfrac{1}{2}b^2 - a^2}. \tag{2.23}$$

Note that for such z_m to exist we required the condition $|c_m| > 1$, and so $|a^2| > |z_m^2|$, to hold for arbitrarily large m. For this to hold in the limit we require $|a^2| > |\frac{1}{2}b^2 - a^2|$ and so the limiting roots in (2.23) may only exist when this condition is met.

We have now seen that, aside from the special case yielding the potential for limiting roots (2.23), all remaining z_m correspond to q_m values which tend to the unit circle. To complete the proof we now translate this result using the q-discriminant condition (2.9). Since q_m tends to the unit circle, the corresponding z_m must tend to the limiting curve defined by (2.9) where $q = e^{i\phi}$ for some $\phi \in [-\pi, \pi]$. This limiting curve in the complex plane is given parametrically as

$$\begin{aligned}
&\frac{B(z)^2}{A(z)} = e^{i\phi} + e^{-i\phi} + 2, && \phi \in [-\pi, \pi] \\
\iff &\frac{(-z^2 + b^2 - a^2)^2}{a^2 z^2} = 4\cos^2\left(\frac{\phi}{2}\right), && \phi \in [-\pi, \pi] \\
\iff & z^2 - b^2 + a^2 = \pm 2az\cos\left(\frac{\phi}{2}\right), && \phi \in [-\pi, \pi] \\
\iff & z^2 - 2a\cos(\theta)z - b^2 + a^2 = 0, && \theta \in [-\pi, \pi] \\
\iff & z = a\cos(\theta) \pm \sqrt{b^2 - a^2\sin^2(\theta)}, && \theta \in [-\pi, \pi].
\end{aligned}$$

Thus, as roots z_m of $p_m(z)$ are eigenvalues λ of $\mathcal{T} \in \mathbb{C}^{2m\times 2m}$, we deduce that the limiting spectrum of $\mathcal{T}$ must lie on the curve defined by (2.10) as $m \to \infty$, except perhaps for the eigenvalues in (2.11) which can only occur if $|a^2| > |\frac{1}{2}b^2 - a^2|$. ■

We note that, while the so-called Szegő formula does not apply in our non-Hermitian case, we have just proven that the limiting eigenvalues of $\mathcal{T}$, except perhaps two, lie on the equivalent curve defined by eigenvalues of the (block) symbol of $\mathcal{T}$, which is precisely that defined in (2.10).

2.4 The one-dimensional problem

We now turn our attention to analysing the one-level method. In this section we study the parallel Schwarz iterative method for the one-dimensional Maxwell's equations with Robin boundary conditions defined on the domain $\Omega = (a_1, b_N)$:

$$\text{(2.24)} \qquad \begin{cases} \mathcal{L}u := -\partial_{xx}u + (ik\tilde{\sigma} - k^2)u = 0, & x \in (a_1, b_N), \\ \mathcal{B}_l u := -\partial_x u + \alpha u = g_1, & x = a_1, \\ \mathcal{B}_r u := \partial_x u + \alpha u = g_2, & x = b_N, \end{cases}$$

where u represents the complex amplitude of the electric field, k is the wave number, and $\tilde{\sigma} = \sigma Z$ with σ being the conductivity of the medium and Z its impedance. Here α is the impedance parameter which is chosen such that the local problems are well-posed and is classically set to ik, in which case the problem corresponds to a *"one-dimensional wave-guide"* and the incoming wave or excitation can be represented by g_1, for example, with g_2 being set to 0. Note that, when $\alpha = ik$, the problem is well-posed even if $\tilde{\sigma} = 0$ but in the following we will assume that $\tilde{\sigma} > 0$. In order to simplify

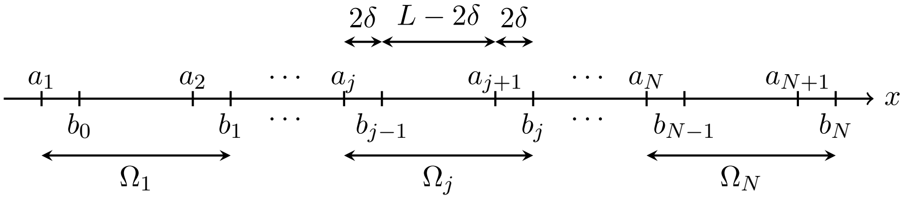

Figure 2.1: Overlapping decomposition of the one-dimensional domain into N subdomains.

notation we will omit the tilde symbol for σ. We remark that (2.24) can also be seen as an absorptive Helmholtz equation where the absorption term $ik\sigma$ comes from the physics of the problem.

Let us also consider two sets of points $\{a_j\}_{j=1,..,N+1}$ and $\{b_j\}_{j=0,..,N}$ defining the overlapping decomposition $\Omega = \cup_{j=1}^{N}\Omega_j$ such that $\Omega_j = (a_j, b_j)$, as illustrated in Figure 2.1 (and considered in [CCGV18]), where

$$b_j - a_j = L + 2\delta, \quad b_{j-1} - a_j = 2\delta, \quad a_{j+1} - a_j = b_{j+1} - b_j = L, \quad \delta > 0. \tag{2.25}$$

Note that the length of each subdomain is fixed and equal to $L + 2\delta$ while the overlap is always 2δ. This means that the family of problems we will consider solving consists of a growing chain of fixed-size subdomains, as in [CCGV18], rather than solving on a fixed problem domain with shrinking subdomain size.

We consider solving (2.24) by a Schwarz iterative algorithm with Robin transmission conditions and denote by u_j^n the approximation to the solution in subdomain j at iteration n, starting from an initial guess u_j^0. We compute u_j^n from the previous values u_j^{n-1} by solving the following local boundary value problem

$$\begin{cases} \mathcal{L}u_j^n = 0, & x \in \Omega_j, \\ \mathcal{B}_l u_j^n = \mathcal{B}_l u_{j-1}^{n-1}, & x = a_j, \\ \mathcal{B}_r u_j^n = \mathcal{B}_r u_{j+1}^{n-1}, & x = b_j, \end{cases} \tag{2.26a}$$

in the case $2 \leq j \leq N$ while for the first ($j = 1$) and last ($j = N$) subdomain we have

$$\begin{cases} \mathcal{L}u_1^n = 0, & x \in \Omega_1, \\ \mathcal{B}_l u_1^n = g_1, & x = a_1, \\ \mathcal{B}_r u_1^n = \mathcal{B}_r u_2^{n-1}, & x = b_1, \end{cases} \quad \begin{cases} \mathcal{L}u_N^n = 0, & x \in \Omega_N, \\ \mathcal{B}_l u_N^n = \mathcal{B}_l u_{N-1}^{n-1}, & x = a_N, \\ \mathcal{B}_r u_N^n = g_2, & x = b_N. \end{cases} \tag{2.26b}$$

In the following we wish to analyse the convergence of the iterative method that is defined by (2.26). We observe this iteration to be a parallel Schwarz method with Robin transmission conditions, a label which we shall adopt in this work. In particular, we will be interested in the convergence properties for a growing number of subdomains N and the absorptive problem, i.e., $\sigma > 0$.[1] This means that we will consider asymptotic bounds for large N and make use of the theory presented in Section 2.3.

In order to do this we define the local errors in each subdomain j at iteration n as $e_j^n = u|_{\Omega_j} - u_j^n$. They verify the boundary value problems (2.26a) for the interior subdomains and the homogeneous analogues of (2.26b) for the first and last subdomains (i.e., (2.26b) but with boundary conditions $g_1 = 0$ and $g_2 = 0$). The convergence study will be done in two steps: first we prove that the Schwarz iteration matrix is a block Toeplitz matrix and then that its spectral radius remains bounded below and away from one in the limit of large N. As mentioned before, we build on the formalism of iteration matrices acting on interface data introduced in [CCGV18]; here this will be Robin data.

Lemma 2.2 (Block Toeplitz iteration matrix)**.** *If $e_j^n = u|_{\Omega_j} - u_j^n$ is the local error in each subdomain j at iteration n and*

$$\mathcal{R}^n := \left[\mathcal{R}_+^n(b_1),\, \mathcal{R}_-^n(a_2),\, \mathcal{R}_+^n(b_2), \ldots, \mathcal{R}_-^n(a_{N-1}),\, \mathcal{R}_+^n(b_{N-1}),\, \mathcal{R}_-^n(a_N)\right]^T,$$

where

$$\mathcal{R}_-^n(a_j) := \mathcal{B}_l e_{j-1}^n(a_j), \qquad \mathcal{R}_+^n(b_j) := \mathcal{B}_r e_{j+1}^n(b_j), \tag{2.27}$$

is the Robin interface data, then

$$\mathcal{R}^n = \mathcal{T}_{1d}\mathcal{R}^{n-1},$$

where $\mathcal{T}_{1d}$ is a block Toeplitz matrix of the form (2.1) *with the complex coefficients a and b being given by*

$$a = \frac{(\zeta+\alpha)^2 e^{2\zeta\delta} - (\zeta-\alpha)^2 e^{-2\zeta\delta}}{(\zeta+\alpha)^2 e^{\zeta(2\delta+L)} - (\zeta-\alpha)^2 e^{-\zeta(2\delta+L)}}, \tag{2.28a}$$

$$b = -\frac{(\zeta^2-\alpha^2)(e^{\zeta L} - e^{-\zeta L})}{(\zeta+\alpha)^2 e^{\zeta(2\delta+L)} - (\zeta-\alpha)^2 e^{-\zeta(2\delta+L)}}, \tag{2.28b}$$

[1] When $\sigma = 0$, impedance transmission conditions are also transparent conditions, with the resulting iteration matrix being nilpotent. Therefore, the algorithm will converge in a number of iterations equal to the number of subdomains in this case and in the sense of the definition from Section 1.2 it is not scalable.

where $\zeta = \sqrt{ik\sigma - k^2}$.

Proof. We first see that the solution to $\mathcal{L}e_j^n = 0$ is given by

$$e_j^n(x) = \alpha_j^n e^{-\zeta x} + \beta_j^n e^{\zeta x}, \qquad \zeta = \sqrt{ik\sigma - k^2}. \tag{2.29}$$

Note that we choose the principle branch of the square root here so that ζ always has positive real and imaginary parts. Now the interface iterations at $x = a_j$ and $x = b_j$ from (2.26) can be written in terms of the error as

$$\begin{bmatrix} \mathcal{B}_l e_j^n(a_j) \\ \mathcal{B}_r e_j^n(b_j) \end{bmatrix} = \begin{bmatrix} \mathcal{B}_l e_{j-1}^{n-1}(a_j) \\ \mathcal{B}_r e_{j+1}^{n-1}(b_j) \end{bmatrix}. \tag{2.30}$$

By introducing (2.29) into the left-hand side of (2.30) and by using the notation from (2.27) we obtain

$$\begin{bmatrix} (\zeta+\alpha)e^{-\zeta a_j} & -(\zeta-\alpha)e^{\zeta a_j} \\ -(\zeta-\alpha)e^{-\zeta b_j} & (\zeta+\alpha)e^{\zeta b_j} \end{bmatrix} \begin{bmatrix} \alpha_j^n \\ \beta_j^n \end{bmatrix} = \begin{bmatrix} \mathcal{R}_-^{n-1}(a_j) \\ \mathcal{R}_+^{n-1}(b_j) \end{bmatrix},$$

which we can solve for the unknowns α_j^n and β_j^n to give

$$\begin{bmatrix} \alpha_j^n \\ \beta_j^n \end{bmatrix} = \frac{1}{D_j} \begin{bmatrix} (\zeta+\alpha)e^{\zeta b_j} & (\zeta-\alpha)e^{\zeta a_j} \\ (\zeta-\alpha)e^{-\zeta b_j} & (\zeta+\alpha)e^{-\zeta a_j} \end{bmatrix} \begin{bmatrix} \mathcal{R}_-^{n-1}(a_j) \\ \mathcal{R}_+^{n-1}(b_j) \end{bmatrix}, \tag{2.31}$$

where $D_j = (\zeta+\alpha)^2 e^{\zeta(b_j-a_j)} - (\zeta-\alpha)^2 e^{\zeta(a_j-b_j)}$. Note that, since $b_j - a_j = L + 2\delta$, then D_j is actually independent of j and thus we simply denote it by D. The algorithm is based on Robin transmission conditions, hence the quantities of interest which are transmitted at the interfaces between subdomains are the Robin data (2.27). Therefore, we need to compute the current interface values $\mathcal{R}_-^n(a_j)$ and $\mathcal{R}_+^n(b_j)$ by replacing the coefficients from (2.31) into (2.29) and then applying the formulae in (2.27), giving

$$\begin{aligned} \mathcal{R}_-^n(a_j) &= \mathcal{B}_l e_{j-1}^n(a_j) = (\zeta+\alpha)\alpha_{j-1}^n e^{-\zeta a_j} - (\zeta-\alpha)\beta_{j-1}^n e^{\zeta a_j} \\ &= \frac{1}{D}\Big[\big((\zeta+\alpha)^2 e^{\zeta(b_{j-1}-a_j)} - (\zeta-\alpha)^2 e^{\zeta(a_j-b_{j-1})}\big)\mathcal{R}_-^{n-1}(a_{j-1}) \\ &\quad + (\zeta^2-\alpha^2)\big(e^{\zeta(a_{j-1}-a_j)} - e^{\zeta(a_j-a_{j-1})}\big)\mathcal{R}_+^{n-1}(b_{j-1})\Big], \end{aligned} \tag{2.32a}$$

$$\begin{aligned} \mathcal{R}_+^n(b_j) &= \mathcal{B}_r e_{j+1}^n(b_j) = -(\zeta-\alpha)\alpha_{j+1}^n e^{-\zeta b_j} + (\zeta+\alpha)\beta_{j+1}^n e^{\zeta b_j} \\ &= \frac{1}{D}\Big[(\zeta^2-\alpha^2)\big(e^{\zeta(b_j-b_{j+1})} - e^{\zeta(b_{j+1}-b_j)}\big)\mathcal{R}_-^{n-1}(a_{j+1}) \\ &\quad + \big((\zeta+\alpha)^2 e^{\zeta(b_j-a_{j+1})} - (\zeta-\alpha)^2 e^{\zeta(a_{j+1}-b_j)}\big)\mathcal{R}_+^{n-1}(b_{j+1})\Big]. \end{aligned} \tag{2.32b}$$

The iteration of interface values (2.32) can be summarised as follows:

$$\begin{aligned}
&\begin{bmatrix} \mathcal{R}_-^n(a_j) \\ \mathcal{R}_+^n(b_j) \end{bmatrix} = T_1 \begin{bmatrix} \mathcal{R}_-^{n-1}(a_{j-1}) \\ \mathcal{R}_+^{n-1}(b_{j-1}) \end{bmatrix} + T_2 \begin{bmatrix} \mathcal{R}_-^{n-1}(a_{j+1}) \\ \mathcal{R}_+^{n-1}(b_{j+1}) \end{bmatrix}, \\
&T_1 = \begin{bmatrix} a & b \\ 0 & 0 \end{bmatrix}, \; T_2 = \begin{bmatrix} 0 & 0 \\ b & a \end{bmatrix},
\end{aligned} \tag{2.33a}$$

where a and b are given by (2.28). Note that since the homogeneous counterparts of the boundary conditions from (2.26b) translate into $\mathcal{R}_-^n(a_1) = 0$ and $\mathcal{R}_+^n(b_N) = 0$ for all n, we can remove these terms. As such, the iterates for $j \in \{1, 2, N-1, N\}$ are prescribed slightly differently as

$$\begin{aligned}
\begin{bmatrix} 0 \\ \mathcal{R}_+^n(b_1) \end{bmatrix} &= T_2 \begin{bmatrix} \mathcal{R}_-^{n-1}(a_2) \\ \mathcal{R}_+^{n-1}(b_2) \end{bmatrix}, \\
\begin{bmatrix} \mathcal{R}_-^n(a_2) \\ \mathcal{R}_+^n(b_2) \end{bmatrix} &= T_1 \begin{bmatrix} 0 \\ \mathcal{R}_+^{n-1}(b_1) \end{bmatrix} + T_2 \begin{bmatrix} \mathcal{R}_-^{n-1}(a_3) \\ \mathcal{R}_+^{n-1}(b_3) \end{bmatrix}, \\
\begin{bmatrix} \mathcal{R}_-^n(a_{N-1}) \\ \mathcal{R}_+^n(b_{N-1}) \end{bmatrix} &= T_1 \begin{bmatrix} \mathcal{R}_-^{n-1}(a_{N-2}) \\ \mathcal{R}_+^{n-1}(b_{N-2}) \end{bmatrix} + T_2 \begin{bmatrix} \mathcal{R}_-^{n-1}(a_N) \\ 0 \end{bmatrix}, \\
\begin{bmatrix} \mathcal{R}_-^n(a_N) \\ 0 \end{bmatrix} &= T_1 \begin{bmatrix} \mathcal{R}_-^{n-1}(a_{N-1}) \\ \mathcal{R}_+^{n-1}(b_{N-1}) \end{bmatrix}.
\end{aligned} \tag{2.33b}$$

With the notation from (2.27), global iteration over interface data belonging to all subdomains becomes $\mathcal{R}^n = \mathcal{T}_{1d}\mathcal{R}^{n-1}$ where

$$\mathcal{T}_{1d} = \begin{bmatrix}
0 & \widehat{T}_2 & & & & & \\
\widetilde{T}_1 & 0_{2\times2} & T_2 & & & & \\
& \ddots & \ddots & \ddots & & & \\
& & T_1 & 0_{2\times2} & T_2 & & \\
& & & \ddots & \ddots & \ddots & \\
& & & & T_1 & 0_{2\times2} & \widetilde{T}_2 \\
& & & & & \widehat{T}_1 & 0
\end{bmatrix} \tag{2.34}$$

with $\widetilde{T}_1 = \begin{bmatrix} b & 0 \end{bmatrix}^T$, $\widetilde{T}_2 = \begin{bmatrix} 0 & b \end{bmatrix}^T$, $\widehat{T}_1 = \begin{bmatrix} a & b \end{bmatrix}$, $\widehat{T}_2 = \begin{bmatrix} b & a \end{bmatrix}$. We conclude from this that the parallel Schwarz algorithm is given by a stationary iteration with iteration matrix $\mathcal{T}_{1d}$ defined by (2.34) and, therefore, convergence is determined by the spectral radius $\rho(\mathcal{T}_{1d})$. We also notice that $\mathcal{T}_{1d}$ is a block Toeplitz matrix precisely of the form in (2.1) where the complex coefficients a and b are given by (2.28) and, as such, the limiting spectral analysis in Section 2.3 will apply. ■

Before proving convergence of the parallel Schwarz algorithm, we first utilise the key result of Theorem 2.1, on the limiting spectrum of $\mathcal{T}_{1d}$, to provide a useful intermediary lemma. This intermediary result will also aid our analysis in the two-dimensional case to follow in Section 2.5.

Lemma 2.3 (Limiting spectral radius and sufficient conditions for convergence)**.** *The following relation holds:*

$$\max_{\theta\in[-\pi,\pi]}\left|a\cos(\theta)\pm\sqrt{b^2-a^2\sin^2(\theta)}\right| = \max\{|a+b|,|a-b|\},$$

and thus the convergence factor $R_{1d} := \lim_{N\to\infty}\rho(\mathcal{T}_{1d})$ *of the Schwarz algorithm as the number of subdomains tends to infinity verifies*

$$R_{1d} \leq \begin{cases} \max\{|a+b|,|a-b|\} & \text{if } \left|a^2-\frac{1}{2}b^2\right|^{1/2} \geq |a|, \\ \max\{|a+b|,|a-b|,|a|\} & \text{if } \left|a^2-\frac{1}{2}b^2\right|^{1/2} < |a|. \end{cases} \tag{2.35}$$

Further, consider the change of variables

$$z = 2\delta\zeta, \qquad l = \frac{L}{2\delta}, \qquad \gamma = 2\delta\alpha, \qquad v = \frac{z-\gamma}{z+\gamma}, \tag{2.36}$$

and let $z := x + iy$ *for* $x, y \in \mathbb{R}^+$. *Then the condition* $g_\pm(z;\delta,l) > 0$, *where*

$$g_\pm(z;\delta,l) = (e^{2lx}-1)(e^{2x}-|v|^2) \pm 4\sin(ly)(\Im v\cos y - \Re v\sin y)e^{x(l+1)}, \tag{2.37}$$

will ensure the desired convergence bound $\max\{|a+b|,|a-b|\} < 1$. *Similarly, the condition* $g(z;\delta,l) > 0$, *where*

$$\begin{aligned} g(z;\delta,l) = {} & (e^{2lx}-1)(e^{2x(l+2)}-|v|^4) + 4\sin(ly) \\ & \cdot \left[((\Re v)^2-(\Im v)^2)\sin(y(l+2)) - 2\Re v\Im v\cos(y(l+2))\right]e^{2x(l+1)}, \end{aligned} \tag{2.38}$$

will ensure that $|a| < 1$.

Proof. Since $\mathcal{T}_{1d}$ is of the form $\mathcal{T}$ in (2.1), Theorem 2.1 provides its limiting spectrum and thus allows us to bound R_{1d} by the largest eigenvalue in magnitude. We first bound $\lambda_\pm(\theta) = a\cos(\theta) \pm \sqrt{b^2 - a^2\sin^2(\theta)}$. It is straightforward to see that these values are the eigenvalues of the matrix

$$T = \begin{pmatrix} a\cos(\theta) & b - a\sin(\theta) \\ b + a\sin(\theta) & a\cos(\theta) \end{pmatrix}.$$

A simple computation shows that the matrix

$$T^*T = \begin{pmatrix} |a|^2 + |b|^2 + (a\bar{b} + \bar{a}b)\sin(\theta) & (a\bar{b} + \bar{a}b)\cos(\theta) \\ (a\bar{b} + \bar{a}b)\cos(\theta) & |a|^2 + |b|^2 - (a\bar{b} + \bar{a}b)\sin(\theta) \end{pmatrix}$$

has the eigenvalues $\mu_\pm = |a \pm b|^2$. We can now conclude that

$$|\lambda_\pm(\theta)| \leq \|T\|_2 = \sqrt{\|T^*T\|_2} = \sqrt{\max\{\mu_+, \mu_-\}} = \max\{|a+b|, |a-b|\},$$

and furthermore note that this bound is attained when $\theta = 0$. Additionally, Theorem 2.1 states that eigenvalues $\lambda = \pm(\frac{1}{2}b^2 - a^2)^{1/2}$ may belong to the limiting spectrum but only if they have magnitude strictly less than $|a|$. Together, these two cases yield (2.35).

Let us consider now the complex-valued functions $F_\pm\colon \mathbb{C} \to \mathbb{C}$

$$F_\pm(z) = \frac{(z+\gamma)^2 e^{z} - (z-\gamma)^2 e^{-z}}{(z+\gamma)^2 e^{(l+1)z} - (z-\gamma)^2 e^{-(l+1)z}} \pm \frac{(z^2-\gamma^2)(e^{lz} - e^{-lz})}{(z+\gamma)^2 e^{(l+1)z} - (z-\gamma)^2 e^{-(l+1)z}}.$$

It is easy to see that $a \mp b = F_\pm(z)$ when z, l and γ are as defined in (2.36). Similarly, we define the function $G\colon \mathbb{C} \to \mathbb{C}$ to be the first term in $F_\pm(z)$ so that $a = G(z)$. Let us simplify in the first instance the expression of $|F_\pm(z)|$ without using any assumption on $z := x + iy$. For this we consider the transformation v along with its polar form

$$\text{(2.39)} \qquad v := \frac{z-\gamma}{z+\gamma}, \qquad v = w(\cos(\varphi) + i\sin(\varphi)), \qquad w = |v|.$$

After some lengthy but elementary calculations we find that

$$\text{(2.40a)} \qquad |F_\pm(z)|^2 = 1 - \frac{(e^{x(l+1)} - w)^2 + 2w(1 \mp \cos((l+1)y - \varphi))e^{x(l+1)}}{(e^{2x(l+1)} - w^2)^2 + 4w^2\sin^2((l+1)y - \varphi)e^{2x(l+1)}} g_\pm(z;\delta,l)$$

$$\text{(2.40b)} \qquad g_\pm(z;\delta,l) = (e^{2lx} - 1)(e^{2x} - w^2) \pm 4w\sin(ly)\sin(\varphi - y)e^{x(l+1)}.$$

We observe that the fraction in (2.40a) is positive, since the individual terms involved are, and thus $\max\{|a-b|, |a+b|\} < 1 \Leftrightarrow |F_\pm(z)|^2 < 1 \Leftrightarrow g_\pm(z;\delta,l) > 0$. We can now rewrite $g_\pm(z;\delta,l)$ in (2.40b) using (2.39) and convert v to Cartesian form to obtain the required expression in (2.37). A near identical argument can be used to derive conditions for $|G(z)|^2 < 1$ and results in the criterion that $g(z; \delta, l) > 0$, where $g(z; \delta, l)$ is defined by (2.38). Thus the required conclusions follow. ■

We are now ready to state our main convergence result for the one-dimensional problem in the case when $\alpha = ik$, namely that of classical impedance conditions.

Theorem 2.2 (Convergence of the Schwarz algorithm in 1D)**.** *If $\alpha = ik$ (the case of classical impedance conditions), then for all $k > 0$, $\sigma > 0$, $\delta > 0$ and $L > 0$ we have that $R_{1d} < 1$. Therefore the convergence will ultimately be independent of the number of subdomains (we say that the Schwarz method will scale).*

Proof. By Lemma 2.3 we see that it is enough to study the sign of $g_{\pm}(z; \delta, l)$ and of $g(z; \delta, l)$. We can see that if $\alpha = ik$ and $\kappa = 2\delta k$ then for $z := x + iy$ (2.39) becomes

$$\Re v = \frac{-\kappa^2 + x^2 + y^2}{(\kappa + y)^2 + x^2}, \qquad \Im v = \frac{-2\kappa x}{(\kappa + y)^2 + x^2}, \qquad |v|^2 = \frac{(\kappa - y)^2 + x^2}{(\kappa + y)^2 + x^2} < 1,$$

the final inequality holding since $\kappa > 0$ and $y > 0$. We emphasise that x and y are the real and imaginary parts of $z = 2\delta\zeta$ and so are positive by the nature of ζ in (2.29). Now we can further simplify (2.37) using these expressions for v to obtain

$$g_{\pm}(z; \delta, l) = \frac{4e^{x(l+1)}}{(\kappa + y)^2 + x^2} \tilde{g}_{\pm}(z; \delta, l) \tag{2.41a}$$

$$\begin{aligned} \tilde{g}_{\pm}(z; \delta, l) = {} & [(\kappa^2 + x^2 + y^2) \sinh(x) + 2\kappa y \cosh(x)] \sinh(lx) \\ & \pm [(\kappa^2 - x^2 - y^2) \sin(y) - 2\kappa x \cos(y)] \sin(ly). \end{aligned} \tag{2.41b}$$

Proving positivity of $g_{\pm}(z; \delta, l)$ is then equivalent to positivity of $\tilde{g}_{\pm}(z; \delta, l)$. To proceed we relate x and y by considering the real part of $z^2 = (x + iy)^2 = 2i\kappa\delta\sigma - \kappa^2$ which yields $y^2 = \kappa^2 + x^2$. Let us now eliminate y using this identity to obtain

$$\begin{aligned} \tilde{g}_{\pm}(z; \delta, l) = {} & 2 \left[(\kappa^2 + x^2) \sinh(x) + \kappa\sqrt{\kappa^2 + x^2} \cosh(x)\right] \sinh(lx) \\ & \mp 2 \left[x^2 \sin(\sqrt{\kappa^2 + x^2}) + \kappa x \cos(\sqrt{\kappa^2 + x^2})\right] \sin(l\sqrt{\kappa^2 + x^2}). \end{aligned}$$

To show that this is positive we want to lower bound the hyperbolic term in the first line (which is positive) while making the trigonometric term in the second line as large as possible in magnitude and negative. To do this we make use of some elementary bounds which hold for $t > 0$:

$$|\sin(t)| < t < \sinh(t), \qquad |\cos(t)| \le 1 < \cosh(t). \tag{2.42}$$

We can now derive the positivity bound on $\tilde{g}_{\pm}(z;\delta,l)$, noting that $x>0$, as follows

$$\tilde{g}_{\pm}(z;\delta,l) > 2\left[(\kappa^2+x^2)x+\kappa\sqrt{\kappa^2+x^2}\right]lx - 2\left[x^2\sqrt{\kappa^2+x^2}+\kappa x\right]l\sqrt{\kappa^2+x^2} = 0.$$

Turning to $g(z;\delta,l)$, we can follow a similar process, simplifying (2.38) to find that

$$g(z;\delta,l) = \frac{4e^{2x(l+1)}}{((\kappa+y)^2+x^2)^2}\tilde{g}(z;\delta,l) \tag{2.43a}$$

$$\begin{aligned} \tilde{g}(z;\delta,l) = {} & \big[((\kappa^2+x^2+y^2)^2+4\kappa^2y^2)\sinh(x(l+2)) \\ & \quad + 4\kappa y(\kappa^2+x^2+y^2)\cosh(x(l+2))\big]\sinh(lx) \\ + {} & \big[((-\kappa^2+x^2+y^2)^2-4\kappa^2x^2)\sin(y(l+2)) \\ & \quad + 4\kappa x(-\kappa^2+x^2+y^2)\cos(y(l+2))\big]\sin(ly). \end{aligned} \tag{2.43b}$$

Using the identity $y^2=\kappa^2+x^2$ along with the elementary bounds (2.42) we obtain

$$\begin{aligned} \tilde{g}(z;\delta,l) &= 4\left[y^2(y^2+\kappa^2)\sinh(x(l+2))+2\kappa y^3\cosh(x(l+2))\right]\sinh(lx) \\ &\quad + 4\left[x^2(x^2-\kappa^2)\sin(y(l+2))+2\kappa x^3\cos(y(l+2))\right]\sin(ly) \\ &> 4\left[y^2(y^2+\kappa^2)x(l+2)+2\kappa y^3\right]lx - 4\left[x^2(x^2+\kappa^2)y(l+2)+2\kappa x^3\right]ly \\ &= 4l(l+2)x^2y^2\kappa^2+8lxy\kappa^3 \\ &> 0. \end{aligned}$$

Thus, we conclude that for any choice of parameters the required sufficient criteria from Lemma 2.3 on $g_{\pm}(z;\delta,l)$ and $g(z;\delta,l)$ hold and hence $R_{1d}<1$. Therefore the algorithm will always converge in a number of iterations ultimately independent of the number of subdomains. Nonetheless, note that as any problem parameter shrinks to zero the bounds become tight and so R_{1d} can be made arbitrarily close to one. ■

In order to verify this result, we compute numerically (using `MATLAB`) the spectrum of the iteration matrix and compare it with the theoretical limit for different values of σ. We choose here $k=30$, $L=1$ and $\delta=L/10$. From Figures 2.2 and 2.3 we notice that the spectrum of the iteration matrix tends to the theoretical limit when the number of subdomains becomes large and the algorithm remains convergent. Additionally, when σ grows the behaviour of the algorithm improves, which is consistent with the fact that when the absorption in the equations is important (solutions are less oscillatory) or the overlap is large (more information is exchanged) the systems are easier to solve. We also remark an empirical observation that the convergence factor monotonically increases towards the limit given in Lemma 2.3, thus indicating that the algorithm will

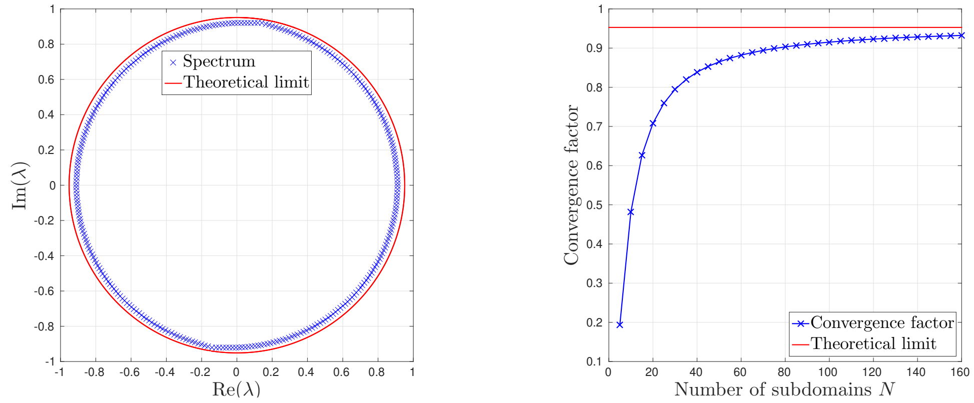

Figure 2.2: The spectrum of the iteration matrix $\mathcal{T}_{1d}$ for $N = 160$ (left) and the convergence factor of the Schwarz algorithm for varying number of subdomains N (right) when $\sigma = 0.1$.

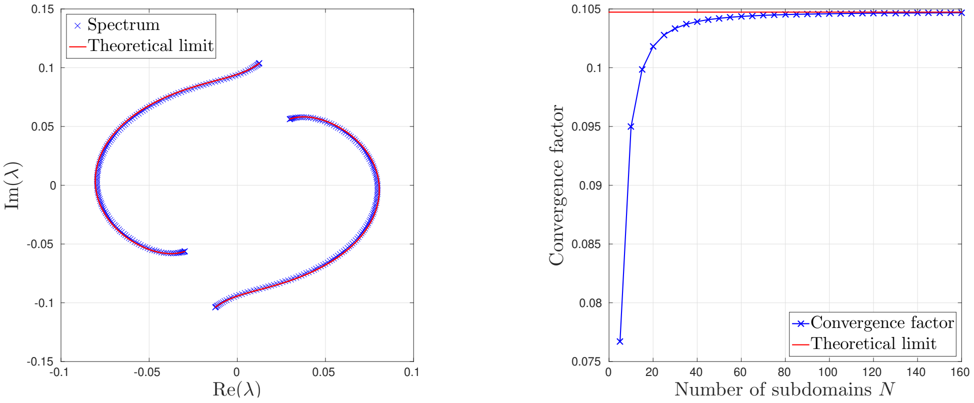

Figure 2.3: The spectrum of the iteration matrix $\mathcal{T}_{1d}$ for $N = 160$ (left) and the convergence factor of the Schwarz algorithm for varying number of subdomains N (right) when $\sigma = 5$.

always converge for any N.

Before moving onto the two-dimensional case, we first derive a simple corollary showing how our results can be extended in the direction of k-independence of the one-level method within certain scenarios. In this case we consider the parameters L and δ being dependent upon the wave number k.

Corollary 2.1 (A case of k-independent convergence)**.** *Suppose $\alpha = ik$ (the case of classical impedance conditions) and that $\sigma = \sigma_0 k$ for some constant σ_0. Consider a k-dependent domain decomposition given by $L = L_0 k^{-1}$ and $\delta = \delta_0 k^{-1}$, that is the subdomain size and overlap shrink inversely proportional to the wave number. Then the convergence of the corresponding Schwarz method is independent of the wave number*

k. Thus the approach is k-robust and convergence will ultimately be independent of the number of subdomains.

Proof. Inserting the relevant k-dependent parameters α, σ, δ and L into (2.28) we find that both coefficients a and b, and thus the iteration matrix $\mathcal{T}_{1d}$, are independent of k. Combining this result with Theorem 2.2 shows that the convergence of the corresponding Schwarz method is both k-independent and, ultimately, independent of the number of subdomains. ■

We note that k-robustness of the one-level method was proved, under certain conditions, in [GSZ20] using rigorous GMRES bounds. Here, our theory is able to directly evidence k-robustness of the algorithm at the continuous level, independent of the discretisation, in a simple one-dimensional scenario. We can also consider the case where k is linked to N such that we now solve on a fixed domain a family of problems with increasing wave number using an increasing number of subdomains, here our theory shows the method to be k-robust and weakly scalable.

Theorem 2.2 shows that weak scalability is achieved in the one-dimensional case as soon as the parameter σ is strictly positive. Intuitively this makes sense since, in the one-dimensional case for $\sigma = 0$, the iteration matrix becomes nilpotent and therefore a classical iterative method will need a number of iterations equal to the number of subdomains to converge. According to the definition from Section 1.2 it is not scalable. The complex shift brought about by σ will aid convergence by damping the waves and, when this damping parameter is large enough, robustness with respect to the wave number can also be achieved as seen in Corollary 2.1.

2.5 The two-dimensional problem

Consider the domain $\Omega = (a_1, b_N) \times (0, \hat{L})$ on which we wish to solve the two-dimensional problem and a decomposition into N overlapping subdomains defined by $\Omega_j = (a_j, b_j) \times (0, \hat{L})$, where a_j and b_j are as given in (2.25). We will analyse the case of the Helmholtz and then Maxwell's equations.

2.5.1 The Helmholtz equation

The definition of the parallel Schwarz method for the iterates u_j^n in the case of the two-dimensional Helmholtz problem is

$$(2.44)\qquad \begin{cases} (ik\sigma - k^2)u_j^n - (\partial_{xx} + \partial_{yy})u_j^n = f, & (x,y) \in (a_j, b_j) \times (0, \hat{L}), \\ \mathcal{B}_l u_j^n(a_j, y) = \mathcal{B}_l u_{j-1}^{n-1}(a_j, y), & y \in (0, \hat{L}), \\ \mathcal{B}_r u_j^n(b_j, y) = \mathcal{B}_r u_{j+1}^{n-1}(b_j, y), & y \in (0, \hat{L}), \\ u_j^n(x, y) = 0, & x \in (a_j, b_j),\ y \in \{0, \hat{L}\}, \end{cases}$$

where the boundary operators $\mathcal{B}_l$ and $\mathcal{B}_r$ are as defined in (2.24). For the first and the last subdomain ($j = 1$ and $j = N$) we impose $\mathcal{B}_l u_1^n = g_1$ when $x = a_1$ and $\mathcal{B}_r u_N^n = g_2$ when $x = b_N$. We consider here the case of impedance conditions, i.e. $\alpha = ik$. Note that this configuration corresponds to a *"two-dimensional wave-guide"* problem. By linearity, it follows that the local errors $e_j^n = u|_{\Omega_j} - u_j^n$ satisfy the homogeneous analogue of (2.44). To proceed, we make use of the Fourier sine expansion of e_j^n, as the solution verifies Dirichlet boundary conditions on the top and bottom of each rectangular subdomain:

$$(2.45)\qquad e_j^n(x,y) = \sum_{m=1}^{\infty} v_j^n(x, \tilde{k}) \sin(\tilde{k}y), \qquad \tilde{k} = \frac{m\pi}{\hat{L}},\ m \in \mathbb{N}.$$

Inserting this expression into the homogeneous counterpart of (2.44) we find that, for each Fourier number $\tilde{k}$, $v_j^n(x, \tilde{k})$ verifies the one-dimensional problem

$$(2.46)\qquad \begin{cases} (ik\sigma + \tilde{k}^2 - k^2)v_j^n - \partial_{xx} v_j^n = 0, & x \in (a_j, b_j), \\ \mathcal{B}_l v_j^n(x, \tilde{k}) = \mathcal{B}_l v_{j-1}^{n-1}(x, \tilde{k}), & x = a_j, \\ \mathcal{B}_r v_j^n(x, \tilde{k}) = \mathcal{B}_r v_{j+1}^{n-1}(x, \tilde{k}), & x = b_j, \end{cases}$$

which is of exactly the same type as (2.26) where $ik\sigma - k^2$ is replaced by $ik\sigma + \tilde{k}^2 - k^2$. Therefore, the result from Lemma 2.2 applies here if we replace α with ik and ζ with

$$(2.47)\qquad \zeta(\tilde{k}) = \sqrt{ik\sigma + \tilde{k}^2 - k^2}.$$

Let us denote the resulting iteration matrix, which propagates information for each Fourier number $\tilde{k}$ independently, by $\mathcal{T}_{1d}^{\mathrm{H}}(\tilde{k})$ and let $R_{1d}^{\mathrm{H}}(\tilde{k}) := \lim_{N\to\infty} \rho(\mathcal{T}_{1d}^{\mathrm{H}}(\tilde{k}))$ with $R_{2d}^{\mathrm{H}} = \sup_{\tilde{k}} R_{1d}^{\mathrm{H}}(\tilde{k})$. We can now state our main convergence result for the two-dimensional Helmholtz problem.

Theorem 2.3 (Convergence of the Schwarz algorithm for Helmholtz in 2D)**.** *If* $\alpha = ik$

(the case of classical impedance conditions), then for all $k > 0$, $\sigma > 0$, $\delta > 0$ and $L > 0$ we have that $R^{\mathrm{H}}_{1d}(\tilde{k}) < 1$ for all evanescent modes $\tilde{k} > k$. Furthermore, under the assumption that between them σ, δ and L are sufficiently large we have that $R^{\mathrm{H}}_{2d} < 1$. In particular, this is true when $\sigma \geq k$ for all $\delta > 0$ and $L > 0$. Therefore the convergence will ultimately be independent of the number of subdomains (we say that the Schwarz method will be weakly scalable according to definitions in Section 1.2).

Proof. By Lemma 2.3 we see that it is enough to study the sign of $g_{\pm}(z;\delta,l)$ and $g(z;\delta,l)$. To assist, we use the scaled notation $\kappa = 2\delta k$, $\tilde{\kappa} = 2\delta\tilde{k}$ and $s = 2\delta\sigma$ akin to (2.36). Now $g_{\pm}(z;\delta,l)$ can be formally simplified identically to (2.41), however, in this case with ζ as in (2.47) the real part of z^2 gives the identity $\tilde{\kappa}^2 - \kappa^2 = x^2 - y^2$. Utilising this identity along with the bounds (2.42) yields

$$\begin{aligned}\tilde{g}_{\pm}(z;\delta,l) &> \left[(\kappa^2 + x^2 + y^2)x + 2\kappa y\right] lx - \left|(\kappa^2 - x^2 - y^2)y - 2\kappa x\right| ly \\ &\geq l(\kappa^2 + x^2 + y^2)(\tilde{\kappa}^2 - \kappa^2).\end{aligned}$$

Hence we always have $\tilde{g}_{\pm}(z;\delta,l) > 0$ for the evanescent modes $\tilde{k} > k$ (equivalent to $\tilde{\kappa} > \kappa$). Similarly, $g(z;\delta,l)$ can be simplified identically to (2.43) and we find that

$$\begin{aligned}\tilde{g}(z;\delta,l) &> l(l+2)\left(x^2((\kappa^2 + x^2 + y^2)^2 + 4\kappa^2 y^2) - y^2|(-\kappa^2 + x^2 + y^2)^2 - 4\kappa^2 x^2|\right) \\ &\quad + 4l\kappa xy\left(\kappa^2 + x^2 + y^2 - |-\kappa^2 + x^2 + y^2|\right) \\ &\geq l(l+2)(\kappa^2 + x^2 + y^2)^2(\tilde{\kappa}^2 - \kappa^2),\end{aligned}$$

and so we always have $\tilde{g}(z;\delta,l) > 0$ for the evanescent modes $\tilde{k} > k$ too. Together this shows that $R^{\mathrm{H}}_{1d}(\tilde{k}) < 1$ for all evanescent modes. Note that, for the remaining modes $\tilde{k} \leq k$, it is possible that $R^{\mathrm{H}}_{1d}(\tilde{k}) \geq 1$ for some choices of problem parameters.

We now refine the above bounds. In order to do so we make use of the identities $4x^2y^2 = \kappa^2 s^2$ and $x^2 + y^2 = \sqrt{(\tilde{\kappa}^2 - \kappa^2)^2 + \kappa^2 s^2}$ which arise since (by considering both real and imaginary parts of $z^2 = (x+iy)^2 = i\kappa s + \tilde{\kappa}^2 - \kappa^2$) we have that

$$(2.48) \quad 2x^2 = \sqrt{(\tilde{\kappa}^2 - \kappa^2)^2 + \kappa^2 s^2} + \tilde{\kappa}^2 - \kappa^2, \quad 2y^2 = \sqrt{(\tilde{\kappa}^2 - \kappa^2)^2 + \kappa^2 s^2} - \tilde{\kappa}^2 + \kappa^2.$$

Now, if we make use of the substitution $\kappa^2 + x^2 = \tilde{\kappa}^2 + y^2$ for the terms involving hyperbolic functions and the substitution $\kappa^2 - y^2 = \tilde{\kappa}^2 - x^2$ for the terms involving trigonometric functions, we obtain the following:

$$\tilde{g}_{\pm}(z;\delta,l) > \left[(\tilde{\kappa}^2 + 2y^2)x + 2\kappa y\right] lx - \left|(\tilde{\kappa}^2 - 2x^2)y - 2\kappa x\right| ly$$

$$
\begin{aligned}
&\geq l\left(x^2(\tilde{\kappa}^2+2y^2)-y^2|\tilde{\kappa}^2-2x^2|\right)\\
&=\begin{cases} l\tilde{\kappa}^2(x^2+y^2) & \text{if } \tilde{\kappa}^2\leq 2x^2,\\ l\left(4x^2y^2+\tilde{\kappa}^2(\tilde{\kappa}^2-\kappa^2)\right) & \text{if } \tilde{\kappa}^2 > 2x^2,\end{cases}\\
&=\begin{cases} l\tilde{\kappa}^2\sqrt{(\tilde{\kappa}^2-\kappa^2)^2+\kappa^2s^2} & \text{if } \tilde{\kappa}^2\leq 2x^2,\\ l\left(\tilde{\kappa}^4+\kappa^2(s^2-\tilde{\kappa}^2)\right) & \text{if } \tilde{\kappa}^2 > 2x^2,\end{cases}
\end{aligned}
$$

and

$$
\begin{aligned}
\tilde{g}(z;\delta,l) &> l(l+2)\left(x^2(\tilde{\kappa}^2+2y^2)^2-y^2(\tilde{\kappa}^2-2x^2)^2\right)\\
&\quad+4l\kappa xy\left(\tilde{\kappa}^2+2y^2-|\tilde{\kappa}^2-2x^2|\right)\\
&= l(l+2)\left(\tilde{\kappa}^4(x^2-y^2)+4x^2y^2(2\tilde{\kappa}^2+y^2-x^2)\right)\\
&\quad+4l\kappa xy\left(\tilde{\kappa}^2+2y^2-|\tilde{\kappa}^2-2x^2|\right)\\
&=\begin{cases} l(l+2)\left(\tilde{\kappa}^4(\tilde{\kappa}^2-\kappa^2)+4x^2y^2(\tilde{\kappa}^2+\kappa^2)\right) \\ \quad +8l\kappa^3xy & \text{if } \tilde{\kappa}^2\leq 2x^2,\\ l(l+2)\left(\tilde{\kappa}^4(\tilde{\kappa}^2-\kappa^2)+4x^2y^2(\tilde{\kappa}^2+\kappa^2)\right) \\ \quad +8l\kappa xy(x^2+y^2) & \text{if } \tilde{\kappa}^2 > 2x^2,\end{cases}\\
&=\begin{cases} l(l+2)\left(\tilde{\kappa}^6+\kappa^4s^2+\tilde{\kappa}^2\kappa^2(s^2-\tilde{\kappa}^2)\right) \\ \quad +4l\kappa^4s & \text{if } \tilde{\kappa}^2\leq 2x^2,\\ l(l+2)\left(\tilde{\kappa}^6+\kappa^4s^2+\tilde{\kappa}^2\kappa^2(s^2-\tilde{\kappa}^2)\right) \\ \quad +4l\kappa^2s\sqrt{(\tilde{\kappa}^2-\kappa^2)^2+\kappa^2s^2} & \text{if } \tilde{\kappa}^2 > 2x^2.\end{cases}
\end{aligned}
$$

From the penultimate expression in each case we see that for evanescent modes $\tilde{k} > k$ (i.e. $\tilde{\kappa} > \kappa$) we always have $\tilde{g}_\pm(z;\delta,l) > 0$ and $\tilde{g}(z;\delta,l) > 0$. Furthermore, from the final expressions we see that all modes $\tilde{k} \leq \sigma$ (i.e. $\tilde{\kappa} \leq s$) also give the desired positivity. Thus we deduce that when $\sigma \geq k$ we have positivity for all modes $\tilde{k}$ and hence $R^{\mathrm{H}}_{2d} < 1$. We also remark that modes $\tilde{k} \leq k$ which are relatively close to k are identified as those giving the worst bounds, suggesting these are the most problematic modes for the algorithm.

If $\sigma < k$ we may still have positivity of $\tilde{g}_\pm(z;\delta,l)$ and $\tilde{g}(z;\delta,l)$ for all modes so long as x or lx is large enough so that the hyperbolic term, which is always positive, is larger than the magnitude of the trigonometric term in both (2.41b) and (2.43b). Using (2.48) and converting back to the original variables we have that

$$x = 2\delta\sqrt{\tfrac{1}{2}\left(\sqrt{(k^2-\tilde{k}^2)^2+\sigma^2k^2}+\tilde{k}^2-k^2\right)}, \tag{2.49}$$

while lx has an identical expression except with 2δ replaced by L. Thus we see that, between the parameters σ, δ and L, so long as they are sufficiently large we will have $\tilde{g}_{\pm}(z;\delta,l) > 0$ and $\tilde{g}(z;\delta,l) > 0$ for all modes $\tilde{k}$ and thus $R^{\mathrm{H}}_{2d} < 1$ as desired. $\blacksquare$

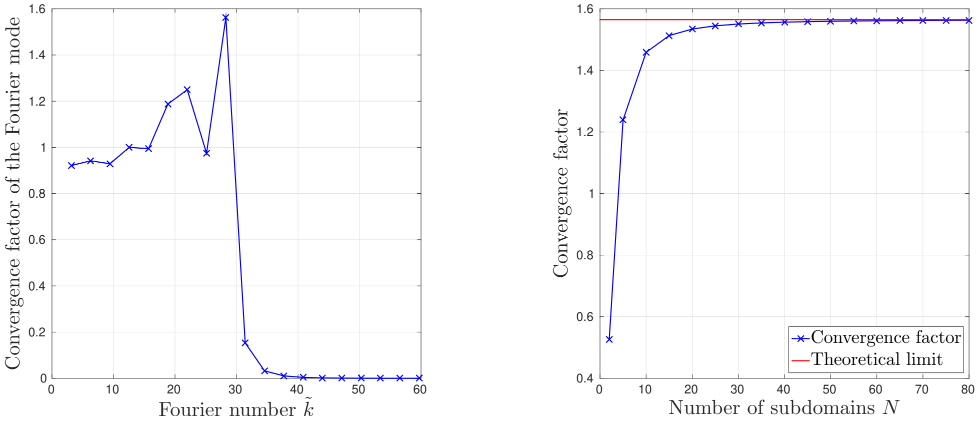

Figure 2.4: The convergence factor of each Fourier mode for $N = 80$ (left) and the convergence factor of the full Schwarz algorithm for varying number of subdomains N (right) when $\sigma = 0.1$, $k = 30$.

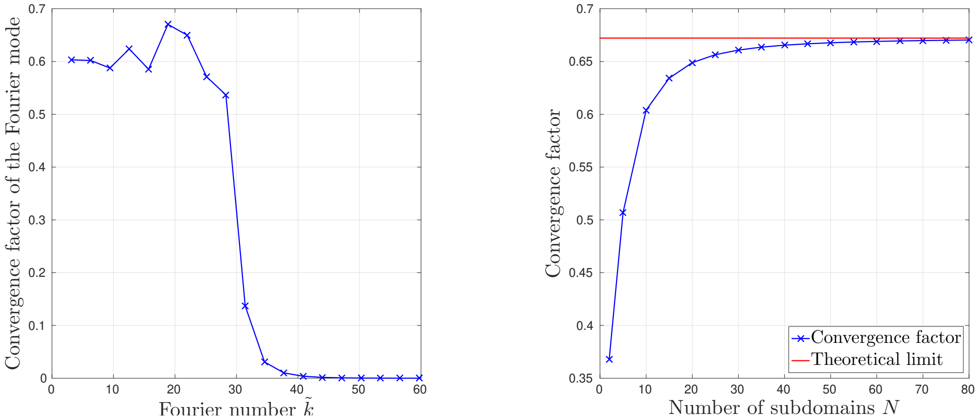

Figure 2.5: The convergence factor of each Fourier mode for $N = 80$ (left) and the convergence factor of the full Schwarz algorithm for varying number of subdomains N (right) when $\sigma = 1$, $k = 30$.

To verify these results, we compare numerically the spectral radius of the iteration matrix with the theoretical limit for different values of σ. We choose here $L = 1$, $\hat{L} = 1$ and $\delta = L/10$. From Figures 2.4 and 2.5 we see that, as predicted, the Schwarz algorithm is not convergent for all Fourier modes when σ is small, but becomes convergent for σ sufficiently large. In particular, we see in Figure 2.5 that the method can be convergent for $\sigma \ll k$. As expected from our theory, the algorithm always converges well for evanescent modes ($\tilde{k} > k$).

Similarly to the one-dimensional case we can also consider the question of k-robustness:

Corollary 2.2 (A case of k-independent convergence)**.** *Suppose $\alpha = ik$ (the case of classical impedance conditions) and that $\sigma = \sigma_0 k$ for some constant σ_0. Consider a k-dependent domain decomposition given by $L = L_0 k^{-1}$ and $\delta = \delta_0 k^{-1}$, that is the subdomain size and overlap shrink inversely proportional to the wave number. Then the convergence factor R_{2d}^{H} can be bounded above by a k-independent value and this bound becomes tight as $k \to \infty$. As such, the convergence of the corresponding Schwarz method is ultimately independent of the wave number k as it increases. Under the additional assumptions of Theorem 2.3 for convergence (now on σ_0, L_0 and δ_0), we thus deduce that the approach will ultimately be k-robust and independent of the number of subdomains.*

Proof. The proof is similar to the one-dimensional case except that now we must consider the Fourier number $\tilde{k}$. To do so, we let $\tilde{k}^2 = \beta k^2$. In this scenario, the coefficients a and b of the iteration matrix depend on k only through β. However, in the final convergence factor R_{2d}^{H} we take the supremum over all $\tilde{k}$, namely now over a discrete set of positive β values. This is bounded above by the supremum over all $\beta \in \mathbb{R}^+$, which is then independent of k, the supremum being finite since the bounds derived in Theorem 2.3 do not rely on the discrete nature of $\tilde{k}$ and so can be readily applied, translated into β. Note that as $k \to \infty$ the discrete set of β values becomes dense in $\mathbb{R}^+$ so this supremum bound becomes tight. Thus we will ultimately have k-robustness. Combining with Theorem 2.3 we further obtain that ultimately the convergence will also be independent of the number of subdomains. ■

Remark 2.1. *We note an empirical observation that, for reasonable values of σ, δ and L (namely when these parameters are not too small, essentially the same conditions required for convergence, but also neither of δ or σ being too large), the value of $\tilde{k}$ giving the supremum of $R_{1d}^{\mathrm{H}}(\tilde{k})$ lies in a small neighbourhood around k (equivalent to $\beta = 1$ in the above proof). This is consistent with other works in the literature, e.g., [GMN02, Con15], where the most problematic modes are those close to the cut-off k. In this case, a series expansion around $\tilde{k} = k$ shows that $k\delta$ and kL being fixed are the requirements on the domain decomposition parameters in order for the algorithm to be k-independent; see the supplementary `Maple` worksheets.*

For more general theory on k-robustness of the one-level method and rigorous GMRES bounds, see [GSZ20]. As in the one-dimensional case, we can link k and N so that we consider solving on a fixed domain a family of problems with increasing wave number using an increasing number of subdomains and, under the conditions of Theorem 2.3 and Corollary 2.2, our theory shows that the Schwarz algorithm will ultimately be k-robust and weakly scalable.

Remark 2.2. *We have focused here on the case of an overlapping domain decomposition. While the algorithm can also work in the non-overlapping case, it typically has a very poor behaviour. It is known from the literature (for example by setting the parameters to zero in formula (3.2) from [GMN02]) that if $\sigma = 0$ in the case of a decomposition into two subdomains, the purely iterative algorithm does not converge for evanescent modes ($\tilde{k} > k$), the convergence factor being equal to* 1*. By increasing σ, the convergence factor can be lowered but only a little (it remains close to one) and the algorithm continues to have very poor convergence properties for evanescent modes. This can be proven by similar techniques to those used in the overlapping case.*

We also note a fundamental difference between the one-dimensional and two-dimensional cases from the scalability point of view. Whereas in the first case independence to the number of subdomains is achieved simply by taking $\sigma > 0$, in the two-dimensional case things become more complex. This is consistent with previous convergence studies, starting from that in the seminal work on optimised transmission conditions [GMN02], where it has been observed that propagative and evanescent modes behave differently and the iterative algorithm does not converge for the cut-off frequency k. The maximum of the convergence factor is usually attained in a neighbourhood of $\tilde{k} = k$ and can be made sufficiently small when σ is taken large enough; in this case we can achieve scalability and k-robustness. We note that this kind of discrepancy, between one- and two-dimensional problems, is typical for the Helmholtz equation and cannot be observed in the case of the Laplace equation.

2.5.2 The transverse electric Maxwell's equations

We now apply the same ideas to the transverse electric Maxwell's equations with damping in the frequency domain. For an electric field $\mathbf{E} = (E_x, E_y)$, these equations are expressed as

$$\begin{aligned} \mathcal{L}\mathbf{E} &:= -k^2\mathbf{E} + \nabla \times (\nabla \times \mathbf{E}) + ik\sigma\mathbf{E} = \mathbf{0} \\ &\Leftrightarrow \begin{cases} -k^2 E_x - \partial_{yy} E_x + \partial_{xy} E_y + ik\sigma E_x = 0, \\ -k^2 E_y - \partial_{xx} E_y + \partial_{xy} E_x + ik\sigma E_y = 0, \end{cases} \end{aligned} \tag{2.50}$$

for $(x, y) \in \Omega$. The boundary conditions on the top and bottom boundaries ($y = 0$ and $y = \hat{L}$) are perfect electric conductor (PEC) conditions, the equivalent of Dirichlet conditions for Maxwell's equations:

$$\mathbf{E} \times \mathbf{n} = \mathbf{0} \Leftrightarrow E_x = 0, \qquad y = \{0, \hat{L}\}. \tag{2.51}$$

On the left and right boundaries ($x = a_1$ and $x = b_N$) we use impedance boundary conditions[2]:

$$\begin{aligned} &(\nabla \times \mathbf{E} \times \mathbf{n}) \times \mathbf{n} + ik\mathbf{E} \times \mathbf{n} = \mathbf{g} \\ \Leftrightarrow &\begin{cases} \mathcal{B}_l\mathbf{E} := (-\partial_x + ik)E_y + \partial_y E_x = g_1, & x = a_1, \\ \mathcal{B}_r\mathbf{E} := (\partial_x + ik)E_y - \partial_y E_x = -g_2, & x = b_N. \end{cases} \end{aligned} \tag{2.52}$$

The same conditions will be used at the interfaces between subdomains, akin to the classical algorithm defined in [DJR92]. The Maxwell problem (2.50)–(2.52) constitutes a *"two-dimensional wave-guide"* model.

Let us denote by $\mathbf{E}_j^n$ the approximation to the solution in subdomain j at iteration n. Starting from an initial guess $\mathbf{E}_j^0$, we compute $\mathbf{E}_j^n$ from the previous values $\mathbf{E}_j^{n-1}$ by solving the following local boundary value problems

$$\begin{cases} \mathcal{L}\mathbf{E}_j^n = \mathbf{0}, & x \in \Omega_j, \\ \mathcal{B}_l\mathbf{E}_j^n = \mathcal{B}_l\mathbf{E}_{j-1}^{n-1}, & x = a_j, \\ \mathcal{B}_r\mathbf{E}_j^n = \mathcal{B}_r\mathbf{E}_{j+1}^{n-1}, & x = b_j, \\ E_{x,j}^n = 0, & y \in \{0, \hat{L}\}, \end{cases} \tag{2.53}$$

for the interior subdomains ($1 < j < N$), while for the first ($j = 1$) and last ($j = N$) subdomain we impose $\mathcal{B}_l\mathbf{E}_1^n = g_1$ when $x = a_1$ and $\mathcal{B}_r\mathbf{E}_N^n = -g_2$ when $x = b_N$. To study the convergence of the Schwarz algorithm we define the local error in each subdomain j at iteration n as $\mathbf{e}_j^n = \mathbf{E}|_{\Omega_j} - \mathbf{E}_j^n$. Note that these errors verify boundary value problems which are the homogeneous counterparts of (2.53).

Due to the PEC boundary conditions on the top and bottom boundaries of each rectangular subdomain we can use the following Fourier series ansatzes to compute the

[2] Note that in rewriting the impedance conditions we can use the three-dimensional definition of the operators, i.e. $\mathbf{E} = (E_x, E_y, 0)$ and $\mathbf{n} = (1, 0, 0)$ for the right boundary and $\mathbf{n} = (-1, 0, 0)$ for the left boundary.

local solutions of $\mathcal{L}\mathbf{e}_j^n = 0$:

$$e_{x,j}^n = \sum_{m=1}^{\infty} v_j^n(x,\tilde{k})\sin(\tilde{k}y), \quad e_{y,j}^n = \sum_{m=0}^{\infty} w_j^n(x,\tilde{k})\cos(\tilde{k}y), \quad \tilde{k} = \frac{m\pi}{\hat{L}},\ m \in \mathbb{N}. \tag{2.54}$$

The first series of $e_{x,j}^n$ contains only the sine basis functions because of the homogeneous Dirichlet boundary condition on the bottom and the top of the boundary. As far as $e_{y,j}^n$ is concerned, the cos terms comes directly from the equations and boundary conditions in which we have replaced the series of $e_{x,j}^n$. Indeed, since these equations involve derivatives of $e_{x,j}^n$ w.r.t y, then the corresponding series for $e_{y,j}^n$ will contain the cos basis functions.

By plugging the expressions for $e_{x,j}^n$ and $e_{y,j}^n$ into $\mathcal{L}\mathbf{e}_j^n = \mathbf{0}$, a simple computation shows that, for each Fourier number $\tilde{k}$, we have the general solutions

$$v_j^n(x,\tilde{k}) = -\alpha_j^n \frac{\tilde{k}}{\zeta} e^{-\zeta x} + \beta_j^n \frac{\tilde{k}}{\zeta} e^{\zeta x}, \qquad w_j^n(x,\tilde{k}) = \alpha_j^n e^{-\zeta x} + \beta_j^n e^{\zeta x}, \tag{2.55}$$

where $\zeta(\tilde{k}) = \sqrt{ik\sigma + \tilde{k}^2 - k^2}$. From these formulae we can see easily that

$$\partial_x v_j^n = \tilde{k} w_j^n, \qquad \partial_x w_j^n = \frac{\zeta^2}{\tilde{k}} v_j^n. \tag{2.56}$$

In order to benefit again from the analysis in the one-dimensional case, we first prove the following result.

Lemma 2.4 (Maxwell reduction)**.** *For each Fourier number $\tilde{k}$, we have that both $v_j^n(x,\tilde{k})$ and $w_j^n(x,\tilde{k})$ are solutions of the following one-dimensional problem:*

$$\left\{ \begin{array}{rl} (ik\sigma + \tilde{k}^2 - k^2)u_j^n - \partial_{xx}u_j^n = 0, & x \in (a_j, b_j), \\ \mathcal{B}_{l,\sigma}u_j^n(x,\tilde{k}) = \mathcal{B}_{l,\sigma}u_{j-1}^{n-1}(x,\tilde{k}), & x = a_j, \\ \mathcal{B}_{r,\sigma}u_j^n(x,\tilde{k}) = \mathcal{B}_{r,\sigma}u_{j+1}^{n-1}(x,\tilde{k}), & x = b_j, \end{array} \right. \tag{2.57}$$

where $\mathcal{B}_{l,\sigma} = -\partial_x + ik + \sigma$ and $\mathcal{B}_{r,\sigma} = \partial_x + ik + \sigma$.

Proof. Let us notice first that, because of (2.56), we have

$$\partial_x e_{x,j}^n + \partial_y e_{y,j}^n = \sum_{m=1}^{\infty} \left(\partial_x v_j^n - \tilde{k} w_j^n\right) \sin(\tilde{k}y) = 0.$$

If we use this in the error equation $\mathcal{L}\mathbf{e}_j^n = \mathbf{0}$ we obtain that both $v_j^n(x,\tilde{k})$ and $w_j^n(x,\tilde{k})$

satisfy, for each $\tilde{k}$, the one-dimensional equation $(ik\sigma + \tilde{k}^2 - k^2)u_j^n - \partial_{xx}u_j^n = 0$. Let us analyse now the boundary conditions. With the help of (2.56), we consider the right boundary and note that the left one can be treated similarly:

$$\begin{aligned}\mathcal{B}_r\mathbf{e}_j^n = (\partial_x + ik)e_{y,j}^n - \partial_y e_{x,j}^n &= \sum_{m=1}^{\infty}((\partial_x + ik)w_j^n - \tilde{k}v_j^n)\cos(\tilde{k}y) \\ = \sum_{m=1}^{\infty}\left(\frac{ik}{\tilde{k}}\partial_x v_j^n + \left(\frac{\zeta^2}{\tilde{k}} - \tilde{k}\right)v_j^n\right)\cos(\tilde{k}y) &= \sum_{m=1}^{\infty}\frac{ik}{\tilde{k}}\mathcal{B}_{r,\sigma}v_j^n\cos(\tilde{k}y).\end{aligned}$$

Thus, imposing transfer of boundary data with $\mathcal{B}_r\mathbf{e}_j^n$ is equivalent to that with $\mathcal{B}_{r,\sigma}v_j^n$, for each Fourier number $\tilde{k}$. ■

It is now clear that the analysis of the two-dimensional case can again be derived from the one-dimensional case. That is, the result from Lemma 2.2 applies here if we replace α with $ik + \sigma$ and with ζ being defined by (2.47). Let us denote the resulting iteration matrix, for each $\tilde{k}$, by $\mathcal{T}_{1d}^{\mathrm{M}}(\tilde{k})$ and let $R_{1d}^{\mathrm{M}}(\tilde{k}) := \lim_{N\to\infty}\rho(\mathcal{T}_{1d}^{\mathrm{M}}(\tilde{k}))$ with $R_{2d}^{\mathrm{M}} = \sup_{\tilde{k}} R_{1d}^{\mathrm{M}}(\tilde{k})$. We can now state our main convergence result for the two-dimensional Maxwell problem.

Theorem 2.4 (Convergence of the Schwarz algorithm for Maxwell in 2D)**.** *For all $k > 0$, $\sigma > 0$, $\delta > 0$ and $L > 0$ we have that $R_{1d}^{\mathrm{M}}(\tilde{k}) < 1$ for all evanescent modes $\tilde{k} > k$. Furthermore, under the assumption that between them σ, δ and L are sufficiently large we have that $R_{2d}^{\mathrm{M}} < 1$. In particular, this is true when $\sigma \geq k$ for all $\delta > 0$ and $L > 0$. Therefore the convergence will ultimately be independent of the number of subdomains (we say that the Schwarz method will scale).*

Proof. By Lemma 2.3 we see that it is enough to study the sign of $g_\pm(z;\delta,l)$ and $g(z;\delta,l)$. To assist, we use the scaled notation $\kappa = 2\delta k$, $\tilde{\kappa} = 2\delta\tilde{k}$ and $s = 2\delta\sigma$ akin to (2.36). We can see that if $\alpha = ik + \sigma$ then for $z := x + iy$ (2.39) becomes

$$\Re v = \frac{-\kappa^2 - s^2 + x^2 + y^2}{(\kappa + y)^2 + (s + x)^2}, \quad \Im v = \frac{2sy - 2\kappa x}{(\kappa + y)^2 + (s + x)^2}, \quad |v|^2 = \frac{(\kappa - y)^2 + (s - x)^2}{(\kappa + y)^2 + (s + x)^2},$$

where $|v|^2 < 1$. We can now simplify $g_\pm(z;\delta,l)$ in (2.37) using these formulae to give

$$g_\pm(z;\delta,l) = \frac{4e^{x(l+1)}}{(\kappa + y)^2 + (s + x)^2}\tilde{g}_\pm(z;\delta,l) \tag{2.58a}$$

$$\begin{aligned}\tilde{g}_\pm(z;\delta,l) = {} & [(\kappa^2 + s^2 + x^2 + y^2)\sinh(x) + 2(\kappa y + sx)\cosh(x)]\sinh(lx) \\ & \pm [(\kappa^2 + s^2 - x^2 - y^2)\sin(y) + 2(sy - \kappa x)\cos(y)]\sin(ly).\end{aligned} \tag{2.58b}$$

Proceeding as before, using $\tilde{\kappa}^2 - \kappa^2 = x^2 - y^2$ and the bounds (2.42), we derive that

$$\tilde{g}_{\pm}(z;\delta,l) > l(\kappa^2 + s^2 + 2s + x^2 + y^2)(\tilde{\kappa}^2 - \kappa^2)$$

which is positive for all evanescent modes $\tilde{k} > k$. Similarly, simplifying $g(z;\delta,l)$ in (2.38) we find that

$$g(z;\delta,l) = \frac{4e^{2x(l+1)}}{((\kappa+y)^2 + (s+x)^2)^2}\tilde{g}(z;\delta,l) \tag{2.59a}$$

$$\begin{aligned} \tilde{g}(z;\delta,l) = {} & \big[((\kappa^2 + s^2 + x^2 + y^2)^2 + 4(\kappa y + sx)^2)\sinh(x(l+2)) \\ & \qquad + 4(\kappa y + sx)(\kappa^2 + s^2 + x^2 + y^2)\cosh(x(l+2))\big]\sinh(lx) \\ & + \big[((-\kappa^2 - s^2 + x^2 + y^2)^2 - 4(\kappa x - sy)^2)\sin(y(l+2)) \\ & \qquad + 4(\kappa x - sy)(-\kappa^2 - s^2 + x^2 + y^2)\cos(y(l+2))\big]\sin(ly), \end{aligned} \tag{2.59b}$$

from which we can obtain the bound

$$\begin{aligned} \tilde{g}(z;\delta,l) > {} & l(l+2)\left((\kappa^2 + s^2 + x^2 + y^2)^2 + 4s^2(x^2+y^2) + 8\kappa sxy\right)(\tilde{\kappa}^2 - \kappa^2) \\ & + 4ls\left(\kappa^2 + s^2 + x^2 + y^2\right)(\tilde{\kappa}^2 - \kappa^2). \end{aligned}$$

Again, this is positive for all evanescent modes and thus we deduce that $R^{\mathrm{M}}_{1d}(\tilde{k}) < 1$ for all $\tilde{k} > k$.

We now refine these bounds, as in the proof of Theorem 2.3 and using the same identities and substitutions. For $g_{\pm}(z;\delta,l)$ we first obtain

$$\tilde{g}_{\pm}(z;\delta,l) > l\left(x^2(s^2 + \tilde{\kappa}^2 + 2y^2) - y^2\left|s^2 + \tilde{\kappa}^2 - 2x^2\right| + 2x(\kappa y + sx) - 2y\left|\kappa x - sy\right|\right),$$

and split into four cases based on the sign of each term we take the absolute value of. Consider first the case $s^2 + \tilde{\kappa}^2 \le 2x^2$ and $\kappa x \le sy$, then

$$\begin{aligned} \tilde{g}_{\pm}(z;\delta,l) &> l\left((s^2 + \tilde{\kappa}^2)(x^2 + y^2) + 4\kappa xy + 2s(x^2 - y^2)\right) \\ &= l\left((s^2 + \tilde{\kappa}^2)\sqrt{(\tilde{\kappa}^2 - \kappa^2)^2 + \kappa^2 s^2} + 2\tilde{\kappa}^2 s\right). \end{aligned}$$

Now consider the case $s^2 + \tilde{\kappa}^2 > 2x^2$ and $\kappa x > sy$ where we find that

$$\begin{aligned} \tilde{g}_{\pm}(z;\delta,l) &> l\left(4x^2y^2 + (s^2 + \tilde{\kappa}^2)(x^2 - y^2) + 2s(x^2 + y^2)\right) \\ &= l\left(\tilde{\kappa}^2(s^2 + \tilde{\kappa}^2 - \kappa^2) + 2s\sqrt{(\tilde{\kappa}^2 - \kappa^2)^2 + \kappa^2 s^2}\right). \end{aligned}$$

The remaining cases follow as combinations of the previous two cases and we deduce, in the case $s^2 + \tilde{\kappa}^2 \leq 2x^2$ and $\kappa x > sy$, that

$$\tilde{g}_{\pm}(z;\delta,l) > l(s^2 + \tilde{\kappa}^2 + 2s)\sqrt{(\tilde{\kappa}^2 - \kappa^2)^2 + \kappa^2 s^2},$$

while the case $s^2 + \tilde{\kappa}^2 > 2x^2$ and $\kappa x \leq sy$ gives

$$\tilde{g}_{\pm}(z;\delta,l) > l\tilde{\kappa}^2(s^2 + \tilde{\kappa}^2 - \kappa^2 + 2s).$$

Turning to $\tilde{g}(z;\delta,l)$, we first derive that

$$\begin{aligned}\tilde{g}(z;\delta,l) > l(l+2)\big[&x^2((s^2+\tilde{\kappa}^2+2y^2)^2 + 4(\kappa y + sx)^2)\\ &- y^2((s^2+\tilde{\kappa}^2-2x^2)^2 + 4(\kappa x - sy)^2)\big]\\ &+ 4l\left(x(\kappa y + sx)(s^2+\tilde{\kappa}^2+2y^2) - y\left|(\kappa x - sy)(s^2+\tilde{\kappa}^2-2x^2)\right|\right),\end{aligned}$$

from which we see that we need to analyse just two sets of combined cases. First consider when both $s^2 + \tilde{\kappa}^2 \leq 2x^2$ and $\kappa x \leq sy$ or both $s^2 + \tilde{\kappa}^2 > 2x^2$ and $\kappa x > sy$, yielding

$$\begin{aligned}\tilde{g}(z;\delta,l) &> l(l+2)\big[(s^2+\tilde{\kappa}^2)^2(x^2-y^2) + 4x^2y^2(2s^2+2\tilde{\kappa}^2+y^2-x^2)\\ &\quad + 4s(x^2+y^2)(2\kappa xy + s(x^2-y^2))\big] + 4l(x^2+y^2)(2\kappa xy + s(s^2+\tilde{\kappa}^2))\\ &= l(l+2)\Big[\kappa^2 s^2(s^2+\kappa^2) + \tilde{\kappa}^2(s^2+\tilde{\kappa}^2)(s^2+\tilde{\kappa}^2-\kappa^2)\\ &\quad + 4\tilde{\kappa}^2 s^2\sqrt{(\tilde{\kappa}^2-\kappa^2)^2+\kappa^2 s^2}\Big] + 4ls(s^2+\tilde{\kappa}^2+\kappa^2)\sqrt{(\tilde{\kappa}^2-\kappa^2)^2+\kappa^2 s^2}.\end{aligned}$$

On the other hand, in the second set of cases when both $s^2 + \tilde{\kappa}^2 \leq 2x^2$ and $\kappa x > sy$ or both $s^2 + \tilde{\kappa}^2 > 2x^2$ and $\kappa x \leq sy$ we have

$$\begin{aligned}\tilde{g}(z;\delta,l) &> l(l+2)\big[(s^2+\tilde{\kappa}^2)^2(x^2-y^2) + 4x^2y^2(2s^2+2\tilde{\kappa}^2+y^2-x^2)\\ &\quad + 4s(x^2+y^2)(2\kappa xy + s(x^2-y^2))\big]\\ &\quad + 4l\left(2\kappa xy(s^2+\tilde{\kappa}^2+y^2-x^2) + s(4x^2y^2 + (s^2+\tilde{\kappa}^2)(x^2-y^2))\right)\\ &= l(l+2)\Big[\kappa^2 s^2(s^2+\kappa^2) + \tilde{\kappa}^2(s^2+\tilde{\kappa}^2)(s^2+\tilde{\kappa}^2-\kappa^2)\\ &\quad + 4\tilde{\kappa}^2 s^2\sqrt{(\tilde{\kappa}^2-\kappa^2)^2+\kappa^2 s^2}\Big] + 4ls\left(\kappa^2(s^2+\kappa^2) + \tilde{\kappa}^2(s^2+\tilde{\kappa}^2-\kappa^2)\right).\end{aligned}$$

Summarising, we see that all cases give $\tilde{g}_{\pm}(z;\delta,l) > 0$ and $\tilde{g}(z;\delta,l) > 0$ for all modes $\tilde{k}$ satisfying $\tilde{k}^2 \geq k^2 - \sigma^2$ (i.e. $\tilde{\kappa}^2 \geq \kappa^2 - s^2$). From this we can deduce that when $\sigma \geq k$ we have positivity for all modes $\tilde{k}$ and hence $R_{2d}^{\mathrm{M}} < 1$. Note that $\sigma \geq k$ is far from a necessary requirement and it is clear that there is some slack in these bounds. We also remark from this analysis that modes $\tilde{k} \leq \sqrt{k^2 - \sigma^2}$ which are relatively close to $\sqrt{k^2 - \sigma^2}$ yield the poorest bounds, suggesting they are the most problematic for the algorithm. Indeed, we may have $R^{\mathrm{M}}_{1d}(\tilde{k}) \geq 1$ when $\tilde{k} \leq \sqrt{k^2 - \sigma^2}$ for some choices of problem parameters. However, as in Theorem 2.4 we can force positivity of $\tilde{g}_{\pm}(z;\delta,l)$ and $\tilde{g}(z;\delta,l)$ for all modes so long as x or lx is large enough. Since x and lx take the same expressions as in Theorem 2.4 we can similarly deduce that, so long as the parameters σ, δ and L between them are sufficiently large, we will have $\tilde{g}_{\pm}(z;\delta,l) > 0$ and $\tilde{g}(z;\delta,l) > 0$ for all modes $\tilde{k}$ and thus the required conclusion that $R^{\mathrm{M}}_{2d} < 1$. ■

2.6 Numerical simulations on the discretised equation

In the following section we will show some numerical simulations which confirm our theory within the more practical setting of using an iterative Krylov method to accelerate convergence, with the Schwarz method being used as a preconditioner. We focus here on the two-dimensional Helmholtz equation, as described in Section 2.5, where a (horizontal) plane wave is incoming from the left boundary and homogeneous Dirichlet boundary conditions are imposed on the top and bottom boundaries, giving a waveguide problem. A second test case we consider is the propagation of such a wave in free space (i.e. when impedance boundary conditions are imposed on the whole boundary). While not covered by our theory, we will nonetheless observe similar conclusions, illustrating that the results apply more widely than within the restrictions of our theoretical assumptions. In our simulations, each subdomain is a unit square split uniformly with a fixed number of grid points in each direction. New subdomains are added on the right so that, with N subdomains, the whole domain is $\Omega = (0, N) \times (0, 1)$.

To discretise we use a uniform square grid in each direction and triangulate to form P1 elements. As we increase k we increase the number of grid points proportional to $k^{3/2}$ in order to ameliorate the pollution effect [BS97]. We use an overlap of size $2h$, with h being the mesh size. All computations are performed using FreeFem (`http://freefem.org/`), in particular using the `ffddm` framework. We solve the discretised problem using GMRES where the parallel Schwarz method with Robin conditions is used as a preconditioner. In particular, we use right-preconditioned GMRES and terminate when a relative residual tolerance of 10^{-6} is reached. The construction of the domain decomposition preconditioner is described in detail in [BDG$^+$19a, DJTO20]. The preconditioner, which arises naturally as the discretised version of the parallel Schwarz method with Robin conditions we have studied (see, e.g., [SCGT07]), is known as the one-level optimised restricted additive Schwarz (ORAS) preconditioner. This ORAS preconditioner is given by

$$\mathbf{M}^{-1} = \sum_{i=1}^{N} \mathbf{R}_i^T \mathbf{D}_i \mathbf{B}_i^{-1} \mathbf{R}_i$$

where $\{\mathbf{R}_i\}_{1\leq i\leq N}$ are the Boolean restriction matrices from the global to the local finite element spaces and $\{\mathbf{D}_i\}_{1\leq i\leq N}$ are local diagonal matrices representing the partition of unity. The key ingredient of the ORAS method is that the local subdomain matrices $\{\mathbf{B}_i\}_{1\leq i\leq N}$ incorporate more efficient Robin transmission conditions.

Note that, unlike in [GSZ20] where the emphasis is placed on the independence of the one-level method to the wave number, we focus here on the scalability aspect, i.e. the independence of the one-level method with respect to the number of subdomains N as soon as the absorption parameter $k\sigma$ is positive. We will observe that, beyond a sufficiently large value of N, the iteration count does not increase further, though in general this value will depend on the parameters of the problem, namely the wave number and absorption as well as the overlap and subdomain size. As a side effect, when the absorption is sufficiently large, i.e. of order k, wave number independence is also achieved.

In Table 2.1 we detail the GMRES iteration count for an increasing number of subdomains N and different values of k for the wave-guide problem and the wave propagation in free space problem. We set the conductivity parameter as $\sigma = 1$ (giving an absorption parameter k). We see that, after an initial increase, the iteration counts become independent of the number of subdomains and also independent of the wave number, which is consistent with the results obtained in [GSZ20] where the absorption parameter for optimal convergence is of order k. Another possible explanation of this is that when the absorption parameter increases, the waves are damped and their amplitude will decrease with the distance to the boundary on which the excitation is imposed. Hence, when additional subdomains are added, the solution will not vary much in these subdomains.

In the following we will provide more extensive numerical evidence on the application of the method.

In Table 2.2, Table 2.3, we perform numerical tests for a given value of the absorption parameter, and we vary the number of subdomains for a particular selection of small wavenumbers and definition of the local degrees of freedom. On each of these numerical computations we use a different size of overlap. We see that the behaviour of the method is not very sensitive to the size of the overlap. If the latter is further increased

Table 2.1: Preconditioned GMRES iteration counts for varying wave number k and number of subdomains N when $\sigma = 1$.

	Wave-guide problem								Free space problem							
k/N	8	16	24	32	40	48	56	64	8	16	24	32	40	48	56	64
20	19	22	25	30	30	30	30	30	19	21	25	25	25	25	25	25
40	18	21	24	29	29	29	29	29	17	19	24	25	25	25	25	25
60	19	21	24	29	29	29	29	29	16	19	24	25	25	25	25	25
80	19	21	24	28	28	28	28	28	16	18	24	25	25	25	25	25
100	19	21	24	28	28	28	28	28	16	18	24	25	25	25	25	24

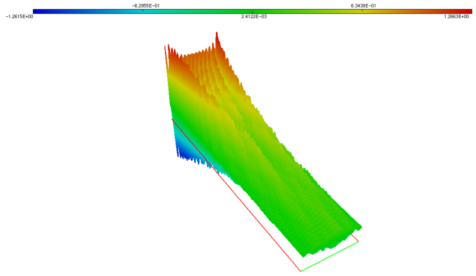

Figure 2.6: Waveguide solution with $\sigma = 1$ and $k = 100$

the performance will not improve. The absorption is defined as $\varepsilon = k\sigma$.

n_{loc}	k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
10	11.6	17	21	24	28	28	28	28	28	11.6
20	18.5	17	22	24	29	29	29	29	29	18.5
40	29.3	19	22	24	29	29	29	29	29	29.3

Table 2.2: GMRES iteration counts for the absorption parameter fixed to k and the overlap equal to 2.

n_{loc}	k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
10	11.6	17	20	24	28	28	28	28	28	11.6
20	18.5	17	22	24	29	29	29	29	29	18.5
40	29.3	19	22	24	29	29	29	29	29	29.3

Table 2.3: GMRES iteration counts for the absorption parameter fixed to k and the overlap equal to 6.

In Table 2.4 we are varying the number of subdomains and using a very small absorption parameter. We notice that the behaviour of the algorithm is very sensitive to the wavenumber although the iteration count tent to stabilise as the number of domains is increasing, as predicted by the theory.

n_{loc}	k	$N=8$	$N=16$	$N=24$	$N=32$	$N=40$	$N=48$	ε
10	11.6	32	71	113	151	177	212	$11.6\ 10^{-6}$
20	18.5	40	82	130	178	223	269	$18.5\ 10^{-6}$
40	29.3	74	147	237	305	388	457	$29.3\ 10^{-6}$

Table 2.4: GMRES iteration counts for very small absorption and the overlap equal to 2.

k	$N=8$	$N=16$	$N=24$	$N=32$	$N=40$	$N=48$	$N=56$	$N=64$	ε
20	19	22	25	30	30	30	30	30	20
40	18	21	24	29	29	29	29	29	40
60	19	21	24	29	29	29	29	29	60
80	19	21	24	28	28	28	28	28	80
100	19	21	24	28	28	28	28	28	100

Table 2.5: GMRES iteration counts for the absorption parameter fixed to k and the overlap equal to 2

In Table 2.5, Table 2.6, Table 2.7, we change the absorption parameter and consider a wider range of wavenumbers. We notice that when the damping (absorption) parameter increases the iteration count is considerably reduced, whereas if we decrease the damping GMRES behaviour will deteriorate.

k	$N=8$	$N=16$	$N=24$	$N=32$	$N=40$	$N=48$	$N=56$	$N=64$	ε
20	8	8	8	8	8	8	8	8	89.44
40	6	6	6	6	6	6	6	6	253
60	5	5	5	5	5	5	5	5	464.76
80	4	4	4	4	4	4	4	4	715.54
100	4	4	4	4	4	4	4	4	1000

Table 2.6: GMRES iteration counts for the absorption parameter fixed to $k^{3/2}$ and the overlap equal to 6.

k	$N=8$	$N=16$	$N=24$	$N=32$	$N=40$	$N=48$	$N=56$	$N=64$	ε
20	32	51	63	71	82	85	86	86	4.47
40	45	68	84	93	101	109	116	117	6.325
60	66	89	106	116	125	137	148	153	7.746
80	65	101	122	138	150	160	172	183	8.9443

Table 2.7: GMRES iteration counts for the absorption parameter fixed to $k^{1/2}$ and the overlap equal to 6.

In all the previous examples we have considered the waveguide problem with Dirichlet conditions on the top and bottom boundaries. In Table 2.8, Table 2.9, Table 2.10 we change the boundary value problem we impose impedance conditions everywhere which simulates the propagation of a wave in the free space. The conclusions are consistent with the ones obtained in the case of the waveguide except that for a smaller absorption parameter, the behaviour doesn't deteriorate as much as in the case of the waveguide.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	19	21	25	25	25	25	25	25	20
40	17	19	24	25	25	25	25	25	40
60	16	19	24	25	25	25	25	25	60
80	16	18	24	25	25	25	25	25	80
100	16	18	24	25	25	25	25	24	100

Table 2.8: GMRES iteration counts for the absorption parameter fixed to k and the overlap equal to 2. Impedance conditions are imposed in all of the boundaries.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	15	15	15	15	15	15	15	15	$20^{\frac{3}{2}}$
40	14	14	14	14	14	14	14	14	$40^{\frac{3}{2}}$
60	13	13	13	13	13	13	13	13	$60^{\frac{3}{2}}$
80	12	12	12	12	12	12	12	12	$80^{\frac{3}{2}}$
100	11	12	12	12	12	12	12	12	$100^{\frac{3}{2}}$

Table 2.9: GMRES iteration counts for the absorption parameter fixed to $k^{3/2}$ and the overlap equal to 2. Impedance conditions are imposed in all of the boundaries.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	20	31	43	44	47	52	57	64	$\sqrt{20}$
40	21	34	47	58	60	62	65	69	$\sqrt{40}$
60	21	34	48	61	67	72	75	78	$\sqrt{60}$
80	20	35	49	64	77	82	82	82	$\sqrt{80}$
100	21	36	51	66	81	95	99	99	$\sqrt{100}$

Table 2.10: GMRES iteration counts for the absorption parameter fixed to $k^{1/2}$ and the overlap equal to 2. Impedance conditions are imposed in all of the boundaries.

In Table 2.11, Table 2.12, Table 2.13, Table 2.14, Table 2.15, Table 2.16 we show the results of a series of tests obtained by varying the absorption and the source function, while we keep the same size of the overlap. We still consider the propagation in a waveguide where Dirichlet conditions are on top and bottom of the domain and Robin conditions on the other two boundaries and the interfaces. The right hand-side is given successively by a line source function

$$f(x, y) = 100 \sin^2(\pi x) e^{-10k(y-\frac{1}{2})^2}$$

and the point source function as

$$f(x,y) = 100\sum_{j=0}^{n} e^{-10k\left((x-(\frac{1}{2}+8j))^2+(y-\frac{1}{2})^2\right)}.$$

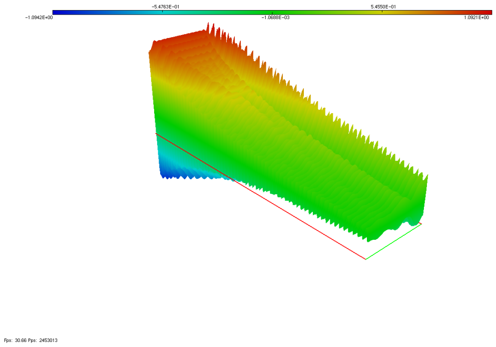

Figure 2.7: Waveguide solution with $\sigma = 1$ and $k = 100$

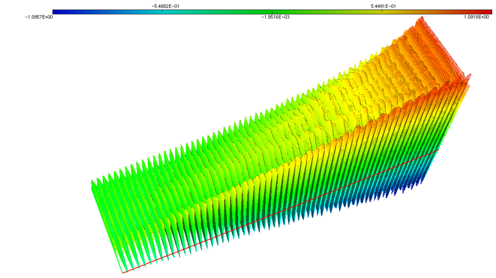

Figure 2.8: Solution using line source, with $\sigma = 1$ and $k = 100$

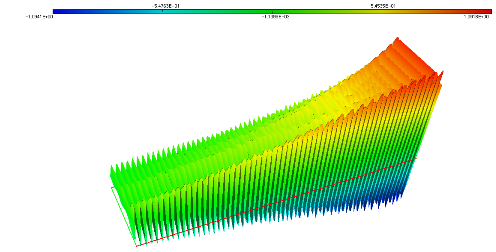

Figure 2.9: Solution using point source, with $\sigma = 1$ and $k = 100$

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	19	22	25	30	30	30	30	30	20
40	18	21	24	29	29	29	29	29	40
60	19	21	24	29	29	29	29	29	60
80	19	21	24	28	28	28	28	28	80
100	19	21	24	28	28	28	28	28	100

Table 2.11: GMRES iteration counts overlap equal to 2. This corresponds to the plane wave problem.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	23	27	30	32	32	31	32	32	20
40	24	27	29	30	31	31	31	31	40
60	25	27	28	30	30	30	31	31	60
80	25	27	28	29	29	29	29	29	80
100	24	27	28	28	29	29	29	29	100

Table 2.12: GMRES iteration counts for the problem with a point source. The overlap is fixed to be 2.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	18	21	23	24	24	24	24	24	20
40	17	20	20	20	20	19	19	19	40
60	18	20	20	20	20	20	20	20	60
80	17	17	18	18	18	18	17	17	80
100	14	15	15	15	15	15	15	15	100

Table 2.13: GMRES iteration counts for the problem with a line source. The overlap is fixed to be 2.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	32	51	63	71	82	85	86	86	4.47
40	45	68	84	93	101	109	116	117	6.325
60	66	89	106	116	125	137	148	153	7.746
80	65	101	122	138	150	160	172	183	8.9443

Table 2.14: GMRES iteration counts for the problem with absorption $k^{\frac{1}{2}}$. The overlap is fixed to be 2. This corresponds to the plane wave problem.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	39	56	71	79	87	93	96	99	$\sqrt{20}$
40	53	78	95	107	115	121	127	132	$\sqrt{40}$
60	86	114	130	140	147	154	162	169	$\sqrt{60}$
80	75	117	140	154	169	180	191	199	$\sqrt{80}$

Table 2.15: GMRES iteration counts for the problem with a point source with absorption $k^{\frac{1}{2}}$. The overlap is fixed to be 2.

k	$N = 8$	$N = 16$	$N = 24$	$N = 32$	$N = 40$	$N = 48$	$N = 56$	$N = 64$	ε
20	25	39	49	55	59	61	65	70	$\sqrt{20}$
40	31	50	59	67	70	74	78	83	$\sqrt{40}$
60	51	87	97	107	111	115	117	118	$\sqrt{60}$
80	39	62	79	91	99	103	108	111	$\sqrt{80}$
100	38	60	75	88	97	105	112	118	$\sqrt{100}$

Table 2.16: GMRES iteration counts for the problem with a line source with absorption $k^{\frac{1}{2}}$. The overlap is fixed to be 2.

Numerical results in this case are consistent with the previous ones: whereas in the case of the line source the overall behaviour degrades less quickly when increasing the wavenumber as in the case with the point source, the key parameter in the convergence of the algorithm remains the absorption. A physical interpretation of this phenomenon, as illustrated in Figure 2.6 is that the waves are damped more quickly when the absorption increases and therefore nothing relevant will be computed in the domains which are far away from the source, hence increasing the number of subdomains won't affect the overall convergence.

2.7 Conclusions

In this chapter we have analysed a purely iterative version of the Schwarz domain decomposition algorithm, in the limiting case of many subdomains, at the continuous level for the one-dimensional and two-dimensional Helmholtz and Maxwell's equations with absorption. The key mathematical tool which facilitated this study is the limiting spectrum of a sequence of block Toeplitz matrices having a particular structure, for which we proved a new result in the non-Hermitian case. The algorithm is convergent in the one-dimensional case as soon as we have absorption and, for sufficiently many subdomains N, its convergence factor becomes independent of the number of subdomains, meaning the algorithm is also scalable. In practice, this is achieved for relatively small N. In the two-dimensional case these conclusions remain true for the evanescent modes of the error (i.e. $\tilde{k} > k$) or when, between them, σ, δ and L are sufficiently large. In particular, we proved that the stationary iteration will always converge when $\sigma \geq k$, giving an absorption parameter k^2. The concept of the limiting spectrum proved to be a very elegant mathematical tool and can be used, for example, in constructing more sophisticated transmission conditions, to analyse the algorithm at the discrete level, or to design improved preconditioners.

Chapter 3

Algorithms for the Magneto telluric approximation of Maxwell's equations

Wave propagation phenomena are ubiquitous in science and engineering. In Geophysics, the magnetotelluric approximation of Maxwell's equations is an important tool to extract information about the spatial variation of electrical conductivity in the Earth's subsurface. This approximation results in a complex diffusion equation [DGH19],

$$\Delta u - (\sigma - i\varepsilon)u = f, \quad \text{in a domain } \Omega, \tag{3.1}$$

where f is the source function, and σ and ε are strictly positive constants[1].

In this chapter we analyse the parallel Schwarz algorithm with Dirichlet transmission conditions in the case of many subdomains. In a second step we design better transmission conditions of Robin type with the purpose of improving the convergence of the algorithm.

[1] In the magnetotelluric approximation we have $\sigma = 0$, but we consider the slightly more general case here. Note also that the zeroth order term in (3.1) is much more benign than the zeroth order term of opposite sign in the Helmholtz equation, see e.g. [EG12].

3.1 One dimensional problem

The purpose of this section is to perform the convergence analysis in one space dimension. We consider that we have a growing number of overlapping subdomains $\Omega_j = (a_j, b_j)$ such that $\Omega = \cup_{j=1}^{N} \Omega_j$ and $a_j = (j-1)L - \delta$ and $b_j = jL + \delta$. We note that $L + 2\delta$ is the width of each subdomain, 2δ is the size of the overlap.

3.1.1 Dirichlet transmission conditions

The Schwarz algorithm with Dirichlet transmission conditions writes:

$$\left\{ \begin{array}{l} (\sigma - i\varepsilon) u_j^n - \dfrac{d^2 u_j^n}{dx^2} = f_j, \ x \in (a_j, b_j) \\ u_j^n(a_j) = u_{j-1}^{n-1}(a_j), \, u_j^n(b_j) = u_{j+1}^{n-1}(b_j). \end{array} \right. \tag{3.2}$$

By linearity, it follows that the error function $e_j^n(x) = u_j^n(x) - u_j(x)$ satisfies the homogeneous counterpart of (3.2)

$$\left\{ \begin{array}{l} (\sigma - i\varepsilon) e_j^n - \dfrac{d^2 e_j^n}{dx^2} = 0, \ x \in (a_j, b_j) \\ e_j^n(a_j) = e_{j-1}^{n-1}(a_j), \, e_j^n(b_j) = e_{j+1}^{n-1}(b_j). \end{array} \right. \tag{3.3}$$

whose solutions are given by

$$e_j^n(x) = A_j^n e^{-\lambda x} + B_j^n e^{\lambda x}, \ \lambda = \sqrt{\sigma - i\varepsilon}. \tag{3.4}$$

By introducing (3.4) into the interface iteration of (3.3) we get

$$\begin{bmatrix} e^{-\lambda a_j} & e^{\lambda a_j} \\ e^{-\lambda b_j} & e^{\lambda b_j} \end{bmatrix} \begin{bmatrix} A_j^n \\ B_j^n \end{bmatrix} = \begin{bmatrix} e_{j-1}^{n-1}(a_j) \\ e_{j+1}^{n-1}(b_j) \end{bmatrix}$$

with the solutions

$$A_j^n = \frac{1}{D}\Big(e^{\lambda b_j} e_{j-1}^{n-1}(a_j) - e^{\lambda a_j} e_{j+1}^{n-1}(b_j) \Big), \ B_j^n = \frac{1}{D}\Big(- e^{-\lambda b_j} e_{j-1}^{n-1}(a_j) + e^{-\lambda a_j} e_{j+1}^{n-1}(b_j) \Big), \tag{3.5}$$

where $D = e^{\lambda(L+2\delta)} - e^{-\lambda(L+2\delta)}$. By replacing (3.5) into (3.4) we obtain

$$\begin{aligned} e_j^n(x) &= \tfrac{1}{D}\Big(e^{\lambda b_j} e_{j-1}^{n-1}(a_j) - e^{\lambda a_j} e_{j+1}^{n-1}(b_j) \Big) e^{-\lambda x} + \tfrac{1}{D}\Big(- e^{-\lambda b_j} e_{j-1}^{n-1}(a_j) + e^{-\lambda a_j} e_{j+1}^{n-1}(b_j) \Big) e^{\lambda x} \\ &= \tfrac{1}{D} e_{j-1}^{n-1}(a_j) \Big(e^{\lambda(b_j - x)} - e^{-\lambda(b_j - x)} \Big) + \tfrac{1}{D} e_{j+1}^{n-1}(b_j) \Big(e^{\lambda(x - a_j)} - e^{-\lambda(x - a_j)} \Big) \end{aligned} \tag{3.6}$$

By using (3.6) into (3.3), we see that the iteration over all interface values a_j and b_j can be written as

$$
\begin{gathered}
\begin{bmatrix} e_{j-1}^{n}(a_j) \\ e_{j+1}^{n}(b_j) \end{bmatrix} = T_1 \begin{bmatrix} e_{j-2}^{n-1}(a_{j-1}) \\ e_{j}^{n-1}(b_{j-1}) \end{bmatrix} + T_2 \begin{bmatrix} e_{j}^{n-1}(a_{j+1}) \\ e_{j+2}^{n-1}(b_{j+1}) \end{bmatrix} \\
T_1 = \begin{bmatrix} a & b \\ 0 & 0 \end{bmatrix}, \, T_2 = \begin{bmatrix} 0 & 0 \\ b & a \end{bmatrix}, \\
a = \frac{e^{2\lambda\delta} - e^{-2\lambda\delta}}{e^{\lambda(2\delta+L)} - e^{-\lambda(2\delta+L)}}, \, b = \frac{e^{\lambda L} - e^{-\lambda L}}{e^{\lambda(2\delta+L)} - e^{-\lambda(2\delta+L)}}.
\end{gathered}
\tag{3.7}
$$

In the case where $j \in \{1, 2, N-1, N\}$ where these are replaced by

$$
\begin{gathered}
\begin{bmatrix} 0 \\ e_{2}^{n}(b_1) \end{bmatrix} = T_2 \begin{bmatrix} e_{1}^{n-1}(a_2) \\ e_{3}^{n-1}(b_2) \end{bmatrix}, \\
\begin{bmatrix} e_{1}^{n}(a_2) \\ e_{3}^{n}(b_2) \end{bmatrix} = \tilde{T}_1 \begin{bmatrix} 0 \\ e_{2}^{n-1}(b_1) \end{bmatrix} + T_2 \begin{bmatrix} e_{2}^{n-1}(a_3) \\ e_{4}^{n-1}(b_3) \end{bmatrix}, \, \tilde{T}_1 = \begin{bmatrix} 0 & b \\ 0 & 0 \end{bmatrix}, \\
\begin{bmatrix} e_{N-2}^{n}(a_{N-1}) \\ e_{N}^{n}(b_{N-1}) \end{bmatrix} = T_1 \begin{bmatrix} e_{N-3}^{n-1}(a_{N-2}) \\ e_{N-1}^{n-1}(b_{N-2}) \end{bmatrix} + \tilde{T}_2 \begin{bmatrix} e_{N-1}^{n-1}(a_N) \\ 0 \end{bmatrix}, \, \tilde{T}_2 = \begin{bmatrix} 0 & 0 \\ b & 0 \end{bmatrix}, \\
\begin{bmatrix} e_{N-1}^{n}(a_N) \\ 0 \end{bmatrix} = T_1 \begin{bmatrix} e_{N-2}^{n-1}(a_{N-1}) \\ e_{N}^{n-1}(b_{N-1}) \end{bmatrix}.
\end{gathered}
\tag{3.8}
$$

The global iteration over all the interfaces can be summarised as follows

$$
\begin{bmatrix} e_{2}^{n}(b_1) \\ e_{1}^{n}(a_2) \\ e_{3}^{n}(b_2) \\ \vdots \\ e_{j-1}^{n}(a_j) \\ e_{j+1}^{n}(b_j) \\ \vdots \\ e_{N-2}^{n}(a_{N-1}) \\ e_{N}^{n}(b_{N-1}) \\ e_{N-1}^{n}(a_N) \end{bmatrix}
=
\underbrace{\begin{bmatrix}
0 & \hat{T}_2 & & & & & \\
\tilde{T}_1 & 0_{2\times 2} & T_2 & & & & \\
 & \ddots & \ddots & \ddots & & & \\
 & & T_1 & 0_{2\times 2} & T_2 & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & T_1 & 0_{2\times 2} & \tilde{T}_2 \\
 & & & & & \hat{T}_1 & 0
\end{bmatrix}}_{\mathcal{T}_{1d}}
\begin{bmatrix} e_{2}^{n-1}(b_1) \\ e_{1}^{n-1}(a_2) \\ e_{3}^{n-1}(b_2) \\ \vdots \\ e_{j-1}^{n-1}(a_j) \\ e_{j+1}^{n-1}(b_j) \\ \vdots \\ e_{N-2}^{n-1}(a_{N-1}) \\ e_{N}^{n-1}(b_{N-1}) \\ e_{N-1}^{n-1}(a_N) \end{bmatrix}
$$

where $\tilde{T}_1 = \begin{bmatrix} b \\ 0 \end{bmatrix}$, $\tilde{T}_2 = \begin{bmatrix} 0 \\ b \end{bmatrix}$, $\hat{T}_1 = \begin{bmatrix} a & b \end{bmatrix}$, $\hat{T}_2 = \begin{bmatrix} b & a \end{bmatrix}$. If define the error vector containing the values at the interfaces by

$$\mathbf{e}^n = [e_2^n(b_1),\, e_1^n(a_2),\, e_3^n(b_2) \ldots e_{j-1}^n(a_j),\, e_{j+1}^n(b_j) \ldots e_{N-2}^n(a_{N-1}),\, e_N^n(b_{N-1}),\, e_{N-1}^n(a_N)]^T$$

the interface iteration can expressed as

$$\mathbf{e}^n = \mathcal{T}_{1d}\mathbf{e}^{n-1} \tag{3.9}$$

with $\mathcal{T}_{1d}$ being the following block Toeplitz matrix:

$$\mathcal{T}_{1d} = \begin{bmatrix} A_0 & A_1 & & & & & \\ A_{-1} & A_0 & A_1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & A_{-1} & A_0 & A_1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & A_{-1} & A_0 & A_1 \\ & & & & & A_{-1} & A_0 \end{bmatrix}, \tag{3.10}$$

where

$$A_0 = \begin{bmatrix} 0 & b \\ b & 0 \end{bmatrix},\; A_1 = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix},\; A_{-1} = \begin{bmatrix} 0 & 0 \\ 0 & a \end{bmatrix}.$$

Since (3.9) is a stationary iteration, its convergence factor is given by the spectral radius of $\mathcal{T}_{1d}$. We can apply the result from Chapter 2 where it was proven that for matrices of this form their characteristic polynomial is verifying a three term recurrence and one can estimate the limiting spectrum:

(3.11)

$$\begin{aligned} R_{1d} := \lim_{N\to\infty} \rho(\mathcal{T}_{1d}) &= \max\left\{ \max_{\theta\in[-\pi,\pi]} \left| a\cos(\theta) \pm \sqrt{b^2 - a^2\sin^2(\theta)} \right|, \left| a^2 - \frac{1}{2}b^2 \right|^{1/2} \right\} \\ &= \max\{|a+b|, |a-b|, |a|\}. \end{aligned}$$

which can be interpreted as the limiting convergence factor of the iterative version of the Schwarz algorithm in the case of many subdomains and can be seen as a measure of the scalability when it is bounded (necessarily independently of N).

Lemma 3.1 (Convergence of the iterative Schwarz algorithm in the one-dimensional case)**.** *The convergence factor of the Schwarz algorithm as the number of domains tends* *to infinity verifies*

$$R_{1d} = \max\{|a+b|, |a-b|, |a|\} < 1, \forall \sigma, \varepsilon, \delta, L > 0.$$

therefore the convergence will ultimately be independent of the number of subdomains (we say that the method will scale).

Proof. Let us consider the complex valued functions $g_{\pm}(z) : \mathbb{C} \to \mathbb{C}$ (g_+ corresponds to the $+$ and g_- corresponds to the $-$) which is given by the formula

$$g_{\pm}(z) = \frac{e^{z} - e^{-z}}{e^{(l+1)z} - e^{-(l+1)z}} \pm \frac{e^{lz} - e^{-lz}}{e^{(l+1)z} - e^{-(l+1)z}}.$$

Note that $a \pm b = g_{\pm}(z)$ with

$$z = 2\lambda\delta = 2\delta\sqrt{\sigma - i\varepsilon},\, l = \frac{L}{2\delta}.$$

Similarly we can define the function g as being the first term in $g_{\pm}(z)$ such that $a = g(z)$. The maximum of $|g_{\pm}(z)|$ and $|g(z)|$ will provide an upper bound for the convergence factor R_{1d}. If $z = x + iy$ after some tedious computations we get that

$$\begin{aligned} |g_{\pm}(z)|^2 &= 1 - \frac{\tilde{g}_{\pm}(x,y,l)\Big(\mathrm{e}^{2x(l+1)} + 1 \mp 2\cos(y(l+1))\,\mathrm{e}^{x(l+1)}\Big)}{e^{4x(l+1)} + 1 - 2\,\mathrm{e}^{2x(l+1)}\cos(2y(l+1))} \\ \tilde{g}_{\pm}(x,y,l) &= (\mathrm{e}^{2x} - 1)(\mathrm{e}^{2lx} - 1) \mp 4\sin(ly)\sin(y)\,\mathrm{e}^{x(l+1)} \end{aligned}$$

Since both denominator $e^{4x(l+1)} + 1 - 2\,\mathrm{e}^{2x(l+1)}\cos(2y(l+1)) = (e^{2x(l+1)} - 1)^2 + 2(1 - \cos(2y(l+1)))\,\mathrm{e}^{2x(l+1)}$ and the second factor in the fraction $e^{2x(l+1)} + 1 \mp 2\,\mathrm{e}^{x(l+1)}\cos(y(l+1)) = (e^{x(l+1)} - 1)^2 + 2(1 \mp \cos(y(l+1)))\,\mathrm{e}^{x(l+1)}$ are obviously positive then we see that $|g_{\pm}(z)|^2 < 1 \Leftrightarrow g(x,y,l) > 0$. We see that this function can be further simplified to:

$$\tilde{g}_{\pm}(l,x,y) := 4\,\mathrm{e}^{x(l+1)}(\sinh(x)\sinh(lx) \mp \sin(ly)\sin(y)) \tag{3.12}$$

In the case when $x = \Re z = 2\delta\Re\sqrt{\sigma - i\varepsilon}$ and $y = 2\delta\Im\sqrt{\sigma - i\varepsilon}$, we see that $x = \sqrt{y^2 + (2\delta)^2\sigma} > |y|$ when $\sigma, \delta > 0$ and therefore $\tilde{g}_{\pm}(l,x,y) > 0$.

■

In order to check this result we will compute numerically the spectrum of the iteration matrix and compare it with the theoretical estimate for different values of the parameters. We have chosen here $L = 1$, $\delta = L/10$, $\varepsilon = 0.1$.

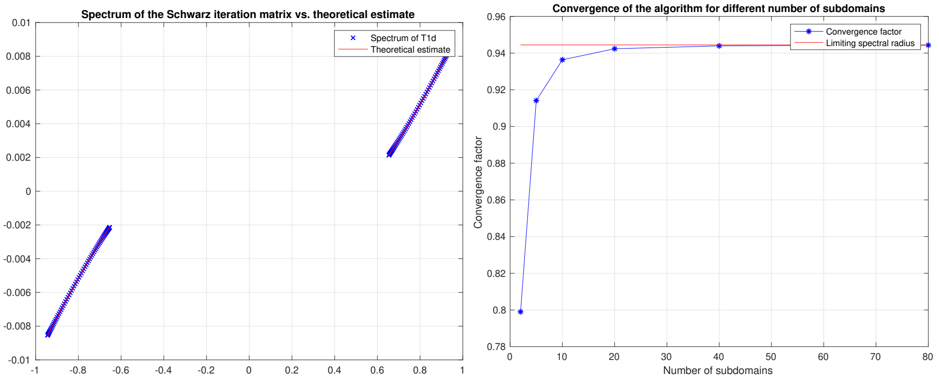

Figure 3.1: Spectrum of the iteration matrix for $N = 80$ $\sigma = 0.6$, $\varepsilon = 0.1$ (left) and the convergence factor vs. the number of subdomains (right)

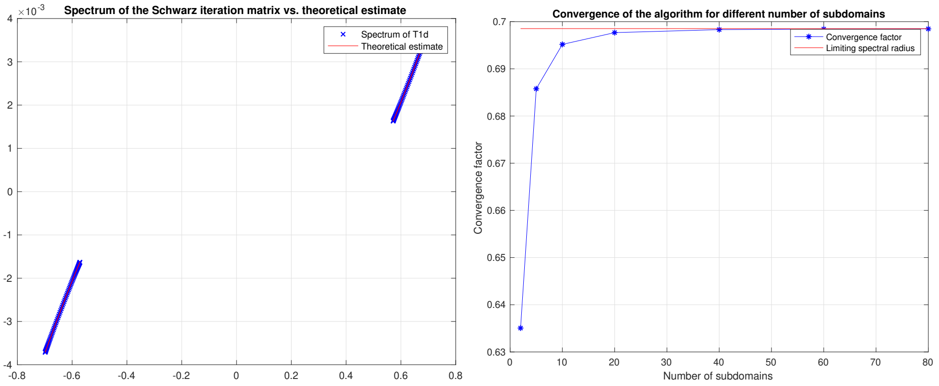

Figure 3.2: Spectrum of the iteration matrix for $N = 80$ $\sigma = 5$, $\varepsilon = 0.1$ (left) and the convergence factor vs. the number of subdomains (right)

A few intermediate conclusions can be drawn from Figure 3.1, 3.2 and the previous formulae

- the spectrum of the iteration matrix tends to the theoretical estimate when N is sufficiently large and the algorithms is convergent.
- when σ increases, the convergence is faster.
- We can notice also an improvement of the convergence if δ increases.
- The convergence factor is always strictly less than 1 as N tends to infinity which proves that the algorithm will scale when all the parameters are strictly positive.

We also see that for a fixed value of ε, the algorithm is converging very slowly when σ is small. We conclude in this case that the Dirichlet transmission conditions do not seem to be sufficient especially in the case of many subdomains as the limiting value of the convergence factor gets closer and closer to 1. At the same time, as we have seen from the proof the positivity of σ is key in having a convergent algorithm when the number of subdomains gets larger and larger.

3.1.2 One dimensional problem with Robin interface conditions

In the same spirit and using the same configuration of the subdomains as in the previous analysis, we investigate the convergence for the complex diffusion problem using different type of transmission conditions in one dimension. More precisely, we use Robin conditions

$$\left\{\begin{array}{rcl} (\sigma - i\varepsilon)u_j^n - \frac{d^2u_j^n}{dx^2} & = & f,\ x \in (a_j, b_j) \\ \mathcal{B}_l u_j^n := -\frac{du_j^n}{dx} + pu_j^n & = & -\frac{du_{j-1}^{n-1}}{dx} + pu_{j-1}^{n-1},\ at\ x = a_j \\ \mathcal{B}_r u_j^n := \frac{du_j^n}{dx} + pu_j^n & = & \frac{du_{j+1}^{n-1}}{dx} + pu_{j+1}^{n-1},\ at\ x = b_j \end{array}\right. \tag{3.13}$$

where p is positive parameter that can be further optimised, $\sigma > 0$, $\tilde{\varepsilon} > 0$. By linearity, it follows that the error function $e_j^n(x) = u_j^n(x) - u_j(x)$ satisfies the homogeneous counterpart of (3.13). The solutions of the latter inside the domain are given by (3.4) as in our former analysis. We introduce some useful notation

$$\mathcal{R}_-^{n-1}(a_j) = \mathcal{B}_l e_{j-1}^{n-1}(a_j),\ \mathcal{R}_+^{n-1}(b_j) = \mathcal{B}_l e_{j+1}^{n-1}(b_j) \tag{3.14}$$

which will allow us to re-write the algorithm. By introducing (3.4) into the interface iterations of (3.13) we get

$$\begin{bmatrix} (\lambda + p)e^{-\lambda a_j} & -(\lambda - p)e^{\lambda a_j} \\ -(\lambda - p)e^{-\lambda b_j} & (\lambda + p)e^{\lambda b_j} \end{bmatrix} \begin{bmatrix} A_j^n \\ B_j^n \end{bmatrix} = \begin{bmatrix} \mathcal{R}_-^{n-1}(a_j) \\ \mathcal{R}_+^{n-1}(b_j) \end{bmatrix}$$

which we can solve for the unknowns A_j^n and A_j^n to give

$$\begin{bmatrix} \alpha_j^n \\ \beta_j^n \end{bmatrix} = \frac{1}{D_j} \begin{bmatrix} (\lambda + p)e^{\lambda b_j} & (\lambda - p)e^{\lambda a_j} \\ (\lambda - p)e^{-\lambda b_j} & (\lambda + p)e^{-\lambda a_j} \end{bmatrix} \begin{bmatrix} \mathcal{R}_-^{n-1}(a_j) \\ \mathcal{R}_+^{n-1}(b_j) \end{bmatrix}, \tag{3.15}$$

where

$$D_j = (\lambda + p)^2 e^{\lambda(b_j - a_j)} - (\lambda - p)^2 e^{\lambda(a_j - b_j)}.$$

Note that since $b_j - a_j = L + 2\delta$ then D_j is actually independent of j and we will further denote it by D. The algorithm is based on Robin transmission conditions, hence the quantities of interest which are transmitted at the interfaces between subdomains are the Robin data. Therefore we need, in order to compute the current interface values $\mathcal{R}_-^n(a_j)$ and $\mathcal{R}_+^n(b_j)$, we replace the coefficients from (3.15) into (3.4) and then apply the formula (3.14).

(3.16)

$$\mathcal{R}_-^n(a_j) = \frac{1}{D}(a\mathcal{R}_-^{n-1}(a_{j-1}) + b\mathcal{R}_+^{n-1}(b_{j-1})),\ \mathcal{R}_+^n(b_j) = \frac{1}{D}(b\mathcal{R}_-^{n-1}(a_{j+1}) + a\mathcal{R}_+^{n-1}(b_{j+1})),$$

where

(3.17)

$$a = \frac{(\lambda+p)^2 e^{2\lambda\delta} - {(\lambda-p)}^2 e^{-2\lambda\delta}}{(\lambda+p)^2 e^{\lambda(L+2\delta)} - (\lambda-p)^2 e^{-\lambda(L+2\delta)}},\ b = -\frac{(\lambda^2-p^2)(e^{\lambda L} - e^{-\lambda L})}{(\lambda+p)^2 e^{\lambda(L+2\delta)} - (\lambda-p)^2 e^{-\lambda(L+2\delta)}}.$$

We can re-write (3.16) in matrix form

$$\begin{aligned}\begin{bmatrix} \mathcal{R}_-^n(a_j) \\ \mathcal{R}_+^n(b_j) \end{bmatrix} &= \frac{1}{D}\begin{bmatrix} a & b \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \mathcal{R}_-^{n-1}(a_{j-1}) \\ \mathcal{R}_+^{n-1}(b_{j-1}) \end{bmatrix} + \frac{1}{D}\begin{bmatrix} 0 & 0 \\ b & a \end{bmatrix}\begin{bmatrix} \mathcal{R}_-^{n-1}(a_{j+1}) \\ \mathcal{R}_+^{n-1}(b_{j+1}) \end{bmatrix} \\ &= T_1\begin{bmatrix} \mathcal{R}_-^{n-1}(a_{j-1}) \\ \mathcal{R}_+^{n-1}(b_{j-1}) \end{bmatrix} + T_2\begin{bmatrix} \mathcal{R}_-^{n-1}(a_{j+1}) \\ \mathcal{R}_+^{n-1}(b_{j+1}) \end{bmatrix},\ T_1 = \frac{1}{D}\begin{bmatrix} a & b \\ 0 & 0 \end{bmatrix}, T_2 = \frac{1}{D}\begin{bmatrix} 0 & 0 \\ b & a \end{bmatrix}.\end{aligned}$$

for $j = \{2, \ldots, N-1\}$. The situation is slightly different for the subdomains Ω_1 Ω_2, Ω_{N-1} and Ω_N

$$\begin{aligned}\begin{bmatrix} 0 \\ \mathcal{R}_+^n(b_1) \end{bmatrix} &= T_2\begin{bmatrix} \mathcal{R}_-^{n-1}(a_2) \\ \mathcal{R}_+^{n-1}(b_2) \end{bmatrix} \\ \begin{bmatrix} \mathcal{R}_-^n(a_2) \\ \mathcal{R}_+^n(b_2) \end{bmatrix} &= \widetilde{T_1}\begin{bmatrix} 0 \\ \mathcal{R}_+^{n-1}(b_1) \end{bmatrix} + T_2\begin{bmatrix} \mathcal{R}_-^{n-1}(a_3) \\ \mathcal{R}_+^{n-1}(b_3) \end{bmatrix} \\ \begin{bmatrix} \mathcal{R}_-^n(a_{N-1}) \\ \mathcal{R}_+^n(b_{N-1}) \end{bmatrix} &= T_1\begin{bmatrix} \mathcal{R}_-^{n-1}(a_{N-2}) \\ \mathcal{R}_+^{n-1}(b_{N-2}) \end{bmatrix} + \widetilde{T_2}\begin{bmatrix} \mathcal{R}_-^{n-1}(a_N) \\ 0 \end{bmatrix} \\ \begin{bmatrix} \mathcal{R}_-^n(a_N) \\ 0 \end{bmatrix} &= T_1\begin{bmatrix} \mathcal{R}_-^{n-1}(a_{N-1}) \\ \mathcal{R}_+^{n-1}(b_{N-1}) \end{bmatrix}\end{aligned}$$

where $\widetilde{T_1} = \frac{1}{D}\begin{bmatrix} 0 & b \\ 0 & 0 \end{bmatrix}, \widetilde{T_2} = \frac{1}{D}\begin{bmatrix} 0 & 0 \\ b & 0 \end{bmatrix}$. As in the case of the Dirichlet conditions, in compact form the algorithm can be written as

$$\begin{bmatrix} \mathcal{R}_+^n(b_1) \\ \mathcal{R}_-^n(a_2) \\ \mathcal{R}_+^n(b_2) \\ \vdots \\ \mathcal{R}_-^n(a_{N-1}) \\ \mathcal{R}_+^n(b_{N-1}) \\ \mathcal{R}_-^n(a_N) \end{bmatrix} = T_{1d}^{OS} \begin{bmatrix} \mathcal{R}_+^{n-1}(b_1) \\ \mathcal{R}_-^{n-1}(a_2) \\ \mathcal{R}_+^{n-1}(b_2) \\ \vdots \\ \mathcal{R}_-^{n-1}(a_{N-1}) \\ \mathcal{R}_+^{n-1}(b_{N-1}) \\ \mathcal{R}_-^{n-1}(a_N) \end{bmatrix}$$

where the iteration matrix T_{1d}^{OS} is of the same form as (3.10) but with a, b given by (3.17). The spectral radius of T_{1d}^{OS} determines the convergence of the algorithm. As in the case of the Dirichlet transmission conditions, the matrix is non-Hermitian but we can still apply the results on the limiting spectrum

$$R_{1opt} := \lim_{N\to\infty} \rho(\mathcal{T}_{1d}^{OS}) = \max\{|a+b|, |a-b|, |a|\}.$$

Note that we could perform a similar analysis in the case of Robin conditions in order to understand whether we can optimise it with respect to the positive parameter p. However, this analysis is of limited interest in the one dimensional case and we will perform it rather in the two-dimensional case.

We will compute numerically the optimal parameter p and spectrum of the iteration matrix. We have chosen here the same values as before for $L = 1$, $\delta = L/10$, and $\varepsilon = 0.1$. We see from the figures 3.3 and 3.4 that the convergence factor has highly improved with respect to 3.1 and 3.2.

3.2 Two-dimensional case with Dirichlet transmission conditions

The analysis that has been done in the one dimensional case is a relatively good start to get an intuition on how a parallel method works. We are going to do the same analysis for the two dimensional complex diffusion problem. The difference now is that we have a chain of many rectangles of height $\hat{L}$. The width of each subdomain, the size of the overlap and the endpoints are unaltered. The domain is defined as $\Omega_j = (a_j, b_j) \times (0, \hat{L})$.

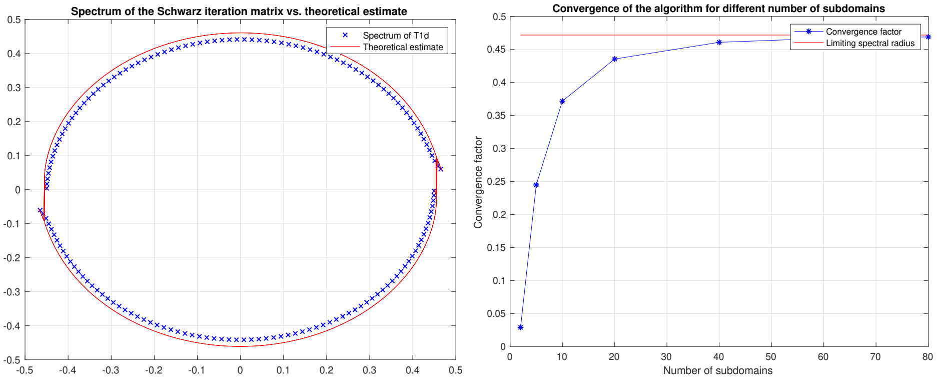

Figure 3.3: Spectrum of the iteration matrix for $N = 80$ $\sigma = 0.6$, $\varepsilon = 0.1$ (left) and the convergence factor vs. the number of subdomains (right)

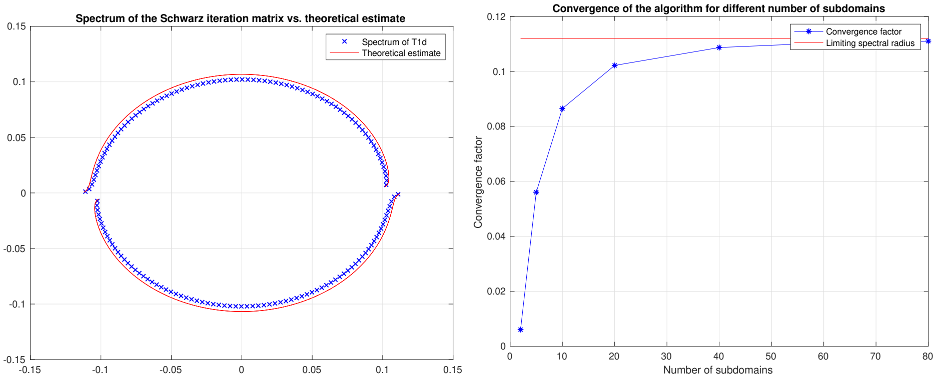

Figure 3.4: Spectrum of the iteration matrix for $N = 80$ $\sigma = 5$, $\varepsilon = 0.1$ (left) and the convergence factor vs. the number of subdomains (right)

The definition of the Schwarz method with Dirichlet transmission conditions for our problem is

$$(3.18)\qquad \left\{\begin{array}{rcl} (\sigma - i\varepsilon)u_j^n - \left(\dfrac{\partial^2 u_j^n}{\partial x^2} + \dfrac{\partial^2 u_j^n}{\partial y^2}\right) & = & f,\ (x,y) \in (a_j, b_j) \times (0, \hat{L}) \\ u_j^n(a_j, y) & = & u_{j-1}^{n-1}(a_j, y),\ y \in (0, \hat{L}), \\ u_j^n(b_j, y) & = & u_{j+1}^{n-1}(b_j, y),\ y \in (0, \hat{L}), \\ u_j^n(0, x) = u_j^n(\hat{L}, x) & = & 0,\ x \in (a_j, b_j). \end{array}\right.$$

By linearity, it follows that the error function e_j^n satisfies the homogeneous counterpart of (3.18).

We use Fourier series expansion for our error function, as we use Dirichlet boundary conditions on the top and the bottom of each rectangle:

$$e_j^n(x,y) = \sum_{m=1}^{\infty} v_j^n(x,\tilde{k})\sin(\tilde{k}y), \quad \tilde{k} = \frac{m\pi}{\hat{L}}. \tag{3.19}$$

By replacing it into the homogeneous counterpart of the equation (3.18) we get that for each Fourier number $\tilde{k}$, $v_j^n(\cdot,\tilde{k})$ verifies the one dimensional problem

$$\left\{ \begin{array}{rcl} (\sigma + \tilde{k}^2 - i\varepsilon)v_j^n - \frac{\partial^2 v_j^n}{\partial x^2} & = & 0, \ x \in (a_j, b_j), \\ v_j^n(a_j,\tilde{k}) & = & v_{j-1}^{n-1}(a_j,\tilde{k}). \\ v_j^n(b_j,\tilde{k}) & = & v_{j+1}^{n-1}(b_j,\tilde{k}). \end{array} \right. \tag{3.20}$$

which is exactly of the same type as (3.3) where η is replaced by $\eta + \tilde{k}^2$.

If define the error vector containing the values at the interfaces by

$$\mathbf{v}^n(\tilde{k}) = [v_2^n(b_1,\tilde{k}),\ v_1^n(a_2,\tilde{k}),\ v_3^n(b_2,\tilde{k}) \ldots v_{N-2}^n(a_{N-1},\tilde{k}),\ v_N^n(b_{N-1},\tilde{k}),\ v_{N-1}^n(a_N,\tilde{k})]^T$$

the interface iteration can expressed as

$$\mathbf{v}^n(\tilde{k}) = \mathcal{T}_{2d}\mathbf{v}^{n-1}(\tilde{k}) \tag{3.21}$$

with $\mathcal{T}_{2d}$ being a Toeplitz matrix of the form (3.10) with a, b defined as in (3.7) in which λ is replaced by $\lambda(\tilde{k})$ defined by

$$\lambda(\tilde{k}) = \sqrt{\sigma + \tilde{k}^2 - i\varepsilon},$$

as in (3.4) where we replaced σ by $\sigma + \tilde{k}^2$. Therefore in this case, the convergence result in Lemma 3.1 will also hold in this case.

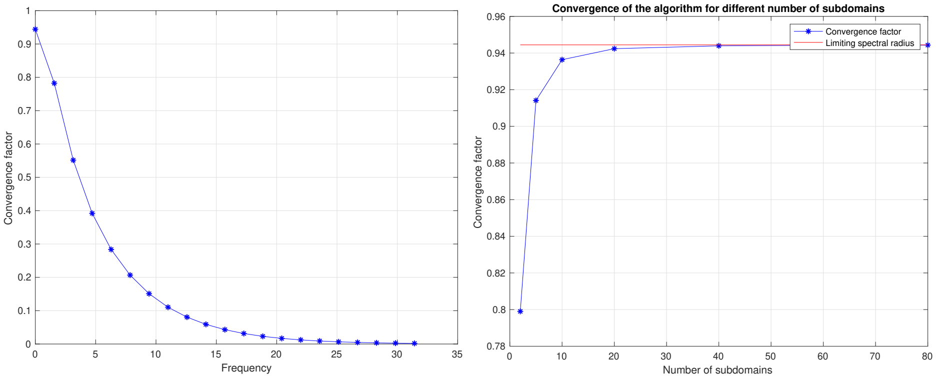

Figure 3.5: Convergence factor for $N = 80$ $\sigma = 0.6$, $\varepsilon = 0.1$ vs. the frequency (left) and the convergence factor vs. the number of subdomains (right)

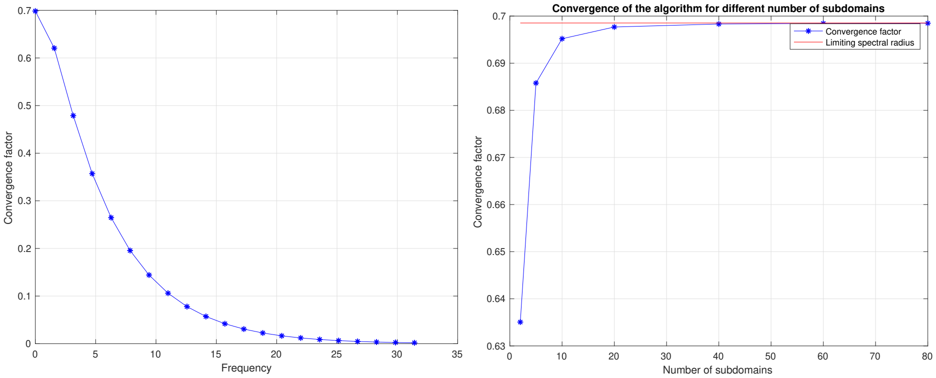

Figure 3.6: Convergence factor for $N = 80$ $\sigma = 5$, $\varepsilon = 0.1$ vs. the frequency (left) and the convergence factor vs. the number of subdomains (right)

Similar conclusions as in the one dimensional case will hold here with respect to the values of σ and ε. We also notice that the algorithm performs well in the case of high frequencies $\tilde{k}$, something that is already well known from the literature.

3.3 Optimizing transmission conditions for multiple subdomains

Classically transmission conditions between subdomains are optimized for a simplified two subdomain decomposition to obtain optimized Schwarz methods for many subdomains. We investigate here if such a simplified optimization suffices for the magnetotelluric approximation of Maxwell's equation which leads to a complex diffusion problem. We start with a direct analysis for 2 and 3 subdomains, and present asymptotically optimized transmission conditions in each case. We then optimize transmission conditions numerically for 4, 5 and 6 subdomains and observe the same asymptotic behavior of optimized transmission conditions. We finally use the technique of limiting spectra to optimize for a very large number of subdomains in a strip decomposition. Our analysis shows that the asymptotically best choice of transmission conditions is the same in all these situations, only the constants differ slightly. It is therefore enough for such diffusive type approximations of Maxwell's equations, which include the special case of the Laplace and screened Laplace equation, to optimize transmission parameters in the simplified two subdomain decomposition setting to obtain good transmission conditions for optimized Schwarz methods for more general decompositions.

To study Optimized Schwarz Methods (OSMs) for (3.1), we use a rectangular domain Ω given by the union of rectangular subdomains $\Omega_j := (a_j, b_j) \times (0, \hat{L})$, $j = 1, 2, \dots, J$, where $a_j = (j-1)L - \frac{\delta}{2}$ and $b_j = jL + \frac{\delta}{2}$, and δ is the overlap, like in [CCGV18]. Our OSM computes for iteration index $n = 1, 2, \dots$

$$\begin{array}{rcll} \Delta u_j^n - (\sigma - i\varepsilon)u_j^n &=& f & \text{in } \Omega_j, \\ -\partial_x u_j^n + p_j^- u_j^n &=& -\partial_x u_{j-1}^{n-1} + p_j^- u_{j-1}^{n-1} & \text{at } x = a_j, \\ \partial_x u_j^n + p_j^+ u_j^n &=& \partial_x u_{j+1}^{n-1} + p_j^+ u_{j+1}^{n-1} & \text{at } x = b_j, \end{array} \tag{3.22}$$

where p_j^- and p_j^+ are strictly positive parameters in the so called 2-sided OSM, see e.g. [GHM07], and we have at the top and bottom homogeneous Dirichlet boundary conditions, and on the left and right homogeneous Robin boundary conditions, i.e we put for simplicity of notation $u_0^{n-1} = u_{J+1}^{n-1} = 0$ in (3.22). The Robin parameters are fixed at the domain boundaries $x = a_1$ and $x = b_J$ to $p_1^- = p_a$ and $p_J^+ = p_b$. By linearity, it suffices to study the homogeneous equations, $f = 0$, and analyze convergence to zero of the OSM (3.22). Expanding the homogeneous iterates in a Fourier series

$$u_j^n(x, y) = \sum_{m=1}^{\infty} v_j^n(x, \tilde{k}) \sin(\tilde{k}y)$$

where $\tilde{k} = \frac{m\pi}{\hat{L}}$ to satisfy the homogeneous Dirichlet boundary conditions at the top and bottom, we obtain for the Fourier coefficients the equations

$$\begin{array}{rcll} \partial_{xx} v_j^n - (\tilde{k}^2 + \sigma - i\varepsilon)v_j^n &=& 0 & x \in (a_j, b_j), \\ -\partial_x v_j^n + p_j^- v_j^n &=& -\partial_x v_{j-1}^{n-1} + p_j^- v_{j-1}^{n-1} & \text{at } x = a_j, \\ \partial_x v_j^n + p_j^+ v_j^n &=& \partial_x v_{j+1}^{n-1} + p_j^+ v_{j+1}^{n-1} & \text{at } x = b_j. \end{array} \tag{3.23}$$

The general solution of the differential equation is

$$v_j^n(x,\tilde{k}) = \tilde{c}_j e^{-\lambda(\tilde{k})x} + \tilde{d}_j e^{\lambda(\tilde{k})x},$$

where $\lambda = \lambda(\tilde{k}) = \sqrt{\tilde{k}^2 + \sigma - i\varepsilon}$. We next define the Robin traces,

$$\mathcal{R}_-^{n-1}(a_j,\tilde{k}) := -\partial_x v_{j-1}^{n-1}(a_j,\tilde{k}) + p_j^- v_{j-1}^{n-1}(a_j,\tilde{k}),\ \mathcal{R}_+^{n-1}(b_j,\tilde{k}) := \partial_x v_{j+1}^{n-1}(b_j,\tilde{k}) + p_j^+ v_{j+1}^{n-1}(b_j,\tilde{k}).$$

Inserting the solution into the transmission conditions in (3.23), we obtain for the remaining coefficients $\tilde{c}_j$ and $\tilde{d}_j$ the linear system

$$\begin{aligned}\tilde{c}_j e^{-\lambda a_j}(p_j^- + \lambda) + \tilde{d}_j e^{\lambda a_j}(p_j^- - \lambda) &= \mathcal{R}_-^{n-1}(a_j,\tilde{k}),\\ \tilde{c}_j e^{-\lambda b_j}(p_j^+ - \lambda) + \tilde{d}_j e^{\lambda b_j}(p_j^+ + \lambda) &= \mathcal{R}_+^{n-1}(b_j,\tilde{k}),\end{aligned}$$

whose solution is

$$\begin{aligned}\tilde{c}_j &= \frac{1}{D_j}(e^{\lambda b_j}(p_j^+ + \lambda)\mathcal{R}_-^{n-1}(a_j,\tilde{k}) - e^{\lambda a_j}(p_j^- - \lambda)\mathcal{R}_+^{n-1}(b_j,\tilde{k})),\\ \tilde{d}_j &= \frac{1}{D_j}(-e^{-\lambda b_j}(p_j^+ - \lambda)\mathcal{R}_-^{n-1}(a_j,\tilde{k}) + e^{-\lambda a_j}(p_j^- + \lambda)\mathcal{R}_+^{n-1}(b_j,\tilde{k})),\end{aligned}$$

where

$$D_j := (\lambda + p_j^+)(\lambda + p_j^-)e^{\lambda(L+\delta)} - (\lambda - p_j^+)(\lambda - p_j^-)e^{-\lambda(L+\delta)}.$$

We thus arrive for the Robin traces in the OSM at the iteration formula

$$\begin{aligned}\mathcal{R}_-^n(a_j,\tilde{k}) &= \alpha_j^- \mathcal{R}_-^{n-1}(a_{j-1},\tilde{k}) + \beta_j^- \mathcal{R}_+^{n-1}(b_{j-1},\tilde{k}),\ j = 2,\ldots,J,\\ \mathcal{R}_+^n(b_j,\tilde{k}) &= \beta_j^+ \mathcal{R}_-^{n-1}(a_{j+1},\tilde{k}) + \alpha_j^+ \mathcal{R}_+^{n-1}(b_{j+1},\tilde{k}),\ j = 1,\ldots,J-1,\end{aligned}$$

where

(3.24)

$$\begin{aligned}\alpha_j^- &:= \frac{(\lambda + p_{j-1}^+)(\lambda + p_j^-)e^{\lambda\delta} - (\lambda - p_{j-1}^+)(\lambda - p_j^-)e^{-\lambda\delta}}{(\lambda + p_{j-1}^+)(\lambda + p_{j-1}^-)e^{\lambda(L+\delta)} - (\lambda - p_{j-1}^+)(\lambda - p_{j-1}^-)e^{-\lambda(L+\delta)}},\ j = 2,\ldots,J,\\ \alpha_j^+ &:= \frac{(\lambda + p_{j+1}^-)(\lambda + p_j^+)e^{\lambda\delta} - (\lambda - p_{j+1}^-)(\lambda - p_j^+)e^{-\lambda\delta}}{(\lambda + p_{j+1}^+)(\lambda + p_{j+1}^-)e^{\lambda(L+\delta)} - (\lambda - p_{j+1}^+)(\lambda - p_{j+1}^-)e^{-\lambda(L+\delta)}},\ j = 1,\ldots,J-1\end{aligned}$$

(3.25)

$$\begin{aligned}\beta_j^- &:= \frac{(\lambda + p_j^-)(\lambda - p_{j-1}^-)e^{-\lambda L} - (\lambda - p_j^-)(\lambda + p_{j-1}^-)e^{\lambda L}}{(\lambda + p_{j-1}^+)(\lambda + p_{j-1}^-)e^{\lambda(L+\delta)} - (\lambda - p_{j-1}^+)(\lambda - p_{j-1}^-)e^{-\lambda(L+\delta)}},\ j = 2,\ldots,J,\\ \beta_j^+ &:= \frac{(\lambda + p_j^+)(\lambda - p_{j+1}^+)e^{-\lambda L} - (\lambda - p_j^+)(\lambda + p_{j+1}^+)e^{\lambda L}}{(\lambda + p_{j+1}^+)(\lambda + p_{j+1}^-)e^{\lambda(L+\delta)} - (\lambda - p_{j+1}^+)(\lambda - p_{j+1}^-)e^{-\lambda(L+\delta)}},\ j = 1,\ldots,J-1.\end{aligned}$$

Defining the matrices

$$T_j^1 := \begin{bmatrix} \alpha_j^- & \beta_j^- \\ 0 & 0 \end{bmatrix}, \; j = 2, .., J \quad \text{and} \quad T_j^2 := \begin{bmatrix} 0 & 0 \\ \beta_j^+ & \alpha_j^+ \end{bmatrix}, \; j = 1, .., J-1,$$

we can write the OSM in substructured form (keeping the first and last rows and columns to make the block structure appear), namely

(3.26)

$$\underbrace{\begin{bmatrix} 0 \\ \mathcal{R}_+^n(b_1,\tilde{k}) \\ \mathcal{R}_-^n(a_2,\tilde{k}) \\ \mathcal{R}_+^n(b_2,\tilde{k}) \\ \vdots \\ \mathcal{R}_-^n(a_j,\tilde{k}) \\ \mathcal{R}_+^n(b_j,\tilde{k}) \\ \vdots \\ \mathcal{R}_-^n(a_{N-1},\tilde{k}) \\ \mathcal{R}_+^n(b_{N-1},\tilde{k}) \\ \mathcal{R}_-^n(a_N,\tilde{k}) \\ 0 \end{bmatrix}}_{\mathcal{R}^n} = \underbrace{\begin{bmatrix} & T_1^2 & & & & \\ T_2^1 & & T_2^2 & & & \\ & \ddots & & \ddots & & \\ & & T_j^1 & & T_j^2 & \\ & & & \ddots & & \ddots \\ & & & T_{N-1}^1 & & T_{N-1}^2 \\ & & & & T_N^1 & \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} 0 \\ \mathcal{R}_+^{n-1}(b_1,\tilde{k}) \\ \mathcal{R}_-^{n-1}(a_2,\tilde{k}) \\ \mathcal{R}_+^{n-1}(b_2,\tilde{k}) \\ \vdots \\ \mathcal{R}_-^{n-1}(a_j,\tilde{k}) \\ \mathcal{R}_+^{n-1}(b_j,\tilde{k}) \\ \vdots \\ \mathcal{R}_-^{n-1}(a_{N-1},\tilde{k}) \\ \mathcal{R}_+^{n-1}(b_{N-1},\tilde{k}) \\ \mathcal{R}_-^{n-1}(a_N,\tilde{k}) \\ 0 \end{bmatrix}}_{\mathcal{R}^{n-1}}.$$

If the parameters $p_j^\pm$ are constant over all the interfaces, and we eliminate the first and the last row and column of T, T becomes a block Toeplitz matrix. The best choice of the parameters minimizes the spectral radius $\rho(T)$ over a numerically relevant range of frequencies $K := [\tilde{k}_{\min}, \tilde{k}_{\max}]$ with $\tilde{k}_{\min} := \frac{\pi}{\hat{L}}$ (or 0 for simplicity) and $\tilde{k}_{\max} := \frac{M\pi}{\hat{L}}$, $M \sim \frac{1}{h}$, where h is the mesh size, and is thus solution of the min-max problem

$$\min_{p_j^\pm} \max_{\tilde{k}\in K} |\rho(T(\tilde{k}, p_j^\pm))|.$$

The traditional approach to obtain optimized transmission conditions for optimized Schwarz methods is to optimize performance for a simple two subdomain model problem, and then to use the result also in the case of many subdomains. We want to study here if this approach is justified, by directly optimizing the performance for two and more subdomains, and then comparing the results.

3.3.1 Optimization for 2, 3, 4, 5 and 6 subdomains

For two subdomains, the general substructured iteration matrix becomes

$$T = \begin{bmatrix} 0 & \beta_1^+ \\ \beta_2^- & 0 \end{bmatrix}.$$

The eigenvalues of this matrix are $\pm\sqrt{\beta_1^+\beta_2^-}$ and thus the square of the convergence factor is $\rho^2 = \left|\beta_1^+\beta_2^-\right|$.

Theorem 3.1 (Two Subdomain Optimization)**.** *Let* $s{:=}\sqrt{\sigma - i\varepsilon}$*, where the complex square root is taken with a positive real part, and let* C *be the real constant*

$$C{:=}\Re\frac{s((p_b+s)(p_a+s)-(s-p_b)(s-p_a)e^{-4sL})}{((s-p_a)e^{-2sL}+s+p_a)((s-p_b)e^{-2sL}+s+p_b)}. \tag{3.27}$$

Then for two subdomains with $p_1^+ = p_2^- {=:} p$ *and* $\tilde{k}_{\min} = 0$*, by equioscillation of the solution, the asymptotically optimized parameter* p *for small overlap* δ *and associated convergence factor are*

$$p = 2^{-1/3}C^{2/3}\delta^{-1/3}, \quad \rho = 1 - 2\cdot 2^{1/3}C^{1/3}\delta^{1/3} + \mathcal{O}(\delta^{2/3}). \tag{3.28}$$

If $p_1^+ \neq p_2^-$ *and* $\tilde{k}_{min} = 0$*, the asymptotically optimized parameters for small overlap* δ *and associated convergence factor are*

$$p_1^+ = 2^{-2/5}C^{2/5}\delta^{-3/5},\ p_2^- = 2^{-4/5}C^{4/5}\delta^{-1/5},\ \rho = 1-2\cdot 2^{-1/5}C^{1/5}\delta^{1/5}+\mathcal{O}(\delta^{2/5}). \tag{3.29}$$

Proof. We obtain that the solution of the min-max problem equioscillates, $\rho(0) = \rho(\tilde{k}^*)$, where $\tilde{k}^*$ is an interior maximum point, and asymptotically $p = C_p\delta^{-1/3}$, $\rho = 1 - C_R\delta^{1/3} + \mathcal{O}(\delta^{2/3})$, and $\tilde{k}^* = C_k\delta^{-2/3}$. By expanding for δ small, and setting the leading term in the derivative $\frac{\partial\rho}{\partial\tilde{k}}(\tilde{k}^*)$ to zero, we get $C_p = \frac{C_k^2}{2}$. Expanding the maximum leads to $\rho(\tilde{k}^*) = \rho(C_k\delta^{-2/3}) = 1 - 2C_k\delta^{1/3} + \mathcal{O}(\delta^{2/3})$, therefore $C_R = 2C_k$. Finally the solution of the equioscillation equation $\rho(0) = \rho(\tilde{k}^*)$ determines uniquely $C_k = 2^{1/3}C^{1/3}$.

In the case with two parameters, we have two equioscillations, $\rho(0) = \rho(\tilde{k}_1^*) = \rho(\tilde{k}_2^*)$ (as seen in Figure 3.7) where $\tilde{k}_j^*$ are two interior local maxima, and asymptotically $p_1 = C_{p1}\delta^{-3/5}$, $p_1 = C_{p1}\delta^{-1/5}$, $\rho = 1 - C_R\delta^{1/5} + \mathcal{O}(\delta^{2/5})$, $\tilde{k}_1^* = C_{k1}\delta^{-2/5}$ and $\tilde{k}_2^* = C_{k2}\delta^{-4/5}$. By expanding for δ small, and setting the leading terms in the derivatives $\frac{\partial\rho}{\partial\tilde{k}}(\tilde{k}_{1,2}^*)$ to zero, and we get $C_{p1} = C_{k2}^2$, $C_{p2} = \frac{C_{k1}^2}{C_{k2}^2}$. Expanding the maxima leads to $\rho(\tilde{k}_1^*) = \rho(C_k\delta^{-2/5}) = 1 - 2\frac{C_{k1}}{C_{k2}^2}\delta^{1/5} + \mathcal{O}(\delta^{2/5})$ and $\rho(\tilde{k}_2^*) = \rho(C_k\delta^{-4/5}) = 1 - 2C_{k_2}\delta^{1/5} + \mathcal{O}(\delta^{2/5})$

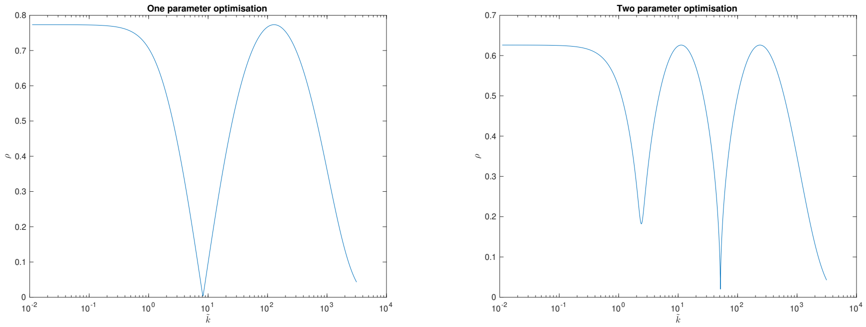

Figure 3.7: Equioscillation in numerical optimisation with one and two optimised parameters

and equating $\rho(\tilde{k}_1^*) = \rho(\tilde{k}_2^*)$ we get $C_{k1} = C_{k2}^3$ and $C_R = 2C_{k2}$. Finally equating $\rho(0) = \rho(\tilde{k}_2^*)$ asymptotically determines uniquely $C_{k2} = 2^{-1/5}C^{1/5}$ and then $C_{k1} = C_{k2}^3$ and $C_{p1} = C_{k2}^2$, $C_{p2} = C_{k2}^4$.

■

Corollary 3.1 (Two Subdomains with Dirichlet outer boundary conditions)**.** *The case of Dirichlet outer boundary conditions can be obtained by letting p_a and p_b go to infinity, which simplifies* (3.27) *to*

$$C = \Re\frac{s(e^{2sL}+1)}{(e^{2sL}-1)} \tag{3.30}$$

and the asymptotic results in Theorem 3.1 simplify accordingly.

For three subdomains, the general substructured iteration matrix becomes

$$T = \begin{bmatrix} 0 & \beta_1^+ & \alpha_1^+ & 0 \\ \beta_2^- & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_2^+ \\ 0 & \alpha_3^- & \beta_3^- & 0 \end{bmatrix},$$

and we obtain for the first time an optimization result for three subdomains:

Theorem 3.2 (Three Subdomain Optimization)**.** *For three subdomains with equal parameters $p_1^+ = p_2^- = p_2^+ = p_3^- = p$, the asymptotically optimized parameter p for small overlap δ and associated convergence factor are*

$$p = 2^{-1/3}C^{2/3}\delta^{-1/3}, \quad \rho = 1 - 2\cdot 2^{1/3}C^{1/3}\delta^{1/3} + \mathcal{O}(\delta^{2/3}), \tag{3.31}$$

where C is a real constant that can be obtained in closed form. If the parameters are different, their asymptotically optimized values for small overlap δ are such that

$$p_1^+, p_2^+, p_2^-, p_3^- \in \{2^{-2/5}C^{2/5}\delta^{-3/5}, 2^{-4/5}C^{4/5}\delta^{-1/5}\},\ p_1^+ \neq p_2^-,\ p_2^+ \neq p_3^-, \tag{3.32}$$

and the associated convergence factor is

$$\rho = 1 - 2 \cdot 2^{-1/5}C^{1/5}\delta^{1/5} + \mathcal{O}(\delta^{2/5}). \tag{3.33}$$

Proof. The characteristic polynomial of the iteration matrix is

$$G(\mu) = \mu^4 - \mu^2(\beta_2^+\beta_3^- + \beta_1^+\beta_2^-) + \beta_1^+\beta_2^-\beta_2^+\beta_3^- - \alpha_1^+\beta_2^+\beta_2^-\alpha_3^-.$$

This biquadratic equation has the roots

$$\mu_1 = \pm\sqrt{\frac{m_1 + \sqrt{m_2}}{2}}, \mu_2 = \pm\sqrt{\frac{m_1 - \sqrt{m_2}}{2}}$$

where

$$m_1 = \beta_2^+\beta_3^- + \beta_1^+\beta_2^-,\ m_2 = (\beta_1^+\beta_2^- - \beta_2^+\beta_3^-)^2 + 4\alpha_1^+\beta_2^+\beta_2^-\alpha_3^-$$

Therefore $\rho(T) = \max\{|\mu_1|, |\mu_2|\}$. Following the same reasoning as in the proof of Theorem 3.1, we observe that the solution equioscillates, and minimizing the maximum asymptotically for δ small then leads to the desired result. ■

Remark 3.1. *Notice that the optimized parameters and the relation between them is the same as in the two-subdomain case, the only difference is the equation whose solution gives the exact value of the constant C. The only difference between a two subdomain optimization and a three subdomain optimization is therefore the constant.*

In order to be able to have a more concrete comparison, we now give a result for Dirichlet boundary conditions at the outer boundaries.

Corollary 3.2 (Three subdomains with Dirichlet outer boundary conditions)**.** *When Dirichlet boundary conditions are used at the end of the computational domain, we obtain for the constant*

$$C = \Re\frac{s(e^{2sL} - e^{sL} + 1)}{e^{2sL} - 1}, \tag{3.34}$$

which is different from the two subdomain constant in (3.30).

Table 3.1: Asymptotic results for four subdomains: $\sigma = \varepsilon = 1, L = 1, p_a = p_b = 1$

	Many parameters							One parameter	
δ	ρ	p_1^+	p_2^-	p_2^+	p_3^-	p_3^+	p_4^-	ρ	p
$1/10^2$	0.5206	13.1269	1.2705	10.1871	0.7748	16.5975	2.1327	0.6202	2.8396
$1/10^3$	0.6708	37.9717	1.4208	42.9379	1.6005	68.1923	2.4896	0.8022	6.0657
$1/10^4$	0.7789	152.9323	2.3266	152.0873	3.1841	161.0389	2.4919	0.9029	13.0412
$1/10^5$	0.8510	651.7536	4.1945	645.0605	4.1519	649.8928	4.1828	0.9537	28.0834

Table 3.2: Asymptotic results for five subdomains : $\sigma = \varepsilon = 1, L = 1, p_a = p_b = 1$

	Many parameters									One parameter	
δ	ρ	p_1^+	p_2^-	p_2^+	p_3^-	p_3^+	p_4^-	p_4^+	p_5^-	ρ	p
$1/10^2$	0.5273	8.5648	1.4619	9.1763	0.8030	9.1398	0.8426	15.5121	2.2499	0.6290	2.6747
$1/10^3$	0.7333	24.6097	0.9209	23.4189	0.4499	37.2200	0.8433	34.8142	0.9181	0.8072	5.7261
$1/10^4$	0.7769	156.0648	2.4223	156.0502	2.4221	161.2036	2.5009	166.3478	2.5941	0.9055	12.3166
$1/10^5$	0.8547	704.4063	4.3378	611.3217	3.7296	611.3217	3.7296	690.8837	4.2116	0.9550	26.5260

For four subdomains, we show in Table 3.1 the numerically optimized parameter values when the overlap δ becomes small.

We observe that again the optimized parameters behave like in Theorem 3.1 and Theorem 3.2 when the overlap δ becomes small. It is in principle possible to continue the asymptotic analysis from two and three subdomains.

Continuing the numerical optimization for five and six subdomains, we get the results in Table 3.2 and Table 3.3, which show again the same asymptotic behavior. We therefore conjecture the following two results for an arbitrary fixed number of subdomains:

1. When all parameters are equal to p, then the asymptotically optimized parameter p for small overlap δ and the associated convergence factor have the same form as for two-subdomains (3.28) in Theorem 3.1, only the constant is different.

2. If all parameters are allowed to be different, the optimized parameters behave for small overlap δ like

$$p_j^+,\, p_{j+1}^- \in \{2^{-2/5}C^{2/5}\delta^{-3/5},\, 2^{-4/5}C^{4/5}\delta^{-1/5}\} \text{ and } p_j^+ \neq p_{j+1}^-\, \forall j = 1.., J-1,$$

   as we have seen in the three subdomain case in Theorem 3.2, and we have again the same asymptotic convergence factor as for two and three subdomains, only the constant is different.

Table 3.3: Asymptotic results for six subdomains: $\sigma = \varepsilon = 1, L = 1, p_a = p_b = 1$

δ	ρ	p_1^+	p_2^-	p_2^+	p_3^-	p_3^+	p_4^-	p_4^+	p_5^-	p_5^+	p_6^-
$1/10^2$	0.5460	10.5283	1.4526	7.7653	1.2124	8.2834	0.6573	7.6445	1.3410	8.0029	0.9586
$1/10^3$	0.7011	30.3314	0.9049	30.3452	1.1096	30.3010	0.9363	30.3458	0.8901	30.1139	1.1307
$1/10^4$	0.7837	145.7147	2.1126	146.4533	2.1231	145.7147	2.1126	149.1802	2.1743	146.7200	2.1909
$1/10^5$	0.8553	660.5326	3.9932	611.9401	3.7012	606.1453	3.6661	606.1144	3.6659	606.0914	3.8534

3.3.2 High frequency vs. low frequency convergence factor

Numerical optimisation for many subdomains has shown that optimised parameters always have the same form (independently of the number of subdomains) and the convergence factor depends on a constant only and the latter depends on the low frequency components. Local maximum values k^* of the convergence factor are also high frequency, that is of the form $k^* = C_k\delta^{-\alpha}$ with $\alpha > 0$. For this reason, all the terms containing $e^{-\lambda(k^*)L}$ will be asymptotically vanishing for small δ and therefore for high frequencies all the terms from Equation (3.24) and Equation (3.25) behave like

$$\alpha_j^{\pm}(k^*) = 0,\ \beta_j^-(k^*) = -\frac{(\lambda - p_j^-)(\lambda + p_{j-1}^-)}{(\lambda{+}p_{j-1}^+)(\lambda{+}p_{j-1}^-)}e^{-\lambda\delta},\ \beta_j^+(k^*) = -\frac{(\lambda - p_j^+)(\lambda + p_{j+1}^+)}{(\lambda{+}p_{j-1}^+)(\lambda{+}p_{j-1}^-)}e^{-\lambda\delta}$$

For this reason, in high frequency regime the general iteration matrix becomes

$$T = \begin{bmatrix} 0 & \beta_1^+ & 0 & 0 & \dots & 0 & 0 \\ \beta_2^- & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \beta_2^+ & \dots & 0 & 0 \\ 0 & 0 & \beta_3^- & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \beta_{J-1}^+ \\ 0 & 0 & \dots & 0 & 0 & \beta_J^- & 0 \end{bmatrix},$$

The eigenvalues of this matrix are given by the pairs $\pm\sqrt{\beta_j^+\beta_{j+1}^-}$, $j = 1, .., J-1$ and therefore the convergence factor is

$$\rho_{hf} = \max_j \left| \frac{\sqrt{(\lambda - p_j^-)(\lambda + p_{j-1}^-)(\lambda - p_j^+)(\lambda + p_{j+1}^+)(\lambda{+}p_{j-1}^+)(\lambda{+}p_{j-1}^-)}}{(\lambda{+}p_{j-1}^+)(\lambda{+}p_{j-1}^-)} \right| e^{-\lambda\delta}$$

We therefore conjecture the following two results for an arbitrary number of subdomains in the high frequency case

1. When all the parameters are equal then

$$\rho_{1,hf} = \left|\frac{\lambda - p}{\lambda + p}\right| e^{-\lambda\delta}$$

2. When the parameters verify p_j^+, $p_{j+1}^- \in \{p_1, p_2\}$ and $p_j^+ \neq p_{j+1}^-$, $j = 1.., J-1$,

$$\rho_{2,hf}^2 = \left|\frac{\lambda - p_1}{\lambda + p_1} \cdot \frac{\lambda - p_2}{\lambda + p_2}\right| e^{-2\lambda\delta}$$

We can now make the study of the convergence factor more systematic no matter the decomposition into subdomains based on the high frequency expression of the convergence factor. In Table 3.4 we have the asymptotic behaviour of the parameters based on the study of the convergence factor in the high frequency regime.

Case	Parameter asymptotics	Local maximum	Convergence factor
One parameter	$p^* = \frac{C_k^2}{2}\delta^{-1/3}$	$k^* = C_k\delta^{-2/3}$	$\rho = 1 - 2C_k\delta^{1/3}$
Two parameters	$\begin{cases} p_1^* = C_{k2}^2\delta^{-3/5} \\ p_2^* = C_{k2}^4\delta^{-1/5} \end{cases}$	$\begin{cases} k_1^* = C_{k2}^3\delta^{-2/5} \\ k_2^* = C_{k2}\delta^{-4/5} \end{cases}$	$\rho = 1 - 2C_{k2}\delta^{1/5}$

Table 3.4: Parameter asymptotics in the high frequency regime

For example in the one parameter case, the optimal parameter $p^* = C_p\delta^{-1/3}$ and as seen from the numerical optimisation, a local maximum can be found in $k^* = C_k\delta^{-2/3}$ and the relation between them, based on the series development of the high frequency component of the convergence factor (which doesn't depend on the number of subdomains) is $C_p = \frac{C_k^2}{2}$. Also the maximum of the convergence factor is given by

$$\rho(k^*) = 1 - 2C_k\delta^{1/3} + \mathcal{O}(\delta^{2/3}). \tag{3.35}$$

On the other side, in the two parameter case, the optimal parameters verify $p_1^*, p_2^* \in \{C_{k2}^2\delta^{-3/5}, C_{k2}^4\delta^{-1/5}\}$ the maximum of the convergence factor is given by

$$\rho(k^*) = 1 - 2C_{k2}\delta^{1/5} + \mathcal{O}(\delta^{2/5}). \tag{3.36}$$

The constants will depend now on the low frequency convergence factor say $\rho(k_{min})$ where k_{min} can be chosen equal to 0 and this will be dependent on the number of subdomains. In order to compute the low frequency component of the convergence factor we will first give the asymptotic values of Equation (3.24) and Equation (3.25) for one and two parameters and to simplify the computation we suppose Dirichlet boundary conditions at the boundaries of the global domain, that is consider the limits of p_a and p_b to infinity.

1. For one optimised parameter $p = C_p\delta^{-1/3} = \frac{C_k^2}{2}\delta^{-1/3}$

$$\begin{aligned}\alpha_j^+(0) = \alpha_j^-(0) &= \frac{4se^{-sL}}{C_p(1-e^{-2sL})}\delta^{1/3} = \frac{8se^{-sL}}{C_k^2(1-e^{-2sL})}\delta^{1/3} := \tilde{a},\\ \beta_j^+(0) = \beta_j^-(0) &= 1 - \frac{2s(e^{-2sL}+1)}{C_p(1-e^{-2sL})}\delta^{1/3} = 1 - \frac{4s(e^{-2sL}+1)}{C_k^2(1-e^{-2sL})}\delta^{1/3} =: \tilde{b}\end{aligned}$$

leading to the low frequency iteration matrix

$$T_{lf,1par} = \begin{bmatrix} 0 & \tilde{b} & \tilde{a} & 0 & \dots & 0 & 0 \\ \tilde{b} & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \tilde{b} & \dots & 0 & 0 \\ 0 & \tilde{a} & \tilde{b} & 0 & \dots & \tilde{a} & 0 \\ 0 & 0 & \dots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \tilde{b} \\ 0 & 0 & \dots & 0 & \tilde{a} & \tilde{b} & 0 \end{bmatrix}. \tag{3.37}$$

By computing the spectral radius of this matrix for $J = 2, 3, 4$ subdomains we get the value for a small δ and then by equating the dominant terms with Equation (3.35) we obtain the constants

$$\begin{aligned}\rho_2(k^*) &= \rho_2(0) = 1 - \frac{1}{C_k^2}\Re\frac{4s(e^{2sL}+1)}{(e^{2sL}-1)}\delta^{1/3}\\ &\Rightarrow \quad C_k = 2^{1/3}\left(\Re\frac{s(e^{2sL}+1)}{(e^{2sL}-1)}\right)^{1/3}\\ \rho_3(k^*) &= \rho_3(0) = 1 - \frac{1}{C_k^2}\Re\frac{4s(e^{2sL}+1-e^{sL})}{(e^{2sL}-1)}\delta^{1/3}\\ &\Rightarrow \quad C_k = 2^{1/3}\left(\Re\frac{s(e^{2sL}+1-e^{sL})}{(e^{2sL}-1)}\right)^{1/3}\\ \rho_4(k^*) &= \rho_4(0) = 1 - \frac{1}{C_k^2}\Re\frac{4s(e^{2sL}+1-\frac{\sqrt{5}-1}{2}e^{sL})}{(e^{2sL}-1)}\delta^{1/3}\\ &\Rightarrow \quad C_k = 2^{1/3}\left(\Re\frac{s(e^{2sL}+1-\frac{\sqrt{5}-1}{2}e^{sL})}{(e^{2sL}-1)}\right)^{1/3}\end{aligned} \tag{3.38}$$

We note that parameters and the convergence factor for a fixed number of domains depend only on one constant, which vary as we increase the number of domains and can be summarised as follows

$$
\begin{aligned}
C_2 &= \Re\frac{s(e^{2sL}+1)}{(e^{2sL}-1)} && \text{2 subdomains,}\\
C_3 &= \Re\frac{s(e^{2sL}+1-e^{sL})}{(e^{2sL}-1)} && \text{3 subdomains}\\
C_4 &= \Re\frac{s(e^{2sL}+1-\frac{\sqrt{5}-1}{2}e^{sL})}{(e^{2sL}-1)} && \text{4 subdomains}
\end{aligned}
\tag{3.39}
$$

2. For two optimised parameter $p_1, p_2 \in \{C_{p1}\delta^{-1/5}, C_{p2}\delta^{-3/5}\} = \{C_{k2}^2\delta^{-1/5}, C_{k2}^4\delta^{-1/5}\}$

$$
\begin{aligned}
\alpha_j^+(0) = \alpha_j^-(0) &= \frac{2se^{-sL}}{C_{p2}(1-e^{-2sL})}\delta^{1/5} = \frac{2se^{-sL}}{C_{k2}^4(1-e^{-2sL})}\delta^{1/5} := \tilde{a},\\
\beta_j^+(0), \beta_{j+1}^-(0) &\in \{\delta^{2/5}C_{k2}^2\tilde{b}, \frac{1}{\delta^{2/5}C_{k2}^2}\tilde{b}\},\ \tilde{b} = 1 - \frac{s(e^{-2sL}+1)}{C_{k2}^4(1-e^{-2sL})}\delta^{1/5}
\end{aligned}
$$

leading to the low frequency iteration matrix

$$
T_{lf,2par} = \begin{bmatrix}
0 & \tilde{b}_+ & \tilde{a} & 0 & \dots & 0 & 0\\
\tilde{b}_- & 0 & 0 & 0 & \dots & 0 & 0\\
0 & 0 & 0 & \tilde{b}_+ & \dots & 0 & 0\\
0 & \tilde{a} & \tilde{b}_- & 0 & \dots & \tilde{a} & 0\\
0 & 0 & \dots & \ddots & \ddots & 0 & 0\\
0 & 0 & \dots & 0 & 0 & 0 & \tilde{b}_+\\
0 & 0 & \dots & 0 & \tilde{a} & \tilde{b}_- & 0
\end{bmatrix}.
\tag{3.40}
$$

where in fact the couple $\tilde{b}_+ \neq \tilde{b}_-$ can vary along the diagonal but still lay in the set $\{\delta^{2/5}C_{k2}^2\tilde{b}, \frac{1}{\delta^{2/5}C_{k2}^2}\tilde{b}\}$ which won't change the eigenvalues of the matrix. By computing the spectral radius of this matrix for $J = 2, 3, 4$ subdomains we get the value for a small δ and then by equating the dominant terms with Equation (3.36) with we obtain the constants

(3.41)

$$
\begin{aligned}
\rho_2(k^*) &= \rho_2(0) = 1 - \frac{1}{C_{k2}^4}\Re\frac{s(e^{2sL}+1)}{(e^{2sL}-1)}\delta^{1/5} \Rightarrow C_{k2} = 2^{-1/5}C_2^{1/5}\\
\rho_3(k^*) &= \rho_3(0) = 1 - \frac{1}{C_{k2}^4}\Re\frac{s(e^{2sL}+1-e^{sL})}{(e^{2sL}-1)}\delta^{1/5} \Rightarrow C_{k2} = 2^{-1/5}C_3^{1/5}\\
\rho_4(k^*) &= \rho_4(0) = 1 - \frac{1}{C_{k2}^4}\Re\frac{s(e^{2sL}+1-\frac{\sqrt{5}-1}{2}e^{sL})}{(e^{2sL}-1)}\delta^{1/5} \Rightarrow C_{k2} = 2^{-1/5}C_4^{1/5}
\end{aligned}
$$

where we note again the dependence of the constants given in Equation (3.39).

Analytical formulae for constants in the general case are out of reach since we cannot compute eigenvalues of matrices (3.37) and (3.40) for more than 4 subdomains. However, asymptotic formulae for these eigenvalues are computable by writing the asymptotic development for small δ of the characteristic polynomial of (3.37) and equating the leading term to 0. Experiments with Maple show that this is doable in all the cases when $N \geq 4$ and the following result can be conjectured (although no proof is available)

$$C_N = \Re \frac{s(e^{2sL} - 2\cos\left(\frac{\pi}{N+1}\right) e^{sL} + 1)}{(e^{2sL} - 1)}, \; N \geq 4. \tag{3.42}$$

In the sequel we will see that this result is fully consistant with the one obtained by using the limiting spectrum approach. In this case a bound on the spectral radius can be obtained and this can be minimised by using the same technique.

3.3.3 Optimization for many subdomains

In order to obtain a theoretical result for many subdomains, we use the technique of limiting spectra to derive a bound on the spectral radius which we can then minimize. To do so, we must however assume that the outer Robin boundary conditions use the same optimized parameter as at the interfaces, in order to have the Toeplitz structure needed for the limiting spectrum approach.

Theorem 3.3 (Many Subdomain Optimization). *With all Robin parameters equal, $p_j^- = p_j^+ = p$, and by equioscillation of the solution, the convergence factor of the OSM satisfies the bound*

$$\rho = \lim_{N \to +\infty} \rho(T_{2d}^{OS}) \leq \max\left\{ |\alpha - \beta|, |\alpha + \beta| \right\} < 1,$$

where

$$\alpha = \frac{(\lambda+p)^2 e^{\lambda\delta} - (\lambda-p)^2 e^{-\lambda\delta}}{(\lambda+p)^2 e^{\lambda(L+\delta)} - (\lambda-p)^2 e^{-\lambda(L+\delta)}}, \; \beta = \frac{(\lambda-p)(\lambda+p)(e^{-\lambda L} - e^{\lambda L})}{(\lambda+p)(\lambda+p)e^{\lambda(L+\delta)} - (\lambda-p)(\lambda-p)e^{-\lambda(L+\delta)}}.$$

The asymptotically optimized parameter and associated convergence factor are

$$p = 2^{-1/3} C^{2/3} \delta^{-1/3}, \quad \rho = 1 - 2 \cdot 2^{1/3} C^{1/3} \delta^{1/3} + \mathcal{O}(\delta^{2/3}) \tag{3.43}$$

with the constant

$$C := \Re \frac{s(1-e^{-sL})}{1+e^{-sL}}.$$

If we allow two-sided Robin parameters, $p_j^- = p^-$ and $p_j^+ = p^+$, the OSM convergence factor satisfies the bound

$$\rho = \lim_{N\to+\infty} \rho(T_{2d}^{OS}) \leq \max\left\{ \left|\alpha - \sqrt{\beta_-\beta_+}\right|, \left|\alpha + \sqrt{\beta_-\beta_+}\right| \right\} < 1,$$

where

$$\alpha = \frac{(\lambda+p^+)(\lambda+p^-)e^{\lambda\delta} - (\lambda-p^+)(\lambda-p^-)e^{-\lambda\delta}}{D}, \; \beta^{\pm} = \frac{(\lambda^2-(p^{\mp})^2)(e^{-\lambda L}-e^{\lambda L})}{D},$$

with

$$D = (\lambda+p^+)(\lambda+p^-)e^{\lambda(L+\delta)} - (\lambda-p^+)(\lambda-p^-)e^{-\lambda(L+\delta)}.$$

The asymptotically optimized parameter choice $p^- \neq p^+$ and the associated convergence factor are

$$p^-, p^+ \in \left\{ C_R{}^{2/5}\delta^{-3/5}, \, C^{4/5}\delta^{-1/5} \right\}, \quad \rho = 1 - 2C_R^{1/5}\delta^{1/5} + \mathcal{O}(\delta^{2/5}),$$

with the same constant

$$C_R := \Re \frac{s(1-e^{-sL})}{1+e^{-sL}}$$

as for one parameter.

Proof. As in the case of two and three subdomains, we observe equioscillation and asymptotically that $p = C_p\delta^{-1/3}$, $\rho = 1 - C_R\delta^{1/3} + \mathcal{O}(\delta^{2/3})$ and the convergence factor has a local maximum at the point $\tilde{k}^* = C_k\delta^{-2/3}$. By expanding for small δ, the derivative $\frac{\partial\rho}{\partial k}(\tilde{k}^*)$ needs to have a vanishing leading order term, which leads to $C_p = \frac{C_k^2}{2}$. Expanding the convergence factor at the maximum point $\tilde{k}^*$ gives $\rho(\tilde{k}^*) = \rho(C_k\delta^{-2/3}) = 1 - 2C_k\delta^{1/3} + \mathcal{O}(\delta^{2/3})$, and hence $C_R = 2C_k$. Equating now $\rho(0) = \rho(\tilde{k}^*)$ determines uniquely C_k and then $C_p = \sqrt{C_k/2}$ giving (3.43). By following the same lines as for two and three subdomains, we also get the asymptotic result in the case of two different parameters. ■

Remark 3.2. *We notice from the previous result that* $\lim_{N\to\infty} C_{N,R} = C_R$ *meaning that when the number of subdomain is large we can use the constant from the limiting spectrum approach. Also the constants $C_{N,R}$ are asymptotic values obtained in the case where Dirichlet boundary conditions are used at the boundaries of the global domain*

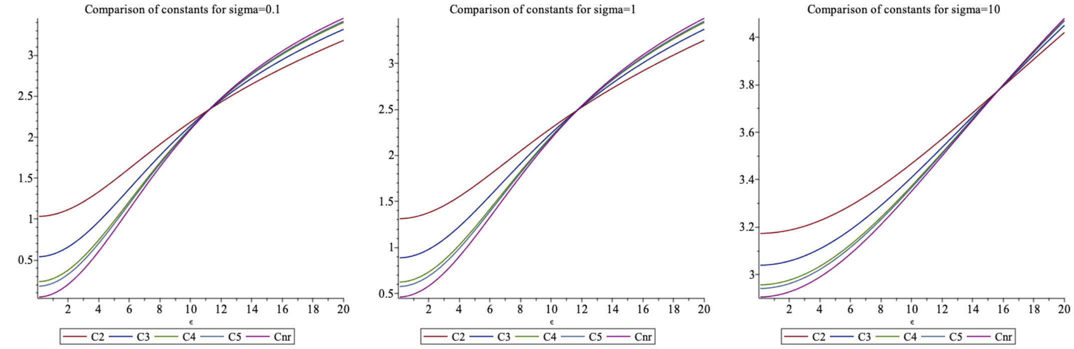

Figure 3.8: Optimised constants for different subdomains for a fixed σ and $L = 1$ as a function of ε

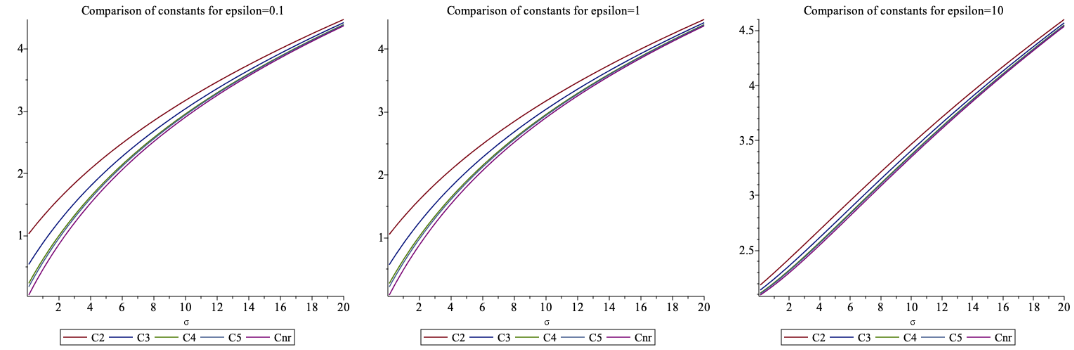

Figure 3.9: Optimised constants for different subdomains for a fixed ε and $L = 1$ as a function of σ

whereas in the limiting spectrum approach we have Robin conditions. In Figures 3.8 and 3.9 we see how these constants evolve for different number of subdomains and how they approach the limiting value as the number of subdomains increases and we notice that as the parameters σ and ε increase, there is no much difference with respect to the two-subdomain case.

We can therefore safely conclude that for the magnetotelluric approximation of Maxwell's equations, which contains the important Laplace and screened Laplace equation as special cases, it is sufficient to optimize transmission conditions for a simple two subdomain decomposition in order to obtain good transmission conditions also for the case of many subdomains, a new result that was not known so far.

3.4 Second order optimised transmission conditions

We can design even better optimised methods if we replace the coefficient $p_j^{\pm}$ (3.22) by second order differential operators along the interface. In this case, by replacing the local solution written as a Fourier series we obtain for Fourier coefficients the equations

$$\begin{array}{rcll} \partial_{xx} v_j^n - (\tilde{k}^2 + \sigma - i\varepsilon) v_j^n & = & 0 & x \in (a_j, b_j), \\ -\partial_x v_j^n + (p_j^- + \tilde{k}^2 q_j^-) v_j^n & = & -\partial_x v_{j-1}^{n-1} + (p_j^- + \tilde{k}^2 q_j^-) v_{j-1}^{n-1} & \text{at } x = a_j, \\ \partial_x v_j^n + (p_j^+ + \tilde{k}^2 q_j^+) v_j^n & = & \partial_x v_{j+1}^{n-1} + (p_j^+ + \tilde{k}^2 q_j^+) v_{j+1}^{n-1} & \text{at } x = b_j. \end{array} \tag{3.44}$$

For these equations the same reasoning from the previous sections will hold including the final form of the Toeplitz iteration matrix. The only difference holds in the fact that this matrix depends now on two sets of parameters $T_2(\tilde{k}, p_j^{\pm}, q_j^{\pm}) := T(\tilde{k}, p_j^{\pm} + \tilde{k}^2 q_j^{\pm})$ and we need to solve now the following min-max problem

$$\min_{p_j^{\pm}, q_j^{\pm}} \max_{\tilde{k} \in K} |\rho(T_2(\tilde{k}, p_j^{\pm}, q_j^{\pm}))|$$

We will follow the same path as in the case of zero order transmission conditions by performing the optimisation in the case of two and three subdomains.

Theorem 3.4 (Two Subdomain Optimization). *Let $s := \sqrt{\sigma - i\varepsilon}$, where the complex square root is taken with a positive real part, and let C be the real constant*

$$C := \Re \frac{s((p_b + s)(p_a + s) - (s - p_b)(s - p_a)e^{-4sL})}{((s - p_a)e^{-2sL} + s + p_a)((s - p_b)e^{-2sL} + s + p_b)}. \tag{3.45}$$

Then for two subdomains with $p_1^+ = p_2^- =: p$, $q_1^+ = q_2^- =: q$ and $\tilde{k}_{\min} = 0$, by equioscillation of the solution , the asymptotically optimized parameters p and q for small overlap δ are

$$p = 2^{-3/5} C^{4/5} \delta^{-1/5}, \; q = 2^{-1/5} C^{-2/5} \delta^{3/5} \tag{3.46}$$

and the associated convergence factor is

$$\rho = 1 - 2 \cdot 2^{3/5} C^{1/5} \delta^{1/5} + \mathcal{O}(\delta^{2/5}).$$

If $p_1^+ \neq p_2^-$, $q_1^+ \neq q_2^-$, and $\tilde{k}_{min} = 0$, the asymptotically optimized parameters for small *overlap δ are*

(3.47)

$$p_1^+ = 2^{-8/9}C^{8/9}\delta^{-1/9},\ q_1^+ = 2^{2/9}C^{-2/9}\delta^{7/9},\ p_2^- = 2^{-2/3}C^{2/3}\delta^{-1/3},\ q_2^- = 2^{4/9}C^{-4/9}\delta^{5/9}$$

and the associated convergence factor is

$$\rho = 1 - 2\cdot 2^{-1/9}C^{1/9}\delta^{1/9} + \mathcal{O}(\delta^{2/9}).$$

Proof. We obtain that the solution of the min-max problem equioscillates. Even with an identical condition we have now two parameters and in this case we have two equioscillations, $\rho(0) = \rho(\tilde{k}_1^*) = \rho(\tilde{k}_2^*)$, where $\tilde{k}_j^*$ are two interior local maxima, and asymptotically $p = C_p\delta^{-1/5}$, $q = C_q\delta^{3/5}$, $\rho = 1 - C_R\delta^{1/5} + \mathcal{O}(\delta^{2/5})$, $\tilde{k}_1^* = C_{k1}\delta^{-2/5}$ and $\tilde{k}_2^* = C_{k2}\delta^{-4/5}$. By expanding for δ small, and setting the leading terms in the derivatives $\frac{\partial\rho}{\partial k}(\tilde{k}_{1,2}^*)$ to zero, and we get $C_p = \frac{2C_{k1}^2}{C_{k2}^2}$, $C_q = \frac{2}{C_{k2}^2}$. Expanding the maxima leads to $\rho(\tilde{k}_1^*) = \rho(C_{k1}\delta^{-2/5}) = 1 - 8\frac{C_{k1}}{C_{k2}^2}\delta^{1/5} + \mathcal{O}(\delta^{2/5})$ and $\rho(\tilde{k}_2^*) = \rho(C_{k2}\delta^{-4/5}) = 1 - 2C_{k_2}\delta^{1/5} + \mathcal{O}(\delta^{2/5})$ and equating $\rho(\tilde{k}_1^*) = \rho(\tilde{k}_2^*)$ we get $C_{k1} = \frac{C_{k2}^3}{4}$ and $C_R = 2C_{k2}$. Finally equating $\rho(0) = \rho(\tilde{k}_2^*)$ asymptotically determines uniquely $C_{k2} = 2^{3/5}C^{1/5}$ and then $C_{k1} = \frac{C_{k2}^3}{4}$ and $C_p = \frac{C_{k2}^4}{8}$, $C_q = \frac{2}{C_{k2}^2}$.

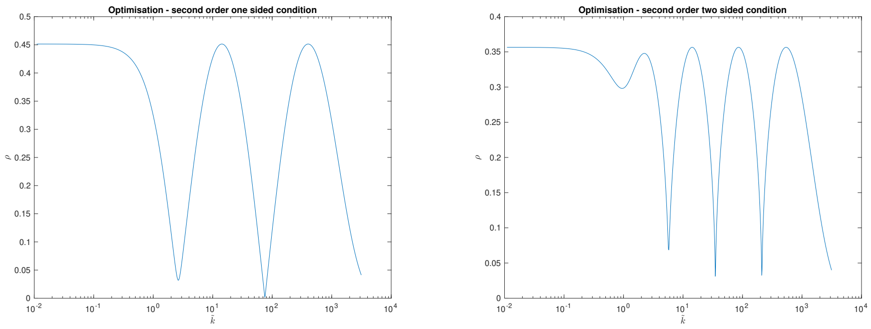

Figure 3.10: Equioscillation in numerical optimisation with one sided and two sided second order optimised conditions

In the case of four different parameters, we have four equioscillations, $\rho(0) = \rho(\tilde{k}_1^*) = \rho(\tilde{k}_2^*) = \rho(\tilde{k}_3^*) = \rho(\tilde{k}_4^*)$, where $\tilde{k}_j^*$ are four interior local maxima, and asymptotically $p_1 = C_{p1}\delta^{-1/9}$, $q_1 = C_{q1}\delta^{7/9}$, $p_2 = C_{p2}\delta^{-3/9}$, $q_2 = C_{q2}\delta^{5/9}$ $\rho = 1 - C_R\delta^{1/9} + \mathcal{O}(\delta^{2/9})$, $\tilde{k}_1^* = C_{k1}\delta^{-2/9}$, $\tilde{k}_2^* = C_{k2}\delta^{-4/9}$, $\tilde{k}_3^* = C_{k3}\delta^{-6/9}$, $\tilde{k}_4^* = C_{k4}\delta^{-8/9}$. By expanding for δ small, and setting the leading terms in the derivatives $\frac{\partial\rho}{\partial k}(\tilde{k}_{1,2,3,4}^*)$ to zero, we get

$$C_{p1} = \frac{C_{k1}^2\cdot C_{k3}^2}{C_{k2}^2\cdot C_{k4}^2},\ C_{p2} = \frac{C_{k2}^2\cdot C_{k4}^2}{C_{k3}^2},\ C_{q1} = \frac{1}{C_{k4}^2},\ ,C_{q2} = \frac{C_{k4}^2}{C_{k3}^2}.$$

Expanding the maxima leads to

$$\begin{aligned}
\rho(\tilde{k}_1^*) &= \rho(C_{k1}\delta^{-2/9}) = 1 - 2\frac{C_{k1}\cdot C_{k3}^2}{C_{k2}^2\cdot C_{k4}^2}\delta^{1/9} + \mathcal{O}(\delta^{2/9}),\\
\rho(\tilde{k}_2^*) &= \rho(C_{k2}\delta^{-4/9}) = 1 - 2\frac{C_{k2}\cdot C_{k4}^2}{C_{k3}^2\cdot C_{k4}^2}\delta^{1/9} + \mathcal{O}(\delta^{2/9}),\\
\rho(\tilde{k}_3^*) &= \rho(C_{k3}\delta^{-6/9}) = 1 - 2\frac{C_{k3}}{C_{k4}^2}\delta^{1/9} + \mathcal{O}(\delta^{2/9}),\\
\rho(\tilde{k}_4^*) &= \rho(C_{k4}\delta^{-8/9}) = 1 - 2C_{k4}\delta^{1/9} + \mathcal{O}(\delta^{2/9}).
\end{aligned}$$

Equating now $\rho(\tilde{k}_1^*) = \rho(\tilde{k}_2^*) = \rho(\tilde{k}_3^*) = \rho(\tilde{k}_4^*)$ we get $C_{k1} = C_{k4}^7, C_{k2} = C_{k4}^5, C_{k3} = C_{k4}^3$ and $C_R = 2C_{k4}$. Finally equating $\rho(0) = \rho(\tilde{k}_4^*)$ asymptotically determines uniquely $C_{k4} = 2^{-1/9}C^{1/9}$ with C given in (3.45) and the other constants are determined accordingly. ■

A similar asymptotic analysis can be performed for three subdomains with similar conclusions from which we can infer the following.

Remark 3.3. *Like in the case of 0th order conditions we can conjecture the following two results for an arbitrary fixed number of subdomains:*

1. *When all pairs (p_j^+, q_j^+) and (p_j^-, q_j^-) of parameters are equal to (p, q), then the asymptotically optimized parameter p for small overlap δ and the associated convergence factor have the same form as for two-subdomains* (3.46) *in Theorem 3.4, only the constant is different.*

2. *If all parameters are allowed to be different, the optimized parameters behave for small overlap δ like*

   $$(p_j^+, q_j^+), (p_{j+1}^-, q_{j+1}^-) \in \{(2^{-8/9}C^{8/9}\delta^{-1/9}, 2^{2/9}C^{-2/9}\delta^{7/9}), (2^{-2/3}C^{2/3}\delta^{-1/3}, 2^{4/9}C^{-4/9}\delta^{5/9})\}$$

   and $(p_j^+, q_j^+) \neq (p_{j+1}^-, q_{j+1}^-)\, \forall j = 1.., J-1$.

3.4.1 High frequency and low frequency analysis

We can make the study of the convergence factor more systematic by performing the low frequency vs. high frequency analysis of the convergence factor like in Section 3.3.2. The equivalent of the summary from Table 3.4 of the optimisation parameters in high frequency regime reads.

Case	Parameter asymptotics	Local maximum	Convergence factor
One sided	$\begin{cases} p^* = \frac{C_{k2}^4}{8}\delta^{-1/5}, \\ q^* = \frac{2}{C_{k2}^2}\delta^{3/5} \end{cases}$	$\begin{cases} k_1^* = \frac{C_{k2}^3}{4}\delta^{-2/5}, \\ k_2^* = C_{k2}\delta^{-4/5} \end{cases}$	$\rho = 1 - 2C_{k2}\delta^{1/5}$
Two sided	$\begin{cases} p_1^* = C_{k4}^8\delta^{-1/9}, \\ q_1^* = \frac{1}{C_{k4}^2}\delta^{7/9} \\ p_2^* = C_{k4}^6\delta^{-3/9}, \\ q_2^* = \frac{1}{C_{k4}^4}\delta^{5/9} \end{cases}$	$\begin{cases} k_1^* = C_{k4}^7\delta^{-2/9} \\ k_2^* = C_{k4}^5\delta^{-4/9} \\ k_3^* = C_{k4}^3\delta^{-6/9} \\ k_4^* = C_{k4}\delta^{-8/9} \end{cases}$	$\rho = 1 - 2C_{k4}\delta^{1/9}$

Table 3.5: Parameter asymptotics for second order transmission conditions in the high frequency regime for second order transmission conditions

Note that in the case of second order conditions even if the same conditions are used on both sides of the interface, we still need two parameters per interface. We will therefore call these conditions one-sided and distinguish them from the two sided where four parameters are needed.

1. For one sided conditions $p^* = \frac{C_{k2}^4}{8}\delta^{-1/5}, q^* = \frac{2}{C_{k2}^2}\delta^{3/5}$ the asymptotic values of $\alpha^{\pm}(0)$ and $\beta^{\pm}(0)$

$$\alpha_j^+(0) = \alpha_j^-(0) = \frac{4se^{-sL}}{C_p(1-e^{-2sL})}\delta^{1/5} = \frac{32se^{-sL}}{C_k^5(1-e^{-2sL})}\delta^{1/5} := \tilde{a},$$
$$\beta_j^+(0) = \beta_j^-(0) = 1 - \frac{2s(e^{-2sL}+1)}{C_p(1-e^{-2sL})}\delta^{1/5} = 1 - \frac{16s(e^{-2sL}+1)}{C_k^4(1-e^{-2sL})}\delta^{1/5} =: \tilde{b}$$

leading to the low frequency iteration matrix of the same form like in Equation (3.37). By computing the spectral radius of this matrix for $J = 2, 3, 4$ subdomains we get the value for a small δ and then by equating the dominant terms with the high frequency components we obtain the constants

$$
\begin{aligned}
\rho_2(k^*) &= \rho_2(0) = 1 - \frac{16s(e^{-2sL}+1)}{C_k^4(1-e^{-2sL})}\delta^{1/5}\\
&\Rightarrow\ C_k = 2^{3/5}\left(\Re\frac{s(e^{2sL}+1)}{(e^{2sL}-1)}\right)^{1/5}\\
\rho_3(k^*) &= \rho_3(0) = 1 - \frac{1}{C_k^4}\Re\frac{16s(e^{2sL}+1-e^{sL})}{(e^{2sL}-1)}\delta^{1/5}\\
&\Rightarrow\ C_k = 2^{3/5}\left(\Re\frac{s(e^{2sL}+1-e^{sL})}{(e^{2sL}-1)}\right)^{1/5}\\
\rho_4(k^*) &= \rho_4(0) = 1 - \frac{1}{C_k^4}\Re\frac{16s(e^{2sL}+1-\frac{\sqrt{5}-1}{2}e^{sL})}{(e^{2sL}-1)}\delta^{1/5}\\
&\Rightarrow\ C_k = 2^{3/5}\left(\Re\frac{s(e^{2sL}+1-\frac{\sqrt{5}-1}{2}e^{sL})}{(e^{2sL}-1)}\right)^{1/5}
\end{aligned}
\tag{3.48}
$$

We can note again that parameters and the convergence factor for a fixed number of domains depend only on one constant, which vary as we increase the number of domains and this constant is given in each case in Equation (3.39).

2. For the two-sided conditions where

$$p_1^* = C_{k4}^8\delta^{-1/9}, q_1^* = \frac{1}{C_{k4}^2}\delta^{7/9}, p_2^* = C_{k4}^6\delta^{-3/9}, q_2^* = \frac{1}{C_{k4}^4}\delta^{5/9}$$

the asymptotic values of $\alpha^{\pm}(0)$ and $\beta^{\pm}(0)$ are

$$\alpha_j^+(0) = \alpha_j^-(0) = \frac{2se^{-sL}}{C_{p2}(1-e^{-2sL})}\delta^{1/9} = \frac{2se^{-sL}}{C_{k4}^4(1-e^{-2sL})}\delta^{1/9} := \tilde{a},$$
$$\beta_j^+(0), \beta_{j+1}^-(0) \in \{\delta^{2/9}C_{k4}^2\tilde{b}, \frac{1}{\delta^{2/9}C_{k4}^2}\tilde{b}\},\ \tilde{b} = 1 - \frac{s(e^{-2sL}+1)}{C_{k4}^8(1-e^{-2sL})}\delta^{1/9}$$

leading to the low frequency iteration matrix of the same form like in Equation (3.40) where in fact the couple $\tilde{b}_+ \neq \tilde{b}_-$ can vary along the diagonal but still lay in the set $\{\delta^{2/9}C_{k4}^2\tilde{b}, \frac{1}{\delta^{2/9}C_{k4}^2}\tilde{b}\}$ which won't change the eigenvalues of the matrix. By computing the spectral radius of this matrix for $J = 2, 3, 4$ subdomains we get the value for a small δ and then by equating the dominant terms with the

high frequency components with we obtain the constants
(3.49)

$$\begin{array}{rcl}\rho_2(k^*) & = & \rho_2(0) = 1 - \dfrac{1}{C_{k4}^8}\Re\dfrac{s(e^{2sL}+1)}{(e^{2sL}-1)}\delta^{1/9} \Rightarrow C_{k4} = 2^{-1/9}C_2^{1/9} \\ \rho_3(k^*) & = & \rho_3(0) = 1 - \dfrac{1}{C_{k2}^8}\Re\dfrac{s(e^{2sL}+1-e^{sL})}{(e^{2sL}-1)}\delta^{1/5} \Rightarrow C_{k4} = 2^{-1/9}C_3^{1/9} \\ \rho_4(k^*) & = & \rho_4(0) = 1 - \dfrac{1}{C_{k2}^8}\Re\dfrac{s(e^{2sL}+1-\frac{\sqrt{5}-1}{2}e^{sL})}{(e^{2sL}-1)}\delta^{1/5} \Rightarrow C_{k4} = 2^{-1/9}C_4^{1/9}\end{array}$$

where we note again the dependence of the constants given in Equation (3.39).

Note that since we have the same constants as in the case of 0th order conditions and the same conclusions hold.

3.5 Conclusions

In this chapter we have analysed the iterative version of Schwarz method by using the idea of limiting spectrum for block Toeplitz matrices. Based on this, we have designed better transmission conditions for the optimised versions of Schwarz algorithms with overlap, obtained closed formulae for the parameters involved in these transmission conditions and asymptotic theoretical results (for a small value of the overlapping parameter) on the predicted convergence factor. It is the first time this kind of results have been obtained for decompositions into many subdomains and complex versions of the diffusion problems as in most of the works from the literature, only the decomposition into two subdomains are considered. These conditions can be one sided and two-sided, zero-th and second order, increasing the complexity usually leads to a better convergence. Numerical implementations of these new algorithms will be shown in the next chapter.

Chapter 4

Numerical assessment of optimised Schwarz methods

In this chapter we focus on the numerical assessment of optimised Schwarz algorithms introduced in the previous chapter when using zero and second order transmission conditions.

4.1 Optimised Schwarz method as a solver

For our tests, first we focus on a decomposition into two overlapping domains, that can be uniform (two rectangles) or a more general decomposition using METIS as shown in Figure 4.1. In the first series of tests we consider one sided Robin interface conditions (i.e. depending only on one parameter) which can be zero-th and second order and we increase locally the number of degrees of freedom (denoted by *nloc* in the tables) which leads to an decreasing value of the mesh size h. We report the iteration number in order to achieve a relative quadratic L_2 norm of the error of 10^{-6}.

In order to fully understand the benefits of the optimised transmission conditions we start by performing a few numerical simulation with the RAS method as an iterative solver. In Figure 4.2 we see that the convergence deteriorates when the mesh size is decreased and the iteration count increases considerably.

In order to quantify this increase we plot the asymptotic dependence of the iteration count with respect to the mesh size in Figure 4.3 and we notice that the iteration count behaves like h^{-1}, hence by refining the mesh size by a factor of 2, the iteration count

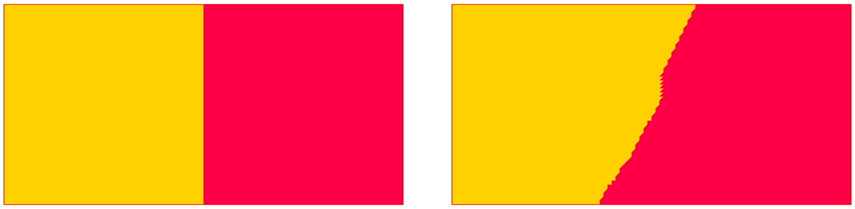

Figure 4.1: Decomposition into two subdomains -uniform and METIS decomposition

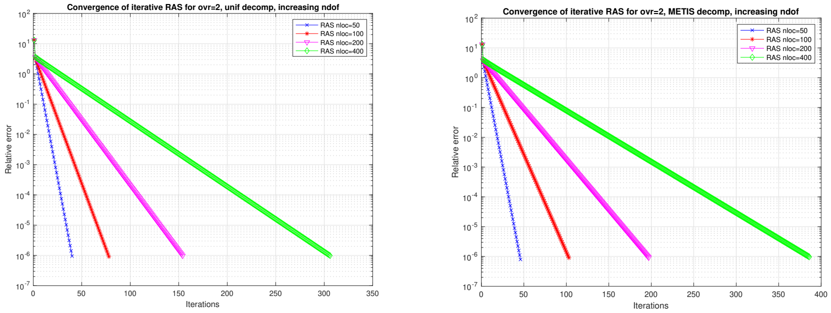

Figure 4.2: Convergence of the RAS algorithm for $\sigma = 1, \varepsilon = 1$ uniform (left) and METIS (right) decomposition

doubles, which is quite a strong dependence.

We move on now to the iterative version of the ORAS algorithm (Optimised Restricted Additive Schwarz). We will see that with the same computational complexity, i.e. by only changing the interface transmission conditions, the algorithm will converge much faster.

We consider three case scenarios for a fixed value of $\sigma = 1$ and different values of $\varepsilon = 0.1, 1, 10$. Results are reported in Tables 4.1, 4.2 and 4.3. We notice that there is slight difference between the uniform and METIS decomposition but that the overall iteration count is far smaller than in the case of the RAS method for the same kind of problem. (we have chosen to show iteration counts for the RAS method only in one case in order to illustrate the stark difference with respect to ORAS). Secondly, when imposing second order transmission conditions in the case of a uniform decomposition we can notice a further decrease in the iteration count whereas in the case of METIS decomposition, the number of iterations might even increase. This can be explained by the presence of tangential derivatives in the interface conditions which are not well approximated in the case of jagged interfaces.

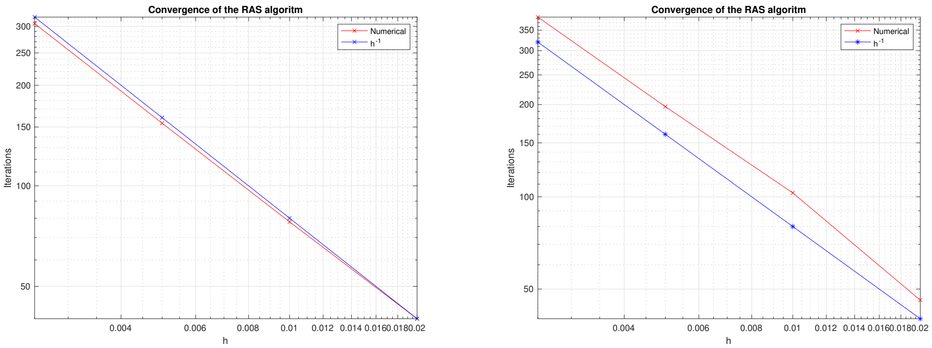

Figure 4.3: Convergence of the RAS algorithm for $\sigma = 1, \varepsilon = 1$ uniform (left) and METIS (right) decomposition for overlap $\delta = 2h$.

		Zero order-one sided		Second order-one sided	
n_{loc}	h	METIS	UNIFORM	METIS	UNIFORM
50	$\frac{1}{49}$	12	12	12	7
100	$\frac{1}{99}$	16	14	18	9
200	$\frac{1}{199}$	19	18	24	11
400	$\frac{1}{399}$	24	22	29	13

Table 4.1: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 0.1$.

		Zero order-one sided		Second order-one sided		RAS	
n_{loc}	h	METIS	UNIFORM	METIS	UNIFORM	METIS	UNIFORM
50	$\frac{1}{49}$	12	12	11	7	46	40
100	$\frac{1}{99}$	16	14	18	9	103	78
200	$\frac{1}{199}$	19	18	23	11	197	154
400	$\frac{1}{399}$	24	22	29	13	386	306

Table 4.2: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 1$.

		Zero order-one sided		Second order-one sided	
n_{loc}	h	METIS	UNIFORM	METIS	UNIFORM
50	$\frac{1}{49}$	10	10	10	9
100	$\frac{1}{99}$	13	12	15	11
200	$\frac{1}{199}$	16	15	19	11
400	$\frac{1}{399}$	21	19	24	11

Table 4.3: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 10$.

The convergence curves for zero-order conditions in the case of the uniform and METIS decompositions corresponding to Tables 4.1 and 4.3 are shown in Figures 4.4 and 4.5 for two different values of ε. We can see that when ε is increasing the convergence of the algorithm improves.

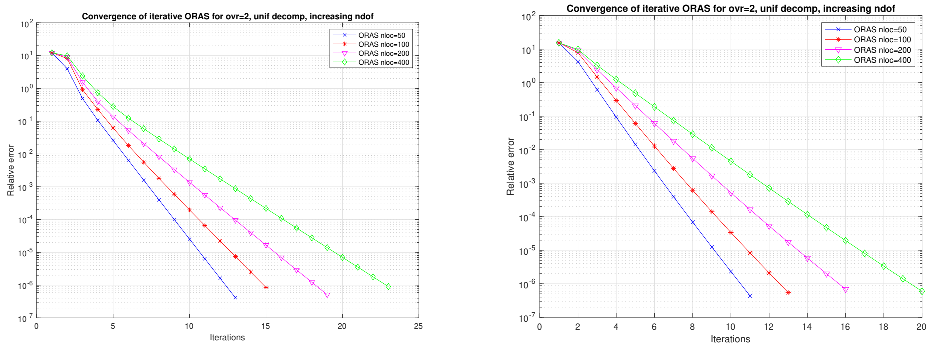

Figure 4.4: Convergence of ORAS for the zero-th order interface conditions $\varepsilon = 0.1$ (left) and $\varepsilon = 10$ (right), uniform decomposition

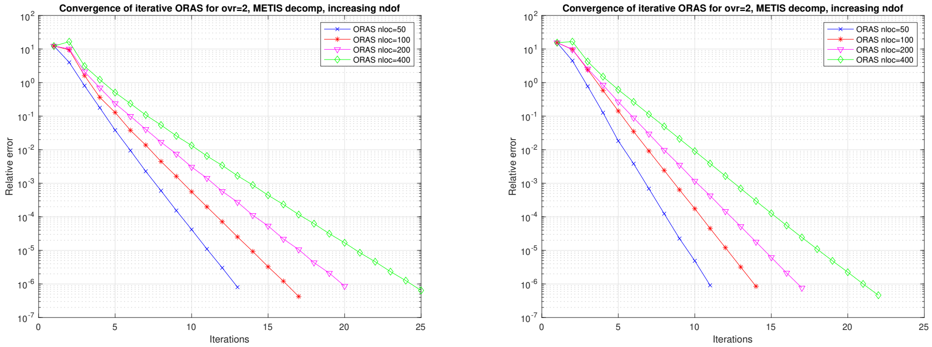

Figure 4.5: Convergence of ORAS for the zero-th order interface conditions $\varepsilon = 0.1$ (left) and $\varepsilon = 10$ (right), METIS decomposition

And finally, if we want to quantify the asymptotic behaviour of the iteration count as a function of the mesh size h, we see in Figure 4.6 that this is consistent with the theory i.e. in the case of zero-th order transmission conditions, the iteration count increases like $h^{-1/3}$ and in the case of second order condition like $h^{-1/5}$. This kind of increase is therefore far weaker than in the case of the RAS algorithm.

We now perform the same kind of experiments but with an increasing number of subdomains and a stripwise decomposition. In Tables 4.4, 4.5 and 4.6 we report the iteration count for $\sigma = 1$ and different values of ε, zero and second order conditions for uniform and METIS decompositions.

We notice that after a slight increase in iterations when the number of subdomains increases, this iteration count stabilises which is consistent to the theoretical results

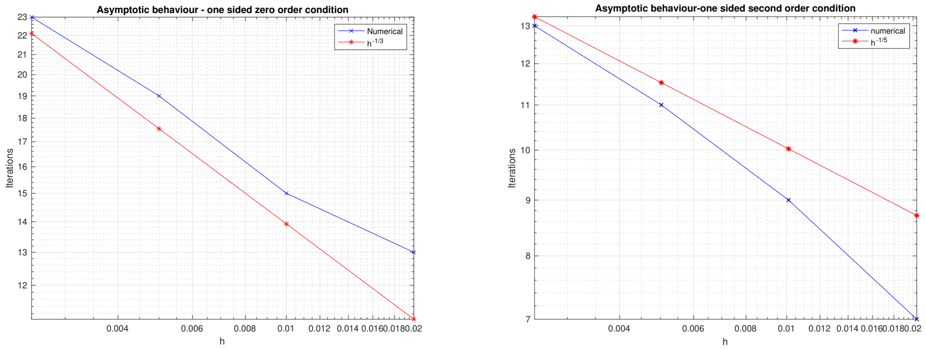

Figure 4.6: Iteration count depending on the mesh size for zero and second order conditions

	Zero order-one sided		Second order-one sided	
N	METIS	UNIFORM	METIS	UNIFORM
2	12	12	12	7
4	17	15	19	9
6	17	16	20	9
8	17	16	20	9
10	18	16	21	9
12	18	16	21	9

Table 4.4: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 0.1$.

	Zero order-one sided		Second order-one sided	
N	METIS	UNIFORM	METIS	UNIFORM
2	12	12	11	7
4	17	15	19	9
6	17	16	20	9
8	17	16	19	9
10	18	16	21	9
12	18	16	20	9

Table 4.5: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 1$.

and show that the method is scalable. This slight increase is barely visible when the parameter ε gets larger. We report illustrate the results from the previous tables in the graphs from Figure 4.7.

We conclude this section by showing a few results with the two-sided conditions. We see in Table 4.7 and Figure 4.8 that as predicted by the theory, the behaviour of zeroth order two-sided condition is very similar (asymptotically) to second order one-

	Zero order-one sided		Second order-one sided	
N	METIS	UNIFORM	METIS	UNIFORM
2	10	10	10	9
4	11	10	12	10
6	11	10	12	10
8	11	10	11	10
10	11	10	12	10
12	11	10	12	10

Table 4.6: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 10$.

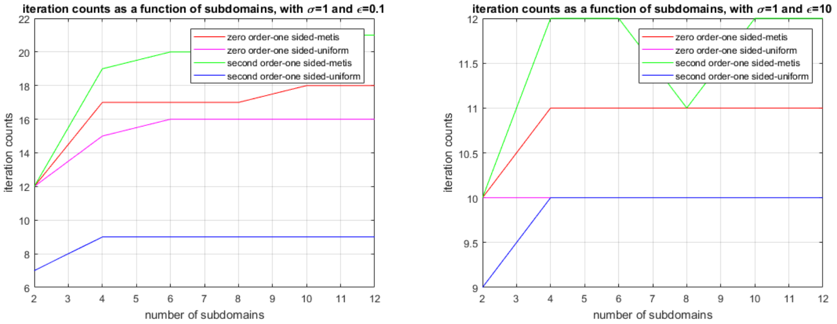

Figure 4.7: Iteration counts as a function of subdomains for various methods.

sided conditions from Figure 4.2 but with an added robustness in the case of METIS decompositions since we don't have any derivatives in the interface conditions.

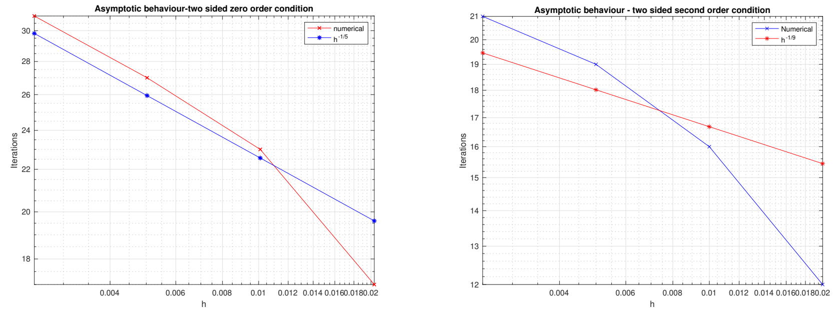

Figure 4.8: Asymptotic behaviour for zero-th and second order 2 sided IC: $\sigma = 1, \varepsilon = 1$

As shown in Table 4.7 In the case two-sided second order transmission conditions, we obtain a better convergence as in the case of the zero-th order method. The asymptotic behaviour is the one prescribed by the theory (iteration count behaving like $h^{-1/9}$). Note that for the results in Table 4.7 we have redefined the tolerance of the iterative to be 10^{-12} whereas in our initial tests, the tolerance has been 10^{-6}. Such a tolerance is

		Zero order- two sided		Second order-two sided	
n_{loc}	h	METIS	UNIFORM	METIS	UNIFORM
50	$\frac{1}{49}$	20	17	15	12
100	$\frac{1}{99}$	26	23	21	16
200	$\frac{1}{199}$	34	27	26	19
400	$\frac{1}{399}$	37	32	33	21

Table 4.7: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 1$.

usually not necessary in practical computations we have use it with the only purpose to be more accurate in our estimates of the asymptotic behaviour since the overall iteration count is not very important.

4.2 Optimised Schwarz method as a preconditioner

In this section we perform exactly the same kind of tests as in the previous one but when using the Schwarz method as a preconditioner in a GMRES iterative method.

In order to fully understand the benefits of the optimised transmission conditions we start by performing a few numerical simulations with the RAS method as preconditioner in a GMRES solver. In Figure 4.9 we see that the convergence deteriorates when the mesh size is decreased and the iteration count increases considerably.

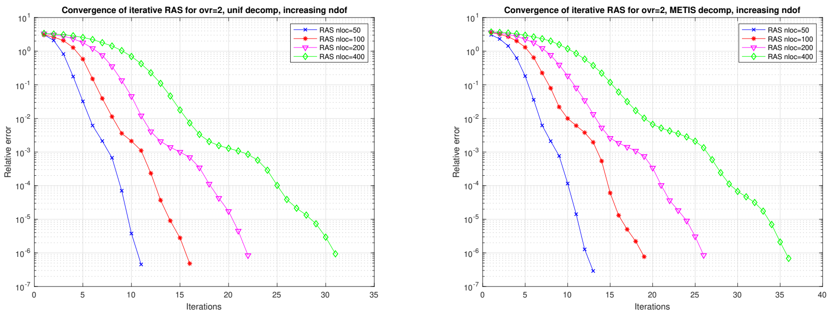

Figure 4.9: Convergence of the RAS algorithm for $\sigma = 1, \varepsilon = 1$ uniform (left) and METIS (right) decomposition

In order to quantify this increase we plot the asymptotic dependence of the iteration count with respect to the mesh size in Figure 4.10 and we notice that the iteration count behaves like $h^{-1/2}$, which is quite a strong dependence, even stronger than the purely iterative version of the optimised algorithm.

We consider again three case scenarios for a fixed value of $\sigma = 1$ and different values of

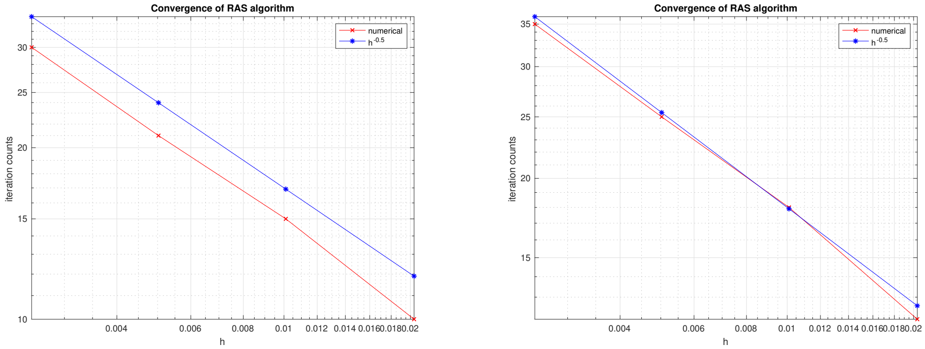

Figure 4.10: Convergence of the RAS algorithm for $\sigma = 1, \varepsilon = 1$ uniform (left) and METIS (right) decomposition

$\varepsilon = 0.1, 1, 10$ and we illustrate now the behaviour of the optimised method when used as a preconditioner in a GMRES iterative solver. Results are reported in Tables 4.8, 4.9 and 4.10. (again we have chosen to show iteration counts for the RAS method only in one case in order to illustrate the difference in iterations with respect to ORAS). We should notice that in this case there is less difference between RAS and ORAS but the advantage of ORAS increases the mesh is refined.

		Zero order-one sided		Second order-one sided	
n_{loc}	h	METIS	UNIFORM	METIS	UNIFORM
50	$\frac{1}{49}$	8	8	8	5
100	$\frac{1}{99}$	10	9	12	6
200	$\frac{1}{199}$	12	12	14	8
400	$\frac{1}{399}$	14	13	16	9

Table 4.8: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 0.1$.

		Zero order-one sided		Second order-one sided		RAS	
n_{loc}	h	METIS	UNIFORM	METIS	UNIFORM	METIS	UNIFORM
50	$\frac{1}{49}$	8	8	8	6	12	10
100	$\frac{1}{99}$	10	10	12	7	18	15
200	$\frac{1}{199}$	12	12	14	8	25	21
400	$\frac{1}{399}$	14	13	16	9	35	30

Table 4.9: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 1$.

We also show the convergence history for two of the cases (zero-th order transmission conditions in the case of uniform and METIS decompositions for $\varepsilon = 0.1$ and $\varepsilon = 10$) in Figures 4.11 and 4.12. We notice that because of the Krylov acceleration, the methods is even less sensitive to the mesh size and the differences between uniform and

		Zero order-one sided		Second order-one sided	
n_{loc}	h	METIS	UNIFORM	METIS	UNIFORM
50	$\frac{1}{49}$	8	8	8	6
100	$\frac{1}{99}$	10	10	11	7
200	$\frac{1}{199}$	12	11	13	8
400	$\frac{1}{399}$	15	14	15	9

Table 4.10: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 10$.

METIS decompositions are barely visible. This is not the case when we use METIS decomposition in the case of second order transmission conditions.

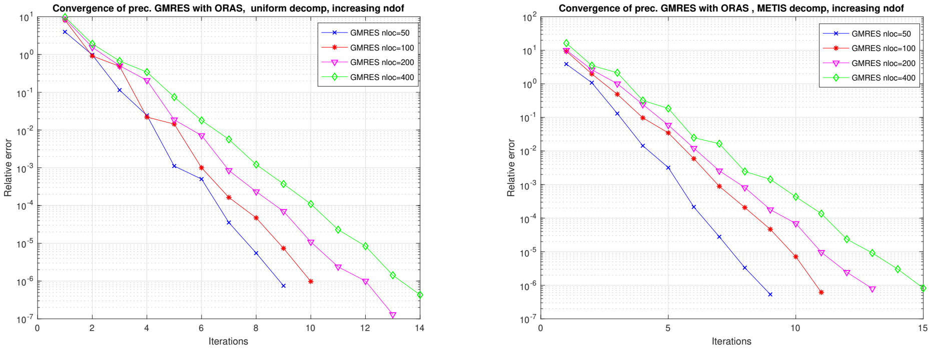

Figure 4.11: Convergence of the prec. GMRES with ORAS, one sided zero order transmission, $\varepsilon = 0.1$ (uniform and METIS decomposition).

We move on now to the case of many subdomains. Results are reported in Tables 4.11,4.12 and 4.13.

	Zero order-one sided		Second order-one sided	
N	METIS	UNIFORM	METIS	UNIFORM
2	8	8	8	5
4	10	10	12	7
6	11	10	12	7
8	11	10	13	7
10	11	10	13	7
12	11	10	13	7

Table 4.11: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 0.1$.

Again in this case we see that the behaviour of the method is consistent with the theory i.e. the iteration count stabilises as the number of subdomains gets larger, which means the method is scalable without further modification.

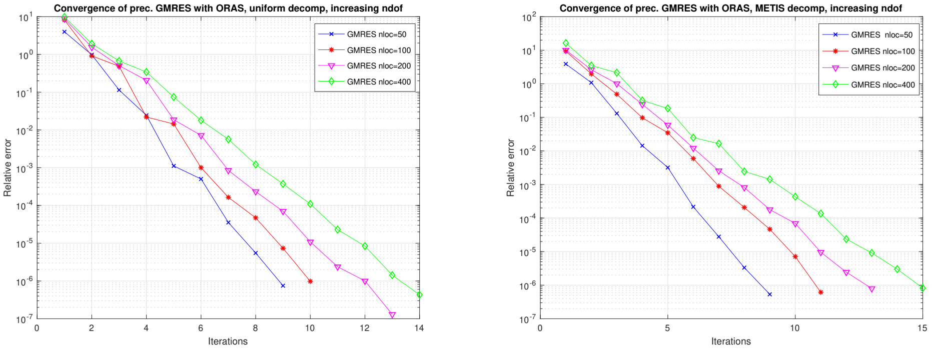

Figure 4.12: Convergence of the prec. GMRES with ORAS, one sided zero order transmission, $\varepsilon = 10$ (uniform and METIS decomposition).

	Zero order-one sided		Second order-one sided	
N	METIS	UNIFORM	METIS	UNIFORM
2	8	8	8	6
4	10	10	11	7
6	11	10	12	7
8	11	10	12	7
10	11	10	12	7
12	11	10	13	7

Table 4.12: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 1$.

	Zero order-one sided		Second order-one sided	
N	METIS	UNIFORM	METIS	UNIFORM
2	8	8	8	6
4	10	9	10	7
6	10	9	9	7
8	9	9	9	7
10	10	9	9	7
12	10	9	9	7

Table 4.13: One sided zeroth and second order IC, with $\sigma = 1$ and $\varepsilon = 10$.

4.3 Conclusions

In this chapter we have seen that developing optimised conditions is very important in this kind of hybrid direct-iterative methods as for the same computational cost we get a much better behaviour. Secondly, even in the case of many subdomains, the analysis shows that stripwise decompositions for this kind of problem can lead to robustness w.r.t to the number of subdomains and hence to scalability without further addition of a coarse space. This is an important feature that can be exploited in larger and more realistic computations. But also, this kind of analysis is applicable to other equations and one could for example design better transmission conditions for the Helmholtz problem for example where the behaviour gets worse as the wave-number increases. And to conclude, all these estimates were possible due to the limiting spectrum technique which proved to be a very accurate theoretical tool for block Toeplitz matrices arising from the decomposition into subdomains.

Appendices

Appendix A

Matlab implementations

A.1 Study of the limiting spectrum

Construction of the iteration matrix.

```
function [M,S] = iteration_matrix(n,a,b)

d1 = b*ones(2*n-3,1);
d1(2:2:end) = 0;
d2 = a*ones(2*n-4,1);
d2(2:2:end) = 0;
d3 = a*ones(2*n-4,1)-d2;

M = diag(d1,1)+diag(d1,-1)+diag(d2,2)+diag(d3,-2);
S = eig(M);
```

Testing the limiting spectrum bounds in the one dimensional case

```
clear all, close all, clc

epsilon = 0.1;
sigma = 5;

L = 1;
delta = L/10;

lambda = sqrt(sigma-1i*epsilon);
Dt = exp(lambda*(2*delta+L))-exp(-lambda*(2*delta+L));
a = (exp(2*lambda*delta) - exp(-2*lambda*delta))/Dt;
b = (exp(L*lambda) - exp(-L*lambda))/Dt;
```

```
rhoest = max(abs(a-b),abs(a+b));

theta = -pi:1/100:pi;
L1 = cos(theta)*a + sqrt(-sin(theta).^2*a^2 + b^2);
L2 = cos(theta)*a - sqrt(-sin(theta).^2*a^2 + b^2);

n = [2 5 10 20 40 60 80];
i = 1;
for ni = n
    [M,S] = iteration_matrix(ni,a,b);
    rho(i) = max(abs(S));
    i = i+1;
    plot(real(S),imag(S),'bx', real(L1),imag(L1),'r-',real(L2),imag(L2),'r-')
end
figure(1)
plot(real(S),imag(S),'bx', real(L1),imag(L1),'r-',real(L2),imag(L2),'r-')
legend('Spectrum of T1d','Theoretical estimate')
title('Spectrum of the Schwarz iteration matrix vs. theoretical estimate')
grid on
saveas(gcf,'Spectrum_Schwarz','epsc')

figure(2)
plot(n,rho,'b*-',n,rhoest*ones(size(n)),'r-')
legend('Convergence factor','Limiting spectral radius')
xlabel('Number of subdomains')
ylabel('Convergence factor')
title('Convergence of the algorithm for different number of subdomains')
grid on
saveas(gcf,'Conv_Schwarz','epsc')
```

Testing the limiting spectrum bounds in the two dimensional case

```
clear all, close all, clc

epsilon = 0.1;
sigma = 0.6;
L = 1;
delta = L/10;

n = [2 5 10 20 40 60 80];
j = 1;
freq = 0:0.5:10;
for ni = n
    i = 1;
    rhoest = 0;
    for m= freq
```

```
        kt = m*pi/L;
        lambda = sqrt(kt^2+sigma-1i*epsilon);
        Dt = exp(lambda*(2*delta+L))-exp(-lambda*(2*delta+L));
        a = (exp(2*lambda*delta) - exp(-2*lambda*delta))/Dt;
        b = (exp(L*lambda) - exp(-L*lambda))/Dt;
        rhoest = max(rhoest,max(abs(a-b),abs(a+b)));

        theta = -pi:1/50:pi;
        L1 = cos(theta)*a + sqrt(-sin(theta).^2*a^2 + b^2);
        L2 = cos(theta)*a - sqrt(-sin(theta).^2*a^2 + b^2);

        [M,S] = iteration_matrix(ni,a,b);
        rho(i) = max(abs(S));
        i = i+1;
    end
    rhomax(j) = max(rho);
    j = j+1;
end

figure(1)
plot(freq*pi/L,rho,'b*-'), grid on
xlabel('Frequency')
ylabel('Convergence factor')
saveas(gcf,'Conv_factor','epsc')

figure(2)
plot(n,rhomax,'b*-',n,rhoest*ones(size(n)),'r-'), grid on
legend('Convergence factor','Limiting spectral radius')
xlabel('Number of subdomains')
ylabel('Convergence factor')
title('Convergence of the algorithm for different number of subdomains')
saveas(gcf,'Conv_Schwarz','epsc')
```

A.2 Optimisation of the transmission conditions: zeroth order case

Convergence factor in the two subdomain case

```
function R = rho2sub(al)
% RHO evaluate the maximum of the convergence factor R = rhoNsub(p);

global kmin kmax sigma epsilon L delta pa pb;

N = 1000;
k = logspace(log10(kmin+0.01),log10(kmax),N);
```

```
%k=linspace(kmin,kmax,N);

lambda = sqrt(k.^2+sigma-1i*epsilon);
%p = al(1)+1i*al(2);
p = al(1);%+k.^2*al(2);

e1 = exp(-2*lambda*L);
e2 = exp(-2*lambda*(L + delta));

r1 = ((lambda+p).*(lambda-pa).*e1-(lambda-p).*(lambda+pa)).*exp(-lambda*delta)./...
    ((lambda+p).*(lambda+pa)-(lambda-p).*(lambda-pa).*e2);
r2 = ((lambda+p).*(lambda-pb).*e1-(lambda-p).*(lambda+pb)).*exp(-lambda*delta)./...
    ((lambda+p).*(lambda+pb)-(lambda-p).*(lambda-pb).*e2);
rho = sqrt(r1.*r2);

semilogx(k,abs(rho),'-');
%plot(k,abs(rho),'-');
drawnow
R = max(abs(rho));
```

and the test file with the numerical optimisation algorithm

```
clear all, close all, clc
global kmin kmax sigma epsilon L delta pa pb;

epsilon = 1;
sigma = 1;
pa = 1;
pb = 1;
L = 1;

% h = [1/10 1/100 1/1000 1/10000 1/100000]
% p = [1.8818,3.7696,7.9749,17.1093, 36.8255]
% ks = [6.5 27.5 127 580 2700]
% R = [0.2877,0.5764,0.7767,0.8896, 0.9472]
% we get ks = Ck*h^(-2/3), R = 1- CR*h^(1/3) and p = Cp*h^(-1/3)

% Constants computed by Maple
% s = sqrt(sigma+1i*epsilon);
% Ck = 2^(1/3)*(real(s*((pb + s)*(pa + s)-(s - pb)*(s - pa)*exp(-4*s*L) )/...
%      (((s - pa)*exp(-2*s*L)+ s + pa)*((s - pb)*exp(-2*s*L)+s+pb))))^(1/3);
% Cp = Ck^2/2;
% CR = 2*Ck;

% loglog(h,p,'rx-', h,Cp*h.^(-1/3),'b*-')
% loglog(h,ks,'rx-', h,Ck*h.^(-2/3),'b*-')
% loglog(h,1-R,'rx-',h,CR*h.^(1/3),'b*-')

kmin = 0.001;

h = 1/1000;
delta = h;

kmax = pi/h;

[p,R] = fminsearch('rho2sub',20)

xlabel('$\tilde{k}$','Interpreter','latex')
ylabel('$\rho$','Interpreter','latex')
title('One parameter optimisation')
```

Convergence factor in the three subdomain case

```
function R = rho3sub(al)
% RHO evaluate the maximum of the convergence factor R = rho3sub(p);

global kmin kmax sigma epsilon L delta pa pb rho;

N = 1000;
k = logspace(log10(kmin+0.01),log10(kmax),N);
%k=linspace(kmin,kmax,N);

p1m = pa; p3p = pb;
p1p = al(1);
p2m = al(2);
p2p = al(3);
p3m = al(4);

lam = sqrt(k.^2+sigma-1i*epsilon);
e1 = exp(-lam*L); e12 = exp(-2*lam*L);
e2 = exp(-lam*delta); e22 = exp(-2*lam*delta);
e3 = exp(-2*lam*(L + delta));

D1 = (p1p+lam).*(p1m+lam)-(p1p-lam).*(p1m-lam).*e3;
D2 = (p2p+lam).*(p2m+lam)-(p2p-lam).*(p2m-lam).*e3;
D3 = (p3p+lam).*(p3m+lam)-(p3p-lam).*(p3m-lam).*e3;

a1p = ((lam+p2m).*(lam+p1p)-(lam-p2m).*(lam-p1p).*e22).*e1./D2;
a3m = ((lam+p2p).*(lam+p3m)-(lam-p2p).*(lam-p3m).*e22).*e1./D2;

b1p = ((lam+p1p).*(lam-p2p).*e12-(lam-p1p).*(lam+p2p)).*e2./D2;
b2m = ((lam+p2m).*(lam-p1m).*e12-(lam-p2m).*(lam+p1m)).*e2./D1;
```

```
b2p = ((lam+p2p).*(lam-p3p).*e12-(lam-p2p).*(lam+p3p)).*e2./D3;
b3m = ((lam+p3m).*(lam-p2m).*e12-(lam-p3m).*(lam+p2m)).*e2./D2;

for zz=1:length(k)
    T3sub=[0 b1p(zz) a1p(zz) 0;
        b2m(zz) 0 0 0;
        0 0 0 b2p(zz);
        0 a3m(zz) b3m(zz) 0];
    rho(zz)=max(abs(eig(T3sub)));
end

semilogx(k,rho,'-');
%plot(k,abs(rho),'-');
drawnow
R = max(rho);
```

and the test file with the numerical optimisation algorithm

```
clear all, close all, clc
global kmin kmax sigma epsilon L delta pa pb;

epsilon = 1;
sigma = 1;
pa = 1;
pb = 1;
L = 1;

% Optimisation for the one parameter case. Asymp:  -1/3, -2/3, 1/3
% h = [1/10 1/100 1/1000 1/10000 1/100000]
% p = [1.5030, 3.1485 6.7083 14.4155 31.0395]
% R = [0.3344, 0.6046 0.7931 0.8982 0.9514]

% Two parameters case (per interface) p1p = p2m, p2p = p3m => only one
% h = [1/10 1/100 1/1000 1/10000 1/100000]
% p1 = [1.5031 3.1485 6.7083 14.4154 31.0395]
% p2 = [1.5029 3.1485 6.7084 14.4155 31.0395]
% R = [0.3312 0.6046  0.7931 0.8982 0.9514]

% Two parameters case (per domain): p1p = p2p = p1, p2m = p3m =p2
% h = [1/100 1/1000 1/10000 1/100000]
% p1 = [1.1611 1.9033 2.9897 4.7031]
% p2 = [9.5285 43.9797 173.4325 686.5970]
% R = [0.5158 0.6497 0.7661 0.8466]

% Four parameter case: p1p, p2m, p2p, p3m => p1p = p2p = p1, p2m = p3m =p2
% h = [1/100 1/1000 1/10000 1/100000]
% p1p = [0.9628 1.9017 2.9496 5.0114]
% p2m = [10.5361 43.9702 172.3793 672.0082]
% p2p = [1.2811 1.9032 3.0352 4.5121]
% p3m = [11.5478 43.9686 172.0470 672.5054]
% R = [0.5025 0.6497 0.7669 0.8481]

kmin = 0.001;

h = 1/100000;
delta = h;

kmax = pi/h;

[p,R] = fminsearch('rho3sub',[5 650 650 5])
```

Convergence factor in the four subdomain case

```
function R = rho4sub(al)
% RHO evaluate the maximum of the convergence factor R = rho3sub(p);

global kmin kmax sigma epsilon L delta pa pb rho;

N = 1000;
k = logspace(log10(kmin+0.01),log10(kmax),N);
%k=linspace(kmin,kmax,N);

p1m = pa; p4p = pb;
p1p = al(1);
p2m = al(2);
p2p = al(3);
p3m = al(4);
p3p = al(5);
p4m = al(6);

lam = sqrt(k.^2+sigma-1i*epsilon);
e1 = exp(-lam*L); e12 = exp(-2*lam*L);
e2 = exp(-lam*delta); e22 = exp(-2*lam*delta);
e3 = exp(-2*lam*(L + delta));

D1 = (p1p+lam).*(p1m+lam)-(p1p-lam).*(p1m-lam).*e3;
D2 = (p2p+lam).*(p2m+lam)-(p2p-lam).*(p2m-lam).*e3;
D3 = (p3p+lam).*(p3m+lam)-(p3p-lam).*(p3m-lam).*e3;
D4 = (p4p+lam).*(p4m+lam)-(p4p-lam).*(p4m-lam).*e3;

a1p = ((lam+p2m).*(lam+p1p)-(lam-p2m).*(lam-p1p).*e22).*e1./D2;
a2p = ((lam+p3m).*(lam+p2p)-(lam-p3m).*(lam-p2p).*e22).*e1./D3;
```

```matlab

a3m = ((lam+p2p).*(lam+p3m)-(lam-p2p).*(lam-p3m).*e22).*e1./D2;
a4m = ((lam+p3p).*(lam+p4m)-(lam-p3p).*(lam-p4m).*e22).*e1./D3;

b1p = ((lam+p1p).*(lam-p2p).*e12-(lam-p1p).*(lam+p2p)).*e2./D2;
b2m = ((lam+p2m).*(lam-p1m).*e12-(lam-p2m).*(lam+p1m)).*e2./D1;
b2p = ((lam+p2p).*(lam-p3p).*e12-(lam-p2p).*(lam+p3p)).*e2./D3;
b3m = ((lam+p3m).*(lam-p2m).*e12-(lam-p3m).*(lam+p2m)).*e2./D2;
b3p = ((lam+p3p).*(lam-p4p).*e12-(lam-p3p).*(lam+p4p)).*e2./D4;
b4m = ((lam+p4m).*(lam-p3m).*e12-(lam-p4m).*(lam+p3m)).*e2./D3;

for l = 1:length(k)
    T4=[0 b1p(l) a1p(l) 0 0 0;
        b2m(l) 0 0 0 0 0;
        0 0 0 b2p(l) a2p(l) 0;
        0 a3m(l) b3m(l) 0 0 0;
        0 0 0 0 0 b3p(l);
        0 0 0 a4m(l) b4m(l) 0];
    rho(l) = max(abs(eig(T4)));
end

semilogx(k,rho,'-');
%plot(k,abs(rho),'-');
drawnow
R = max(rho);
```

and the test file with the numerical optimisation algorithm

```matlab
clear all, close all, clc
global kmin kmax sigma epsilon L delta pa pb rho;

epsilon = 1;
sigma = 1;
pa = 1;
pb = 1;
L = 1;

% One parameter p1p = p2p = p3p = p2m = p3m = p4m = p
% h = [1/100 1/1000 1/10000 1/100000];
% p = [2.8396 6.0657 13.0412 28.0834];
% loglog(h,p,'rx-', h,h.^(-1/3),'b*-')
% R = [0.6202 0.8022 0.9029 0.9537];
% loglog(h,1-R,'rx-',h,h.^(1/3),'b*-')

% Two parameters p1p = p2p = p3p  = p1, p2m = p3m = p4m = p2
% h = [1/100 1/1000 1/10000 1/100000];
% p1 = [7.0602 40.9958 163.1013 646.3896]
% loglog(h,p1,'rx-', h,h.^(-3/5),'b*-')
% p2 = [1.1809 1.6560 2.6452 4.1691]
% loglog(h,p2,'rx-', h,h.^(-1/5),'b*-')
% R = [0.5604 0.6599 0.7725 0.8509]
% loglog(h,1-R,'rx-',h,h.^(1/5),'b*-')

% Different parameters  => only 2 and same asymptotes as before
% h = [1/100 1/1000 1/10000 1/100000];
% p1p = [13.1269 37.9717 152.9323 651.7536]
% p2m = [1.2705 1.4208 2.3266 4.1945]
% p2p = [10.1871 42.9379 152.0873 645.0605]
% p3m = [0.7748 1.6005 3.1841 4.1519]
% p3p = [16.5975 68.1923 161.0389 649.8928]
% p4m = [2.1327 2.4896 2.4919 4.1828]

% R = [0.5206 0.6708 0.7789 0.8510]

kmin = 0.001;

h = 1/100000;
delta = h;

kmax = pi/h;

[p,R] = fminsearch('rho4sub',[2.3461  396.3355 1.6427  414.2898  397.1038 1.5748])
```

Convergence factor in the five subdomain case

```
function R = rho5sub(al)
% RHO evaluate the maximum of the convergence factor R = rho3sub(p);

global kmin kmax sigma epsilon L delta pa pb rho;

N = 1000;
k = logspace(log10(kmin+0.01),log10(kmax),N);
%k=linspace(kmin,kmax,N);

p1m = pa; p5p = pb;
p1p = al(1);
p2m = al(2);
p2p = al(3);
p3m = al(4);
p3p = al(5);
p4m = al(6);
p4p = al(7);
```

```
p5m = al(8);

lam = sqrt(k.^2+sigma-1i*epsilon);
e1 = exp(-lam*L); e12 = exp(-2*lam*L);
e2 = exp(-lam*delta); e22 = exp(-2*lam*delta);
e3 = exp(-2*lam*(L + delta));

D1 = (p1p+lam).*(p1m+lam)-(p1p-lam).*(p1m-lam).*e3;
D2 = (p2p+lam).*(p2m+lam)-(p2p-lam).*(p2m-lam).*e3;
D3 = (p3p+lam).*(p3m+lam)-(p3p-lam).*(p3m-lam).*e3;
D4 = (p4p+lam).*(p4m+lam)-(p4p-lam).*(p4m-lam).*e3;
D5 = (p5p+lam).*(p5m+lam)-(p5p-lam).*(p5m-lam).*e3;

a1p = ((lam+p2m).*(lam+p1p)-(lam-p2m).*(lam-p1p).*e22).*e1./D2;
a2p = ((lam+p3m).*(lam+p2p)-(lam-p3m).*(lam-p2p).*e22).*e1./D3;
a3p = ((lam+p4m).*(lam+p3p)-(lam-p4m).*(lam-p3p).*e22).*e1./D4;

a3m = ((lam+p2p).*(lam+p3m)-(lam-p2p).*(lam-p3m).*e22).*e1./D2;
a4m = ((lam+p3p).*(lam+p4m)-(lam-p3p).*(lam-p4m).*e22).*e1./D3;
a5m = ((lam+p4p).*(lam+p5m)-(lam-p4p).*(lam-p5m).*e22).*e1./D4;

b1p = ((lam+p1p).*(lam-p2p).*e12-(lam-p1p).*(lam+p2p)).*e2./D2;
b2m = ((lam+p2m).*(lam-p1m).*e12-(lam-p2m).*(lam+p1m)).*e2./D1;
b2p = ((lam+p2p).*(lam-p3p).*e12-(lam-p2p).*(lam+p3p)).*e2./D3;
b3m = ((lam+p3m).*(lam-p2m).*e12-(lam-p3m).*(lam+p2m)).*e2./D2;
b3p = ((lam+p3p).*(lam-p4p).*e12-(lam-p3p).*(lam+p4p)).*e2./D4;
b4m = ((lam+p4m).*(lam-p3m).*e12-(lam-p4m).*(lam+p3m)).*e2./D3;
b4p = ((lam+p4p).*(lam-p5p).*e12-(lam-p4p).*(lam+p5p)).*e2./D5;
b5m = ((lam+p5m).*(lam-p4m).*e12-(lam-p5m).*(lam+p4m)).*e2./D4;

for l = 1:length(k)
    T5 = [0 b1p(l) a1p(l) 0 0 0 0 0;
        b2m(l) 0 0 0 0 0 0 0;
        0 0 0 b2p(l) a2p(l) 0 0 0;
        0 a3m(l) b3m(l) 0 0 0 0 0;
        0 0 0 0 0 b3p(l) a3p(l) 0;
        0 0 0 a4m(l) b4m(l) 0 0 0;
        0 0 0 0 0 0 0 b4p(l);
        0 0 0 0 0 a5m(l) b5m(l) 0];
    rho(l) = max(abs(eig(T5)));
end

semilogx(k,rho,'-');
%plot(k,abs(rho),'-');
drawnow
```

```
R = max(rho);
```

and the test file with the numerical optimisation algorithm

```
clear all, close all, clc
global kmin kmax sigma epsilon L delta pa pb rho;

epsilon = 1;
sigma = 1;
pa = 1;
pb = 1;
L = 1;

% One parameter p1p = p2p = p3p = p2m = p3m = p4m = p4p = p5m = p
% h = [1/100 1/1000 1/10000 1/100000];
% p = [];
% loglog(h,p,'rx-', h,h.^(-1/3),'b*-')
% R = [];
% loglog(h,1-R,'rx-',h,h.^(1/3),'b*-')

% Two parameters p1p = p2p = p3p  = p1, p2m = p3m = p4m = p2
% h = [1/100 1/1000 1/10000 1/100000];
% p1 = [7.0602 40.9958 163.1013 646.3896]
% loglog(h,p1,'rx-', h,h.^(-3/5),'b*-')
% p2 = [1.1809 1.6560 2.6452 4.1691]
% loglog(h,p2,'rx-', h,h.^(-1/5),'b*-')
% R = [0.5604 0.6599 0.7725 0.8509]
% loglog(h,1-R,'rx-',h,h.^(1/5),'b*-')

% Different parameters => only 2 and same asymptotes as before
% h = [1/100 1/1000 1/10000 1/100000];
% p1p = [13.1269 37.9717 152.9323 651.7536]
% p2m = [1.2705 1.4208 2.3266 4.1945]
% p2p = [10.1871 42.9379 152.0873 645.0605]
% p3m = [0.7748 1.6005 3.1841 4.1519]
% p3p = [16.5975 68.1923 161.0389 649.8928]
% p4m = [2.1327 2.4896 2.4919 4.1828]

% R = [0.5206 0.6708 0.7789 0.8510]

kmin = 0.001;

h = 1/100000;
delta = h;

kmax = pi/h;
```

```

[p,R] = fminsearch('rho5sub',[651.7536 4.1945 645.0605 4.1519 649.8928 4.1828 650 4])
```

Convergence factor in the six subdomain case

```
function R = rho6sub(al)
% RHO evaluate the maximum of the convergence factor R = rho3sub(p);

global kmin kmax sigma epsilon L delta pa pb rho;

N = 1000;
k = logspace(log10(kmin+0.01),log10(kmax),N);
%k=linspace(kmin,kmax,N);

p1m = pa; p6p = pb;
p1p = al(1);
p2m = al(2);
p2p = al(3);
p3m = al(4);
p3p = al(5);
p4m = al(6);
p4p = al(7);
p5m = al(8);
p5p = al(9);
p6m = al(10);

lam = sqrt(k.^2+sigma-1i*epsilon);
e1 = exp(-lam*L); e12 = exp(-2*lam*L);
e2 = exp(-lam*delta); e22 = exp(-2*lam*delta);
e3 = exp(-2*lam*(L + delta));

D1 = (p1p+lam).*(p1m+lam)-(p1p-lam).*(p1m-lam).*e3;
D2 = (p2p+lam).*(p2m+lam)-(p2p-lam).*(p2m-lam).*e3;
D3 = (p3p+lam).*(p3m+lam)-(p3p-lam).*(p3m-lam).*e3;
D4 = (p4p+lam).*(p4m+lam)-(p4p-lam).*(p4m-lam).*e3;
D5 = (p5p+lam).*(p5m+lam)-(p5p-lam).*(p5m-lam).*e3;
D6 = (p6p+lam).*(p6m+lam)-(p6p-lam).*(p6m-lam).*e3;

a1p = ((lam+p2m).*(lam+p1p)-(lam-p2m).*(lam-p1p).*e22).*e1./D2;
a2p = ((lam+p3m).*(lam+p2p)-(lam-p3m).*(lam-p2p).*e22).*e1./D3;
a3p = ((lam+p4m).*(lam+p3p)-(lam-p4m).*(lam-p3p).*e22).*e1./D4;
a4p = ((lam+p5m).*(lam+p4p)-(lam-p5m).*(lam-p4p).*e22).*e1./D5;

a3m = ((lam+p2p).*(lam+p3m)-(lam-p2p).*(lam-p3m).*e22).*e1./D2;
a4m = ((lam+p3p).*(lam+p4m)-(lam-p3p).*(lam-p4m).*e22).*e1./D3;
a5m = ((lam+p4p).*(lam+p5m)-(lam-p4p).*(lam-p5m).*e22).*e1./D4;
```

```
a6m = ((lam+p5p).*(lam+p6m)-(lam-p5p).*(lam-p6m).*e22).*e1./D5;

b1p = ((lam+p1p).*(lam-p2p).*e12-(lam-p1p).*(lam+p2p)).*e2./D2;
b2m = ((lam+p2m).*(lam-p1m).*e12-(lam-p2m).*(lam+p1m)).*e2./D1;
b2p = ((lam+p2p).*(lam-p3p).*e12-(lam-p2p).*(lam+p3p)).*e2./D3;
b3m = ((lam+p3m).*(lam-p2m).*e12-(lam-p3m).*(lam+p2m)).*e2./D2;
b3p = ((lam+p3p).*(lam-p4p).*e12-(lam-p3p).*(lam+p4p)).*e2./D4;
b4m = ((lam+p4m).*(lam-p3m).*e12-(lam-p4m).*(lam+p3m)).*e2./D3;
b4p = ((lam+p4p).*(lam-p5p).*e12-(lam-p4p).*(lam+p5p)).*e2./D5;
b5m = ((lam+p5m).*(lam-p4m).*e12-(lam-p5m).*(lam+p4m)).*e2./D4;

b5p = ((lam+p5p).*(lam-p6p).*e12-(lam-p5p).*(lam+p6p)).*e2./D6;
b6m = ((lam+p6m).*(lam-p5m).*e12-(lam-p6m).*(lam+p5m)).*e2./D5;


for l = 1:length(k)
    T6 = [0 b1p(l) a1p(l) 0 0 0 0 0 0 0;
        b2m(l) 0 0 0 0 0 0 0 0 0;
        0 0 0 b2p(l) a2p(l) 0 0 0 0 0;
        0 a3m(l) b3m(l) 0 0 0 0 0 0 0;
        0 0 0 0 0 b3p(l) a3p(l) 0 0 0;
        0 0 0 a4m(l) b4m(l) 0 0 0 0 0;
        0 0 0 0 0 0 0 b4p(l) a4p(l) 0;
        0 0 0 0 0 a5m(l) b5m(l) 0 0 0;
        0 0 0 0 0 0 0 0 0 b5p(l);
        0 0 0 0 0 0 0 a6m(l) b6m(l) 0];
    rho(l) = max(abs(eig(T6)));
end

semilogx(k,rho,'-');
%plot(k,abs(rho),'-');
drawnow
R = max(rho);
```

and the test file with the numerical optimisation algorithm

```
clear all, close all, clc
global kmin kmax sigma epsilon L delta pa pb rho;

epsilon = 1;
sigma = 1;
pa = 1;
pb = 1;
L = 1;

% One parameter p1p = p2p = p3p = p2m = p3m = p4m = p4p = p5m = p
% h = [1/100 1/1000 1/10000 1/100000];
% p = [];
% loglog(h,p,'rx-', h,h.^(-1/3),'b*-')
% R = [];
% loglog(h,1-R,'rx-',h,h.^(1/3),'b*-')

% Two parameters p1p = p2p = p3p  = p1, p2m = p3m = p4m = p2
% h = [1/100 1/1000 1/10000 1/100000];
% p1 = [7.0602 40.9958 163.1013 646.3896]
% loglog(h,p1,'rx-', h,h.^(-3/5),'b*-')
% p2 = [1.1809 1.6560 2.6452 4.1691]
% loglog(h,p2,'rx-', h,h.^(-1/5),'b*-')
% R = [0.5604 0.6599 0.7725 0.8509]
% loglog(h,1-R,'rx-',h,h.^(1/5),'b*-')

% Different parameters  => only 2 and same asymptotes as before
% h = [1/100 1/1000 1/10000 1/100000];
% p1p = [13.1269 37.9717 152.9323 651.7536]
% p2m = [1.2705 1.4208 2.3266 4.1945]
% p2p = [10.1871 42.9379 152.0873 645.0605]
% p3m = [0.7748 1.6005 3.1841 4.1519]
% p3p = [16.5975 68.1923 161.0389 649.8928]
% p4m = [2.1327 2.4896 2.4919 4.1828]

% R = [0.5206 0.6708 0.7789 0.8510]

kmin = 0.001;

h = 1/100000;
delta = h;

kmax = pi/h;

[p,R] = fminsearch('rho6sub',[600 4 600 4 600 4 600 4 600 4])
```

Convergence factor in the N subdomain case using the limiting spectrum

```
function R = rhoNsub(al)
% RHO evaluate the maximum of the convergence factor R = rhoNsub(p);

global kmin kmax sigma epsilon L delta;

N = 1000;
k = logspace(log10(kmin+0.01),log10(kmax),N);
%k=linspace(kmin,kmax,N);

```

```
lambda = sqrt(k.^2+sigma-1i*epsilon);
p = al(1);

e1 = exp(-lambda*L);
e2 = exp(-2*lambda*(L + delta));

D = (lambda+p).^2-(lambda-p).^2.*e2;
a = ((lambda+p).^2- (lambda-p).^2.*exp(-2*delta*lambda)).*e1./D;
b = (lambda.^2-p^2).*(e1.^2-1).*exp(-lambda*delta)./D;
rho = max(abs(a-b),abs(a+b));

semilogx(k,abs(rho),'-');
%plot(k,abs(rho),'-');
drawnow
R = max(abs(rho));
```

Appendix B

FreeFem++ implementations

In this section we discuss the FreeFem++ implementation of the methods from Chapter 3 and show the main parts of codes we used. All the details about the choice the solver, discrete spaces, boundary conditions, are described in the exhaustive comments from the codes. We use the build-in polynomial finite element spaces.

The main program needs the routines (of `decomp.idp` and `createPartition.idp`) to create a decomposition of the domain and to build the restriction and partition of unity matrices. We do not include here the files `decomp.idp` and `createpartitionVec.idp` as they can be downloaded here `http://www.victoritadolean.com/p/book.html`.

These methods are used as solvers and preconditioners.

B.1 Data files and definitions of macros

The data file in both cases is `dataMagnetotelluric.edp`

```
load "metis"
load "medit"

//Boundary conditions
int Dirichlet = 1;
int Robin = 2;
int order = 2;

string method = "ORAS";     // preconditioner RAS or ORAS
int nn = 2, mm = 1;              // number of the domains in each direction
int npart = nn*mm;        // total number of domains
bool withmetis = 0;       // = 1 (Metis decomp) =0 (uniform decomp)
real sizeovr = 2;         // size of the overlap
int nloc = 400;            // local no of dof per domain in one direction
mesh Th;
real allong;

real L = 1;
func f = 1;
real h = 1./(nloc-1);
real delta = sizeovr*h; // size of the overlap
real sigma = 1;
real epsilon = 1;
real kmin = pi;
//kmin = 0;
complex s = sqrt(sigma +kmin^2-1i*epsilon);
complex C;
complex p,q;

if (nn == 2){
    C = real(s*(1+exp(2*s*L))/(exp(2*s*L)-1));
}
else if (nn ==3){
    C = real(s*(1+exp(2*s*L)-exp(s*L))/(exp(2*s*L)-1));
}
else {
    C = real(s*(1+exp(2*s*L)-2*cos(pi/(nn+1))*exp(s*L))/(exp(2*s*L)-1));
}

if (order == 0)
{
    p = 2.^(-1./3.)*C^(2./3.)*delta^(-1./3.);
    q = 0;
}
else
{
    p = 2.^(-3/5.)*C^(4./5.)*delta^(-1./5.);
    q = 2.^(-1/5.)*C^(-2./5.)*delta^(3./5.);
}

complex pbc = 10.0;
cout << "p= " << p << endl;

int[int] chlab = [1,Dirichlet   ,2,Dirichlet  ,3,Dirichlet   ,4,Dirichlet]; //Robin conditions for label = 2
```

```
macro Grad(u) [dx(u),dy(u)]                              // EOM

// Iterative solver parameters
real tol = 1e-12;    // tolerance for the iterative method
int maxit = 400;     // maximum number of iterations
```

The two sided equivalent of the data file is `dataMagnetotelluric-2sd.edp`

```
load "metis"
load "medit"

//Boundary conditions
int Dirichlet = 1;
int Robin = 2;
int order = 2;

string method = "ORAS";     // preconditioner RAS or ORAS
int nn=2, mm=1;               // number of the domains in each direction
int npart = nn*mm;          // total number of domains
bool withmetis = 0;         // =1 (Metis decomp) =0 (uniform decomp)
real sizeovr = 2;           // size of the overlap
int nloc = 50;              // local no of dof per domain in one direction
mesh Th;
real allong;

real L = 1;
func f = 1;
real h = 1./(nloc-1);
real delta = sizeovr*h; // size of the overlap
real sigma = 1;
real epsilon = 1;
real kmin = pi;
kmin = 4;
complex s = sqrt(sigma +kmin^2-1i*epsilon);
complex C;
complex[int] p(2),q(2);

if (nn == 2){
    C = real(s*(1+exp(2*s*L))/(exp(2*s*L)-1));
}
else if (nn ==3){
    C = real(s*(1+exp(2*s*L)-exp(s*L))/(exp(2*s*L)-1));
}
else {
    C = real(s*(1+exp(2*s*L)-2*cos(pi/(nn+1))*exp(s*L))/(exp(2*s*L)-1));
}
```

```

if (order == 0){
    p[0] = 2.^(-2./5.)*C^(2./5.)*delta^(-3./5.);
    p[1] = 2.^(-4./5.)*C^(4./5.)*delta^(-1./5.);
    // filep << "p1=" << real(p[0]) << endl;
    // filep << "p2=" << real(p[1]) << endl;
    q = 0.;
}
else{
    p[0] = 2.^(-8./9.)*C^(8./9.)*delta^(-1./9.);
    p[1] = 2.^(-2./3.)*C^(2./3.)*delta^(-1./3.);
    q[0] = 2.^(2./9.)*C^(-2./9.)*delta^(7./9.);
    q[1] = 2.^(4./9.)*C^(-4./9.)*delta^(5./9.);
}

complex pbc = 10.0;
cout << "p= " << p << endl;

int[int] chlab=[1,Dirichlet  ,2,Dirichlet ,3,Dirichlet  ,4,Dirichlet]; //Robin conditions for label = 2
macro Grad(u) [dx(u),dy(u)]                              // EOM

// Iterative solver parameters
real tol = 1e-12;    // tolerance for the iterative method
int maxit = 30;    // maximum number of iterations
```

We also need to define the domain decomposition data structures and the global variational formulation as shown in `defMagnetotelluric.edp`

```
// Definition ingredients - numerical solution of magnetotelluric equation
// Mesh of a rectangular domain

allong = real(nn)/real(mm); // aspect ratio of the global domain
Th = square(nn*nloc,mm*nloc,[x*allong,y]);

fespace Ph(Th,P0);
fespace Vh(Th,P1);   // scalar fem space
fespace Uh(Th,P1);        // scalar fem space
Ph part;                  // piecewise constant function
int[int] lpart(Ph.ndof);  // giving the decomposition

// Domain decomposition data structures
mesh[int] aTh(npart),aTh0(npart);                        // sequence of ovr. meshes
matrix<complex>[int] Rih(npart);           // local restriction operators
matrix<complex>[int] Dih(npart);           // partition of unity operators
```

```
matrix[int] Dihreal(npart),Rihreal(npart),Dihreal0(npart),Rihreal0(npart),Dihrealr(npart);
int[int] Ndeg(npart),Ndeg0(npart);                          // number of dof for each mesh
real[int] AreaThi(npart),AreaThi0(npart);                     // area of each subdomain
matrix<complex>[int] aA(npart),aR(npart);   // local Dirichlet/Robin matrices

Th=change(Th,refe=chlab);
// global variational formulation
    varf vaglobal(u,v) =
  int2d(Th)(Grad(u)'*Grad(v)+(sigma-1i*epsilon)*u*v)
                                  + int1d(Th,Robin)(pbc*u*v)
      + int2d(Th)(f*v)
                                  + on(Dirichlet,u=0); // EOM

matrix<complex> Aglobal;
Vh<complex>rhsglobal,uglob;
```

B.2 RAS/ORAS

The main script file for the iterative versions of RAS and ORAS algorithms for the one-sided interface conditions is `Solver-Magnetotelluric.edp`

```
/*# debutPartition #*/
include "./dataMagnetotelluric.edp"
include "./defMagnetotelluric.edp"
include "./decomp.idp"
include "./createPartition.idp"
SubdomainsPartitionUnity(Th,part[],sizeovr,aTh,Rihreal,Dihreal,Ndeg,AreaThi);

//Build a new partition of unity
SubdomainsPartitionUnity(Th,part[],0,aTh0,Rihreal0,Dihreal0,Ndeg0,AreaThi0);
for (int i=0; i < npart; i++) {
  mesh Thi = aTh[i];
  mesh Thi0 = aTh0[i];
   matrix Maux1, Maux2, Maux3;
   Maux1  = Rihreal0[i]*Rihreal[i]';
   Maux2  = Dihreal0[i]*Maux1;
   Maux3  = Rihreal0[i]'*Maux2;
   Dihrealr[i] = Rihreal[i]*Maux3;
   // plot(Thi0,wait=1);
   //plot(Thi,wait=1);
}
```

```
for (int i=0; i<npart; i++) {
        Rih[i] = Rihreal[i];
        Dih[i] = Dihrealr[i];
  //Dih[i] = Dihreal[i];
  // test the partition of unity
  /* Vh<complex> ux,vx;
  ux[]=1.;
  matrix Maux1, Maux2;
        Maux1 = Dihrealr[i]*Rihreal[i];
  mesh Thi = aTh[i];
  fespace Vhi(Thi,P1);
  Vhi<complex> uxi;
  uxi[] = Maux1*ux[];
        plot(uxi,value=1,fill=1,dim=3,wait=1); */
        //Maux2 = Rihreal[i]'*Maux1;
  //vx[] = Maux2*ux[];
        //plot(vx,value=1,fill=1,dim=3,wait=1);
}

/*# endPartition #*/
/*# debutGlobalData #*/
Aglobal = vaglobal(Vh,Vh,solver = UMFPACK);  // global matrix
rhsglobal[] = vaglobal(0,Vh);                                 // global rhs
uglob[] = Aglobal^-1*rhsglobal[];
real ugl2 = uglob[].l2;
// Vh realuglob = real(uglob);
//medit("Solution",Th,realuglob);
/*# finGlobalData #*/

/*# debutLocalData #*/
for(int i = 0;i<npart;++i){
         mesh Thi = aTh[i];
     fespace Vhi(Thi,P1);
     cout << " Domain :" << i << "/" << npart << endl;
     if (method == "ORAS"){
         varf valocal(u,v) =
       int2d(Thi)(Grad(u)'*Grad(v)+(sigma-1i*epsilon)*u*v)
             + int1d(Thi,Robin)(pbc*u*v)
             + int1d(Thi,10)(p*u*v)
             + int1d(Thi,10)(q*dy(u)*dy(v))
             + on(Dirichlet,u=0);
              aR[i] = valocal(Vhi,Vhi,solver = UMFPACK);
      }
      if (method == "RAS"){
        matrix<complex> temp = Aglobal*Rih[i]';
```

```
    aR[i] = Rih[i]*temp;
    set(aR[i],solver = UMFPACK);
   }
}
/*# finLocalData #*/
/*# debutSchwarzIter #*/
ofstream filei (method+"_ovr"+sizeovr+"_sig"+sigma+"_ep"+epsilon+"_n"+nloc+".m");
Vh<complex> un = 0;                              // initial guess
for(int i = 0;i<Vh.ndof;++i)
{
 un[][i] = randreal1()+1i*randreal1();
}
Vh intern;
intern = (x>0) && (x<allong ) && (y>0) && (y<1);
un[].re = un[].re.*intern[];
Vh<complex> rn = rhsglobal;
Vh<complex> er,dr;
for(int iter = 0;iter<maxit;++iter)
 {
  real err = 0, res;
  dr = 0;
  for(int i = 0;i<npart;++i)
   {
    complex[int] bi = Rih[i]*rn[];     // restriction to the local domain
    complex[int] ui = aR[i] ^-1 * bi; // local solve
    bi = Dih[i]*ui;
    dr[] += Rih[i]'*bi;
   }
  un[] += dr[];               // build new iterate
  rn[] = Aglobal*un[];        // computes global residual
  rn[] = rn[] - rhsglobal[];
  rn[] *= -1;
        er[] = un[]-uglob[];
        //cout << "Error = "<< er[][25] << endl;
  err = er[].l2/ugl2;
  res = rn[].l2/ugl2;
  cout << "It: "<< iter << " Residual = " << res  <<  " Relative L2 Error = "<<  err
     << endl;
  Vh [abser] = [abs(er)];
  plot(abser,value=1,dim=3,fill=1,wait=0,cmm="error");
        //plot(abser,value=1,dim=3,fill=1,cmm="error");
  int j = iter+1;
  // Store the error and the residual in Matlab/Scilab/Octave form
  filei << method+"("+j+")=" << err << ";" << endl;
  if(err < tol) break;
  }
/*medit("Error",Th,abs(er));
medit("Absolute value of the solution",Th,abs(un));*/
```

and its two-sided equivalent

```
// Implementation of the iterative version
// of the two-sided interface transmission conditions

include "./dataMagnetotelluric-2sd.edp"
include "./defMagnetotelluric.edp"
include "./decomp.idp"
include "./createPartition.idp"
SubdomainsPartitionUnity(Th,part[],sizeovr,aTh,Rihreal,Dihreal,Ndeg,AreaThi);
// plot(part,wait=1,fill=1,ps="Partition");

//Build a new partition of unity
SubdomainsPartitionUnity(Th,part[],1,aTh0,Rihreal0,Dihreal0,Ndeg0,AreaThi0);
for (int i=0; i < npart; i++) {
    matrix Maux1, Maux2, Maux3;
    Maux1  = Rihreal0[i]*Rihreal[i]';
    Maux2  = Dihreal0[i]*Maux1;
    Maux3  = Rihreal0[i]'*Maux2;
    Dihrealr[i] = Rihreal[i]*Maux3;
}

plot(part,wait=1,fill=1,ps="Partition");

for (int i=0; i<npart; i++) {
        Rih[i] = Rihreal[i];
        //Dih[i] = Dihreal[i];
  Dih[i] = Dihrealr[i];
}

/*# endPartition #*/
/*# debutGlobalData #*/
Aglobal = vaglobal(Vh,Vh,solver = UMFPACK);  // global matrix
rhsglobal[] = vaglobal(0,Vh);                               // global rhs
uglob[] = Aglobal^-1*rhsglobal[];
real ugl2 = uglob[].l2;
/*# finGlobalData #*/

/*# debutLocalData #*/

complex[int] init(Vh.ndof);
for(int i = 0;i<Vh.ndof;++i){
      init[i] = randreal1()+1i*randreal1();
      }

for(int i = 0;i<npart;++i){
         mesh Thi = aTh[i];
     fespace Vhi(Thi,P1);
     int i0 = fmod(i,2);
     if (method == "ORAS"){
         varf valocal(u,v) =
          int2d(Thi)(Grad(u)'*Grad(v)+(sigma-1i*epsilon)*u*v)
             + int1d(Thi,Robin)(pbc*u*v)
             + int1d(Thi,10)(p[i0]*u*v)
             + int1d(Thi,10)(q[i0]*dy(u)*dy(v))
             + on(Dirichlet,u=0);
              aR[i] = valocal(Vhi,Vhi,solver = UMFPACK);
      }
      if (method == "RAS"){
        matrix<complex> temp = Aglobal*Rih[i]';
        aR[i] = Rih[i]*temp;
        set(aR[i],solver = UMFPACK);
       }
}
/*# finLocalData #*/
/*# debutSchwarzIter #*/
ofstream filei (method+"_ovr"+sizeovr+"_sig"+sigma+"_ep"+epsilon+"_n"+nloc+"_2sd.m");
Vh<complex> un = 0;                              // initial guess
un[] = init;
Vh<complex> rn = rhsglobal;
Vh<complex> er,dr;
int niter;
real err = 0, res;
for(int iter = 0;iter<maxit;++iter){
      dr = 0;
      for(int i = 0;i<npart;++i){
        complex[int] bi = Rih[i]*rn[];    // restriction to the local domain
        complex[int] ui = aR[i] ^-1 * bi; // local solve
        bi = Dih[i]*ui;
        dr[] += Rih[i]'*bi;
      }
      un[] += dr[];              // build new iterate
      rn[] = Aglobal*un[];       // computes global residual
      rn[] = rn[] - rhsglobal[];
      rn[] *= -1;
                  er[] = un[]-uglob[];
                  //cout << "Error = "<< er[][25] << endl;
```

```
    err = er[].l2/ugl2;
    res = rn[].l2/ugl2;
    cout << "It: "<< iter << " Residual = " << res  <<  " Relative L2 Error =  "<<  err << endl;
    Vh [abser] = [abs(er)];
    plot(abser,value=1,dim=3,fill=1,wait=0,cmm="error");
                //plot(abser,value=1,dim=3,fill=1,cmm="error");
    niter = iter+1;
    // Store the error and the residual in Matlab/Scilab/Octave form
    filei << method+"("+niter+")=" << err << ";" << endl;
    if(err < tol) break;
}
/*# finSchwarzIter #*/
```

and of the preconditioned version is `Precond-GMRES-Magnetotelluric.edp`

```
/*# debutPartition #*/
include "./dataMagnetotelluric.edp"
include "./defMagnetotelluric.edp"
include "./decomp.idp"
include "./createPartition.idp"
SubdomainsPartitionUnity(Th,part[],sizeovr,aTh,Rihreal,Dihreal,Ndeg,AreaThi);

//Build a new partition of unity
SubdomainsPartitionUnity(Th,part[],1,aTh0,Rihreal0,Dihreal0,Ndeg0,AreaThi0);
for (int i=0; i < npart; i++) {
   matrix Maux1, Maux2, Maux3;
   Maux1  = Rihreal0[i]*Rihreal[i]';
   Maux2  = Dihreal0[i]*Maux1;
   Maux3  = Rihreal0[i]'*Maux2;
   Dihrealr[i] = Rihreal[i]*Maux3;
}

//plot(part,wait=1,fill=1,ps="Partition");

for (int i=0; i<npart; i++) {
        Rih[i] = Rihreal[i];
        Dih[i] = Dihrealr[i];
  //Dih[i] = Dihreal[i];
  // test the partition of unity
  /* Vh<complex> ux,vx;
  ux[]=1.;
  matrix Maux1, Maux2;
        Maux1 = Dihrealr[i]*Rihreal[i];
  mesh Thi = aTh[i];
  fespace Vhi(Thi,P1);
  Vhi<complex> uxi;
  uxi[] = Maux1*ux[];
        plot(uxi,value=1,fill=1,dim=3,wait=1); */
        //Maux2 = Rihreal[i]'*Maux1;
  //vx[] = Maux2*ux[];
        //plot(vx,value=1,fill=1,dim=3,wait=1);
}

/*# endPartition #*/
/*# debutGlobalData #*/
Aglobal = vaglobal(Vh,Vh,solver = UMFPACK);  // global matrix
rhsglobal[] = vaglobal(0,Vh);                              // global rhs
uglob[] = Aglobal^-1*rhsglobal[];
real ugl2 = uglob[].l2;
// Vh realuglob = real(uglob);
//medit("Solution",Th,realuglob);
/*# finGlobalData #*/

/*# debutLocalData #*/
for(int i = 0;i<npart;++i){
         mesh Thi = aTh[i];
     fespace Vhi(Thi,P1);
     cout << " Domain :" << i << "/" << npart << endl;
     if (method == "ORAS"){
         varf valocal(u,v) =
       int2d(Thi)(Grad(u)'*Grad(v)+(sigma-1i*epsilon)*u*v)
             + int1d(Thi,Robin)(pbc*u*v)
             + int1d(Thi,10)(p*u*v)
             + int1d(Thi,10)(q*dy(u)*dy(v))
             + on(Dirichlet,u=0);
              aR[i] = valocal(Vhi,Vhi,solver = UMFPACK);
      }
      if (method == "RAS"){
        matrix<complex> temp = Aglobal*Rih[i]';
        aR[i] = Rih[i]*temp;
        set(aR[i],solver = UMFPACK);
       }
}
/*# finLocalData #*/

ofstream filei (method+"_ovr"+sizeovr+"_sig"+sigma+"_ep"+epsilon+"_n"+nloc+".m");

include "MTGMRES.idp"
Vh<complex> un;                          // initial guess
for(int i = 0;i<Vh.ndof;++i)
```

```
{
  un[][i] = randreal1()+1i*randreal1();
}
Vh intern;
intern = (x>0) && (x<allong ) && (y>0) && (y<1);
un[].re = un[].re.*intern[];
un[].im = un[].im.*intern[];

Vh <complex> sol,er;
sol[] =  GMRES(un[], tol, maxit);
er[] = un[]-uglob[];
```

The details of the implementation of these preconditioners as well as the complex version of the Krylov solver used here (GMRES with a left preconditioning) are shown in `MTGMRES.idp`

```

// Preconditioned GMRES algorithm Applied to the system
// M^{-1}Aglobal x = M^{-1}b
// Here Aglobal denotes the global matrix
// M^{-1} is the RAS preconditioner based on domain decomposition
// In order to use the GMRES routine define first the matrix-vector product
/*# debutGlobalMatvec #*/
func complex[int] A(complex[int] &vec)
{
        // Matrix vector product with the global matrix
        Vh<complex> Ax;
        Ax[]= Aglobal*vec;
        return Ax[];
}
/*# finGlobalMatvec #*/
/*# debutRASPrecond #*/
// and the application of the preconditioner
func complex[int] PREC(complex[int] &l)
{
   // Application of the preconditioner
   // M^{-1}*y = \sum Ri^T*Di*Ai^{-1}*Ri*y
   // Ri restriction operators, Ai local matrices
   Vh<complex>  s = 0;
   for(int i=0; i<npart; ++i)  {
       complex[int] bi = Rih[i]*l;          // restricts rhs
       complex[int] ui = aR[i] ^-1 * bi;    // local solves
       bi = Dih[i]*ui;                      // partition of unity
       s[] += Rih[i]'*bi;                   // prolongation
      }
   return s[];
```

```
}
/*# finRASPrecond #*/

/*# debutGMRESsolve #*/
func complex[int] GMRES(complex[int] x0, real eps, int nbiter)
{
        int intmetis = withmetis;
        ofstream filei("GMRES_ovr"+sizeovr+"_sig"+sigma+"_ep"+epsilon+"_n"+nloc+".m");

        Vh<complex> r, z,v ,w,er,un;

        Vh<complex>[int] [V](nbiter+1), [Vp](nbiter+1);  // orthonormal basis
        complex[int,int] Hn(nbiter+2,nbiter+1);      // Hessenberg matrix
        Hn = 0.;
        complex[int,int] rot(2,nbiter+2);
        rot = 0.;
        complex[int] g(nbiter+1),g1(nbiter+1);
        g = 0.;  g1 = 0.;
        r[] = A(x0);
        r[] -= rhsglobal[];
        r[] *= -1.0;

        z[] = r[];
        g[0] = z[].l2;       // initial residual norm

        //filei << "relres("+1+")=" << g[0] << ";" << endl;
        V[0][]=1/g[0]*z[];              // first basis vector
        for(int it=0; it<nbiter; it++){
                   Vp[it][] = PREC(V[it][]);
                 v[] = Vp[it][];
                 w[] = A(v[]);   // w = A*M^{-1}V_it

                 for(int i=0; i<it+1; i++) {
            Hn(i,it) = w[]'*V[i][];
            w[] -=  conj(Hn(i,it))*V[i][];
         }
                 Hn(it+1,it) = sqrt(real(w[]'*w[]));

                 complex aux = Hn(it+1,it);
                 for(int i=0; i<it; i++){        // QR decomposition of Hn
                         complex aa = conj(rot(0,i))*Hn(i,it)+conj(rot(1,i))*Hn(i+1,it);
                         complex bb = -rot(1,i)*Hn(i,it)+rot(0,i)*Hn(i+1,it);
                         Hn(i,it) = aa;
                         Hn(i+1,it) = bb;
       }
```

```
        complex sq = sqrt( conj(Hn(it,it))*Hn(it,it) + Hn(it+1,it)*Hn(it+1,it) );
        rot(0,it) = Hn(it,it)/sq;
        rot(1,it) = Hn(it+1,it)/sq;
            Hn(it,it) = conj(rot(0,it))*Hn(it,it)+conj(rot(1,it))*Hn(it+1,it);
            Hn(it+1,it) =  0.;
                    g[it+1] = -rot(1,it)*g[it];
          g[it] = conj(rot(0,it))*g[it];

                    complex[int] y(it+1);           // Reconstruct the solution
                    for(int i=it; i>=0; i--) {
                             g1[i] = g[i];
                             for(int j=i+1; j<it+1; j++){
                                     g1[i] = g1[i]-Hn(i,j)*y[j];
                              }
                             y[i]=g1[i]/Hn(i,i);
         }
                    un[] = x0;
                    for(int i=0;i<it+1;i++){
              un[]= un[]+ conj(y[i])*Vp[i][];
                       }
        er[] = un[] - uglob[];
        real relres = abs(g[it+1]);
    real relerr = er[].l2/uglob[].l2;
                             Vh abser = abs(er);
                             Vh absun = abs(un);
                      //plot(abser, dim=3, cmm="Error at step " + it, value=1, fill=1);
                             //plot(abser, dim=3, cmm="Error at step " + it, value=1, fill=1, wait=1);
                              //plot(absun, dim=3, cmm="Solution at step " + it, value=1, fill=1);
        cout << "It: "<< it << " Residual = " << relres  <<  " Relative L2 Error =  "<< relerr << endl;
        int j = it+2;
        int l = j-1;
                             filei << "GMRES_ovr"+sizeovr+"_sig"+sigma+"_ep"+epsilon+"_n"+nloc+" ("+l+")=" << relerr << ";" << endl;
        if(relerr < eps) {//relres
               cout << "GMRES has converged in " + it + " iterations " << endl;
               cout << "The relative residual is " +  relres << endl;
               break;      }
        V[it+1][]=1/aux*w[];

      }
      return un[];
}
```

```
/*# finGMRESsolve #*/
```

Bibliography

[Amd67] Gene M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *AFIPS Proceedings*, pages 483–485, 1967.

[BDG$^+$19a] M. Bonazzoli, V. Dolean, I. G. Graham, E. A. Spence, and P.-H. Tournier. A 2-level domain decomposition preconditioner for the time-harmonic Maxwell's equations. *Math. Comp.*, 88:2559–2604, 2019.

[BDG19b] Romain Brunet, V. Dolean, and M. J. Gander. Natural domain decomposition algorithms for the solution of time-harmonic elastic waves. *arXiv:1904.12158*, 2019.

[BDJT21] Niall Bootland, Victorita Dolean, Pierre Jolivet, and Pierre-Henri Tournier. A comparison of coarse spaces for helmholtz problems in the high frequency regime. *Computers and Mathematics with Applications*, 98:239–253, 2021.

[BO99] C. M. Bender and S. A. Orszag. *Advanced Mathematical Methods for Scientists and Engineers I: Asymptotic Methods and Perturbation Theory.* Springer-Verlag, New York, 1999.

[BS97] Ivo M. Babuska and Stefan A. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Journal on numerical analysis*, 34(6):2392–2423, 1997.

[CCGV18] F. Chaouqui, G. Ciaramella, M. J. Gander, and T. Vanzan. On the scalability of classical one-level domain-decomposition methods. *Vietnam J. Math.*, 46:1053–1088, 2018.

[CG17] G. Ciaramella and M. J. Gander. Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part I. *SIAM J. Numer. Anal.*, 55(3):1330–1356, 2017.

[CG18a] G. Ciaramella and M. J. Gander. Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part II. *SIAM J. Numer. Anal.*, 56(3):1498–1524, 2018.

[CG18b] Gabriele Ciaramella and Martin J. Gander. Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part III. *Electron. Trans. Numer. Anal.*, 49:210–243, 2018.

[CHS20] Gabriele Ciaramella, Muhammad Hassan, and Benjamin Stamm. On the scalability of the Schwarz method. *SMAI J. Comput. Math.*, 6:33–68, 2020.

[CMS13] Eric Cances, Yvon Maday, and Benjamin Stamm. Domain decomposition for implicit solvation models. *J. Chem. Phys.*, 139(5), 2013.

[Con15] Lea Conen. Domain decomposition preconditioning for the Helmholtz equation: a coarse space based on local Dirichlet-to-Neumann maps. *Ph.D. Thesis, Faculty of Informatics, Università della Svizzera italiana*, 2015.

[CS99] X.-Ch. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21(2):792–797 (electronic), 1999.

[DGG09] V. Dolean, L. Gerardo Giorda, and M. J. Gander. Optimized Schwarz methods for Maxwell equations. *SIAM J. Scient. Comp.*, 31(3):2193–2213, 2009.

[DGH19] Fabrizio Donzelli, Martin J. Gander, and Ronald D. Haynes. A Schwarz method for the magnetotelluric approximation of Maxwell's equations, 2019.

[DGL$^+$15] Victorita Dolean, Martin J. Gander, Stephane Lanteri, Jin-Fa Lee, and Zhen Peng. Effective transmission conditions for domain decomposition methods applied to the time-harmonic curl-curl Maxwell's equations. *J. Comput. Phys.*, 280:232–247, 2015.

[DJN15] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015. Algorithms, theory, and parallel implementation.

[DJR92] Bruno Després, Patrick Joly, and Jean E. Roberts. A domain decomposition method for the harmonic Maxwell equations. In *Iterative methods in linear algebra (Brussels, 1991)*, pages 475–484. North-Holland, Amsterdam, 1992.

[DJTO20] V. Dolean, P. Jolivet, P.-H. Tournier, and S. Operto. Iterative frequency-domain seismic wave solvers based on multi-level domain-decomposition preconditioners. In *82^{th} Annual EAGE Meeting (Amsterdam)*, volume arXiv:2004.06309, 2020.

[DLP08] V. Dolean, S. Lanteri, and R. Perrussel. A domain decomposition method for solving the three-dimensional time-harmonic Maxwell equations discretized by discontinuous Galerkin methods. *J. Comput. Phys.*, 227(3):2044–2072, 2008.

[DNSC12] Marco Donatelli, Maya Neytcheva, and Stefano Serra-Capizzano. Canonical eigenvalue distribution of multilevel block Toeplitz sequences with non-Hermitian symbols. In *Spectral Theory, Mathematical System Theory, Evolution Equations, Differential and Difference Equations*, pages 269–291. Springer, 2012.

[Dru00] Paul Drude. Zur elektronentheorie der metalle. *Annalen der Physik*, 306(3):566–613, 1900.

[EDGL12] M. El Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri. Optimized Schwarz methods for the time-harmonic Maxwell equations with dampimg. *SIAM J. Scient. Comp.*, 34(4):2048–2071, 2012.

[EG12] O. G. Ernst and M. J. Gander. Why it is difficult to solve Helmholtz problems with classical iterative methods. In *Numerical analysis of multiscale problems*, volume 83 of *Lect. Notes Comput. Sci. Eng.*, pages 325–363. Springer, Heidelberg, 2012.

[GC21] M. Gander and G. Ciaramella, editors. *Iterative Methods and Preconditioners for Systems of Linear Equations*. SIAM, 2021.

[GGS20] Shihua Gong, Ivan G. Graham, and Euan A. Spence. Domain decomposition preconditioners for high-order discretisations of the heterogeneous Helmholtz equation. *arXiv:2004.03996*, 2020.

[GHM07] Martin J Gander, Laurence Halpern, and Frédéric Magoules. An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. *International journal for numerical methods in fluids*, 55(2):163–175, 2007.

[GMN02] Martin J Gander, Frédéric Magoulès, and Frédéric Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.

[GSV17] I. G. Graham, E. A. Spence, and E. Vainikko. Recent results on domain decomposition preconditioning for the high-frequency Helmholtz equation using absorption. *Lahaye D., Tang J., Vuik K. (eds) Modern Solvers for Helmholtz Problems. Geosystems Mathematics. Birkhäuser, Cham*, pages 3–26, 2017.

[GSZ20] I. G. Graham, E. A. Spence, and J. Zou. Domain decomposition with local impedance conditions for the Helmholtz equation. *SIAM J. Numer. Anal.*, 58(5):2515–2543, 2020.

[Gus88] John L. Gustafson. Reevaluating amdahl's law. *Communications of the ACM*, pages 532–533, 1988.

[GZ19] Martin J. Gander and Hui Zhang. A class of iterative solvers for the Helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods. *SIAM Review*, 61(1):3–76, 2019.

[Hir67] I. I. Hirschman. The spectra of certain Toeplitz matrices. *Illinois J. Math.*, 11:145–159, 1967.

[KK98] G. Karypis and V. Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Technical report, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN, 1998.

[Kra99] S. G. Krantz. *Handbook of Complex Variables.* Birkhäuser, Boston, MA, 1999.

[Kry31] A. N. Krylov. On the numerical solution of equation by which are determined in technical problems the frequencies of small vibrations of material systems. *Izvestiia Akademii nauk SSSR*, pages 491–539, 1931.

[Lio89] P.-L. Lions. On the Schwarz alternating method. II. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, pages 47–70, Philadelphia, PA, 1989. SIAM.

[Lio90] P.-L. Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.

[LS13] J. Liesen and Z. Strakos, editors. *Krylov Subspace Methods: Principles and Analysis.* Oxford University Press, 2013.

[MDG19] Vanessa Mattesi, Marion Darbas, and Christophe Geuzaine. A high-order absorbing boundary condition for 2D time-harmonic elastodynamic scattering problems. *Computers & Mathematics with Applications*, 77(6):1703–1721, 2019.

[Mor62] C. S. Morawetz. The limiting amplitude principle. *Comm. Pure Appl. Math*, 15:349–361, 1962.

[Ned01] J.-C. Nedelec. *Acoustic and electromagnetic equations. Integral representations for harmonic problems.* Applied Mathematical Sciences, 144. Springer Verlag, 2001.

[QV99] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford Science Publications, 1999.

[Saa03] Y. Saad, editor. *Iterative Methods for Sparse Linear Systems- 2nd edition.* SIAM, 2003.

[SBG96] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[SC02] S. Serra-Capizzano. More inequalities and asymptotics for matrix valued linear positive operators: the noncommutative case. In *Toeplitz Matrices and Singular Integral Equations*, pages 293–315. Springer, 2002.

[SCGT07] Amik St-Cyr, Martin J. Gander, and Stephen J. Thomas. Optimized Multiplicative, Additive, and Restricted Additive Schwarz Preconditioning. *SIAM J. Sci. Comput.*, 29(6):2402–2425 (electronic), 2007.

[SM13] Aliaksei Sandryhaila and Jose M. F. Moura. Eigendecomposition of Block Tridiagonal Matrices. *arXiv e-prints*, page arXiv:1306.0217, June 2013.

[SS60] P. Schmidt and F. Spitzer. The Toeplitz matrices of an arbitrary Laurent polynomial. *Math. Scand.*, 8:15–38, 1960.

[Til98] P. Tilli. A note on the spectral distribution of Toeplitz matrices. *Linear Multilin. Algebra*, 45(2-3):147–159, 1998.

[Tis87] M. Tismenetsky. Determinant of block-Toeplitz band matrices. *Linear Algebra and its Applications*, 85:165–184, 1987.

[Tra14] Khang Tran. Connections between discriminants and the root distribution of polynomials with rational generating function. *Journal of Mathematical Analysis and Applications*, 410(1):330 – 340, 2014.

[TW05] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.

[Wid74] H. Widom. Asymptotic behavior of block Toeplitz matrices and determinants. *Adv. Math.*, 13:284–322, 1974.

[Yee66] Kane S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, AP-14(3):302–307, 1966.