Domain decomposition preconditioning for the Helmholtz equation

A coarse space based on local Dirichlet-to-Neumann maps

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Lea Conen

under the supervision of
Prof. Rolf Krause

January 2015

Dissertation Committee

Prof. Martin Gander	Université de Genève, Geneva, Switzerland
Prof. Helmut Harbrecht	Universität Basel, Basel, Switzerland
Prof. Igor Pivkin	Università della Svizzera italiana, Lugano, Switzerland
Prof. Olaf Schenk	Università della Svizzera italiana, Lugano, Switzerland

Dissertation accepted on 7 January 2015

Research Advisor	PhD Program Director
Prof. Rolf Krause	**Prof. Igor Pivkin, Prof. Stefan Wolf**

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Lea Conen
Lugano, 19 January 2015

Divide each difficulty into as many parts as
is feasible and necessary to resolve it.

René Descartes (1596–1650)

Abstract

In this thesis, we present a two-level domain decomposition method for the iterative solution of the heterogeneous Helmholtz equation. The Helmholtz equation governs wave propagation and scattering phenomena arising in a wide range of engineering applications. Its discretization with piecewise linear finite elements results in typically large, ill-conditioned, indefinite, and non-Hermitian linear systems of equations, for which standard iterative and direct methods encounter convergence problems. Therefore, especially designed methods are needed. The inherently parallel domain decomposition methods constitute a promising class of preconditioners, as they subdivide the large problems into smaller subproblems and are hence able to cope with many degrees of freedom. An essential element of these methods is a good coarse space. Here, the Helmholtz equation presents a particular challenge, as even slight deviations from the optimal choice can be fatal.

We develop a coarse space that is based on local eigenproblems involving the Dirichlet-to-Neumann operator. Our construction is completely automatic, ensuring good convergence rates without the need for parameter tuning. Moreover, it naturally respects local variations in the wave number and is hence suited also for heterogeneous Helmholtz problems. Apart from the question of how to design the coarse space, we also investigate the question of how to incorporate the coarse space into the method. Also here the fact that the stiffness matrix is non-Hermitian and indefinite constitutes a major challenge. The resulting method is parallel by design and its efficiency is investigated for two- and three-dimensional homogeneous and heterogeneous numerical examples.

Acknowledgements

I would never have been able to finish my dissertation without the guidance and support of many people. Foremost, I would like to express my gratitude to my supervisor, Prof. Rolf Krause, who did not only introduce me to the field of numerical mathematics during my Diploma studies in Bonn, but also guided me through the last years of research. I am especially indebted to Dr. Victorita Dolean (University of Strathclyde, UK) and Prof. Frédéric Nataf (Université Pierre et Marie Curie, France) for their interest in my work and many stimulating discussions. I am grateful to my colleagues at the ICS for always having an open ear for my problems and in particular for the countless coffee breaks. Special thanks go to the members of my committee, for agreeing to review this thesis. Last, but not least, I would also like to thank my friends and family for their support.

Contents

Figures

Tables

List of algorithms

Introduction

The Helmholtz equation $-\Delta u - k^2 u = f$, where u is the unknown function, f the right-hand side and $k > 0$ the wave number, governs wave propagation and scattering phenomena arising in a wide range of engineering applications such as aeronautics, acoustics and geophysical seismic imaging. Even though it looks deceivingly similar to the well-understood Laplace problem, the additional zeroth order term completely changes its behavior. As the wave number k becomes larger, there is an increasing number of negative eigenvalues in the spectrum of the Helmholtz operator and its condition number deteriorates. Therefore, at higher frequencies, the indefinite, ill-conditioned, and non-Hermitian system of linear equations arising from the finite element (FE) discretization of the Helmholtz equation is difficult to solve numerically. Developing an efficient iterative solution method for this system is the goal of this thesis.

An additional challenge for the numerical solution is the typically large size of the system of linear equations. While choosing a minimum number of grid points per wave length is sufficient to interpolate the wave-like solutions, it is not enough for the discretized system to accurately represent the continuous one due to the pollution effect [8]. Therefore, with growing wave number k, not only does the problem become more difficult, but also the size of the linear system of equations increases quickly, particularly in the three-dimensional case. Thus efficient preconditioners are of utter importance, especially for high-frequent problems.

Direct methods such as incomplete factorization preconditioners yield fast black-box preconditioners for a wide range of problems [132], but are not efficient for the Helmholtz equation [46]. While various modifications of this class of methods for the Helmholtz equation have been studied, resulting for example in incomplete factorization methods [13, 105, 120] and the "sweeping preconditioner" [39], in this work we focus on iterative solution strategies.

Unfortunately, standard iterative methods also suffer from convergence problems when applied to indefinite, non-Hermitian problems. Therefore, special care needs to be taken when designing iterative algorithms for the Helmholtz equation. Among the iterative methods that have been considered are multigrid methods. The problems that occur when applying geometric multigrid methods to the Helmholtz equation have been analyzed in detail [18, 37]. Based on this understanding, various modifications of the standard components have been proposed, see e.g. [37, 47, 90]. Particularly interesting is the wave-ray multigrid [18]. Here, special levels based on plane waves are introduced, designed to represent the oscillatory part of the solution. Preconditioning with a shifted, easier problem is investigated e.g. in [10, 43].

In this work, we concentrate on domain decomposition methods (DDMs) [60, 147]. Their inherent parallelism helps to cope with the typically large systems of linear equations. Unfortunately, the classical DDMs are not effective for the Helmholtz equation. One important, problematic component are the transmission conditions, specifying the information exchange between neighboring subdomains. Early work on this part was done in [32], where a first order approximation to the Sommerfeld radiation condition is employed. In the sequel, different, more advanced techniques have been used; including the perfectly matched layer (PML) method at the interfaces [134, 146], non-local transmission conditions [27], optimized Schwarz methods [57, 64], and others, e.g. [15]. The second important component is the coarse space, allowing for global transfer of information. Here, plane waves have received a lot of interest. Apart from having been used in the multigrid context [18], they have been successfully employed as coarse space basis functions for DDMs. Their evaluation at the interfaces of the subdomains are used in variants of the finite element tearing and interconnecting (FETI) method [49, 52]. They have also been utilized in other DDMs [92, 101] and as deflation vectors [4]. Plane waves, to our knowledge, have been employed mainly for homogeneous problems; the extension to the heterogeneous case is not obvious.

The goal of this thesis is the development of a two-level DDM for the heterogeneous Helmholtz equation. Our method is based on a standard one-level restricted additive Schwarz (RAS) method with Robin-type transmission conditions, and the emphasis lies on the definition of the second level. In a first step, we investigate different ways how to add a second level to the indefinite, non-Hermitian one-level DDM. While this question is well-understood for the symmetric positive definite (s.p.d.) case, for the Helmholtz equation problems arise. In particular, we discuss under which circumstances it is possible to ensure that the convergence rates for the two-level method are not worse than for the one-level method. In a second step, we define a new coarse space for the Helmholtz equation. For that purpose, we adapt an idea for elliptic problems [33, 116]: The coarse space is based on local functions, the solutions of eigenproblems involving the Dirichlet-to-Neumann (DtN) operator on the subdomains' interfaces. Its construction is completely automatic, refraining from the need for parameter tuning. This feature is crucial for indefinite problems as in contrast to the elliptic case, even slight deviations from the optimal choice can be fatal [53]. We investigate the resulting two-level DDM, and in particular its robustness with respect to heterogeneous coefficients, numerically for two- and three-dimensional examples.

The thesis is structured as follows. Chapter 1 is an introduction to the Helmholtz equation and its discretization. We review the related literature with a particular focus on DDMs in Chapter 2. In Chapter 3, we present the one-level RAS method that serves as the basic, one-level preconditioner and discuss its convergence behavior with the help of Fourier analysis. Different ways to add a second level to a one-level method are discussed in Chapter 4. Here, the emphasis lies on the fact that the systems of interest are typically indefinite and non-Hermitian, which imposes additional difficulties in this step. In Chapter 5, we introduce the new coarse space based on DtN eigenvalue problems and motivate it via some preliminary numerical experiments. We test the coarse space and the resulting preconditioners extensively for different two-dimensional experiments and compare the performance to a coarse space based on plane waves in Chapter 6. In Chapter 7, we finally do three-dimensional tests and look at larger problems and examine the scalability of our approach.

Chapter 1

The Helmholtz problem

This chapter introduces all the prerequisites that are necessary for the rest of this manuscript. After introducing the basic notations and definitions in Section 1.1, in Section 1.2 we present the problem we are interested in, the Helmholtz equation. We discuss its discretization with piecewise linear finite elements (FEs) in Section 1.3. In Section 1.4, we define the model problems used throughout this manuscript in the numerical experiments.

1.1 Basic definitions and notation

In this section, we introduce some basic definitions and notation. For details see any textbook on functional analysis, e.g. [2].

Fields We denote by $\mathbb{R}$ the field of real numbers, by $\mathbb{C}$ the field of complex numbers, by $\mathbb{N}$ the set of natural numbers and by $\mathbb{N}_0$ the set $\mathbb{N}\cup\{0\}$. Furthermore, we write $\mathbb{K}$ for $\mathbb{K}=\mathbb{R}$ or $\mathbb{K}=\mathbb{C}$. Let $|\alpha|$ be the *absolute value* of a number $\alpha\in\mathbb{K}$ and $\overline{\alpha}$ be the complex conjugate of α. We denote by $\imath$ the imaginary unit and for $\alpha=\alpha_1+\imath\alpha_2\in\mathbb{C}$, $\alpha_1,\alpha_2\in\mathbb{R}$, let $\mathrm{re}(\alpha):=\alpha_1$ be the *real part* of α and $\mathrm{im}(\alpha):=\alpha_2$ be the *imaginary part* of α.

Euclidean product Let $\langle .,.\rangle:\mathbb{K}^n\times\mathbb{K}^n\to\mathbb{K}$ denote the *scalar product* on $\mathbb{K}^n$ defined for $\mathbf{x}=(x_i)_{i=1}^n,\mathbf{y}=(y_i)_{i=1}^n\in\mathbb{K}^n$ by $\langle\mathbf{x},\mathbf{y}\rangle=\sum_{i=1}^n y_i\cdot\overline{x}_i$.

Sets Let (X,d) be a metric space. For $A\subset X$, let $\mathring{A}$ denote the *interior of* A and $\overline{A}$ the *closure of* A. The *boundary* ∂A of A is defined as $\partial A:=\overline{A}\setminus\mathring{A}$.

Functions and derivatives Let $\Omega\subset\mathbb{R}^n$ and let Y be a Banach space. For $x\in\Omega$, we denote by $\partial_i f(x)$ or $\frac{\partial}{\partial x_i}f(x)$ the *i-th partial derivative* of $f:\Omega\to Y$ in the point x and by $\partial_v f$ or $\frac{\partial}{\partial v}f:\mathbb{R}^n\to Y$ the *directional derivative* of f in direction $v\in\mathbb{R}^n$.

Spaces of differentiable functions For each multi-index $s=(s_1,\ldots,s_d)\in\mathbb{N}^d$ we set $|s|=\sum_{i=1}^d s_i$ and $\partial^s=\frac{\partial^{|s|}}{\partial_1^{s_1}\ldots\partial_d^{s_d}}v$. Let $\Omega\subset\mathbb{R}^n$ open and bounded, $m\geq 0$, and Y a Banach space. By $C^m\left(\overline{\Omega},Y\right)$ or $C^m\left(\overline{\Omega}\right)$ we denote the vector space of functions $f:\overline{\Omega}\to Y$ that are m times

continuously differentiable in Ω and for which $\partial^s f$ can be extended continuously to $\overline{\Omega}$ for all multi-indices $s \in \mathbb{N}^d$ satisfying $|s| \leq m$. We additionally define

$$C^\infty(\Omega) = \bigcap_{m\in\mathbb{N}} C^m(\Omega).$$

$C_0^\infty(\Omega)$ are those functions in $C^\infty(\Omega)$ that have compact support.

Lebesgue spaces Let Y be a Banach space and $(\Omega, \mathcal{B}, \mu)$ be a measure space. For $1 \leq p < \infty$, we denote by $L^p(\mu, Y)$ the *Lebesgue space of order p*. Here and in the following all integrals are Lebesgue integrals and measures refer to the Lebesgue measure. If all others choices are clear, we also write $L^p(\Omega)$ for $L^p(\mu, Y)$. We define the norm

$$\|u\|_{L^p} = \left(\int_\Omega |u(\mathbf{x})|^p \, \mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}.$$

Moreover, for $p = \infty$ we denote by $L^\infty(\Omega)$ the space of all measurable, essentially bounded functions $u : \Omega \to Y$ with norm

$$\|u\|_{L^\infty} = \inf\{C \geq 0 : |u(x)| \leq C \text{ for almost every } x \in \Omega\}.$$

For $p = 2$, $L^p(\Omega)$ is a Hilbert space with inner product

$$(u, v)_{L^2} = \int_\Omega u(x) \cdot \overline{v(x)} \, \mathrm{d}x.$$

Sobolev spaces For $m \in \mathbb{N}$ and $1 \leq p \leq \infty$ the Sobolev space $H^{m,p}(\Omega)$ is defined by

$$H^{m,p}(\Omega) = \Big\{ f \in L^p(\Omega) : \text{ For each } s \in \mathbb{N}^d, |s| \leq m \quad \text{there exists an } f^{(s)} \in L^p(\Omega) \\ \text{with } \int_\Omega f \cdot \partial^s \zeta = (-1)^{|s|} \int_\Omega f^{(s)} \zeta \ \forall \zeta \in C^\infty(\Omega) \Big\},$$

equipped with the norm

$$\|f\|_{H^{m,p}(\Omega)} := \left(\sum_{|s|\leq m} \left\| f^{(s)} \right\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}}.$$

For the frequently used case $p = 2$, we set $H^m(\Omega) := H^{m,2}(\Omega)$. This is a Hilbert space endowed with the inner product $(.,.)_{H^m(\Omega)}$ defined by

$$(u, v)_{H^m(\Omega)} = \sum_{|\alpha|\leq m} \int_\Omega \partial^\alpha u(x) \overline{\partial^\alpha v(x)} \, \mathrm{d}x.$$

Furthermore, let

$$H_0^{m,p}(\Omega) := \{ f \in H^{m,p}(\Omega) : \exists f_k \in C_0^\infty(\Omega), k \in \mathbb{N} \\ \text{with } \|f - f_k\|_{H^{m,p}} \to 0 \text{ for } k \to \infty \}.$$

Landau symbols We introduce the Landau symbols $O(.)$ and $o(.)$. We write $f(x) = O(g(x))$ for $x \to \infty$, if there exist a positive constant $C \in \mathbb{R}$ and a real number $x_0 \in \mathbb{R}$ such that

$$\|f(x)\| \leq C\,\|g(x)\| \qquad \forall x > x_0.$$

We write $f(x) = o(g(x))$ for $x \to \infty$ if for every positive constant ε there exists a constant x_0 such that

$$\|f(x)\| \leq \varepsilon\,\|g(x)\| \qquad \forall x > x_0.$$

Kronecker delta For $i, j \in \mathbb{N}$, δ_{ij} denotes the *Kronecker delta*, i.e.

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j. \end{cases}$$

Matrices Let $A \in \mathbb{K}^{n\times n}$, $n \in \mathbb{N}$, be a matrix. By $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ we denote the maximum and minimum (non-zero) eigenvalue by modulus, respectively. If A is normal, the condition number $\kappa(A)$ of A is given by

$$\kappa(A) = \left|\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}\right|.$$

1.2 Helmholtz equation

In this section, we introduce Helmholtz equation, a partial differential equation (PDE) of the form

$$-\Delta u - k^2 u = f \tag{1.2.1}$$

in some domain $\Omega \subseteq \mathbb{R}^d$, where Δ is the Laplace operator, f is a right-hand side function, u is the unknown solution and k is the wave number that might either be constant or depend on the position $x \in \mathbb{R}^d$.

1.2.1 Relation to the wave equation

The Helmholtz equation is a special case of the *wave equation*, which describes the propagation of waves, such as sound, light or water waves, in a medium. Denoting by c a constant that is related to the propagation speed of the waves in the given medium, the wave equation reads

$$\Delta U - \frac{1}{c^2}\frac{\partial^2}{\partial t^2} U = 0. \tag{1.2.2}$$

The unknown U is a scalar function, whose values model the displacement of the wave. With the assumption of time harmonic waves, i.e.

$$U(\mathbf{x}, t) = u(\mathbf{x})\mathrm{e}^{-\imath\omega t}, \tag{1.2.3}$$

where ω is the *circular frequency*, we get

$$\Delta U - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}U = \left(\Delta u(\mathbf{x}) - \frac{\omega^2}{c^2}u(\mathbf{x})\right)\mathrm{e}^{-\imath\omega t} = 0$$

and consequently, since the exponential is non-zero, we arrive at the Helmholtz equation

$$\Delta u + k^2 u = 0 \qquad \text{with } k := \frac{\omega}{c}. \tag{1.2.4}$$

Here, as in Equation (1.2.1), k is the *wave number*. The Helmholtz equation describes hence the propagation of time-harmonic waves in a medium.

1.2.2 The Sommerfeld radiation condition

In an unbounded domain Ω, energy that is emitted from a source must scatter to infinity. Likewise, incoming waves might represent nonphysical behavior as no waves should be reflected from infinity. The *Sommerfeld radiation condition* [139] enforces these properties for the solution of the PDE. Suppose that the unbounded domain is truncated by a sphere S_R of sufficiently large radius R. The Sommerfeld radiation condition then reads for the solution u of Equation (1.2.4) using the Landau symbols $O(.)$ and $o(.)$ [79, Chapter 3]

$$u = O\Big(R^{-(d-1)/2}\Big), \quad \imath k u - \frac{\mathrm{d}u}{\mathrm{d}R} = o\Big(R^{-(d-1)/2}\Big), \quad R \to \infty. \tag{1.2.5}$$

These two equations characterize the decay and the directional character, respectively, of the stationary solution in the far field. Any function that satisfies both the Helmholtz equation and the second equation in the Sommerfeld condition in Equation (1.2.5), the *radiation condition*, automatically satisfies the first equation in Equation (1.2.5), the *decay condition*, cf. [79, Remark 1.4] and references therein. If the unbounded domain Ω is truncated to a bounded one to facilitate numerical computations, Equation (1.2.5), which gives asymptotical formulas, needs to be approximated. We discuss this in Subsection 1.3.1.

1.2.3 Formulation of the Helmholtz problem

Using the definitions of the preceding section, the interior Helmholtz problem , which this work investigates, is of the following form: Let $\Omega \subset \mathbb{R}^d$, $d = 2,3$, be a possibly unbounded domain with polygonal boundary. Find $u : \Omega \to \mathbb{C}$ s.t.

$$-\Delta u - k^2 u = f \qquad \text{in } \Omega, \tag{1.2.6a}$$

$$u = 0 \qquad \text{on } \Gamma_D, \tag{1.2.6b}$$

$$\frac{\partial u}{\partial n} = g \qquad \text{on } \Gamma_N, \tag{1.2.6c}$$

where u satisfies the Sommerfeld radiation condition in Equation (1.2.5) and $\Gamma_D \cup \Gamma_N = \Gamma := \partial\Omega$ is a disjoint partition of the boundary $\partial\Omega$. The wave number k is given by $k(x) = \omega/c(x)$, where ω is the angular frequency and c is the speed of propagation that might depend on the location $x \in \Omega$.

1.3 The discretized problem

Section 1.2 introduced the Helmholtz equation in the continuous setting. Infinite dimensional objects can in general not be simulated on a computer, where all quantities need to be of finite size. The standard procedure for obtaining a numerical solution of any PDE is therefore first replacing the PDE by its discrete formulation and then solving the discretized problem numerically [127]. In case of the Helmholtz equation, there are various methods that realize this, including the finite difference method [73, 99, 104, 137, 142, 145], the FE method [16, 19, 26] and the boundary element method [133, 141, 152]. References [6–8, 31, 79–82, 95, 109], among others, discuss FE methods for the Helmholtz equation. This section introduces the FE method that we use for discretization.

1.3.1 Truncation of the computational domain and absorbing boundary conditions

The domain Ω in the continuous formulation of the Helmholtz equation is possibly unbounded. For that case, in Equation (1.2.5), we introduced the Sommerfeld radiation condition. When discretization techniques such as FEs or finite differences are used, working with an unbounded domain is not feasible due to, e.g. restricted hardware resources. Therefore, the domain Ω needs to be truncated to a computational domain $\tilde{\Omega}$ in such a way that a computational and physical compromise is reached. In this truncated domain, the Sommerfeld radiation condition in Equation (1.2.5) is not applicable, as it defines the asymptotic behavior of the solution and hence cannot be used without modifications in a bounded domain.

Various techniques have been developed in order to simulate the radiation condition in the finite computational domain. Infinite element schemes, for example, employ complex-valued basis functions with outwardly propagating wave-like behavior to represent the unbounded complement [71]. Another approach is the use of absorbing layers. Here, the computational domain $\tilde{\Omega}$ is enlarged by an additional layer of finite thickness, which is used to damp the outgoing waves. In the ideal case, waves arriving from any direction are not reflected at the layer. Due to the damping, at the outer boundary of the layer, hard-wall boundary conditions can be employed. One possible implementation of this technique is the perfectly matched layer (PML) formulation [12]. Other discretization techniques such as the boundary element method [133, 141, 152] are by nature able to work with unbounded domains. They have other restrictions, though, for example on the geometry or on the variations in the coefficients.

In this work, we use absorbing boundary conditions. Here, on the artificial boundary of the truncated domain $\tilde{\Omega}$, a relation of the unknown solution and its derivatives is specified. *Non*-local boundary conditions can be used to mimic the radiation condition in Equation (1.2.5) [67, 88, 130, 149]. Despite their accuracy, these conditions are not practical for our aims because of their non-locality – FE discretizations rely on the locality of basis functions. A remedy is to use local approximations, see e.g. [9, 38, 67]; for higher order ones cf. the survey in [68]. As we are interested in an efficient iterative solution strategy, and not in the accuracy of the discrete solution, we use a simple, first-order approximation of the Sommerfeld radiation condition [32]

$$\frac{\partial u}{\partial n} + \imath k u = 0$$

on the part of the boundary, where the domain has been truncated. Assuming additionally that both the domain and its truncation are polygonal and denoting from now on the truncated domain simply by Ω, the full system of equations then reads: Let $\Omega \subset \mathbb{R}^d$, $d = 2,3$, be a polygonal, bounded domain. Find $u : \Omega \to \mathbb{C}$ s.t.

$$-\Delta u - k^2 u = f \qquad \text{in } \Omega, \tag{1.3.1a}$$

$$u = 0 \qquad \text{on } \Gamma_D, \tag{1.3.1b}$$

$$\frac{\partial u}{\partial n} = g \qquad \text{on } \Gamma_N, \tag{1.3.1c}$$

$$\frac{\partial u}{\partial n} + \imath k u = 0 \qquad \text{on } \Gamma_R, \tag{1.3.1d}$$

where Γ_R is the part of the boundary of the domain that has been obtained by truncation of the unbounded domain and $\Gamma_D \cup \Gamma_N \cup \Gamma_R = \Gamma := \partial\Omega$ is a disjoint partition of the boundary Γ. We abbreviate the boundary conditions in the form $\mathcal{C}(u) = 0$ on Γ. The wave number k is given by $k(x) = \omega/c(x)$, where ω is the angular frequency and c is the speed of propagation, cf. Equation (1.2.6).

1.3.2 Finite element discretization

In Equation (1.3.1) of Section 1.2, we have given the strong form of the Helmholtz problem we are interested in. In this section we derive its FE formulation for a bounded domain Ω, assuming that the initially possibly unbounded domain has been truncated, as explained in Subsection 1.3.1. FE methods are based on a weak, variational formulation. Using the notation introduced in Section 1.1, the variational formulation of Equation (1.3.1) is: Find $u \in \mathcal{V} := \{u \in H^1(\Omega) : \ u = 0 \text{ on } \Gamma_D\}$ s.t.

$$a(u,v) = F(v) \qquad \forall v \in \mathcal{V}, \tag{1.3.2}$$

where $a(.,.) : \mathcal{V} \times \mathcal{V} \to \mathbb{C}$ and $F : \mathcal{V} \to \mathbb{C}$ are defined by

$$a(u,v) = \int_\Omega \left(\nabla u \overline{\nabla v} - k^2 u \bar{v}\right) \mathrm{d}x + \int_{\Gamma_R} \imath k u \bar{v} \mathrm{d}s, \quad F(v) = \int_\Omega f \bar{v} \mathrm{d}x.$$

Problem (1.3.2) is well-posed if $\Gamma_R \neq \emptyset$ [79, Chapter 2]. If a solution exists, we call the problem *weakly solvable* and the solution u a *variational* or a *weak solution*.

As a next step, we define the discretized system based on the variational formulation in Equation (1.3.2). Let the polygonal domain Ω be discretized with a uniform triangular mesh $\mathcal{T}_h$, where h is the maximum diameter of the triangles in the mesh. We use piecewise linear FEs, see e.g. [16], to keep the setting as simple as possible. Denoting by $\mathcal{V}_h \subset \mathcal{V}$ the corresponding FE space associated to the mesh $\mathcal{T}_h$, the variational formulation for the discrete spaces reads: Find $u_h \in \mathcal{V}_h$ such that

$$a(u_h, v_h) = F(v_h) \qquad \forall v_h \in \mathcal{V}_h. \tag{1.3.3}$$

With $\{\phi_k\}_{k=1}^n$ the nodal linear FE basis for $\mathcal{V}_h$, $n := \dim(\mathcal{V}_h)$, we rewrite Equation (1.3.3) in matrix form:

$$A\mathbf{x} = \mathbf{b}, \tag{1.3.4}$$

where the coefficients of the stiffness matrix $A = (A_{ij})_{i,j=1}^n \in \mathbb{C}^{n\times n}$ and the right-hand side $\mathbf{b} = (b_i)_{i=1}^n \in \mathbb{C}^n$ are given by $A_{kl} = a(\phi_l, \phi_k)$ and $b_k = F(\phi_k)$.

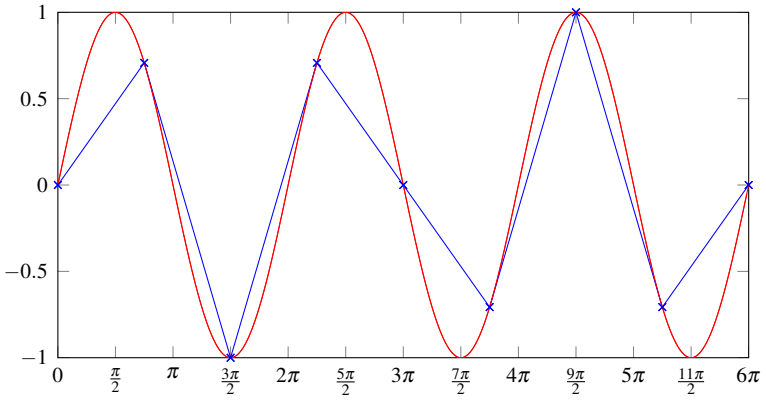


Figure 1.3.1. *Illustration of the requirement of having a minimum number of grid points per wavelength. In this case, there are clearly too few grid points chosen such that the discretized wave is not a good approximation of the continuous one.*

1.3.3 Properties of the discrete system

In order to better understand the problems that iterative solvers have with the solution of the stiffness matrix A arising from the discretization of the Helmholtz equation, cf. Chapter 2, in this section, we examine its properties. One of the main difficulties when applying classical iterative solvers to the Helmholtz equation is its indefiniteness[1]. Both the Laplacian term $-\Delta$ and the identity are positive definite. For the Helmholtz operator $-\Delta - k^2$, the positive eigenvalues of the Laplacian are hence shifted by k^2 to the left, eventually making some of them negative. The larger k is, the more negative eigenvalues the spectrum of the Helmholtz stiffness matrix contains. This complicates issues for classical iterative schemes, which have been mostly developed for definite or only slightly indefinite problems. Moreover, the matrix might be singular if the shift is equal to an eigenvalue of the (discrete) Laplace operator. This is only possible if $\Gamma_R = \emptyset$, otherwise the problem is non-singular and well-posed [79]. Compounding these difficulties is the ill-conditioning of the stiffness matrix, which gets worse the smaller the grid size h and the larger the wave number k are. Moreover, the matrix is in general not Hermitian, but only complex symmetric. This makes the system difficult to solve with iterative methods and specialized preconditioners are needed.

The size of the discrete system increases rapidly with the wave number k. This is partially due to the wave character of the solution. For the one-dimensional free space problem, plane waves $\mathrm{e}^{\pm \imath kx}$ are the elementary solutions. If the wave number k is large, the continuous solution has thus highly oscillatory parts and is periodic with wave length $\lambda = 2\pi/k$. In order for the discrete solution to resolve these waves, the mesh must contain a minimum number n_{res} of grid points per wave length, see Figure 1.3.1. This leads to a "rule of thumb" of the form

[1] Recently, Moiola and Spence [111] showed that the sign-indefiniteness is not inherent to the Helmholtz equation and can be avoided by using a non-standard variational formulation. In this thesis, we will however use a standard approach.

$$n_{\text{res}} = \frac{\lambda}{h} \approx \text{constant}. \tag{1.3.5}$$

For second order finite differences or piecewise linear FEs, the choice $n_{\text{res}} = 10$ is for example recommended in [13]. This rule leads to the accurate *interpolation* of an oscillatory function, or equivalently [79, (4.4.6)] to a good best approximation in the FE space $\mathcal{V}_h$. Hence the interpolation error is controlled by Equation (1.3.5).

When solving the discretized Helmholtz equation, however, not only the discrete FE space $\mathcal{V}_h$ needs to allow for an sufficiently accurate approximation of the solution, but also the solution of the discretized system needs to be sufficiently close to the best approximation in the FE space $\mathcal{V}_h$. The FE solution and the best approximation in the FE space $\mathcal{V}_h$ are connected by Céa's lemma [16]. It provides an estimate of the form

$$\|u - u_{\text{FE}}\| \leq C \inf_{v_h \in \mathcal{V}_h} \|u - v_h\|,$$

where u is the exact solution and u_{FE} the FE solution. Unfortunately, for the Helmholtz equation the constant C depends on the wave number k in such a way that C increases with k even if kh is constant [79]. Therefore, using only the requirement in Equation (1.3.5), with increasing wave number k, the obtained solution will become more and more inaccurate as the discretized operator gives a solution with substantial phase error. This is known as the *pollution effect* [8] and is related to the fact that the discrete solution, as opposed to the continuous one, is *dispersive*, i.e. its phase velocity depends on the angular frequency ω. The polluting term can be shown to be of the order k^3h^2 in one space dimension [8, 80]. Hence in order for the FE solution to be a good approximation of the continuous one in 1D, a condition of the form

$$k^3h^2 \leq \text{constant} \tag{1.3.6}$$

is necessary[2]. We will also use this condition in the higher-dimensional case, even though here the theoretical foundation is missing.

For a one-dimensional example it is easy to see that the discrete solution is dispersive [79]: Consider the following model problem, discretized on the uniform mesh $X_h = \{x_i = ih, i \in \mathbb{N}_0\}$. Find $u \in H^1(\Omega)$, such that

$$-u'' - k^2u = f \quad \text{on } \Omega = (0,1), \qquad u(0) = 0, \qquad u'(1) - \imath k u(1) = 0.$$

Assuming a solution of the form

$$u(x_h) = \mathrm{e}^{\imath k_h x_h}, \quad x_h \in X_h,$$

the unknown *discrete wave number* k_h of the propagating solution can be computed as

$$k_h = k - \frac{k^3h^2}{24} + \mathcal{O}\left(k^5h^4\right). \tag{1.3.7}$$

For details see [79, Section 4.5.1]. Hence the phase velocity of the "numerical wave" differs from the one of the exact wave, and the phase difference is characterized by Equation (1.3.7).

[2] There exist modifications to the Galerkin method in order to reduce this effect, cf. for example [5, 8, 55, 72, 80, 82]. A survey can be found in [71].

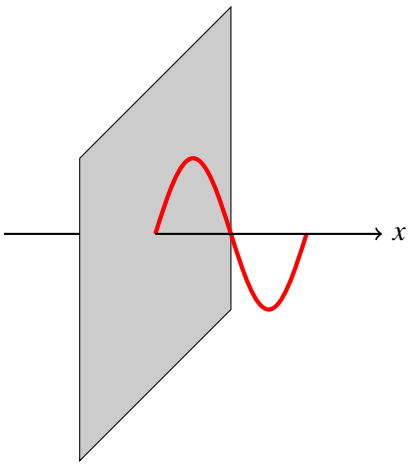


Figure 1.4.1. *Uniaxial propagation of a plane wave. Figure adapted from [79].*

1.4 Model problems

In this section, we introduce the model problems that are used for the numerical experiments throughout this thesis.

One-dimensional model problem Due to its simplicity, the one-dimensional model problem, which we introduce in this paragraph, is used at a few places for illustration purposes. For the numerical results in Chapter 6 and Chapter 7, only the multi-dimensional examples in the next paragraphs will be used.

Problem 1 (Uniaxial propagation of a plane wave [79])**.** The propagation of a time-harmonic plane wave along the x-axis, see Figure 1.4.1, leads to a boundary value problem of the form

$$-u'' - k^2 u = f \qquad \text{on } \Omega = (0,1), \tag{1.4.1a}$$

$$u(0) = 0, \tag{1.4.1b}$$

$$u'(1) - \imath k u(1) = 0. \tag{1.4.1c}$$

Remark 1.4.1 (Green's function). For right-hand side $f \in L^2(0,1)$, the solution of the boundary value problem in Equation (1.4.1) can be written in the form

$$u(x) = \int_0^1 G(x,s) f(s)\,\mathrm{d}s,$$

using the Green's function

$$G(x,s) = \frac{1}{k} \begin{cases} \sin(kx)\mathrm{e}^{\imath k x}, & 0 \le x \le s, \\ \sin(kx)\mathrm{e}^{\imath k s}, & s \le x \le 1. \end{cases}$$

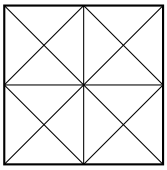
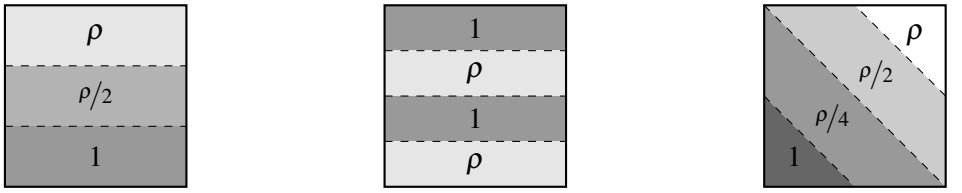


Figure 1.4.2. *Mesh.*



(a) c_1, increasing layers **(b)** c_2, alternating layers **(c)** c_3, diagonal layers

Figure 1.4.3. *Velocity profiles for Problem 2.*

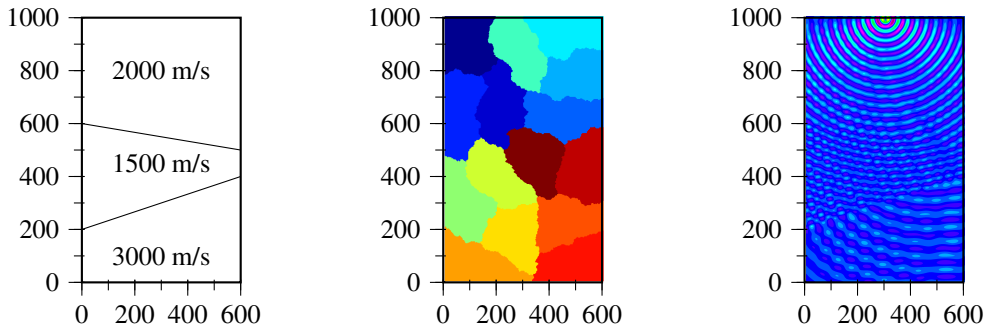
The Green's function for the one-dimensional Helmholtz equation is hence composed of waves with wave number k. Similar results hold for the multi-dimensional case if sufficiently simple boundary conditions are chosen. These waves of the form $e^{\imath kx}$ are called *plane waves* and play an important role for the Helmholtz equation, cf. Subsection 2.4.2 and Section 6.3.

Two-dimensional model problems Here we define the model problems that are used in the two-dimensional numerical experiments throughout this thesis, in particular in Chapter 6. They are all based on the Helmholtz equation, Equation (1.3.1). The first example [64] is the one that we investigate in most detail. For the discretization of the unit square, we use a mesh of the type shown in Figure 1.4.2.

Problem 2 (Wave guide problem)**.** In Equation (1.3.1), let $\Omega := [0,1]^2$, $\Gamma_D := \{0,1\} \times [0,1]$, and $\Gamma_R := [0,1] \times \{0,1\}$. The right-hand side f is a point source at $(0.5, 0.5)$. The wave number $k(\mathbf{x}) = \omega/c(\mathbf{x})$ is either constant, or the wave speed $c = c_i$, $1 \leq i \leq 3$ is piecewise constant according to Figure 1.4.3, where $\rho \in \mathbb{R}$, $\rho > 1$.

Problem 3 (Free space problem)**.** In Equation (1.3.1), let $\Omega := [0,1]^2$, $\Gamma_R := \partial\Omega$, i.e. we simulate an unbounded region. The right-hand side f is a point source in the center $(0.5, 0.5)$ of the domain Ω. The wave number k is constant.

Problem 4 (Wedge problem)**.** This example mimics three layers with a simple heterogeneity. It was introduced in [121]. Let $\Omega = (0,600) \times (0,1000)$ m^2 and $\Gamma_R = \partial\Omega$ in Equation (1.3.1). The right hand-side f is a point source located at $(300, 980)$. The wave number is given by $k(\mathbf{x}) = \omega/c(\mathbf{x})$, where c is defined in Figure 1.4.4a.



(a) Velocity c **(b)** Example partition **(c)** Solution with $\omega = 360$

Figure 1.4.4. *Two-dimensional wedge problem.*

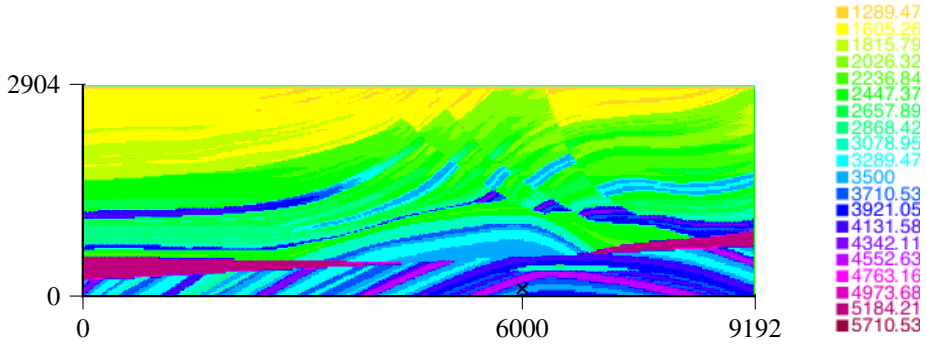


Figure 1.4.5. *Wave speed c for the Marmousi problem, Problem 5. The black cross marks the location of the point source f.*

Problem 5 (Marmousi problem)**.** This is the Marmousi problem, which mimics subsurface geology. A part of this problem is also used in [41]. The data used here was downloaded from [108]. The domain Ω is the rectangle $\Omega = [0,9192] \times [0,2904]$ with $\Gamma_R = \partial\Omega$, i.e. non-reflecting boundary conditions everywhere. The wave number is given by $k(\mathbf{x}) = \omega/c(\mathbf{x})$, where the wave speed c is sampled on a grid composed of squares of size 24×24, that is it contains 384 samples in the x-direction and 122 samples in the y-direction. The wave speed c varies between 1500 and 5500, and is plotted in Figure 1.4.5. The right-hand side f is a point source located at $(6000, 104)$.

Three-dimensional model problems We here define the three-dimensional model problems that will be used for the numerical experiments in Chapter 7.

Problem 6 (Capacitor problem in 3D)**.** In Equation (1.3.1), let $\Omega := [0,1]^3$, the Dirichlet boundary $\Gamma_D := \{(x,y,z) \in \Gamma : y = 0 \text{ or } y = 1\}$, and the Robin boundary $\Gamma_R := \Gamma \setminus \Gamma_D$. The right-hand side f is a point source at $(0.5, 0.5, 0.5)$. The wave number $k(\mathbf{x}) = \omega/c(\mathbf{x})$ is constant.

Problem 7 (Layer problem in 3D)**.** In Equation (1.3.1), let $\Omega := [0,1]^3$, the Dirichlet boundary $\Gamma_D := \emptyset$, and the Robin boundary $\Gamma_R := \Gamma$. The right-hand side f is a point source at $(0.5, 0.5, 0.5)$. The wave number is $k(\mathbf{x}) = \omega / c(\mathbf{x})$, where the wave speed c for $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ is defined as

$$c(x_1, x_2, x_3) = \begin{cases} \frac{5}{6} & \text{for } 0 \leq x_2 < \frac{1}{3}, \\ 1 & \text{for } \frac{1}{3} \leq x_2 < \frac{2}{3}, \\ \frac{2}{3} & \text{for } \frac{2}{3} \leq x_2 \leq 1. \end{cases}$$

Hence the unit cube is divided into three layers of equal size perpendicular to the x_2-axis. This example is adapted from [41, Section 7.3.2].

Problem 8 (Wedge problem in 3D)**.** In Equation (1.3.1), let $\Omega := [0,1]^3$, the Dirichlet boundary $\Gamma_D := \emptyset$, and the Robin boundary $\Gamma_R := \Gamma$. The right-hand side f is a point source at $(0.5, 0.5, 0.5)$. The wave number is $k(\mathbf{x}) = \omega / c(\mathbf{x})$, where the wave speed c for $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ is defined as

$$c(x_1, x_2, x_3) = \begin{cases} \frac{5}{6} & \text{for } 0 \leq x_2 < \frac{2}{5} - \frac{1}{5}x_1 - \frac{3}{20}x_3, \\ 1 & \text{for } \frac{2}{5} - \frac{1}{5}x_1 - \frac{3}{20}x_3 \leq x_2 < \frac{3}{5} + \frac{1}{10}x_1 + \frac{1}{5}x_3, \\ \frac{2}{3} & \text{for } \frac{3}{5} + \frac{1}{10}x_1 + \frac{1}{5}x_3 \leq x_2 \leq 1. \end{cases}$$

Hence as in Problem 7, the unit cube is divided into three layers, but this time the layers are not perpendicular to any of the coordinate axes. This example is adapted from [41, Section 7.3.3].

Chapter 2

Literature review

The properties of the stiffness matrix A arising from the FE discretization of the Helmholtz equation, cf. Section 1.3, adversely affect the performance of standard iterative methods [46]. Consequently, special care needs to be taken when constructing preconditioners. In this chapter, we give an overview over the work that has been done on the iterative solution and preconditioning of the Helmholtz equation in the past. We discuss several different preconditioners. Special emphasis lies on Section 2.4, where domain decomposition methods (DDMs) are presented.

2.1 Preconditioning with multigrid methods

Geometric multigrid methods [20, 70, 148] solve a PDE by employing a hierarchy of grids. They have been extensively studied and optimized mainly for symmetric positive definite (s.p.d.) problems. When applying these techniques naively to the system of linear equations arising to the Helmholtz equation, that is substantially different in nature, they fail [17, 18, 46, 103]. Therefore, specialized variants of these methods are necessary, tailored to meet the needs of the indefinite and non-Hermitian problem. In this section, we discuss the problems that multigrid methods encounter in detail and present some approaches on how to tackle them.

Smoothing is one of the two main ingredients of multigrid methods. It is based on the fact that it is beneficial to treat low- and high-frequent components separately on different grids. In fact, standard relaxation methods, such as the Jacobi method, exhibit different rates of convergence for low- and high-frequent components. Whether a mode is of low or high frequency depends on the mesh width. Therefore, error components that are difficult to smooth on one level are easy to eliminate on another appropriate level. This holds true for standard smoothers and s.p.d. problems, but gets more involved when solving non-Hermitian, indefinite problems. Brandt and Livshits [18] and Livshits [103] identify the main reason for the arising difficulties: After a regular multigrid procedure is applied, "near-kernel[1]" Fourier error components of the form

$$\mathrm{e}^{\pm i\omega x},\ \omega^2 \approx k^2$$

[1] They are called "near-kernel", as they lie in the kernel of the Helmholtz operator $-\Delta - k^2$, if ω^2 is exactly equal to k^2.

remain unreduced. On the fine grids, these components have a very small residual and are consequently almost invisible for relaxation. On the coarse grids, they are subject to large phase errors due to their short wave length. So there is a range of components that converge slowly on the fine grids and that are poorly approximated on the coarser grids. These components can thus not be efficiently eliminated by a standard multigrid method. Compounding these difficulties is the fact that the dimension of the subspace of troublesome components increases with the wave number k. Ernst and Gander [47] and Elman, Ernst, and O'Leary [37] furthermore observe that if the damping parameter in the smoother, a Jacobi method or a Gauß-Seidel method [132], respectively, is optimized for the oscillatory part of the spectrum, the smooth components might be amplified. Moreover, the smoothing properties of the Gauß-Seidel method depend on the relation between the mesh resolution h and the wave number k. For intermediate resolutions the smoother diverges [37, 102].

The other important building block of multigrid methods is the coarse grid correction, where one exploits that the problem on a coarse mesh is usually cheaper to solve than the original problem. For the Helmholtz equation, this is also problematic [37, 47]. As each grid needs to resolve the shortest wave-length present in the problem, multigrid solvers can only employ a limited number of grids. This makes the coarse problem still expensive to solve. Adding to this difficulty is the fact that the computed corrections – if done in the straightforward way – are not always helpful: Due to dispersion effects, cf. Subsection 1.3.3, the eigenvalue of the fine grid function and the one of its restriction to a coarser grid might differ. The larger this difference is, the worse the computed correction is. If they have different signs, the computed correction is even in the wrong direction [37].

Various works adapt the multigrid method to the Helmholtz equation in order to tackle some or all of the above mentioned problems. Because of the shifting of eigenvalues when transferring between different levels in the multigrid hierarchy, Ernst and Gander [47] suggest to use a modified wave number on the coarse grid to account for dispersion effects. This works and is theoretically sound in one space dimension; however, so far it has not been extended to the higher-dimensional case. Kim and Kim [90] use a standard multigrid algorithm on the finer levels, where it is effective, and switch to an optimized Schwarz DDM on the coarser levels, where the multigrid convergence behavior is not satisfactory. Elman, Ernst, and O'Leary [37] use standard grid transfer operators, but replace the standard smoother by the generalized minimal residual (GMRES) method on the coarser grids. Furthermore, they employ the multigrid method as a preconditioner inside a GMRES method. Lee, Manteuffel, McCormick, and Ruge [97] propose a nonstandard multigrid method for a first-order system least-squares formulation of the Helmholtz problem. Brandt and Livshits [18] and Livshits [103] introduce and examine the *wave-ray multigrid method*. Their main idea is to treat the critical, near-kernel error components, which are not reduced by the standard multigrid cycle on the "wave grids", separately on appropriately designed grids, the "ray grids". This approach achieves k-independent convergence rates. However, it involves the use of analytical properties of the functions, and is hence difficult to extend to the case of non-constant wave number k. Since the construction is rather technical and difficult to both understand and implement, Livshits [102] additionally introduces a "slimmed", allegedly easier to implement variant of the wave-ray algorithm and tests it on a different set of problems.

In the following, we give a brief overview of some algebraic multigrid methods for the Helmholtz equation. Most of the presented approaches do not solve the Helmholtz equation directly but a shifted problem [43], cf. Section 2.2. Following the work of Erlangga, Oosterlee, and Vuik [44] on a multigrid method for the shifted problem, Airaksinen, Heikkola, Pennanen, and Toivanen [1] propose a preconditioner based on an algebraic multigrid approach. Umetani, MacLachlan, and Oosterlee [150] use incomplete *LU* smoothing and full weighting restriction for solving the shifted Laplacian. While the coarse grid is chosen based on geometric multigrid principles, the interpolation operator is based on algebraic multigrid principles. Olson and Schroder [119] present a smoothed aggregation algebraic multigrid method for 1D and 2D scalar Helmholtz problems, using the original equation and not the shifted one. They employ plane waves in their approach, cf. Section 2.4 for a description of plane waves and DDMs that use it. Notay [118] proposes an aggregation-based algebraic multigrid method using a double pairwise aggregation scheme [117].

2.2 Preconditioning with elliptic operators

A recently very popular idea is to precondition the Helmholtz operator by a similar operator that is easier to solve numerically. Already in 1983, Bayliss, Goldstein, and Turkel [10] consider an approximate inverse of the Laplacian as a preconditioner. Laird and Giles [96] further extend this idea, proposing a Helmholtz preconditioner with a positive sign in front of the Helmholtz term. The class of *shifted Laplacian preconditioners* [41, 43], where the lower order term is multiplied by some (complex) factor, thereby adding absorption to the problem, constitutes a generalization of these approaches. Erlangga [41] defines the shifted problem as

$$-\Delta u - (\beta_1 - i\beta_2)\, k^2 u = f, \qquad \beta_1, \beta_2 \in \mathbb{R}. \tag{2.2.1}$$

Choosing the shift appropriately, Equation (2.2.1) is easily solvable with standard methods; e.g. if $\beta_1 = \beta_2 = 0$, Equation (2.2.1) is the well-understood, s.p.d. Laplace problem. The challenge is thus to choose the shift in such a way that at the same time the shifted problem is still a good preconditioner for the Helmholtz equation.

The question of what parameters β_1 and β_2 yield the best results has been examined both theoretically and numerically in different settings. In his PhD thesis introducing this class of preconditioners, Erlangga [41] uses a preconditioned Krylov method with the shifted problem solved by a multigrid method as a preconditioner. An appropriate choice of the parameters achieves to a certain extent both goals – the shifted problem is a good preconditioner for the original one and it is easily solvable – at the same time, for the low- to mid-frequency regime. Multigrid methods have been further examined in this context, see e.g. [29, 129, 135] and [13], where the emphasis in the latter work, however, is not on the shift. Calandra, Gratton, Pinel, and Vasseur [24] combine the GMRES smoothing idea of Elman, Ernst, and O'Leary [37] with the shifted Laplacian preconditioner for the coarse problem to obtain a two-level algorithm with inexact coarse solves that they apply to three-dimensional Helmholtz problems in heterogeneous media.

Unfortunately, the shift parameters that are best for the solution with multigrid methods or other standard methods are not necessarily best for preconditioning the Helmholtz operator. In fact, this

conflict can be shown to be impossible to overcome asymptotically: By means of Fourier analysis, Ernst and Gander [46] conclude that the shift needs to satisfy two conditions that exclude each other in order to simultaneously achieve a favorable spectrum of the preconditioned operator and good convergence rates for the multigrid method. This is also observed numerically e.g. by Gijzen, Erlangga, and Vuik [66] and Airaksinen, Heikkola, Pennanen, and Toivanen [1]. The question of how to choose the shift in order for the shifted equation to be a good preconditioner for the original one is further studied by Gander, Graham, and Spence [65].

2.3 Preconditioning with incomplete factorization methods

While incomplete LU factorization preconditioners yield fast black-box preconditioners for a wide range of problems [132], for the Helmholtz equation they are not efficient [46]. Especially for larger wave numbers k, incomplete LU preconditioners may fail, e.g. because of ill-conditioned incomplete factors or large fill-in [45]. In the spirit of the shifted Laplacian preconditioners described in Section 2.2, these problems might be partially removed by using a shifted problem as a preconditioner [13, 105, 120]. Several other attempts have been made in order to develop an efficient incomplete factorization preconditioner for the Helmholtz equation [58, 59, 87, 120].

Particularly interesting are the *sweeping preconditioners*. Engquist and Ying [39, 40] define two different approaches, both based on approximating a block LDL^T decomposition of the Helmholtz operator layer by layer, exploiting the radiation boundary conditions [83]. The first approach [39] uses an $\mathcal{H}$-matrix approximation [11, 14, 69] of the blocks in the diagonal matrix D. In 2D, its theoretical foundation builds on the fact that the inverse of each of these blocks is the discretization of a half-space Green's function. In 3D, a similar argument fails to give a theoretical justification. In the second approach [40], the Schur complement of the factorization is approximated using auxiliary problems on layers equipped with artificial radiation boundary conditions. This approach has been further studied by Poulson, Engquist, Li, and Ying [122] and Poulson [123], where the authors successfully apply the method to larger three-dimensional problems.

2.4 Preconditioning with domain decomposition methods

DDMs [126, 138, 147] solve boundary value problems by splitting the original computational domain into subdomains. On each of these subdomains, a smaller boundary value problem is defined that can be solved more easily. In an iterative process, values between the subdomains are exchanged to coordinate the solution between adjacent subdomains. If both subproblems and information exchange are defined appropriately, the method converges to the global solution. The problems on the subdomains can be solved independently of each other, which makes DDMs suitable for parallel computing. When applying DDMs to indefinite problems such as the Helmholtz equation, a number of difficulties arise, cf. [147, Chapter 11.5.2] and [46]. We elaborate on those and on possible solutions in the remainder of this section.

2.4.1 Transmission conditions

Transmission conditions are an important ingredient of DDMs. If information cannot travel efficiently between neighboring subdomains, the 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. For the Helmholtz equation, however, they fail to reduce the error in large parts of the spectrum and are therefore not suitable [63]. Furthermore, in case local Dirichlet or Neumann problems are employed on the subdomains, local problems may become singular. One easy way to avoid that is to bound the diameter of each subdomain from above by half a wavelength. However, such a requirement is preposterous in today's practice, where high-frequent problems are of particular interest. An early attempt to resolve these problems has been made in Després's Ph.D. thesis [32], using a first-order approximation to the Sommerfeld radiation condition [139] at the subdomain interfaces. We will use his approach in this work, cf. Subsection 3.2.1 and refrain from employing the more involved approaches presented in the following, as we focus our attention rather on the coarse space than on the transmission conditions.

More advanced transmission conditions include conditions that are based on the PML method as examined e.g. by Toselli [146] for an overlapping Schwarz method. He concludes that it is best to use very thin PMLs. However, this conclusion might be caused by the fact that the incoming fields are coupled at the *external* boundary of the PML layer and thus are damped before being transferred to the neighboring subdomain [134]. In a subsequent work, Schädle and Zschiedrich [134] examine a similar approach, solving the afore-mentioned problem by coupling the incoming field at the *internal* boundary of the PML layer via an approximation of the Dirichlet-to-Neumann (DtN) operator and achieving better results.

For *optimized Schwarz methods*, an approximation to optimal transmission conditions is computed via Fourier analysis of a simplified, continuous problem; for an overview see [57]. These methods can also be applied to the Helmholtz equation. Gander, Magoulès, and Nataf [63], for example, examine low order boundary conditions that optimize transmission between subdomains via Fourier analysis and prove asymptotic convergence bounds. Although these are only derived for a model case, numerical experiments support their claim that transmission conditions of optimized Schwarz type are also suitable for quite arbitrary domain decompositions. See also [34, 61, 140] and Section 3.3 for related work.

2.4.2 Coarse spaces

Since one-level DDMs exchange information only between neighboring subdomains, they converge at best in a number of iterations that is proportional to the number of subdomains per direction [151]. Therefore, it is crucial in the design of DDMs to include a coarse component. Using a global problem with only a few unknowns per subdomain allows information to propagate through the whole domain in one step. The question of how to construct it for DDMs is closely related to the corresponding question for multigrid methods. If the coarse space is defined as the FE space on a coarser mesh, for Helmholtz problems the mesh width h needs to be sufficiently fine or the polynomial degree p

(a) Plot **(b)** Schematic view

Figure 2.4.1. *A plane wave for* $k = 2$, $\theta = \frac{\sqrt{5}}{2}(1/2 \;\; 1)^T$.

of the FE functions needs to be sufficiently large in order for the two-level method to be effective. This gives a still expensive global problem. Hence more sophisticated choices for the second level are necessary. The underlying idea of the wave-ray multigrid method [18], that is splitting the solution into waves traveling into different directions on the "ray grids", cf. Section 2.1, is similar to a popular choice in DDMs. These coarse spaces based on *plane waves*[2] are used in a couple of works [49, 52, 92, 93]. We will discuss them in detail in this section as they are closely related to the work presented in this thesis and will be used for comparison in the numerical experiments.

A plane wave is a function of the form

$$p(\mathbf{x}) = \mathrm{e}^{\imath k\theta\cdot\mathbf{x}}, \tag{2.4.1}$$

where $\theta \in \mathbb{R}^d$ is a vector of unit length, $\|\theta\|_2 = 1$, specifying the direction into which the plane wave is traveling. For a 2D plot see Figure 2.4.1. The wave front is a straight line in 2D (a plane in higher dimensions) with normal θ. Plane waves are solutions to the homogeneous Helmholtz equation. In the one-dimensional case, the situation is particularly easy. There are only two linearly independent solutions, the plane waves associated to the two different directions. Each linear combination of these two fundamental solutions is a solution of Equation (1.2.4). The general solution hence has the form

$$u(x) = \alpha\mathrm{e}^{\imath kx} + \beta\mathrm{e}^{-\imath kx} \tag{2.4.2}$$

with coefficients $\alpha, \beta \in \mathbb{C}$. In higher dimensions, i.e. for $d = 2, 3$, the situation is more complicated, as plane waves can travel into infinitely many directions θ. A solution to the Helmholtz equation can be any superposition of these infinitely many plane waves.

The basic principle in all the works that use plane wave based coarse spaces in DDMs for the Helmholtz equation is the same. The (global) coarse space is built out of local components. For that purpose, the computational domain Ω is divided into subdomains that do not necessarily coincide with

[2]Plane waves play an important role for the Helmholtz equation. They are also employed e.g. for discretization [3, 21, 25, 50, 77, 95, 110] and for iterative solution techniques [49, 103].

the subdomains used for the domain decomposition, but are strongly related: While Farhat, Macedo, and Lesoinne [49] and Farhat, Avery, Tezaur, and Jing [52] use the original subdomains, Kimn and Sarkis [91–93] use the original subdomains but with a different overlap size and Leong [100] uses unions of neighboring subdomains. For each of these subdomains, a finite number of plane waves is chosen and they are evaluated on the mesh points of the subdomain [91–93] or on a subset associated to boundary degrees of freedom [49, 52, 100]. The resulting local vectors – possibly after extension to the interior of the subdomains [100] and multiplication with a partition of unity function [91–93] – are then extended by 0 to global ones, and combined in a global system.

The strategy how to choose a finite number of plane waves on each subdomain is straightforward in the one-dimensional case. As there are only exactly two plane waves traveling in negative and positive x-direction, respectively, the coarse space based is spanned by the discretizations of exactly these two plane waves. For the higher dimensional cases, the situation is more involved as there are infinitely many plane waves. To choose a finite subset of them, all of the works implement a simple approach: In two dimensions, a uniform discretization of the unit circle into circular sectors is used. Thus for m_i directions θ_k, $1 \leq k \leq m_i$, we get (up to rotation of all directions θ_k by a fixed angle)

$$\theta_k := \begin{pmatrix} \cos(t_k) \\ \sin(t_k) \end{pmatrix}, \quad \text{where } t_k = \frac{2\pi(k-1)}{m_i}, \quad 1 \leq k \leq m_i.$$

For the three-dimensional case, the definition of uniformly distributed directions a bit more complicated, see e.g. [144].

Another important, even though to our knowledge open question is how to determine reliably in a general setting the number of modes m_i per subdomain that should enter the coarse space. This question has two aspects. On the one hand, m_i has to be large enough for the second level to be beneficial. As opposed to the s.p.d. case [112, 115], for the indefinite Helmholtz case, an incomplete coarse space can even deteriorate the convergence of the two-level method [125]. On the other hand, m_i should not be too large: If the two directions θ_1 and θ_2 are "close" to each other, the corresponding plane waves are almost linearly dependent. That is, in a discrete setting, if we evaluate those two plane waves at a finite number of points, the matrix having the resulting two vectors as column vectors is ill-conditioned. Here, what "close" means depends on the size of the wave number k; the smaller k is, the more linearly dependent the plane waves are. The coarse matrix based on plane waves can hence become rank deficient [52, 95]. This causes in the worst case divergence of the whole iterative scheme. For a general problem, it is not well-understood how to determine a priori the minimum number of plane wave directions that are needed for convergence and the maximum number of plane waves that can be used before conditioning problems occur.

As a remedy to the ill-conditioning problems, Farhat, Avery, Tezaur, and Jing [52] propose to "filter" the vectors that serve as a generating set for the coarse space via a QR decomposition by dropping all those vectors whose associated diagonal entry in the R matrix is smaller than a prescribed tolerance ε. This is a local procedure, which is performed on each subdomain separately. A too small value of ε can cause the coarse matrix to be still rank deficient. Therefore, it should be rather chosen too large than too small.

We now discuss in more detail the single methods that employ plane waves. Finite element tearing and interconnecting (FETI) methods [48] belong to the class of iterative substructuring methods. A

variant of the FETI method that uses a second level based on plane waves is the FETI for Helmholtz (FETI-H) method [49]. It is subject to two main modifications: As the arising problems on the subdomains might be singular, they are regularized by suitable interface matrices. This is equivalent to imposing Dirichlet and radiation boundary conditions at parts of the subdomains' boundary. A global problem is defined by evaluating plane waves on the interfaces of the original subdomains. As FETI methods work with interface degrees of freedom, it is not necessary to explicitly extend them to the interiors of the subdomains.

The closely related dual-primal FETI (FETI-DP) method [51] has also been adapted to the Helmholtz equation. This variant is called FETI-DP for Helmholtz (FETI-DPH) [52]. In this method, there are two different sets of degrees of freedom for the coarse space. The primal degrees of freedom, which are the values of the functions at some nodes (the "corners") in the mesh, are the same as in the s.p.d. case. The dual degrees of freedom, which arise from the evaluation of plane waves on the interface nodes, are the ones that are tailored for the Helmholtz equation. Orthogonality of the iterates to the dual component is enforced by the use of Lagrange multipliers. For the FETI-DPH method, the subdomain problems are not regularized and thus might become singular. This is why the authors require the diameter of each subdomain to be bounded from above, which makes using the method for the high frequency regime unfeasible. However, both methods have been tested successfully for examples in the low- to mid-frequency regime for homogeneous media.

For Schwarz methods, as opposed to FETI methods, also the degrees of freedom associated to the interior of the subdomains are part of the system. When computing the coarse space, the plane waves are hence either evaluated on all degrees of freedom of the subdomain immediately [91–93], or they are evaluated on a subset and then extended in some sense to the remaining degrees of freedom on the subdomain Leong [100]. For the extension, Leong uses either the shifted problem $-\left(\Delta - k^2\right) u = 0$ or the harmonic extension $-\Delta u = 0$. From these works, several different Schwarz methods for the Helmholtz equation arise. They are different in their exact formulations, but share the same ideas for the coarse space. We here do not give the full details. All these methods have been tested numerically and their positive effect on the iteration counts has been shown.

Concluding, for any of these methods employing plane waves some open problems remain. Firstly, it is not clear a priori how many plane waves the coarse space should contain per subdomain. Secondly, as the global problem is usually rather big compared to the s.p.d. case, where typically only very few coarse degrees of freedom per subdomain are chosen [147]. Therefore, it would make sense to solve the coarse problem iteratively, see e.g. [94]. Thirdly, it is not clear how the plane waves should be used for heterogeneous media. It is mainly the first and the third problem that this thesis will tackle by defining a different coarse space. As they are closest to our method, we will compare quite frequently to the plane wave approach, e.g. in the numerical experiments in Chapter 6 and Chapter 7.

Chapter 3

The one-level method

Standard iterative solvers do not converge well for the Helmholtz equation [46]. Therefore, the definition of a suitable preconditioner is important, cf. the overview on existing work in Chapter 2. The present chapter introduces the fine level of the iterative method that we employ for the solution of the Helmholtz equation. It consists of two parts. The outer iterative solver is a GMRES method [131, 132], see Section 3.1. Even though this type of Krylov method is suited to solve also non-Hermitian, indefinite matrices, the ill-conditioning of the Helmholtz matrix causes the convergence of the GMRES method to be very slow without preconditioning. For that reason, in Section 3.2, we introduce the second part of the one-level method, the preconditioner. It is a restricted additive Schwarz (RAS) method [22, 147] with special transmission conditions adapted to the Helmholtz equation. RAS type methods are overlapping DDMs. Along with other DDMs, they are particularly well-suited for the Helmholtz equation, as they subdivide the typically large system of linear equations into smaller ones, which are then solved in parallel. In Section 3.3, we will analyze the RAS method by means of Fourier analysis to gain a clearer understanding of which parts of the spectrum do not converge well. This is the point of departure for the construction of the coarse space, which is introduced in Chapter 5.

3.1 The GMRES method

The GMRES method [131, 132] will be used as the outer iterative solver for the numerical experiments in this manuscript. We start with the introduction of the basic GMRES algorithm without restart. It is given in Algorithm 3.1.1 [132, Chapter 6]. For practical implementation, in particular for the complex case, see [132, Section 6.5.9]. In this form, the algorithm always performs $k^{\max}$ iteration steps. In practice, we need an additional convergence criterion that is checked in regular intervals, e.g. in each iteration step. Usually, the stopping criterion is based on the relative residual and has the form $\rho \leq \varepsilon \|b\|_2$, where ε is the desired accuracy of the solution and the residual ρ is computed during the GMRES method.

Algorithm 3.1.1 GMRES method for complex linear systems

Input: initial iterate $\mathbf{x}^{(0)}$, matrix A, right-hand side $\mathbf{b}$, maximum number of iterations $k^{\max}$
Output: approximate solution $\tilde{\mathbf{x}}$, norm of residual $\rho = \|\mathbf{b} - A\tilde{\mathbf{x}}\|_2$

function GMRES($\mathbf{x}^{(0)}, A, \mathbf{b}, k^{\max}$)
  Compute $\mathbf{r}^{(0)} \leftarrow \mathbf{b} - A\mathbf{x}^{(0)}$, $\beta \leftarrow \left\|\mathbf{r}^{(0)}\right\|_2$, $\mathbf{v}^{(1)} \leftarrow \mathbf{r}^{(0)}/\beta$.
  for $k \leftarrow 1, 2, \ldots, k^{\max}$ **do**
    Compute $\mathbf{w}^{(k)} \leftarrow A\mathbf{v}^{(k)}$
    for $i \leftarrow 1, 2, \ldots, k$ **do**
      $h_{ik} \leftarrow \langle \mathbf{w}^{(k)}, \mathbf{v}^{(i)} \rangle$
      $\mathbf{w}^{(k)} \leftarrow \mathbf{w}^{(k)} - h^{ik}\mathbf{v}^{(i)}$
    end for
    $h_{k+1,k} \leftarrow \left\|\mathbf{w}^{(k)}\right\|_2$
    $\mathbf{v}^{(k+1)} \leftarrow \mathbf{w}^{(k)}/h_{k+1,k}$
  end for
  Define $\overline{H}^{k^{\max}} \leftarrow$ the $(k^{\max}+1) \times k^{\max}$ Hessenberg matrix with entries h_{ij}.
  Compute $y^{k^{\max}} \leftarrow \operatorname{argmin}_{\mathbf{y} \in \mathbb{C}^{k^{\max}}} \left\|\beta \mathbf{e}^1 - \overline{H}^{k^{\max}}\mathbf{y}\right\|_2$.
  Define $V^{k^{\max}} \leftarrow$ the matrix with columns $\mathbf{v}^{(j)}$, $1 \le j \le k^{\max}$.
  Compute $\tilde{\mathbf{x}} \leftarrow \mathbf{x}^{(0)} + V^{k^{\max}}\mathbf{y}^{k^{\max}}$.
  $\rho \leftarrow \left\|\beta \mathbf{e}^1 - \overline{H}^{k^{\max}}\mathbf{y}^{k^{\max}}\right\|_2 = \|\mathbf{b} - A\tilde{\mathbf{x}}\|_2$
  return $\tilde{\mathbf{x}}$, ρ
end function

The convergence behavior of the GMRES method is important to better understand the effect of the preconditioner and of the coarse level. Maybe most importantly, the GMRES algorithm has the solver property, that is, for a non-singular problem it always converges to the solution.

Theorem 3.1.1 ([89, Theorem 3.1.2])**.** *Let A be a non-singular, $n \times n$ matrix. Then the* GMRES *algorithm will find the solution within n iterations.*

Apart from this, its convergence behavior is not as well understood as for example for the conjugate gradient (CG) method [76, 136]. However, clustering of the eigenvalues has a positive effect on the convergence. Among the many results, we here give the following theorem examining the relation between the eigenvalues and the convergence behavior of the GMRES method.

Theorem 3.1.2 ([131, Proposition 4 and Theorem 5])**.** *Assume that A is a diagonalizable $n \times n$ matrix, so that $A = X\Lambda X^{-1}$, where Λ is a diagonal matrix with non-zero entries $\lambda_1, \ldots, \lambda_n$. Let*

$$\varepsilon^{(m)} = \min_{p^m \in P^m} \max_{\lambda \in \sigma(A)} |p^m(\lambda)|,$$

where P^m for $m \in \mathbb{N}$ is the space of monic polynomials of degree at most m with $p(0) = 1$ for all $p \in P^m$. Assume that the eigenvalues $\lambda_1, \ldots, \lambda_\nu$ are contained in the left half plane for some

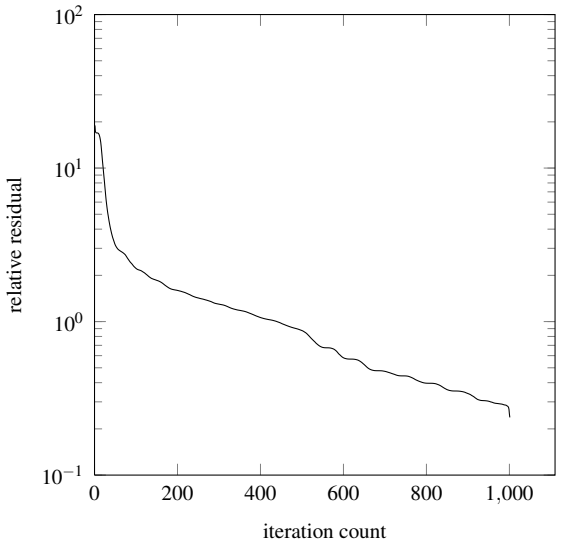


Figure 3.1.1. *Relative residual in each iteration step for the* GMRES *method applied to Problem 1 with random right-hand side.* $k = 50$, $h = 10^{-3}$.

$0 < \nu < n$. *Moreover, assume that the eigenvalues* $\lambda_{\nu+1}, \ldots, \lambda_n$ *are confined to a closed disk with center* $C > 0$ *and radius* $R < C$. *Then*

$$\varepsilon^{(m)} \leq \left(\frac{D}{d}\right)^{\nu} \left(\frac{R}{C}\right)^{m-\nu},$$

where

$$D = \max_{1 \leq i \leq \nu,\ \nu+1 \leq j \leq n} \left|\lambda_i - \lambda_j\right| \quad \textit{and} \quad d = \min_{1 \leq i \leq \nu} |\lambda_i|.$$

Moreover, the residual norm provided at the m-th step of the GMRES *algorithm satisfies*

$$\left\| r^{(m+1)} \right\| \leq \kappa(X) \varepsilon^{(m)} \left\| \mathbf{r}^{(0)} \right\|,$$

where $\kappa(X) = \|X\| \left\|X^{-1}\right\|$.

Even though the GMRES method in Algorithm 3.1.1 finds a solution to the linear system of equations in at most n iteration steps, for the usually large and sparse matrices that arise from the FE discretization of the Helmholtz equation this is not enough. As the GMRES method is used as an iterative, rather than a direct solver, a sufficiently accurate approximation to the solution has to be found in significantly less than n iteration steps in order to keep the costs reasonable. Whether or not the GMRES method succeeds in doing this depends on the properties of the system matrix. Unfortunately, for the matrix arising for the discretization of the Helmholtz equation, the non-preconditioned GMRES method suffers from very slow convergence. In Figure 3.1.1, we show an example for this behavior. For the one-dimensional model problem, Problem 1, with wave

number $k = 50$ and uniform mesh with mesh width $h = 10^{-3}$, the residual is hardly reduced by the GMRES method. The long plateau is due to the bad conditioning of the stiffness matrix related to the Helmholtz problem. For that reason, a preconditioner is needed. The first level of the preconditioner will be introduced in Section 3.2.

Even if convergence within a reasonable number of iteration steps is achieved with the use of a preconditioner, to be suitable for larger systems, the GMRES algorithm as introduced in Algorithm 3.1.1 needs to be modified: Its storage requirements grow quickly with the size of the system and the number of iterations, as the complete Krylov subspace basis must be stored for the Arnoldi process in Lines 3 to 11 of Algorithm 3.1.1. This means that in order to perform k GMRES iterations, k vectors of the size of the matrix A need to be stored. This is in contrast to e.g. the CG method [136], which however works only for s.p.d. matrices. This is feasible as long as the number of iterations k or the size of the system A are small. However, when one or both of these quantities are large, storing all vectors is prohibitively expensive. Therefore, the iteration is restarted when there is no more space to store all the previous basis vectors. This can significantly reduce the storage costs of the iteration. The restarted GMRES algorithm, called the GMRES(m) method, where m is the number of GMRES iterations that are performed before the method is restarted is given in Algorithm 3.1.2 [132]. Restarting can slow down the convergence as the information about the previously built Krylov subspace is lost. Therefore, also the solver property of Theorem 3.1.1 does no longer hold true when using the restarted variant.

Algorithm 3.1.2 GMRES(m): restarted GMRES method

Input: initial iterate $\mathbf{x}^{(0)}$, matrix A, right-hand side $\mathbf{b}$, tolerance ε, number m of GMRES iterations before restart, maximum number $k^{\max}$ of GMRES calls

Output: approximate solution $\tilde{\mathbf{x}}$, norm of residual $\rho = \|\mathbf{b} - A\tilde{\mathbf{x}}\|_2$

1: **function** GMRESM($\mathbf{x}^{(0)}$,A, $\mathbf{b}$, ε, m, $k^{\max}$)
2: $\tilde{\mathbf{x}}$, ρ = GMRES($\mathbf{x}^{(0)}$,A,$\mathbf{b}$,m)
3: $k \leftarrow 1$
4: **while** $\rho > \varepsilon \|\mathbf{b}\|_2$ and $k < k^{\max}$ **do**
5: $\tilde{\mathbf{x}}$, ρ = GMRES($\tilde{\mathbf{x}}$,A,$\mathbf{b}$,m)
6: $k \leftarrow k + 1$
7: **end while**
8: **return** $\tilde{\mathbf{x}}$, ρ
9: **end function**

3.2 The restricted additive Schwarz method

DDMs [147] are well suited to solve the Helmholtz equation, as they split the typically very large system into smaller subsystems, which can be solved in parallel, independently of each other. They

can be divided into two groups. In a *non-overlapping* DDM, the subdomains in the splitting of the domain are disjoint except for their boundaries, see e.g. [48, 106, 107]. In an *overlapping* DDM, on the other hand, neighboring subdomains share also some interior degrees of freedom. This can in many cases help to facilitate information exchange between the subdomains. In this section, we define the DDM that we use as a preconditioner for the Helmholtz equation throughout this thesis. It is a RAS method [22, 147] with special transmission conditions as introduced by Després [32].

3.2.1 Domain decomposition

We introduce the partitioning of the computational domain and some definitions needed for the RAS method [22, 147]. Let the domain Ω be decomposed into a set of non-overlapping subdomains $\{\Omega'_j\}_{j=1}^N$ resolved by the mesh $\mathcal{T}_h$. The overlapping subdomains Ω_j are defined by adding one or several layers of mesh elements to Ω'_j in the following sense, cf. [75]:

Definition 3.2.1 (Overlapping subdomains)**.** Given a subdomain $D' \subset \Omega$, which is resolved by the FE mesh $\mathcal{T}_h$, the extension D of D' by $n_{\text{ov}} \in \mathbb{N}$ layers of elements is

$$D = \operatorname{supp}\left(\left(\Pi_1^2\Pi_0^2\right)^{n_{\text{ov}}} \mathbb{1}_{D'}\right).$$

Here $\operatorname{supp}(\mathrm{f}) := \overline{\{x \in \Omega : f(x) \neq 0\}}$ denotes the *support* of the function f; $\mathbb{1}_{D'}$ denotes the characteristic function on D', i.e.

$$\mathbb{1}_{D'}(x) = \begin{cases} 1 & \text{for } x \in D' \\ 0 & \text{else} \end{cases}$$

for $x \in \Omega$; and Π_1^2 and Π_0^2 denote the L^2-projections onto the affine continuous and constant FE spaces on the mesh $\mathcal{T}_h$, respectively.

This gives an overlapping partition $\{\Omega_j\}$ of Ω. Figure 3.2.1 shows a plot of an example partition into non-overlapping subdomains and the extension of the central subdomain by two layers of elements. Let $\mathcal{V}_h(\Omega_j) = \left\{v|_{\Omega_j} : v \in \mathcal{V}_h\right\}$, $1 \leq j \leq N$, denote the space of functions in the FE space $\mathcal{V}_h$ restricted to the subdomain Ω_j. Let $n := |\operatorname{dof}(\Omega)|$ and $n_j := \left|\operatorname{dof}(\Omega_j)\right|$, $1 \leq j \leq N$, where for $D \subseteq \Omega$ we define

$$\operatorname{dof}(D) := \left\{k : \operatorname{supp}(\phi_k) \subset \overline{D}, \phi_k \text{ is a basis function of } \mathcal{V}_h\right\}.$$

We also define a partition of unity subordinate to the domain decomposition following [75]. As illustrated in Figure 3.2.2a, after the construction of the overlapping subdomains, some degrees of freedom in the system belong to several subdomains. Via the partition of unity, we assign to each degree of freedom a weight. These weights sum up to one, when adding the contributions from all subdomains. Using the notation of Definition 3.2.1 and defining for $1 \leq j \leq N$

$$\pi_j^* = \left(\Pi_1^2\Pi_0^2\right)^m \mathbb{1}_{\Omega'_j},$$

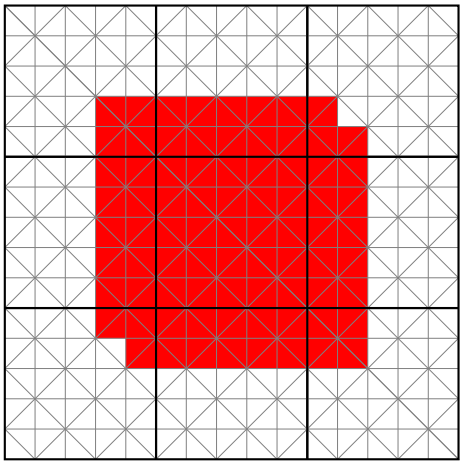


Figure 3.2.1. *Example partition of the square into* 9 *non-overlapping subdomains. To the central subdomain two layers of elements are added in the way defined in Definition 3.2.1.*

where $\mathbb{1}_{\Omega'_j}$ is the characteristic function on Ω'_j as defined in Definition 3.2.1, we define the partition of unity functions as

$$\pi_j = \frac{\pi_j^*}{\sum_{k=1}^N \pi_k^*}. \tag{3.2.1}$$

Each function π_j is greater than zero on Ω_j and equal to 0 outside of Ω_j, cf. Figure 3.2.2b. The partition of unity property $\sum_{j=1}^N \pi_j(x) = 1 \ \forall x \in \Omega$ is obvious from the definition.

3.2.2 Discrete method

This section introduces the RAS method [22]. We start with the definition of the restriction operators that map from the global degrees of freedom to the local ones associated to one subdomain. For $1 \leq j \leq N$, we define the restriction operator $\mathcal{R}_j : \mathcal{V}_h \to \mathcal{V}_h(\Omega_j)$ by injection, i.e. for $u \in \mathcal{V}_h$ we set

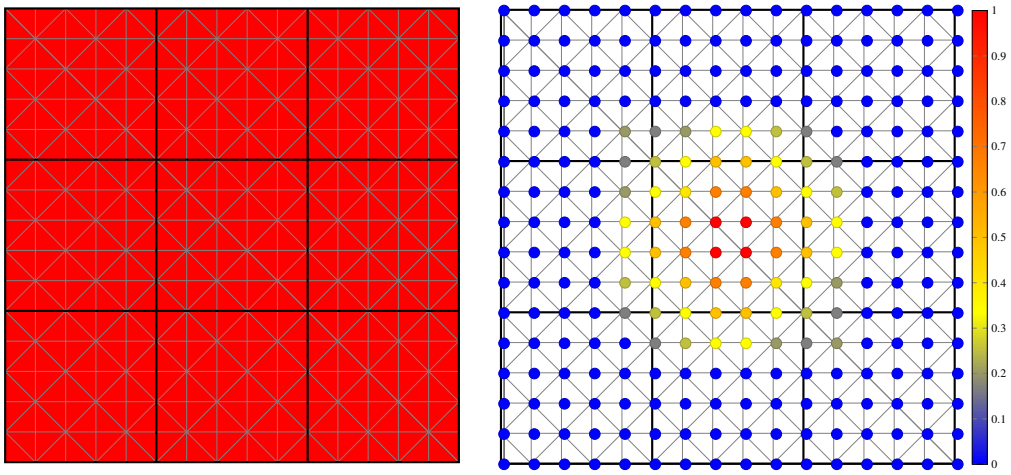
$$(\mathcal{R}_j u)(\mathbf{x}) = u(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega_j.$$

We denote the corresponding matrix in $\mathbb{R}^{n_j \times n}$ that maps coefficient vectors of functions in $\mathcal{V}_h$ to coefficient vectors of functions in $\mathcal{V}_h(\Omega_j)$ by R_j.

In the next step, we define the matrices corresponding to a partition of unity subordinate to the domain decomposition, cf. Equation (3.2.1). Let $D_j \in \mathbb{R}^{n_j \times n_j}$ be a diagonal matrix corresponding to a partition of unity in the sense that

$$\sum_{j=1}^N \tilde{R}_j^T R_j = I, \quad \tilde{R}_j := D_j R_j.$$

Hence each matrix $\tilde{R}_j^T R_j$ assigns a weight to each (global) degree of freedom that is 1 in the interior of the subdomain away from the overlap, 0 outside of the subdomain and between 0 and 1 in the



(a) The overlapping domains

(b) Partition of unity function with support on central subdomain

Figure 3.2.2. *The partition of unity is illustrated for an overlap of $n_{\text{ov}} = 1$ mesh element. The original partition into non-overlapping subdomains is marked with the fat black lines.*

regions where several subdomains overlap. While in principal any choice of such matrices D_j defines a DDM, for the parallel implementation the fact that the matrix D_j vanishes at $\partial\Omega_j$ for all j is crucial, cf. Subsection 7.1.1 and Subsection 7.1.2. For this reason, in this thesis, we use matrices D_j associated to the partition of unity defined in Equation (3.2.1), which have the desired property.

Using the above definitions, the RAS preconditioner reads

$$M^{-1} := \sum_{j=1}^{N} \tilde{R}_j^T A_j^{-1} R_j. \tag{3.2.2}$$

It remains to define the subdomain matrices A_j. The classical choice $A_j = R_j A R_j^T$ corresponds to Dirichlet transmission conditions in the continuous equations. Hence frequencies in the error smaller than the wave number k are not damped [64], cf. also Subsection 2.4.1 and the short discussion in Subsection 3.2.3. This might cause slow convergence or even stagnation of the iterative method. Therefore, we define the matrices in A_j in Equation (3.2.2) to be the stiffness matrices of the local Robin problems [23, 32]

$$\left(-\Delta - k^2\right)(u_j) = f \qquad \text{in } \Omega_j, \tag{3.2.3a}$$

$$\mathcal{C}(u_j) = 0 \qquad \text{on } \partial\Omega \cap \partial\Omega_j, \tag{3.2.3b}$$

$$\left(\frac{\partial}{\partial n_j} + \imath k\right)(u_j) = 0 \qquad \text{on } \partial\Omega_j \setminus \partial\Omega. \tag{3.2.3c}$$

More advanced techniques such as the discretized optimized boundary conditions [57] are also possible, but not considered here.

In a serial code the implementation of the RAS method is straightforward. The main feature of DDMs is however the possibility to parallelize them. In a parallel code, the global matrices and vectors are never assembled. This is in particular true for the global stiffness matrix A and the restriction operators R_j. We will discuss the implementation of the method in a parallel code in Chapter 7, where the results for three-dimensional examples are discussed.

3.2.3 Continuous method

The continuous counterpart of the RAS method introduced in Subsection 3.2.2 is the *Jacobi-Schwarz method*. Under certain assumptions, it is possible to show the equivalence of the discrete and the continuous methods [140]. We note however that these assumptions are not necessarily fully satisfied in our setting; in particular cross points and the fact that the partition of unity in Equation (3.2.1) is non-zero on several subdomains on the layer next to the interface impose problems, cf. [140, Assumption 1] and [35]. The Jacobi-Schwarz method is defined for continuous quantities, and is hence useful when one wants to analyze the method via Fourier analysis, see Section 3.3. For $N = 2$ subdomains – the general case is defined analogously – the Jacobi-Schwarz method reads [140, Equation (3.2)]

$$\begin{aligned}
\mathcal{L}u_1^{n+1} &= f && \text{in } \Omega_1, & \qquad \mathcal{L}u_2^{n+1} &= f && \text{in } \Omega_2, \\
\mathcal{C}\left(u_1^{n+1}\right) &= g && \text{on } \partial\Omega_1, & \qquad \mathcal{C}\left(u_2^{n+1}\right) &= g && \text{on } \partial\Omega_2, \\
\mathcal{B}_{12}u_1^{n+1} &= \mathcal{B}_{12}u_2^{n} && \text{on } \Gamma_{12}, & \qquad \mathcal{B}_{21}u_2^{n+1} &= \mathcal{B}_{21}u_1^{n} && \text{on } \Gamma_{21}.
\end{aligned} \tag{3.2.4}$$

Here $\mathcal{L}$ is the partial differential operator in question, i.e. in our case $\mathcal{L} = -\Delta - k^2$ is the Helmholtz operator. The Jacobi-Schwarz method hence consists of the solution of a local problem followed by the exchange of information between the two subdomains via the boundary operators $\mathcal{B}_{ij}$. These boundary operators are in the simplest case just the identity, i.e. Dirichlet transmission conditions. For the Helmholtz equation, Dirichlet transmission conditions do not succeed to damp frequencies that are close to the wave number k, see Section 3.3. Therefore, other, more advanced transmission conditions are used, involving derivatives of arbitrary order into tangential and normal direction on the interface, cf. Section 2.4. Those can be approximations to the Sommerfeld radiation condition or optimized conditions, cf. [64]. Note also that the question of transmission conditions for the Helmholtz equation is closely related to the question of non-reflecting boundary conditions, cf. Section 2.4. Under the assumptions given in [140], the subdomain matrices defined in Equation (3.2.3) are equivalent to choosing

$$\mathcal{B}_{ij} = \frac{\partial}{\partial n_j} + \imath k \tag{3.2.5}$$

in Equation (3.2.4). These transmission conditions can be derived in a simplified setting as a zeroth order approximation of the Taylor series at the frequency $\xi = 0$ of the optimal transmission conditions [63, Section 2]. The optimal transmission conditions are computed using Fourier analysis, cf. also Section 3.3.

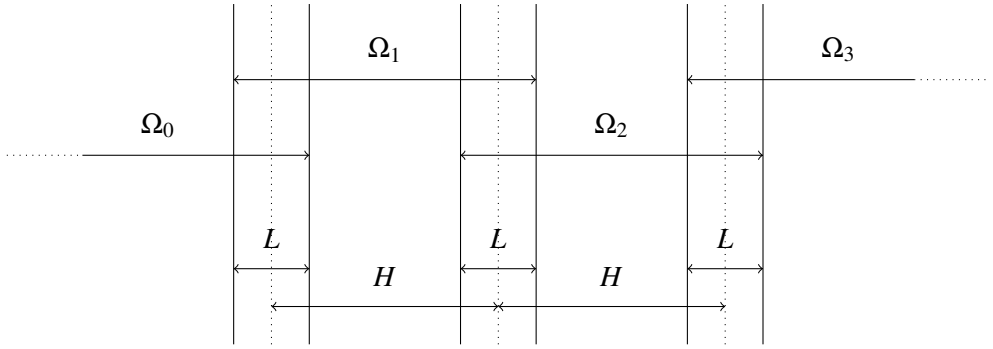


Figure 3.3.1. *Decomposition of the plane into $N = 4$ overlapping strips as defined in Equation (3.3.1).*

3.3 Fourier analysis

Fourier analysis can be used as a tool to analyze the convergence rates of RAS methods and to develop better transmission conditions, see e.g. [57]. While the resulting *optimized Schwarz* methods show optimal convergence rates in the model case analyzed, that is with the plane divided into two subdomains, they yield non-optimal behavior if the number of subdomains is increased [60]. The same is true for the transmission conditions presented above. In order to understand which Fourier modes cause the slow convergence in the multi-subdomain case, we here examine the properties of the one-level method introduced in Section 3.2 via Fourier analysis.

3.3.1 Fourier analysis for zeroth order transmission conditions

We examine the Jacobi-Schwarz algorithm introduced in Subsection 3.2.1 equipped with the zeroth order Robin transmission conditions defined in Equation (3.2.5). The computations are inspired by [36], where a similar analysis has been done for a different PDE. The domain $\Omega = \mathbb{R}^2$ is assumed to be the real plane decomposed into strips of infinite lengths as follows, cf. Figure 3.3.1:

$$\Omega_0 = (-\infty, 0 + \frac{L}{2}) \times \mathbb{R} \tag{3.3.1a}$$

$$\Omega_j = ((j-1)H - \frac{L}{2}, jH + \frac{L}{2}) \times \mathbb{R},\ 1 \leq j \leq \tilde{N} \tag{3.3.1b}$$

$$\Omega_{\tilde{N}+1} = (\tilde{N}H - \frac{L}{2}, \infty) \times \mathbb{R}, \tag{3.3.1c}$$

where H is the width of the non-overlapping strips, and L is the size of the overlap. For ease of presentation, we define the x-coordinates of the boundary of the subdomain Ω_j in x-direction as

$$x_j^- := (j-1)H - \frac{L}{2}, \qquad x_j^+ = jH + \frac{L}{2},$$

and set $N := \tilde{N} + 2$ to be the total number of subdomains. The Jacobi-Schwarz method with Robin

transmission conditions for the decomposition defined in Equation (3.3.1) reads

$$-\Delta u_j^{n+1} - k^2 u_j^{n+1} = f \qquad \text{in } \Omega_j \tag{3.3.2a}$$

$$\left(\frac{\partial}{\partial n_j} + \imath k\right) u_j^{n+1}(x_j^-, y) = \left(\frac{\partial}{\partial n_j} + \imath k\right) u_{j-1}^{n}(x_j^-, y), \qquad y \in \mathbb{R} \tag{3.3.2b}$$

$$\left(\frac{\partial}{\partial n_j} + \imath k\right) u_j^{n+1}(x_j^+, y) = \left(\frac{\partial}{\partial n_j} + \imath k\right) u_{j+1}^{n}(x_j^+, y), \qquad y \in \mathbb{R} \tag{3.3.2c}$$

for $1 \leq j \leq \tilde{N}$. The "boundary" subdomains Ω_j, $j \in \{0, \tilde{N}+1\}$, satisfy a similar system of equations, where the transmission condition to the non-existing neighbor is missing.

For the analysis it suffices to consider by linearity the case with right-hand side $f = 0$ and to analyze convergence to the zero solution. We consider subdomain Ω_j, $1 \leq j \leq \tilde{N}$, and apply a Fourier transformation in the y-direction with Fourier variable ξ to Equation (3.3.2). The Fourier transformation of a function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ is defined by

$$\hat{g}(x, \xi) = \mathcal{F}_y\left[g(x,y)\right](\xi) = \int_{-\infty}^{\infty} g(x,y) \mathrm{e}^{iy\xi} \mathrm{d}y, \quad \xi \in \mathbb{R}.$$

We obtain from Equation (3.3.2) using that $\mathcal{F}_y\left[\frac{\partial}{\partial x} g(x,y)\right](\xi) = -\imath\xi \mathcal{F}_y\left[g(x,y)\right](\xi)$

$$\frac{\partial^2}{\partial x^2} \hat{u}_j^{n+1}(x, \xi) + \left(\xi^2 - k^2\right) \hat{u}_j^{n+1}(x, \xi) = 0, \tag{3.3.3a}$$

$$\left(-\frac{\partial}{\partial x} + \imath k\right) \hat{u}_j^{n+1}\left(x_j^-, \xi\right) = \left(-\frac{\partial}{\partial x} + \imath k\right) \hat{u}_{j-1}^{n}\left(x_j^-, \xi\right), \tag{3.3.3b}$$

$$\left(\frac{\partial}{\partial x} + \imath k\right) \hat{u}_j^{n+1}\left(x_j^+, \xi\right) = \left(\frac{\partial}{\partial x} + \imath k\right) \hat{u}_{j+1}^{n}\left(x_j^+, \xi\right). \tag{3.3.3c}$$

The ansatz

$$\hat{u}_j^n(x, \xi) = A_j^n \mathrm{e}^{\lambda(\xi)(x - M_j)} + B_j^n \mathrm{e}^{-\lambda(\xi)(x - M_j)}, \tag{3.3.4}$$

where $M_j = \left(j - \frac{1}{2}\right) H$ is the midpoint of subdomain Ω_j and $\lambda(\xi) = \sqrt{\xi^2 - k^2}$, gives with

$$\lambda_1 := \lambda(\xi) + \imath k$$
$$\lambda_2 := -\lambda(\xi) + \imath k$$

the following linear system

$$\begin{pmatrix} \lambda_2 \mathrm{e}^{-\lambda(\xi)\frac{H+L}{2}} & \lambda_1 \mathrm{e}^{\lambda(\xi)\frac{H+L}{2}} \\ \lambda_1 \mathrm{e}^{\lambda(\xi)\frac{H+L}{2}} & \lambda_2 \mathrm{e}^{-\lambda(\xi)\frac{H+L}{2}} \end{pmatrix} \begin{pmatrix} A_j^{n+1} \\ B_j^{n+1} \end{pmatrix}$$

$$= \begin{pmatrix} \lambda_2 \mathrm{e}^{\lambda(\xi)\frac{H-L}{2}} & \lambda_1 \mathrm{e}^{-\lambda(\xi)\frac{H-L}{2}} & 0 & 0 \\ 0 & 0 & \lambda_1 \mathrm{e}^{-\lambda(\xi)\frac{H-L}{2}} & \lambda_2 \mathrm{e}^{\lambda(\xi)\frac{H-L}{2}} \end{pmatrix} \begin{pmatrix} A_{j-1}^n \\ B_{j-1}^n \\ A_{j+1}^n \\ B_{j+1}^n \end{pmatrix}.$$

Defining

$$\begin{aligned} d &= \lambda_1^2 \mathrm{e}^{\lambda(\xi)(L+H)} - \lambda_2^2 \mathrm{e}^{-\lambda(\xi)(L+H)}, \\ \alpha_1 &= -\lambda_2^2 \mathrm{e}^{-\lambda(\xi)L}, & \alpha_2 &= -\lambda_1\lambda_2 \mathrm{e}^{-\lambda(\xi)H}, \\ \alpha_3 &= \lambda_1^2 \mathrm{e}^{\lambda(\xi)L}, & \alpha_4 &= \lambda_1\lambda_2 \mathrm{e}^{\lambda(\xi)H}, \end{aligned}$$

we get

$$\begin{pmatrix} A_j^{n+1} \\ B_j^{n+1} \end{pmatrix} = \frac{1}{d} \begin{pmatrix} \alpha_1 & \alpha_2 & 0 & 0 & \alpha_3 & \alpha_4 \\ \alpha_4 & \alpha_3 & 0 & 0 & \alpha_2 & \alpha_1 \end{pmatrix} \begin{pmatrix} A_{j-1}^n \\ B_{j-1}^n \\ A_j^n \\ B_j^n \\ A_{j+1}^n \\ B_{j+1}^n \end{pmatrix}. \tag{3.3.5}$$

For the boundary subdomains, we get

$$A_{\tilde{N}+1}^{n+1} = 0, \quad B_{\tilde{N}+1}^{n+1} = \begin{pmatrix} \frac{\lambda_2}{\lambda_1}\mathrm{e}^{-\lambda(\xi)L} & \mathrm{e}^{-\lambda(\xi)H} & 0 & 0 \end{pmatrix} \begin{pmatrix} A_{\tilde{N}}^n \\ B_{\tilde{N}}^n \\ A_{\tilde{N}+1}^n \\ B_{\tilde{N}+1}^n \end{pmatrix}$$

and

$$A_0^{n+1} = \begin{pmatrix} 0 & 0 & \mathrm{e}^{-\lambda(\xi)H} & \frac{\lambda_2}{\lambda_1}\mathrm{e}^{-\lambda(\xi)L} \end{pmatrix} \begin{pmatrix} A_0^n \\ B_0^n \\ A_1^n \\ B_1^n \end{pmatrix}, \quad B_0^{n+1} = 0.$$

The above formulas define the iteration matrix Ψ that maps the coefficients of the local functions in one iteration to the coefficients of the local functions in the next iteration:

$$\begin{pmatrix} A_0^{n+1} \\ B_0^{n+1} \\ \vdots \\ A_{\tilde{N}+1}^{n+1} \\ B_{\tilde{N}+1}^{n+1} \end{pmatrix} = \Psi \begin{pmatrix} A_0^n \\ B_0^n \\ \vdots \\ A_{\tilde{N}+1}^n \\ B_{\tilde{N}+1}^n \end{pmatrix}$$

The columns and rows of Ψ associated to B_0^n and $A_{\tilde{N}+1}^n$ can be eliminated as these values are zero and do not contribute to the other coefficients. As by assumption we investigate convergence to the zero solution, the spectral radius of the iteration matrix $\rho(\Psi)$ has to be smaller than 1 in order for the iteration to converge.

3.3.2 Interpretation of the results

As a first step, we look at the case of $N = 2$ subdomains and verify that our results coincide with those derived in other works before, see [62, 63]. In fact, the matrix Ψ in this case has only the eigenvalue $\frac{\lambda_2}{\lambda_1} e^{-\lambda(\xi)L}$, which means that the absolute value of the convergence rate ρ_{T0} of the two subdomain method defined by

$$\hat{u}_j^{n+1} = \rho_{T0}^2(\xi, L)\,\hat{u}_j^{n-1}$$

is

$$|\rho_{T0}(\xi, L)| = \begin{cases} e^{-L\sqrt{\xi^2-k^2}}, & \xi^2 \geq k^2 \\ \frac{\left|\sqrt{k^2-\xi^2}-k\right|}{\left|\sqrt{k^2-\xi^2}+k\right|}, & \xi^2 < k^2. \end{cases} \tag{3.3.6}$$

If $k > 0$, we hence have

$$|\rho_{T0}(\xi, L)| \begin{cases} < 1, & \xi^2 < k^2 \\ = 1, & \xi^2 = k^2 \text{ or } (L = 0 \text{ and } \xi^2 > k^2) \\ < 1, & \xi^2 > k^2 \text{ and } L > 0 \end{cases}$$

and the convergence rates are only good for frequencies away from the wave number k. Figure 3.3.2 shows the convergence rates for the method analyzed above and additionally for Dirichlet transmission conditions, that is [62]

$$\rho(\xi, L)^2 = e^{-2L\sqrt{\xi^2-k^2}}. \tag{3.3.7}$$

Hence in the special case of two subdomains, convergence rates for Fourier frequencies $\xi > k$ are the same with Dirichlet and Robin transmission conditions. Robin transmission conditions consequently offer an improvement only for the low-frequent Fourier modes; the convergence of the higher frequencies depends solely on the width of the overlap. This deficiency could for example be resolved by optimized Schwarz methods [57], compare also the overview in Section 2.4.

As a next step, we examine the eigenvalues of the iteration matrix Ψ for the general case of N subdomains. Figure 3.3.3 shows the modulus of the maximum eigenvalue of the matrix Ψ for different values of the number of subdomains N and the size of the overlap L. Figure 3.3.4 illustrates the dependence of the spectral radius on the width of the subdomains H. The figures allow to draw the following conclusions:

For high frequent Fourier modes with $\xi > k$, the maximum eigenvalue is not affected by the number of subdomains N, but depends solely (among the parameters varied) on the size of the overlap L. In this case, increasing the size of the overlap reduces the maximum eigenvalue of the iteration matrix; i.e. with a larger overlap the method will eventually converge faster.

For low frequent Fourier modes with $\xi < k$, on the contrary, the number of subdomains employed has a huge influence on the eigenvalues; if it increases, so does the spectral radius of the matrix Ψ. Moreover, while the size of the overlap L has no influence on the spectral radius in the two-subdomain case, for $N > 2$ subdomains, an increase of the size of the overlap L causes an increase in the spectral

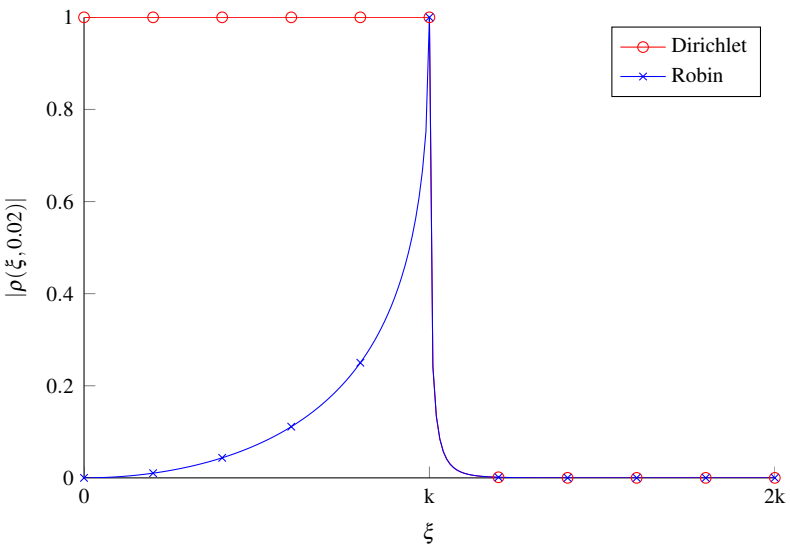


Figure 3.3.2. *The convergence rates computed with Fourier analysis for the* RAS *method with Robin and Dirichlet transmission conditions, respectively. The plot shows* $|\rho(\xi, 0.02)|$ *for* $k = 5$ *and* $H = 1$ *defined in Equation (3.3.6) and Equation (3.3.7).*

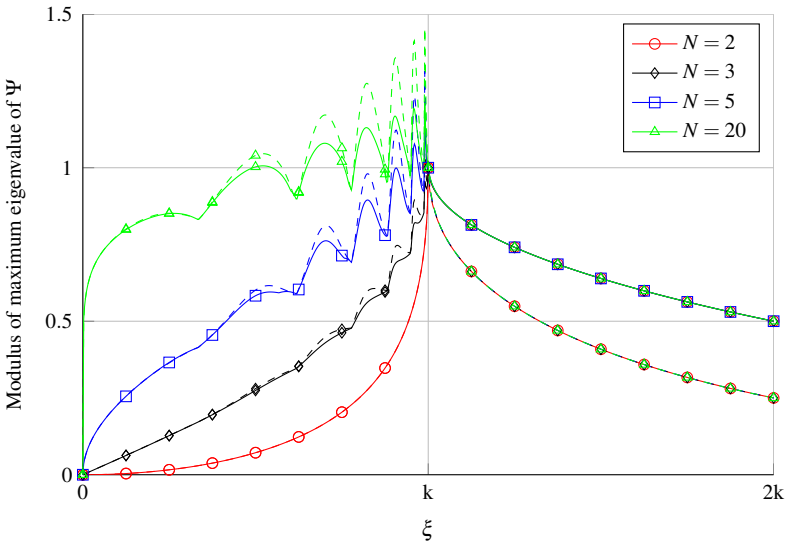


Figure 3.3.3. *Convergence rates computed with Fourier analysis for a strip decomposition, varying the number of subdomains.* $k = 20$, $H = 1$. *The solid lines correspond to an overlap of* $L = 2 \cdot 10^{-2}$, *the dashed ones to* $L = 4 \cdot 10^{-2}$. *Please note that only every* 50*th data point has a marker.*

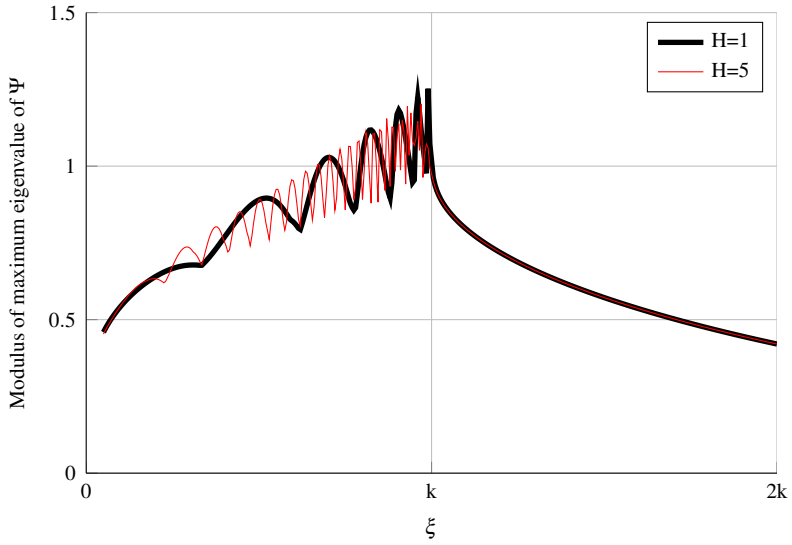


Figure 3.3.4. *Convergence rates computed with Fourier analysis for a strip decomposition, varying the width of the subdomains H. $k = 20$, $N = 10$.*

radius of the iteration matrix Ψ. Hence the method converges slower for these modes if the overlap is increased. Maybe most importantly, for some modes the spectral radius of the iteration matrix is larger than 1. This means that the method diverges for these modes. The more subdomains are used, the larger is the number of divergent modes. Moreover, the spectral radius of the matrix Ψ does not depend monotonically on the Fourier frequency ξ. Instead, the larger the width H of the subdomains is, that is the more wavelengths fit into one subdomain, the more oscillations are present in the spectral radius. Comparing Figure 3.3.3 with Figure 3.3.4 shows furthermore that the slow convergence is mainly caused by the increase in the number of subdomains and not by the presence of more wavelengths in the problem, as the maximum spectral radius seems to be hardly influenced by H in Figure 3.3.4.

In order to better understand the sobering results for the low-frequent Fourier modes, we investigate the eigenvalues of the iteration matrix Ψ in more detail in Figure 3.3.5 and Figure 3.3.6. In Figure 3.3.5, both the minimum and the maximum eigenvalue of the matrix Ψ are plotted for all frequencies. While they coincide for Fourier frequencies $\xi > k$, for the low-frequent modes, the eigenvalues differ in their absolute value. The minimal eigenvalues show the same oscillatory behavior as the maximal ones, but their absolute value always remains bounded from above by one. In Figure 3.3.6, all the eigenvalues of the matrix Ψ are shown for a few frequencies ξ. The closer the Fourier frequency ξ is to the wave number k, the larger the radius of the circle on which the eigenvalues lie in the complex plane becomes. For some frequencies, the circle then gets distorted and seems to be split up into two arcs, parts of which might then lie outside of the unit circle, hence

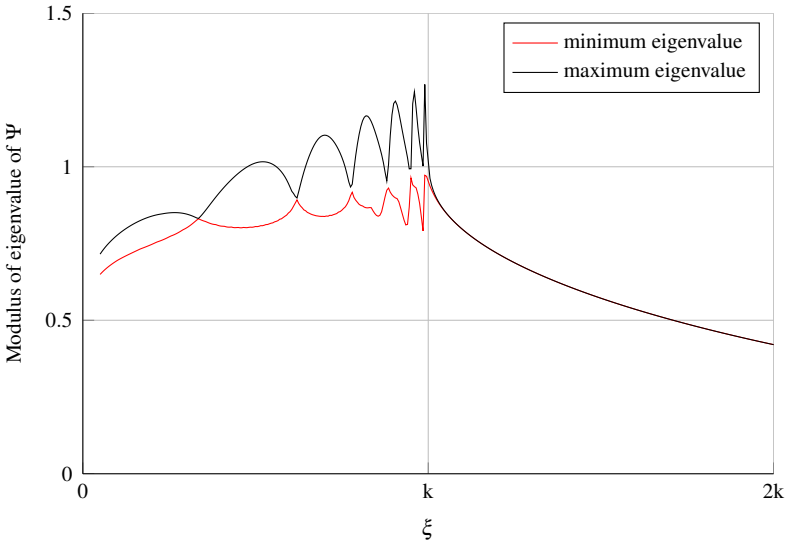
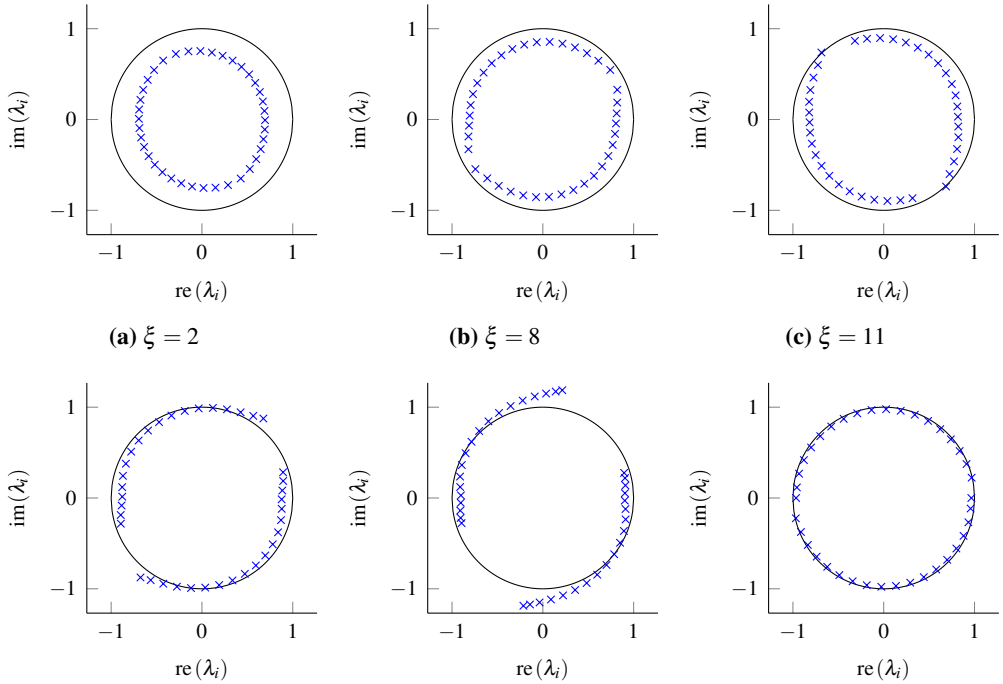


Figure 3.3.5. *Minimum and maximum eigenvalues of the iteration matrix* Ψ. 20 *subdomains,* $k = 20$, *overlap of* $L = 2 \cdot 10^{-2}$.



(a) $\xi = 2$ **(b)** $\xi = 8$ **(c)** $\xi = 11$

(d) $\xi = 16$ **(e)** $\xi = 18$ **(f)** $\xi = 19$

Figure 3.3.6. *Eigenvalues* λ_i *of the iteration matrix* Ψ *for different Fourier frequencies* ξ. *The black line is the unit circle.* 20 *subdomains,* $k = 20$, $H = 1$, *overlap of* $L = 2 \cdot 10^{-2}$.

yielding a spectral radius of Ψ that is bigger than one. This "splitting" of the circle apparently causes the spectral radius of the iteration matrix Ψ to oscillate.

Concluding, in contrast to the $N = 2$ subdomains case, there are several Fourier modes that converge at a low rate or not at all. All these modes are associated to Fourier frequencies that are smaller than the wave number k. The subspace of problematic modes becomes bigger the more subdomains are employed in the strip domain decomposition. This is in line with the numerical experiments in [61, Section 4]. The exactly same kind of analysis could also be applied to any other kind of transmission condition. However, the numerical experiments of [61] suggest that this would only partially remove the convergence problems that we encounter. Gander and Zhang [61] have proposed a way to modify the optimized parameters in order to account for the multi-domain case, but did not achieve better results than with low order Taylor conditions. Therefore, the development of a second level that tackles the subspace of slowly converging modes is important.

Chapter 4

Adding a second level

The convergence rates of one-level DDMs as introduced in Chapter 3 depend on the number of subdomains in the domain decomposition [147]. A standard remedy for this problem is to add a second level to the method. It captures global features that the purely locally acting RAS method is not able to treat. This second level is called the *coarse space*, accounting for the fact that it has typically less degrees of freedom than the first level and is cheaper to solve. Despite its name, it is not necessarily related to a coarser mesh. In order to design and test a coarse space for the Helmholtz equation, it is indispensable to understand how it influences the eigenvalues of the preconditioned operator and hence the convergence behavior of the iterative method. Contrarily to the s.p.d. case [114, 115], for indefinite matrices there seems to be no way to ensure that using a two-level method with an *arbitrarily* chosen second level always accelerates convergence [53, 124]. This is why choosing the right, problem dependent coarse space is utterly important for indefinite systems as those arising from the Helmholtz equation. This question will be investigated in Chapter 5.

Besides the problem of *which modes* the coarse space should contain, there is the equally important question of *how* it should be *incorporated* into the method. This is the focus of this chapter. Depending on the context, several alternative approaches have been proposed. In geometric multigrid methods, see e.g. [70], the coarse space is often associated to the coarsest grid in the grid hierarchy and transition to it is done as transition to all the other levels. However, different choices are possible, as e.g. in the wave-ray multigrid method [18], where due to the special form of the coarse levels, completely different operators are used. In DDMs, adding the coarse space is less straightforward. Besides additive and multiplicative Schwarz methods, where the coarse space can be treated exactly as the spaces associated to the subdomains [138], and deflation and balancing methods [42, 106, 112], various other approaches are also possible, see e.g. [138, Section 2.3] for an overview, and [114, 143] for a comparison of different approaches for s.p.d. systems. Due to the strong connection to our work, we note that in [92] yet another method is used.

This chapter presents three methods to use the coarse space, the deflation and the balancing preconditioners and the approach presented in [74]. The latter is to our knowledge the only one that results in a provably invertible preconditioner with the desired filtering properties for non-symmetric, indefinite linear systems of equations. The results in Section 4.2 and Section 4.3 are

presented in condensed form by Conen, Dolean, Krause, and Nataf [28]; some of the text has been copied verbatim.

4.1 Basic definitions

The first step towards the introduction of the methods in this chapter is the definition of some matrices that appear in the following sections. Let $B \in \mathbb{C}^{n\times n}$, and let $Z, Y \in \mathbb{C}^{n\times r}$, $r < n$, have full column rank. The matrix B is related to the stiffness matrix A either directly via $B = A$ or by using the preconditioned matrix $B = M^{-1}A$. Define the matrices

$$E = Y^* BZ, \qquad \Xi = ZE^{-1}Y^*, \tag{4.1.1a}$$

$$P_D = I - B\Xi, \qquad Q_D = I - \Xi B, \tag{4.1.1b}$$

where we assume that E is invertible and $* \in \{T, \dagger\}$, where T denotes the transpose and $\dagger$ the conjugate transpose. We usually choose $* = \dagger$, unless mentioned otherwise. The additional assumption on E is not necessary if B is Hermitian positive definite, $Y = Z$ has full column rank and $*$ is the Hermitian transpose. The following lemma states a few basic identities that can be found e.g. in [114].

Lemma 4.1.1. *With the definitions in Equation (4.1.1), the following identities hold:*

$$P_D^2 = P_D, \qquad Q_D^2 = Q_D, \qquad P_D B = BQ_D, \qquad Q_D\Xi = 0,$$
$$\Xi B\Xi = \Xi, \qquad \Xi BZ = Z, \qquad Y^* B\Xi = Y^*.$$

Proof. The proof is an easy calculation employing the definitions. □

For the methods introduced in this chapter, the matrix Z (and Y, respectively) implicitly defines the coarse space $\mathcal{Z}$, i.e. the columns of Z (and Y, respectively) represent the basis vectors of $\mathcal{Z}$. We only consider the case $Z = Y$. While this is the standard choice in the Hermitian case, for the non-Hermitian problem considered in this thesis, one could possibly achieve better results choosing $Z \neq Y$, as left and right eigenvectors of the matrix B might differ. However, it is unclear how it should be constructed in practice, when the eigenvectors are not known.

4.2 The deflation operator

The possibly simplest approach to add a second level to a one-level method is *deflation*. The deflation operator is a projection to the complement of the coarse space. It thus moves the deflated, critical eigenvalues to zero; the deflated system is singular while still being consistent. This imposes additional difficulties when applying a Krylov subspace method such as the GMRES method. Because of this, even though the GMRES method can be adapted to singular systems, see e.g. [128], we do not consider the deflation operator in the numerical experiments in Chapter 6 and Chapter 7. We include it in the theoretical discussion as it is the easiest to analyze and is a building block of all the other methods presented.

4.2.1 Definition and basic properties

In this section, we introduce the deflation operator, see e.g. [114], and state its basic properties. Deflation is defined as follows:

Definition 4.2.1 (Deflation)**.** Let $A \in \mathbb{C}^{n\times n}$ and let $Z, Y \in \mathbb{C}^{n\times r}$, $r < n$, and use the definitions in Equation (4.1.1) with $B := A$. The matrix P_D is called the *deflation matrix*. It is used to precondition the system $A\mathbf{u} = \mathbf{b}$ as follows: The solution $\mathbf{u}$ can be decomposed as

$$\mathbf{u} = \Xi A\mathbf{u} + Q_D\mathbf{u} = \Xi\mathbf{b} + Q_D\mathbf{u}.$$

If $\tilde{\mathbf{u}}$ is the solution of the *deflated system*

$$P_D A\tilde{\mathbf{u}} = P_D\mathbf{b}, \tag{4.2.1}$$

then $Q_D\tilde{\mathbf{u}}$ solves

$$A\left(Q_D\tilde{\mathbf{u}}\right) = P_D\mathbf{b},$$

and hence $Q_D\tilde{\mathbf{u}} = Q_D\mathbf{u}$ and $\mathbf{u}$ can be computed from $\tilde{\mathbf{u}}$ and $\Xi\mathbf{b}$.

Deflation follows a quite simple principle. Lemma 4.1.1 implies

$$P_D AZ = (I - A\Xi)AZ = 0, \tag{4.2.2a}$$

and similarly

$$Y^* P_D A = Y^* A Q_D = Y^* A(I - \Xi A) = 0. \tag{4.2.2b}$$

Hence deflation removes the column vectors of the matrix Z (of the matrix Y, respectively) from the system by putting them into the kernel of the deflated operator (into the kernel of the Hermitian transpose of the deflated operator, respectively). If Z and Y contain some of the right and left eigenvectors of A, respectively, the corresponding eigenvalues are shifted to zero [42, Theorem 2.11]. Consequently, problematic modes can simply be "removed" from the linear system in this simple setting. Without the assumption that the matrices Y and Z contain eigenvectors of the stiffness matrix A, the situation is more complicated and will be examined in Subsection 4.2.2.

4.2.2 Spectral properties for the symmetric positive definite case

For s.p.d. matrices, the question of how the coarse space influences the convergence rates of the two-level method has been examined extensively e.g. in [113, 115]. In particular, Nabben, Tang, and Vuik [115] showed that the spectrum never deteriorates compared to the non-deflated operator for any choice of the coarse matrix Z. Here, not deteriorating means that the smallest non-zero eigenvalue does not decrease and the largest eigenvalue does not increase.

Theorem 4.2.2 ([115, Theorem 2.2])**.** *Let $A \in \mathbb{R}^{n\times n}$ be an s.p.d. matrix. Let $Z \in \mathbb{R}^{n\times r}$ with $r < n$ have full column rank. Let M be a real, $n \times n$ s.p.d. matrix. Then the following inequality holds:*

$$\kappa_{\text{eff}}(M^{-1}P_D A) < \kappa(M^{-1}A),$$

where $\kappa(C)$ for a matrix $C \in \mathbb{C}^{n\times n}$ denotes the condition number *of the matrix C and $\kappa_{\text{eff}}(C)$ denotes* effective condition number *of the matrix C, that is the ratio of its largest to smallest nonzero eigenvalue*

$$\kappa_{\text{eff}}(C) = \frac{\lambda_{\max}(C)}{\lambda_{\min}(C)},$$

where $\lambda_{\min}(C) = \min_{i:|\lambda_i|\neq 0} |\lambda_i|$ and $\lambda_{\max}(C) = \max_i |\lambda_i|$. Here λ_i, $1 \leq i \leq n$, are the eigenvalues of the matrix C.

For s.p.d. matrices, the deflated matrix has hence always a better condition number than the original, non-deflated matrix, no matter how the matrix Z is chosen. Moreover, this remains true if additionally a preconditioner is applied to those matrices, provided that the preconditioning matrix M is also s.p.d. When the matrices cannot be assumed to be (semi-)definite, the situation gets more complicated and a similar behavior can only be concluded under additional assumptions, see Subsection 4.2.3 for an investigation of this issue. In the following, we give an example, where deflation deteriorates the spectrum of an indefinite matrix.

Example 4.2.3. Let

$$A = \begin{pmatrix} -4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 8 \end{pmatrix}, \quad Y = Z = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

The matrix A is hence symmetric, but indefinite with condition number 8. We get

$$P_D A = \begin{pmatrix} -8 & 0 & 8 \\ 0 & 1 & 0 \\ 8 & 0 & -8 \end{pmatrix}.$$

The eigenvalues of $P_D A$ are -16, 1, and 0. It follows that $\kappa_{\text{eff}}(P_D A) = 16 > \kappa(A) = 8$ and Theorem 4.2.2 does *not* hold in the indefinite case.

While the positive definiteness is consequently a crucial assumption in Theorem 4.2.2, we can easily extend the theorem to complex, Hermitian, positive definite matrices:

Theorem 4.2.4. *Let $A \in \mathbb{C}^{n\times n}$ be Hermitian positive definite. Let $Z \in \mathbb{C}^{n\times r}$ with $r < n$ have full column rank. Using Definition 4.2.1 with $* = \dagger$, the following inequality holds:*

$$\kappa_{\mathit{eff}}(P_D A) < \kappa(A).$$

Proof. The proof follows literally the one of [115, Theorem 2.1]. □

4.2.3 Spectral analysis for indefinite, Hermitian matrices

Example 4.2.3 shows that adding a second level to a preconditioner for an indefinite matrix may negatively affect the spectrum of the preconditioned operator and hence the convergence rates of the iterative method. This section explains *why* this is the case, i.e. why the results from [115], presented

in Subsection 4.2.2, do not hold true in the indefinite case. To be able to provide results, we restrict to a simpler setting: Use Definition 4.2.1 with $* = \dagger$ the conjugate transpose, let $Z = Y$, and let the matrix A be Hermitian[1]. Let $\mathbf{v}_i$, $1 \leq i \leq n$ be an orthonormal basis of eigenvectors of A with corresponding eigenvalues $\lambda_i \in \mathbb{R}$. Let $|\lambda_i| \leq |\lambda_{i+1}|$ for all $1 \leq i < n$. We may consider the columns of Z separately in a recursive procedure, using a variant of [85, Theorem 3.2]:

Theorem 4.2.5. *Let* $P^{(k)} = I - AZ_k \left(Z_k^\dagger A Z_k\right)^{-1} Z_k^\dagger$ *with* $Z_k = [\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_k]$, *where* $\tilde{Z}_j \in \mathbb{C}^{n \times l_j}$ *has full rank* l_j. *Let* $\tilde{Z}_i^\dagger \tilde{A}_{i-1} \tilde{Z}_i$ *and* $Z_i^\dagger A Z_i$ *be non-singular for all* $1 \leq i \leq k$. *Then*

$$P^{(k)} A = P_k P_{k-1} \ldots P_1 A,$$

where $P_{i+1} = I - \tilde{A}_i \tilde{Z}_{i+1} \left(\tilde{Z}_{i+1}^\dagger \tilde{A}_i \tilde{Z}_{i+1}\right)^{-1} \tilde{Z}_{i+1}^\dagger$, $\tilde{A}_i = P_i \tilde{A}_{i-1}$, $\tilde{A}_0 = A$.

Proof. In [85, Theorem 3.2], the positive definiteness is only used for the non-singularity of $\tilde{Z}_i^\dagger \tilde{A}_{i-1} \tilde{Z}_i$ and $Z_i^\dagger A Z_i$ that we added to the assumptions. The rest of the proof for our situation is literally the same, as the symmetry is not used. □

We consequently restrict without loss of generality (w.l.o.g.) to $Z \in \mathbb{C}^{n \times 1}$ with only one column, $Z := \sum_{i \in I} \alpha_i \mathbf{v}_i$, where $\alpha_i \neq 0 \in \mathbb{C}$ are coefficients and $I \subseteq \{1, \ldots, n\}$.

It is clear that when $|I| = 1$ and hence $Z = \mathbf{v}$ for some eigenvector $\mathbf{v}$ with eigenvalue λ of A, the deflation operator P_D removes exactly this eigenvalue from the spectrum of A, i.e. $\sigma(P_D A) = \sigma(A) \setminus \{\lambda\} \cup \{0\}$. In this case, the effective condition number of the deflated matrix cannot be worse than the one of the original matrix A. Even though the situation is similar to the s.p.d. case for this very simple setting, it changes substantially, if the columns of Z are linear combinations of eigenvectors associated to eigenvalues with different signs. As a first step towards understanding this in more detail, we compute $P_D A \mathbf{v}_k$ for all k and get the following lemma:

Lemma 4.2.6 (Structure of $P_D A$)**.** *W.l.o.g. assume* $I = \{1, 2, \ldots, |I|\}$. *In the basis of eigenvectors* $(\mathbf{v}_i)_{1 \leq i \leq n}$, $P_D A$ *is a block diagonal matrix with the two blocks C and D, i.e. there is a basis transformation V such that*

$$V^\dagger P_D A V = \begin{pmatrix} C & 0 \\ 0 & D \end{pmatrix},$$

where $C \in \mathbb{C}^{|I| \times |I|}$ *is the block associated to* $(\mathbf{v}_i)_{i \in I}$ *and is defined by*

$$C_{ii} = \frac{\sum_{k \in I \setminus \{i\}} |\alpha_k|^2 \lambda_i \lambda_k}{\sum_{k \in I} |\alpha_k|^2 \lambda_k}, \quad C_{ij} = -\frac{\alpha_j \bar{\alpha}_i \lambda_j \lambda_i}{\sum_{k \in I} |\alpha_k|^2 \lambda_k}, \qquad \forall i, j \in I, i \neq j,$$

and D is a diagonal matrix with diagonal entries $\lambda_{|I|+1}, \ldots, \lambda_n$.

The following theorem treats the simple case in which bounds on the eigenvalues of the deflated operator can be guaranteed.

[1] The matrix associated to the Helmholtz problem defined in Equation (1.3.4) is Hermitian if $\Gamma_R = \emptyset$.

Theorem 4.2.7. *If all λ_i, $i \in I$, have the same sign, then*

$$\lambda_{\max}(P_D A) \leq \lambda_{\max}(A) \quad \textit{and} \quad \lambda_{\min}(P_D A) \geq \lambda_{\min}(A).$$

Proof. Let V be the matrix whose columns are the eigenvectors $\mathbf{v}_i$, $1 \leq i \leq n$. According to Lemma 4.2.6, after reordering, $V^\dagger P_D A V$ has block structure with a block C associated to $\{\mathbf{v}_i\}_{i\in I}$ and a diagonal block D. As the two blocks are decoupled and all eigenvalues of D are eigenvalues of A, we can consider only C. Eigenvalues are invariant under change of basis and by assumption, all λ_i, $i \in I$, have the same sign. Since either C or $-C$ is Hermitian positive semi-definite, we can use the result for the real, s.p.d. case [115, Theorem 2.1], whose proof is literally the same for complex matrices, to prove the claim. □

Consequently, if all eigenvalues associated to eigenvectors that contribute to the vector Z have the same sign, the spectrum of the deflated operator can be bounded by the one of the original operator. Deterioration of the spectrum could thus be avoided if an orthonormal basis of eigenvectors of the global operator was known. This is not feasible in practice; knowing the full spectral information of the global operator, there is no more need for employing an iterative solution technique.

The remaining question is what happens to the eigenvalues if the λ_i, $i \in I$, have different signs. For simplicity, we restrict to the case $|I| = 2$.

Theorem 4.2.8. *Let $|I| = 2$, i.e. $Z = \alpha_i \mathbf{v}_i + \alpha_j \mathbf{v}_j$ for some $1 \leq i, j \leq n$. Then $\lambda_{\max}(P_D A) > \lambda_{\max}(A)$ holds, if and only if i and j are chosen such that λ_i and λ_j have different signs and*

$$\frac{\left(|\alpha_i|^2 + |\alpha_j|^2\right) |\lambda_i| \, |\lambda_j|}{\left||\alpha_i|^2 \lambda_i + |\alpha_j|^2 \lambda_j\right|} > |\lambda_k| \;\; \forall \; 1 \leq k \leq n. \tag{4.2.3}$$

Proof. This follows directly from Theorem 4.2.7 and Lemma 4.2.6, observing that the matrix C has the eigenvalues 0 and $\frac{\left(|\alpha_i|^2+|\alpha_j|^2\right)\lambda_i\lambda_j}{|\alpha_i|^2\lambda_i+|\alpha_j|^2\lambda_j}$. □

Theorem 4.2.8 implies that if eigenvectors associated to eigenvalues with different signs enter the coarse space, the eigenvalues of the deflated matrix might become arbitrarily large. Here we give a small example showing that the condition given in Theorem 4.2.8 can be fulfilled easily:

Example 4.2.9. Use the definitions in Example 4.2.3. The matrix

$$Z = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

is the sum of the (orthonormalized) eigenvectors associated to the eigenvalues -4 and 8, respectively. As predicted by the theory, $P_D A$ has the eigenvalue

$$-16 = \frac{(1+1)\cdot(-4)\cdot 8}{-4+8}$$

that is larger in modulus than any of the eigenvalues of A.

4.2.4 Spectral analysis of a modified deflation operator

In this section, we examine a variant of the deflation operator used e.g. in [4, 49] for the iterative solution of the Helmholtz equation, where $* = T$ in Definition 4.2.1. Even though the transpose T seems to be more suitable for complex symmetric matrices than the conjugate transpose $\dagger$, as it is closely related to the structure of the matrix, we show that the situation with this choice is even worse. We consider a complex symmetric, possibly non-Hermitian matrix A as it arises from the discretization of the Helmholtz equation in Equation (1.3.1) and assume that A is diagonalizable. It hence has an eigenvector matrix V such that $V^T AV$ is diagonal and $V^T V = I$ [78, Theorem 4.4.13]. Under these assumptions, a modified version of Theorem 4.2.8 holds, where the condition in Equation (4.2.3) is substituted by

$$\left| \frac{\left(\alpha_i^2 + \alpha_j^2\right) \lambda_i \lambda_j}{\alpha_i^2 \lambda_i + \alpha_j^2 \lambda_j} \right| > |\lambda_k| \quad \forall 1 \leq k \leq n.$$

Note that here the condition includes coefficients α_k^2 instead of $|\alpha_k|^2$, $k = i, j$. This means that the denominator $\left|\alpha_i^2 \lambda_i + \alpha_j^2 \lambda_j\right|$ might become arbitrarily large independently of the signs of the eigenvalues. Thus in contrast to the Hermitian transpose case, no sign restriction on λ_i and λ_j can prevent the eigenvalues of the deflated operator from becoming huge. This is illustrated in the following example:

Example 4.2.10. Let $A = \begin{pmatrix} 1 & \imath & 0 \\ \imath & 1 & 2\imath \\ 0 & 2\imath & 1 \end{pmatrix}$ with (orthogonal) eigenvectors $\mathbf{v}_1 = \begin{pmatrix} 1 & \sqrt{5} & 2 \end{pmatrix}^T$, $\mathbf{v}_2 = \begin{pmatrix} 2 & 0 & -1 \end{pmatrix}^T$, $\mathbf{v}_3 = \begin{pmatrix} 1 & -\sqrt{5} & 2 \end{pmatrix}^T$ and eigenvalues $\lambda_1 = 1 + \imath\sqrt{5}$, $\lambda_2 = 1$, $\lambda_3 = 1 - \imath\sqrt{5}$. Choosing $Z = \alpha \frac{1}{\|v_1\|} \mathbf{v}_1 + \frac{1}{\|v_2\|} \mathbf{v}_2$, $\alpha = \sqrt{\frac{\varepsilon - \lambda_2}{\lambda_1}}$ for some real number $0 < \varepsilon < 1$, we get

$$\lambda_{\max}(P_D A) = \left|1 + \imath\varepsilon^{-1}\sqrt{5}\right| > \lambda_{\max}(A) = \left|1 + \imath\sqrt{5}\right|.$$

These theoretical results suggest to use the Hermitian transpose instead of the transpose. Some numerical results on this issue will be given in Section 4.6.

4.3 The balancing Neumann-Neumann method

A well-known way to add a coarse space to the one-level RAS method is the balancing Neumann-Neumann (BNN) method introduced by Mandel [106]. For the more general version that we are using in this work see [112]. While this kind of preconditioner is well-understood for s.p.d. problems, for the indefinite, non-Hermitian matrix arising from the discretization of the Helmholtz equation difficulties arise. Even though Erlangga and Nabben [42] generalize the BNN idea to this case and claim invertibility of the resulting matrix, their results contain an error, for details see Subsection 4.3.2, that we were not aware of when starting the work presented in this manuscript. The

resulting preconditioned matrix is possibly singular and hence the GMRES method might encounter problems to find the right solution. Building on those wrong results, we have used the balancing preconditioner for a lot of the numerical experiments. We will comment on this in more detail in Chapter 6 and Chapter 7, where the numerical experiments are presented. In this section, we present the BNN preconditioner and discuss its properties.

4.3.1 Definition and basic properties

The BNN preconditioner reads for s.p.d. systems in its abstract form [114]

$$P_{\mathrm{B}} = P_D^T M^{-1} P_D + \Xi, \tag{4.3.1}$$

where $Z = Y$ and $B = A$ in Equation (4.1.1). This preconditioner and similar ones are suitable for Krylov methods as the residual r_n at any step n of the Krylov method remains orthogonal to the vector space spanned by the columns of Z [74]: $Z^* \mathbf{r}_n = 0$. Erlangga and Nabben [42] extend this approach to non-symmetric systems, using the formula

$$P_{\mathrm{B}} = Q_D M^{-1} P_D + \Xi, \tag{4.3.2}$$

where M^{-1} is the one-level preconditioner in Equation (3.2.2) and using the definitions in Equation (4.1.1).

The BNN preconditioners in Equation (4.3.1) and Equation (4.3.2) can be rewritten as a three-step method. Setting

$$\mathbf{u}^{n+1} \leftarrow \mathbf{u}^n + P_{\mathrm{B}}(\mathbf{f} - A\mathbf{u}^n) \tag{4.3.3}$$

is equivalent to iterating on the residuals $\mathbf{r}^i := \mathbf{f} - A\mathbf{u}^i$ in the following way

$$\mathbf{r}^{n+1} \leftarrow (I - AP_{\mathrm{B}})\mathbf{r}^n. \tag{4.3.4}$$

This can be rewritten as

$$\mathbf{r}^{n+1/3} \leftarrow P_D \mathbf{r}^n \tag{4.3.5a}$$

$$\mathbf{r}^{n+2/3} \leftarrow (I - AM^{-1})\mathbf{r}^{n+1/3} \tag{4.3.5b}$$

$$\mathbf{r}^{n+1} \leftarrow P_D \mathbf{r}^{n+2/3}. \tag{4.3.5c}$$

Hence, the balancing preconditioner consists of two ingredients, which can be separated from each other: application of the deflation operator P_D, see Definition 4.2.1, incorporating the second level, and application of the one-level preconditioner M^{-1}. In Subsection 4.3.3, we will examine the relation between the eigenvalues of the balancing and the deflation operators. Note that for the BNN preconditioner defined in Equation (4.3.2) a right-filtering property holds as an easy calculation shows:

$$P_B A Z = Z.$$

Assuming that P_B is non-singular, this can be rewritten as

$$AZ = P_B^{-1} Z.$$

4.3.2 A problematic observation

Apart from the filtering property derived in Subsection 4.3.1, another important property of the preconditioned matrix is its non-singularity. On the one hand, for a singular system the solution is not uniquely determined and, in contrast to the construction for the deflated matrix, cf. Section 4.2, it is not clear how the solution of a singular system would be related to the solution of the original system. On the other hand, the GMRES method encounters problems when naively applied to singular systems: Convergence might stagnate before a sufficiently accurate approximation of the discrete solution has been computed [128]. Solving a singular system with GMRES requires hence modifications of the solver in order to enable it to deal with the additional difficulties. For these reasons, a non-singular preconditioned matrix would be favorable. Erlangga and Nabben [42] examine this question. In particular, they state that the preconditioned system

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

is non-singular and consequently Krylov methods such as the GMRES method can be used without further modifications:

Theorem (Theorem 2.9 of [42])**.** *Let Z and Y be full ranked. Let M be non-singular. Then P_BA is non-singular. In addition, any zero eigenvalue of $M^{-1}P_DA$ is shifted to one in P_BA.*

Unfortunately, as we discovered after having done a significant portion of the work presented here, this is simply *wrong*. In general P_B and P_BA are singular: Consider

$$A = \begin{pmatrix} 2 & 5 & 2 \\ 0 & 6 & 0 \\ 0 & 1 & 4 \end{pmatrix}$$

with eigenvalues $\sigma(A) = \{2,4,6\}$, hence A is real, non-singular, positive definite, diagonalizable and all its eigenvalues are mutually distinct. Furthermore, let

$$Z = Y = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \qquad M^{-1} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

A, M, and $E = Y^TAZ = \begin{pmatrix} 11 \end{pmatrix}$ are clearly non-singular, but

$$\begin{pmatrix} 15 \\ -4 \\ 7 \end{pmatrix}$$

is an eigenvector of P_BA with eigenvalue 0 and hence P_BA is singular. Moreover, we have

$$\sigma\left(M^{-1}P_DA\right) = \left\{0, 0, \frac{52}{11}\right\},$$

and

$$\sigma\left(P_B A\right) = \left\{1, 0, \frac{52}{11}\right\}.$$

Consequently, not all zero eigenvalues of $M^{-1}P_D A$ are shifted to one in $P_B A$; the second claim of [42, Theorem 2.9] is also wrong.

In the previous example, the matrix A was definite, but not Hermitian. We also give an example for the case where A is Hermitian, but not positive definite. Let

$$M^{-1} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad Z = Y = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

Both matrices A and M^{-1} are real, symmetric and indefinite. Since they have mutually different eigenvalues, they are diagonalizable. Then

$$M^{-1}P_D A = \frac{1}{2}\begin{pmatrix} 1 & 2 & -2 \\ -1 & 2 & -2 \\ -2 & 0 & 0 \end{pmatrix}$$

has eigenvalues $0, 0, \frac{3}{2}$ with eigenvectors

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 3 \\ 1 \\ -2 \end{pmatrix},$$

where the eigenspace corresponding to the eigenvalue 0 is only one-dimensional and

$$P_{\mathrm{B}} A = \frac{1}{2}\begin{pmatrix} 1 & 2 & -2 \\ \frac{3}{2} & 1 & 1 \\ \frac{1}{2} & -1 & 3 \end{pmatrix}$$

has eigenvalues $0, 1, \frac{3}{2}$ with corresponding eigenvectors

$$\begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}.$$

Thus, here again, the balancing preconditioner is not guaranteed to be non-singular as opposed to the s.p.d. case [113].

The possible singularity of the BNN preconditioner P_B and the preconditioned matrix $P_B A$ is problematic. The GMRES method used to solve the preconditioned system might break down for singular systems without providing a sufficiently good approximation to the solution [128]. Assume that convergence is tested via the non-preconditioned residual $\mathbf{r}^{(n)} := \mathbf{f} - A\mathbf{u}^{(n)}$ or via the error $\mathbf{e}^{(n)} := \mathbf{u} - \mathbf{u}^{(n)}$, where $\mathbf{u}$ denotes the exact discrete solution and $\mathbf{u}^{(n)}$ denotes the approximation

to the solution $\mathbf{u}$ in step n of the iterative method. Then *if* the method converges, it converges to the correct solution. If, however, the method does not converge, it is unclear whether the reason is the indefiniteness of the system causing a breakdown of GMRES or a deficiency of the coarse space/RAS method, whose performances are under investigation. Therefore, the use of this preconditioner makes it difficult to correctly interpret cases where divergence or stagnation of convergence occurs.

4.3.3 Relation to the deflation operator

From the form of the BNN preconditioner given in Equation (4.3.5), there is reason to believe that the deflation and the BNN operators are closely connected. The exact relation between these two operators has been examined for the s.p.d. case.

Theorem 4.3.1 (Theorem 2.8 of [113])**.** *Let A and M^{-1} be s.p.d. matrices. If the spectrum of $M^{-1}P_DA$ is given by*

$$\sigma(M^{-1}P_DA) = \{0,\dots,0,\mu_{r+1},\dots,\mu_n\},$$

then

$$\sigma(P_{\mathrm{B}}A) = \{1,\dots,1,\mu_{r+1},\dots,\mu_n\}.$$

For the more general case of complex, non-Hermitian, indefinite matrices, we are only able to prove a weaker result. It is stated in the following theorem. We will comment on the relation to the similar, stronger and presumably incorrect result [42, Theorem 2.8] after the proof of the theorem.

Theorem 4.3.2 (Relation between the BNN preconditioner and deflation)**.** *Let $A,M \in \mathbb{C}^{n\times n}$. Let $Z,Y \in \mathbb{C}^{n\times r}$ have full column rank. Let P_D be as in Definition 4.2.1 and P_B be defined in Equation (4.3.2). Then the following relations between the eigenvalues of $M^{-1}P_DA$ and $P_{\mathrm{B}}A$ hold:*

1. *The r columns of the matrix Z are linearly independent eigenvectors of both $M^{-1}P_DA$ and $P_{\mathrm{B}}A$ with corresponding eigenvalues 0 and 1, respectively.*
2. *If $\lambda \neq 0$ is an eigenvalue of $M^{-1}P_DA$, then it is also an eigenvalue of $P_{\mathrm{B}}A$.*
3. *If $\lambda \neq 1$ is an eigenvalue of $P_{\mathrm{B}}A$, then it is also an eigenvalue of $M^{-1}P_DA$.*

Proof. The proof follows the main line of [113, Theorem 2.8] and [42, Theorem 2.8]. However, some more work is required in our setting. We start with proving Item 1. Equation (4.2.2a) implies that

$$P_DAZ = 0,$$

and hence also $M^{-1}P_DAZ = 0$, so the columns of the matrix Z are eigenvectors of $M^{-1}P_DA$ corresponding to zero eigenvalues. On the other hand,

$$P_{\mathrm{B}}AZ = \left(Q_DM^{-1}P_D + \Xi\right)AZ = Q_DM^{-1}P_DAZ + Z = 0 + Z = Z.$$

This implies that the columns of the matrix Z are eigenvectors corresponding to the eigenvalue 1 of the preconditioned matrix $P_{\mathrm{B}}A$.

For the proof of Item 2, let $\lambda \neq 0$ be an eigenvalue of $M^{-1}P_DA$. Suppose that v is a corresponding eigenvector and thus $M^{-1}P_DA\mathbf{v} = \lambda\mathbf{v}$. Note that such a vector $\mathbf{v}$ always exists, but that the algebraic multiplicity of λ might be higher than its geometric multiplicity. Since

$$0 \neq \lambda\mathbf{v} = M^{-1}P_DA\mathbf{v} = M^{-1}P_D^2A\mathbf{v} = M^{-1}P_DAQ_D\mathbf{v},$$

the vector $Q_D\mathbf{v}$ is nonzero. It follows that

$$\begin{aligned} P_{\mathrm{B}}A(Q_D\mathbf{v}) &= Q_DM^{-1}P_DAQ_D\mathbf{v} + \Xi AQ_D\mathbf{v} \\ &= Q_DM^{-1}P_D^2A\mathbf{v} + 0 \\ &= Q_DM^{-1}P_DA\mathbf{v} \\ &= \lambda(Q_D\mathbf{v}). \end{aligned}$$

So the vector $Q_D\mathbf{v}$ is an eigenvector of $P_{\mathrm{B}}A$ corresponding to the eigenvalue λ.

We proceed with the proof of Item 3. Let $\mathbf{v}$ be an eigenvector of $P_{\mathrm{B}}A$ such that

$$P_{\mathrm{B}}A\mathbf{v} = \lambda\mathbf{v}$$

for some eigenvalue $\lambda \neq 1$. Then $Q_D\mathbf{v} \neq 0$. Indeed, assuming $Q_D\mathbf{v} = 0$, i.e. $\mathbf{v} = \Xi A\mathbf{v}$ and using the fact that $P_DA = AQ_D$, it follows that

$$\begin{aligned} \lambda\mathbf{v} =& P_{\mathrm{B}}A\mathbf{v} \\ =& Q_DM^{-1}P_DA\mathbf{v} + \Xi A\mathbf{v} \\ =& Q_DM^{-1}AQ_D\mathbf{v} + \mathbf{v} \\ =& 0 + \mathbf{v}, \end{aligned}$$

which is obviously false for $\lambda \neq 1$ and $Q_D\mathbf{v} \neq 0$ follows. As a next step, we compute

$$\begin{aligned} \lambda\left(Q_D\mathbf{v}\right) = Q_D\left(\lambda\mathbf{v}\right) =& Q_D\left(P_{\mathrm{B}}A\mathbf{v}\right) \\ =& Q_DM^{-1}P_DA\mathbf{v} + Q_D\Xi A\mathbf{v} \\ =& Q_DM^{-1}P_DA\mathbf{v} + 0 \\ =& \left(Q_DM^{-1}A\right)\left(Q_D\mathbf{v}\right). \end{aligned}$$

Using the relation $P_DA = AQ_D$ yields $M^{-1}P_DA = \left(M^{-1}A\right)\left(Q_DM^{-1}A\right)\left(M^{-1}A\right)^{-1}$ and hence the matrix $M^{-1}P_DA$ is similar to the matrix $Q_DM^{-1}A$ that appears in the calculation. Similar matrices have the same eigenvalues. Thus λ is also an eigenvalue of $M^{-1}P_DA$. □

We now comment on the relation of our result to the sharper one of Erlangga and Nabben [42]. In particular, they state the following theorem:

Theorem (Theorem 2.8 of [42])**.** *Suppose that the spectrum of $M^{-1}P_DA$ is given by*

$$\sigma\left(M^{-1}P_DA\right) = \{0\ldots,0,\mu_{r+1},\ldots,\mu_n\},$$

then

$$\sigma(P_B A) = \{1, \ldots, 1, \mu_{r+1}, \ldots, \mu_n\}.$$

Conversely, if the spectrum of $P_B A$ is given by

$$\sigma(P_B A) = \{1, \ldots, 1, \mu_{r+1}, \ldots, \mu_n\},$$

then

$$\sigma\left(M^{-1} P_D A\right) = \{0 \ldots, 0, \mu_{r+1}, \ldots, \mu_n\}.$$

This result is stronger in the sense that it does not only claim that the eigenvalues of the two operators are the same, but additionally states that their multiplicities coincide. Even though our proof follows closely the line of theirs, we are only able to prove a weaker result. Unfortunately, we are not aware of a counterexample to the stronger result. So it is unclear whether or not it may hold in this form.

Remark 4.3.3. We have seen in the proof of Theorem 4.3.2 that the balancing preconditioner P_B shifts some of the eigenvalues of the original operator to one. This is not good for the convergence of a stationary iterative method applied to the system preconditioned with P_B, and could be avoided by adding the coarse space in a multiplicative instead of an additive way. However, we will not use the preconditioner for a stationary iterative method, but only inside of a GMRES method, where the clustering of the eigenvalues is important, and not their actual values.

4.4 A non-singular way to add a coarse space

The deflation operator introduced in Section 4.2 and the BNN preconditioner introduced in Section 4.3 are both singular. In the case of the deflation operator, the singularity is introduced on purpose and the solution of the deflated system still provides enough information to construct from it the solution of the original system. In the case of the BNN preconditioner, however, the singularity comes in unintentionally and it is not clear if there is an easy and efficient way to reconstruct the solution of the original system from the one of the system preconditioned with P_B in case $P_B A$ is singular. An additional difficulty in both cases is that the GMRES method is not guaranteed to converge without breakdown for singular systems, unless it is suitably modified [128]. Therefore, in this section, we present a way to use the coarse space such that the preconditioned system is invertible [74].

The method presented here stems from Havé, Masson, Nataf, Szydlarski, Xiang, and Zhao [74] and is tailored for non-symmetric, indefinite problems. To our knowledge, this is the only approach that allows to prove invertibility of the preconditioned operator for arbitrary matrices A, M^{-1} and $Z = Y$. The preconditioner reads

$$Q_G := Q_D + \Xi, \tag{4.4.1}$$

where the matrices Q_D and Ξ are defined in Equation (4.1.1) with $B = M^{-1}A$, i.e. using the matrix preconditioned with the one-level preconditioner. The definition in this thesis differs slightly from the original one in [74], as it allows for different matrices $Y \neq Z$. We denote this preconditioner by Q_G. We use the letter Q for the matrix to indicate that it is used as a right preconditioner as opposed to the

left preconditioners denoted by P in this thesis, cf. also Q_D and P_D in Equation (4.1.1). The subscript "G" stands for "generic" as it can be applied to any kind of square matrix. This approach has the advantage that the resulting two-level right-preconditioned matrix $BQ_G = M^{-1}AQ_G$ is non-singular if $Z = Y$ as opposed to the BNN preconditioner, where such a result cannot be guaranteed.

Theorem 4.4.1 (cf. Lemma 4.1 of [74])**.** *The coarse correction Q_G defined in Equation (4.4.1) has the left-filtering property*

$$Y^*BQ_G = Y^*.$$

Furthermore, it is invertible if $Z = Y$ or if Y^*Z is invertible. In this case, the left-filtering property can be rewritten as

$$Y^*B = Y^*Q_G^{-1}.$$

Proof. The proof is similar to the one of [74, Lemma 4.1]. However, as our definition differs slightly, allowing for two coarse space matrices Z and Y, we repeat it here and point out the place where the assumption $Z = Y$ or Y^*Z invertible is needed.

The first step is to prove that Q_G is invertible by contradiction. Assume that there is a vector $\mathbf{u}$ such that $Q_G\mathbf{u} = Q_D\mathbf{u} + \Xi\mathbf{u} = 0$. Left multiplying by Q_D and using Lemma 4.1.1 gives $Q_D\mathbf{u} = 0$ and consequently also $\Xi\mathbf{u} = 0$. $Q_D\mathbf{u} = 0$ implies $\mathbf{u} = \Xi B\mathbf{u} = ZE^{-1}Y^*B\mathbf{u}$. Setting $\mathbf{w} := E^{-1}Y^*B\mathbf{u}$ yields $\mathbf{u} = Z\mathbf{w}$. Then $\Xi\mathbf{u} = 0$ implies $\Xi\mathbf{u} = \Xi Z\mathbf{w} = ZE^{-1}Y^*Z\mathbf{w} = 0$. Left-multiplying by Y^*B yields $Y^*Z\mathbf{w} = 0$. This is the place, where the assumption $Z = Y$ or Y^*Z invertible is necessary, as it is sufficient to conclude that $Z\mathbf{w} = 0$ and hence $\mathbf{u} = 0$.

The next step is to prove the left-filtering property $Y^*BQ_G = Y^*$. Transposing the equation and using the definition of Q_G, this can be equivalently written as $(Q_D^* + \Xi^*)B^*Y = Y$. We have $Q_D^*B^*Y = 0$ as

$$Q_D^*B^*Y = B^*Y - B^*\Xi^*B^*Y = B^*Y - B^*Y\left(Z^*B^*Y\right)^{-1}Z^*B^*Y = B^*Y - B^*Y = 0.$$

It remains to prove that $\Xi^*B^*Y = Y$:

$$\Xi^*B^*Y = Y\left(Z^*B^*Y\right)^{-1}Z^*B^*Y = Y. \qquad \square$$

Moreover, the following result on the residual of the GMRES method holds:

Theorem 4.4.2 (cf. Lemma 4.2 of [74])**.** *Let $\mathbf{x}_0$ be an initial guess, and let $\mathbf{r}_0 := \mathbf{b} - B\mathbf{x}_0$ be the initial residual such that $Y^*\mathbf{r}_0 = 0$. Let $\mathcal{K}_m(BQ_G, \mathbf{r}_0)$ denote the Krylov space of dimension m, i.e.*

$$\mathcal{K}_m(BQ_G, \mathbf{r}_0) := \operatorname{span}\left\{\mathbf{r}_0, BQ_G\mathbf{r}_0, \ldots, (BQ_G)^{m-1}\mathbf{r}_0\right\}.$$

Then, for any $\mathbf{x}_m \in \{\mathbf{x}_0\} \oplus Q_G K_m(BQ_G, \mathbf{r}_0)$ it holds that

$$Y^*(\mathbf{b} - B\mathbf{x}_m) = 0,$$

i.e. in particular in any step of the Krylov method, the residual is perpendicular to the coarse space spanned by the columns of the matrix Y.

Proof. The proof is almost literally the same as [74, Lemma 4.2]. □

Note that in Theorem 4.4.2 the assumption that $Y = Z$ or that Y^*Z is invertible is not necessary. Moreover as explained in [74], it is not difficult to satisfy the assumption $Y^*\mathbf{r}_0 = 0$. Indeed, for an arbitrary initial guess $\tilde{\mathbf{x}}_0$, the vector $\mathbf{x}_0 = Q_G(\mathbf{b} + P_D\tilde{\mathbf{x}}_0)$ satisfies the assumption as

$$Y^*\mathbf{r}_0 = Y^*(\mathbf{b} - BQ_G(\mathbf{b} + P_D\tilde{\mathbf{x}}_0)) = Y^*\mathbf{b} - Y^*(\mathbf{b} + P_D\tilde{\mathbf{x}}_0) = Y^*P_D\tilde{\mathbf{x}}_0 = 0.$$

As opposed to the previous sections on the deflation operator and the BNN preconditioner, in this section we do not give a spectral analysis. The results that we were able to obtain do not provide any additional insight. The analysis is complicated by a couple of factors. On the one hand, to the best of our knowledge, there are no positive results for the spectrum of this preconditioner similar to those presented in Subsection 4.2.2, even if additional assumptions on the properties of the matrix B are made. On the other hand, the technique used in the previous sections, that is restricting to a single column coarse matrix Z, is not justified in this case, as we do not know whether an analogue of Theorem 4.2.5 holds also in this case. Moreover, the results are difficult to compare to the previous ones as they would probably use eigenvalues of $B = M^{-1}A$ and not those of A as before. Therefore, we here just present a counterexample, showing that also the Q_G preconditioner may deteriorate the spectrum of the preconditioned operator. Note that this example is analogous to the one used for the deflation operator, Example 4.2.3.

Example 4.4.3. Let

$$B := \begin{pmatrix} -4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 8 \end{pmatrix}, \qquad Z = Y := \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

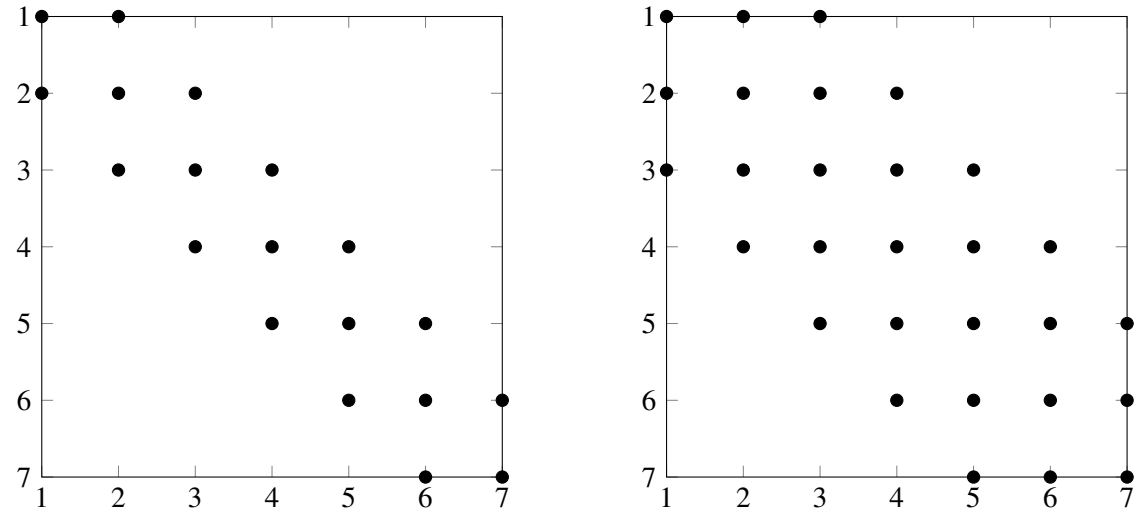
Then

$$BQ_G = \begin{pmatrix} -9 & 0 & 7 \\ 0 & 1 & 0 \\ 10 & 0 & -6 \end{pmatrix}$$

has eigenvalues $\rho(BQ_G) = \{-16, 1, 1\}$, while B has eigenvalues $\rho(B) = \{-4, 1, 8\}$. Hence the maximum eigenvalue in modulus has doubled by applying the preconditioner Q_G.

4.5 Sparsity structure of the coarse matrix E

As opposed to the deflation operator and the BNN preconditioner, in the case of the preconditioner Q_G, not the matrix $E = Z^\dagger AZ$ is used as the coarse matrix, but the matrix $E = Z^\dagger M^{-1}AZ$. This has an influence on the sparsity structure of the matrix E, which we will explain in this section. For simplicity, consider the one-dimensional Laplace problem discretized with piecewise linear FEs,



(a) $E = Z^*AZ$ **(b)** $E = Z^*M^{-1}AZ$

Figure 4.5.1. *Sparsity structure of the coarse matrix E defined with the stiffness matrix A of a* 1*D Laplace problem and with the preconditioned matrix $M^{-1}A$, where M^{-1} denotes the one-level* RAS *method as defined in Equation (3.2.2). Each dot represents a non-zero entry in the matrices.*

such that the stiffness matrix A has the form

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

Let the computational domain $\Omega = (0,7)$ be divided into seven overlapping subdomains $\Omega_i = ((i-1)-0.2, i+0.2) \cap \Omega$ and use a uniform grid of width $h = 0.1$. Assume that Z satisfies Assumption 7.1.1 and has 7 columns, one for each subdomain. The values for the block $\tilde{W}_j$ with $W_j = D_j\tilde{W}_j$ a block in Z are random values drawn from a normal distribution in the unit interval. The one-level preconditioner M^{-1} is the RAS method defined in Equation (3.2.2) equipped with Dirichlet transmission conditions, i.e. $A_i = R_iAR_i^T$. In Figure 4.5.1, we plot the sparsity structure of the matrix E for the classical $E = Z^*AZ$ and for $E = Z^*M^{-1}AZ$ that is employed in the definition of the preconditioner Q_G, cf. Equation (4.4.1). While for $E = Z^*AZ$ each coarse degree of freedom couples with the degrees of freedom on its neighboring subdomains only – i.e. for a one-dimensional problem, each row/column has at most three entries – for $E = Z^*M^{-1}AZ$, each coarse degree of freedom couples with two neighbors on each side, i.e. the coarse degree of freedom associated to Ω_i couples with those of Ω_{i-2}, Ω_{i-1}, Ω_{i+1} and Ω_{i+2}. Thus a row in this matrix has typically five entries and the matrix E has significantly more non-zero entries. The reason for the additional entries is the

n_{loc}	k	1-lev	Preconditioning	DtN, $* = \dagger$		DtN, $* = T$	
20	18.5	80	$P_B A\mathbf{x} = P_B\mathbf{b}$	15	(144)	15	(144)
			$M^{-1}AQ_G\mathbf{x} = M^{-1}\mathbf{b}$	16	(144)	38	(144)
40	29.3	116	$P_B A\mathbf{x} = P_B\mathbf{b}$	18	(224)	18	(224)
			$M^{-1}AQ_G\mathbf{x} = M^{-1}\mathbf{b}$	20	(224)	45	(224)
80	46.5	156	$P_B A\mathbf{x} = P_B\mathbf{b}$	29	(299)	29	(299)
			$M^{-1}AQ_G\mathbf{x} = M^{-1}\mathbf{b}$	31	(299)	78	(299)
160	73.8	217	$P_B A\mathbf{x} = P_B\mathbf{b}$	39	(508)	38	(508)
			$M^{-1}AQ_G\mathbf{x} = M^{-1}\mathbf{b}$	43	(508)	127	(508)

Table 4.6.1. *Comparison of different methods to use the coarse space. Number of iterations (dimension of coarse space) for Problem 2, see Section 1.4, decomposed into* 5×5 *subdomains. The DtN coarse space used here will be introduced in Chapter 5.*

presence of the inverses of local problems in the preconditioner M^{-1} that are – as opposed to the stiffness matrix A – fully populated and hence distribute values from the overlap of Ω_i and Ω_{i+1} also to $\Omega_{i+1} \cap \Omega_{i+2}$. This difference gets even more significant for higher-dimensional problems.

The additional values in the matrix E cause the parallel implementation of the assembly of E to be more involved as not only communication with neighbors, but also with neighbors of neighbors is needed, cf. Subsection 7.1.2. The solution or factorization of the matrix E is in general the more expensive the more entries it has. This is especially true when using iterative methods, as direct methods often result in fully populated matrices anyhow. Moreover, the less sparse structure of E might impose problems when storage is critical.

4.6 Comparison and conclusions

In this chapter, we have presented and theoretically examined different ways to use the coarse space. From the analysis, it is clear that the use of the BNN preconditioner in this form is problematic, as it yields a possibly singular matrix and hence to an underdetermined system of equations. However, the alternatives also suffer from difficulties: In case of the deflation operator, the GMRES solver has to deal with a singular problem; in case of the preconditioner Q_G, memory requirements for the coarse matrix E increase and more communication is necessary for its assembly and application.

In Table 4.6.1, we compare the two main approaches in this work: the BNN preconditioner and the non-singular operator Q_G. For these two preconditioners, we examine both the use of the Hermitian transpose and of the simple transpose. The latter might seem to be the natural choice due to the complex symmetry of the stiffness matrix. We do not consider deflation here, as it would require to modify the GMRES solver and has almost the same eigenvalues as the BNN preconditioner. The two ways to precondition the system yield pretty much the same results for all cases but one, the preconditioner Q_G using the simple transpose, which yields considerably worse results. The bad

performance for this case is similar to the results in [4, Section 4.1.2], where the authors observe that the conjugate transpose outperforms the transpose, even though in a different setting. These conclusions depend of course on a lot of choices, e.g. the problem that is solved, the coarse space, and the way the second level is incorporated into the method. However, as in our setting the simple transpose at least never outperforms the Hermitian transpose, in the following we work with the latter only, that is $* = \dagger$. Additional comparisons between the two preconditioners P_B and Q_G will be presented in the numerical experiments in Chapter 6.

Concluding, to our knowledge, the question of how to best add a second level to an iterative method for non-Hermitian problems in such a way that the resulting operator is easier to solve iteratively *independently of the coarse space* is still open. Complicating this issue is the fact that it is not even clear how exactly the convergence rates of the GMRES method depend on the properties of the system that is to be solved. We have examined in detail a few of the approaches present in the literature. While none of them is perfect in theory, the first numerical experiments showed that they work reasonably well in practice. The numerical experiments in this thesis will use the two preconditioners P_B and Q_G concurrently. This is partially due to the fact that a lot of the results were produced before we discovered that P_B might be singular. Apart from this, we think it provides additional insight into the differences of the two approaches. In contrast to the simple setting in Table 4.6.1, in the more complicated experiments in Chapter 6, differences between the two approaches become apparent.

Chapter 5

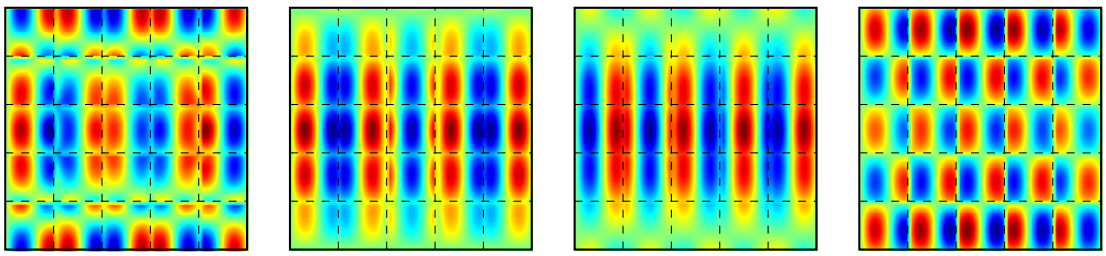
Construction of a second level: The Dirichlet-to-Neumann operator based coarse space

The coarse space $\mathcal{Z}$ influences the convergence speed of a two-level method significantly. Whereas for certain elliptic problems choices are known that turn the resulting two-level DDMs into optimal solvers with subdomain independent convergence rates [147], for the Helmholtz problem the situation is much more complicated and the correctness of the coarse space is especially important. Contrarily to the elliptic case [114], any deviation from the optimal setting might be fatal, as convergence rates can deteriorate if the wrong or too few coarse space functions are used, cf. Section 6.2 and [53]. Hence particular emphasis has to be put on the design of the coarse space.

There exist various approaches in the literature that aim at designing efficient two-level methods [4, 18, 49, 52, 92, 101], often using plane waves as basis functions for the coarse space, cf. also the literature review in Section 2.4. Although very elegant by design, the plane wave construction does not cover the case of varying coefficients and requires an a priori choice of certain parameters. Here, we thus intend to construct a coarse space with the following properties: On the one hand, for constant coefficients it behaves similarly as the one based on plane waves. On the other hand, it is also efficient for heterogeneous coefficients and can be constructed in an automatic, parameter-free fashion. The construction is based on local eigenproblems involving the DtN operator. This idea is based on a similar construction for s.p.d. systems, cf. [33, 116]. Part of the presentation in this chapter has been taken or adapted from [28].

5.1 What should the coarse space look like?

As a first step towards the design of the coarse space $\mathcal{Z}$, we investigate numerically the properties that it should have. While Fourier analysis detects the flaws of the one-level method in Equation (3.2.2) in a simplified setting, cf. Section 3.3, we here aim at getting a better understanding of how the second level should be designed by looking at the functions that it contains in the optimal case directly.



(a) $\mathrm{Re}\,(v_1)$ **(b)** $\mathrm{Re}\,(v_2)$ **(c)** $\mathrm{Re}\,(v_3)$ **(d)** $\mathrm{Re}\,(v_4)$

Figure 5.1.1. *Real part of optimal coarse space function v_i associated to eigenvalue λ_i, $|\lambda_i| \geq |\lambda_{i+1}|$ for all i, of Equation (5.1.1) on Ω. 5×5 subdomains, $n_{\mathrm{loc}} = 40$, $k = 29.3$.*

The Richardson iteration (without relaxation) for the system $A\mathbf{x} = \mathbf{f}$ preconditioned with M^{-1} reads $\mathbf{r}^{k+1} = \left(I - M^{-1}A\right)\mathbf{r}^k$, where $\mathbf{r}^k = \mathbf{f} - A\mathbf{x}^k$ is the residual. The one-level RAS preconditioner M^{-1} is consequently not efficient for eigenfunctions of $I - M^{-1}A$ associated to eigenvalues with large modulus. These are the functions that should enter the coarse space. To compute these "optimal" functions, we solve the eigenproblem

$$\text{Find } (\mathbf{v}_i, \lambda_i) \in \mathbb{C}^n \times \mathbb{C},\ 1 \leq i \leq n, \text{ such that } \left(I - M^{-1}A\right)\mathbf{v}_i = \lambda_i \mathbf{v}_i, \tag{5.1.1}$$

and then choose those functions v_i for which the modulus of the associated eigenvalue $|\lambda_i|$ is maximal. From the investigation of these eigenfunctions, the guidelines for the coarse space construction in the remainder of this section follow.

The subdomain structure should be reflected in the coarse space.

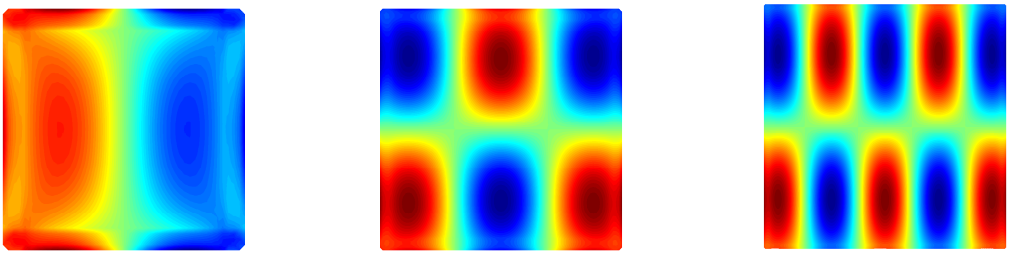
Even though the eigenfunctions $\mathbf{v}_i$ are global functions, they have a subdomain structure that is introduced by the preconditioner M^{-1}, see Figure 5.1.1. This structure clearly reflects the original domain partitioning, suggesting a subdomain based construction.

In the interior of each subdomain, the Helmholtz problem with zero right-hand side should be solved.

One can check numerically that the optimal coarse functions solve the local problems with zero right-hand side in the interior of each subdomain away from the overlap. Moreover, a coarse space correction that solves the homogeneous equation inside each subdomain does not introduce additional errors in the RAS method, which solves local problems exactly.

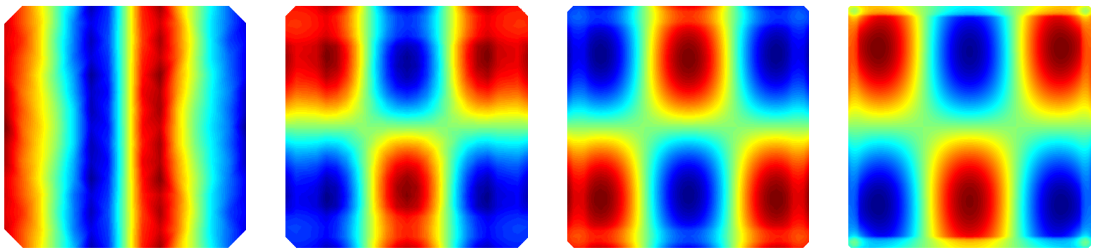
Frequencies close to the wave number k are not handled well by the one-level method.

Fourier analysis for a two subdomain model problem shows that Fourier frequencies close to the wave number k have the worst convergence rates, cf. [64] and Section 3.3. Varying the wave number for the eigenproblem in Equation (5.1.1), in Figure 5.1.2 we see that also in our setting of several overlapping subdomains in the x-direction, which is the direction orthogonal to the Dirichlet boundary conditions, the eigenfunctions clearly depend on the wave number.



(a) $n_{\text{loc}} = 40$, $k = 29.3$ **(b)** $n_{\text{loc}} = 80$, $k = 46.5$ **(c)** $n_{\text{loc}} = 160$, $k = 73.8$

Figure 5.1.2. *Real part of optimal coarse space function associated to the largest eigenvalue of Equation (5.1.1) for different wave numbers on the central subdomain in a* 5×5 *subdomain decomposition.*



(a) $n_{\text{loc}} = 10$ **(b)** $n_{\text{loc}} = 20$ **(c)** $n_{\text{loc}} = 40$ **(d)** $n_{\text{loc}} = 80$

Figure 5.1.3. *Dependence of the optimal coarse space functions on the grid width. The plots show the real part of the eigenfunction associated to the largest eigenvalue of Equation (5.1.1) on central subdomain in a* 5×5 *subdomain decomposition with* $k = 46.3$.

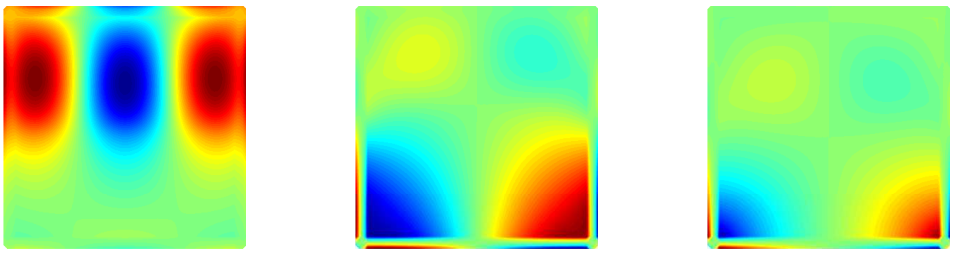
The coarse space functions are independent of the mesh width.

Figure 5.1.3 shows that the optimal coarse space functions are basically independent of the grid width h. They only change if the grid width h is chosen so large that the waves are no longer resolved. Nevertheless, as the overlap is given in terms of number of elements, its physical size changes with the grid width. This effect is clearly visible in Figure 5.1.3, where in the overlap of the subdomains, the optimal coarse space functions are blurred.

Eigenfunctions on the interface are well-suited to capture varying coefficients.

As local problems are solved exactly, all the work is done on the interface/in the overlap. Therefore, also the eigenproblems should be posed in that region. This has been proven true in a number of works, considering however definite problems, see e.g. [33, 56, 116].

Additionally to the preceding observations, in Figure 5.1.4, the behavior of the optimal coarse space functions in the presence of a heterogeneity is shown. For that purpose, the unit square $\Omega = [0,1]^2$ is divided into two parts: $[0,1] \times [0,0.5]$ and $[0,1] \times [0.5,1]$. We choose $k = 46.3$ in the upper part and



(a) $\rho = 2$ **(b)** $\rho = 5$ **(c)** $\rho = 10$

Figure 5.1.4. *Behavior of the optimal coarse space functions in the presence of a heterogeneous wave number. The plots show the real part of the eigenfunction associated to the largest eigenvalue of Equation (5.1.1) on central subdomain in a* 5×5 *subdomain decomposition,* $n_{loc} = 80$*,* $k = 46.3$*, Problem 2 with* $c = \rho$ *in* $[0,1] \times [0,0.5]$ *and* $c = 1$ *in* $[0,1] \times [0.5,1]$*.*

$k = 46.3/\rho$ in the lower part. The resulting optimal coarse space function associated to the largest eigenvalue is plotted in Figure 5.1.4. The presence of a sufficiently large contrast destroys the nicely looking functions that have been observed before, even though the general structure is still visible.

5.2 Definition of the Dirichlet-to-Neumann coarse space

The coarse space introduced in this section is based on eigenproblems involving local DtN maps. The underlying idea originates from work on elliptic problems [33, 56, 116]. This construction respects the main principles outlined in Section 5.1: Apart from including all the important modes, in the interior of each subdomain the coarse functions lie in the kernel of the Helmholtz operator and the construction is based on local problems only. The latter makes an efficient parallel implementation possible. We describe and motivate the construction of the coarse space both in the continuous and in the discrete case.

5.2.1 Continuous formulation

For ease of presentation, we introduce the coarse functions in a continuous setting before giving their discrete definitions in Subsection 5.2.2. The construction is similar to the one for elliptic problems examined in [33, 56, 116] and to the ones described in the literature review in Subsection 2.4.2 for plane wave coarse spaces. It uses almost exclusively local computations. As a first step, on each subdomain Ω_j we define local interface functions. They are the eigenfunctions of an eigenproblem on the interface $\Gamma_j := \partial\Omega_j \setminus \partial\Omega$ involving the DtN operator, which we define in the following. For that purpose, we first define the extension of a function on the boundary of a subdomain to a function defined everywhere on the subdomain.

Definition 5.2.1 (Helmholtz extension)**.** Let $D \subset \Omega$, $\Gamma_D = \partial D \setminus \partial\Omega$. Let $v_{\Gamma_D} : \Gamma_D \to \mathbb{C}$. The extension $u : D \to \mathbb{C}$ of v with respect to the Helmholtz operator is defined by

$$\begin{aligned} -\Delta u - k^2 u &= 0 && \text{in } D, \\ \mathcal{C}(u) &= 0 && \text{on } \partial\Omega \cap \partial D, \\ u &= v_{\Gamma_D} && \text{on } \Gamma_D. \end{aligned}$$

The DtN operator, which relates Dirichlet values of a solution of the homogeneous Helmholtz equation to Neumann values, is then defined as follows:

Definition 5.2.2 (DtN operator)**.** Let $D \subset \Omega$, let $\Gamma_D = \partial D \setminus \partial\Omega$. Let $v_{\Gamma_D} : \Gamma_D \to \mathbb{C}$. Then

$$\mathrm{DtN_D}\left(v_{\Gamma_D}\right) = \left.\frac{\partial u}{\partial n}\right|_{\Gamma_D},$$

where $u : D \to \mathbb{C}$ is the extension of v_{Γ_D} in the sense of Definition 5.2.1.

With these definitions, the eigenproblem that is used for the coarse space construction reads on the interface Γ_j: Find $(u_{\Gamma_j}, \lambda) \in \mathcal{V}(\Gamma_j) \times \mathbb{C}$ such that

$$\mathrm{DtN}_{\Omega_j}\left(u_{\Gamma_j}\right) = \lambda u_{\Gamma_j}. \tag{5.2.2}$$

Even after discretization, this problem still gives n_{Γ_j} functions $u_{\Gamma_j} : \Gamma_j \to \mathbb{C}$, where n_{Γ_j} is the number of degrees of freedom on the interface Γ_j. Even though the number of interface degrees of freedom n_{Γ_j} might be significantly smaller than the number n_j of total degrees of freedom associated to the subdomain Ω_j, using all the eigenfunctions in the coarse space is still too costly. Additionally, some modes might be useless or even harmful for the convergence. It is thus important to choose the right modes. We will motivate our choice in Section 5.4 with numerical experiments, and here just state the strategy that we use. We choose $m_j \in \mathbb{N}$ eigenfunctions for each subdomain Ω_j according to the following criterion. It provides a way to automatically construct the coarse space $\mathcal{Z}$ without the need to tune its dimension, a crucial parameter for the convergence of the two-level method.

Criterion 5.2.3 (Choice of DtN eigenfunctions)**.** On each subdomain Ω_j, we choose all eigenfunctions v of the DtN eigenproblem in Equation (5.2.2), for which the associated eigenvalue λ satisfies

$$\mathrm{Re}(\lambda) < k_i.$$

Here $k_i := \max_{\mathbf{x} \in \Omega_i} k(\mathbf{x})$ is the maximum wave number on Ω_i. If no eigenvalue satisfies this condition, the eigenvalue with smallest real part is chosen.

As a next step, we extend the interface functions that arise from the DtN eigenproblem to the interior of the subdomain Ω_j. For that purpose, the extension defined in Definition 5.2.1 is used. This is motivated by the observation that the coarse space functions should solve the homogeneous Helmholtz equation in the interior of the subdomains, see Section 5.1. Nevertheless, in Section 5.3,

we give numerical evidence that this is indeed the right extension operator. Please note that the computation of the extension might be a singular problem, cf. Remark 5.2.5.

In practice, instead of looking for the pair $\left(u_{\Gamma_j}, \lambda\right)$ solving the eigenvalue problem in Equation (5.2.2) and then computing the extension u_{Ω_j} of u_{Γ_j} from the interface Γ_j to the interior of the subdomain Ω_j with Definition 5.2.1, it is possible to directly compute the pair $(u_{\Omega_j}, \lambda) \in (\mathcal{V}(\Omega_j), \mathbb{C})$. It satisfies

$$-\Delta u_{\Omega_j} - k^2 u_{\Omega_j} = 0 \qquad \text{in } \Omega_j, \tag{5.2.3a}$$

$$u_{\Omega_j} = 0 \qquad \text{on } \Gamma_D, \tag{5.2.3b}$$

$$\frac{\partial}{\partial n} u_{\Omega_j} = \lambda u_{\Omega_j} \qquad \text{on } \Gamma_j. \tag{5.2.3c}$$

The variational formulation of Equation (5.2.3) is: Find $\left(u_{\Omega_j}, \lambda\right) \in (\mathcal{V}(\Omega_j), \mathbb{C})$ such that

$$\int_{\Omega_j} \nabla u_{\Omega_j} \nabla v \, \mathrm{d}x - \int_{\Omega_j} k^2 u_{\Omega_j} v \, \mathrm{d}x = \lambda \int_{\Gamma_j} u_{\Omega_j} v \, \mathrm{d}s \quad \text{for all } v \in H^1(\Omega_j). \tag{5.2.4}$$

For each subdomain Ω_j, the previous construction yields m_j locally defined functions. To compute the global coarse space functions, each of them is first multiplied by the partition of unity function for this subdomain, cf. Subsection 3.2.1, and then extended by zero to the whole computational domain Ω. The additional multiplication by the partition of unity is motivated by the fact that in the overlap contributions from the single subdomains should be weighted just as in the RAS algorithm. This strategy is also used in a number of related works, see e.g. [33, 93]. Concluding, we have constructed $\sum_{j=1}^{n} m_j$ global functions that have non-zero values only locally. Those are the functions that span the coarse space $\mathcal{Z}$.

5.2.2 Discrete formulation

In Subsection 5.2.1, we have defined the DtN coarse space in the continuous setting. While this formulation aids understanding its structure and properties, we are ultimately interested in the discrete problem and hence also in the discrete formulation of the DtN eigenproblem. For that reason, we here explain how to construct the rectangular matrix $Z \in \mathbb{C}^{n \times \sum_{j=1}^{N} m_j}$, whose columns span the discrete coarse space.

The matrix Z is a block matrix with blocks W_j, $1 \leq j \leq N$, of the form

$$Z = \begin{pmatrix} \square & & \\ & \square & \\ & & \square \end{pmatrix}. \tag{5.2.5}$$

The block $W_j \in \mathbb{C}^{n_j \times m_j}$ is associated to subdomain Ω_j. Its columns contain the discretizations of the local, continuous functions defined in Subsection 5.2.1. The columns of the matrix Z are set to the extension of these local functions to the global domain by zero, i.e. the columns $1+\sum_{k=1}^{j-1} m_k, \ldots, \sum_{k=1}^{j} m_k$ of Z are set to $R_j^T W_j$ for $1 \leq j \leq N$. Due to the overlap in the domain decomposition, also the rows of the blocks overlap. The definition of the blocks W_j is given in Algorithm 5.2.1 and is equivalent to the construction of the local functions in Subsection 5.2.1.

Algorithm 5.2.1 Construction of the block W_j of the DtN coarse matrix

1: Solve the discrete DtN eigenproblem in Equation (5.2.6) on subdomain Ω_j.
2: Choose m_j eigenvectors $\mathbf{g}_j^l \in \mathbb{C}^{n_{\Gamma_j}}$, $1 \leq l \leq m_j$ by the discrete analogue of Criterion 5.2.3.
3: **for** $l \leftarrow 1$ to m_j **do**
4: Compute the extension $\mathbf{u}_j^l \in \mathbb{C}^{n_j}$ of $\mathbf{g}_j^l$ according to Definition 5.2.4.
5: **end for**
6: Define the matrix $W_j \in \mathbb{C}^{n_j \times m_j}$ as $W_j := \left(D_j \mathbf{u}_j^1, \quad \ldots, \quad D_j \mathbf{u}_j^{m_j}\right)$, where D_j is the matrix corresponding to the partition of unity function on Ω_j defined in Subsection 3.2.2.

We now define the components of Algorithm 5.2.1. For Line 1 of Algorithm 5.2.1, we need the discrete formulation of the DtN eigenproblem in Equation (5.2.2): Let I and Γ_i be the sets of indices corresponding to the interior and boundary degrees of freedom, respectively. Let n_I and n_{Γ_i} be their cardinalities. We define $a_i : H^1(\Omega_i) \times H^1(\Omega_i) \to \mathbb{R}$,

$$a_i(v,w) = \int_{\Omega_i} \left(\nabla v \cdot \nabla w - k^2 v w\right) \mathrm{d}x.$$

Using the FE basis $\{\phi_k\}$ for $\mathcal{V}(\Omega)$, let $A^{(i)}$ be the coefficient matrix of a Neumann boundary value problem on Ω_i, $A_{kl}^{(i)} = a_i(\phi_k, \phi_l)$, with boundary conditions defined by $\mathcal{C}$ on $\partial\Omega_i \cap \partial\Omega$. With the usual block notation, the subscripts I and Γ_i for the matrices A and $A^{(i)}$ denote the entries of these matrices associated to the respective degrees of freedom. Let

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

be the mass matrix on the interface of subdomain Ω_i. The discrete formulation of the eigenproblem in Equation (5.2.2) is, cf. [33]: For $1 \leq i \leq N$ find $(\mathbf{u}, \lambda) \in \mathbb{C}^{n_{\Gamma_i}} \times \mathbb{C}$, s.t.

$$\left(A_{\Gamma_i\Gamma_i}^{(i)} - A_{\Gamma_i I} A_{II}^{-1} A_{I\Gamma_i}\right) \mathbf{u} = \lambda M_{\Gamma_i} \mathbf{u}. \tag{5.2.6}$$

Now we define the extension operator required in Line 4 of Algorithm 5.2.1:

Definition 5.2.4 (Discrete Helmholtz extension)**.** The extension of a vector $\mathbf{g} \in \mathbb{C}^{n_{\Gamma_i}}$ defined on the interface Γ_i to all degrees of freedom on subdomain Ω_i is the vector $\mathbf{u} \in \mathbb{C}^{n_i}$ given by $\mathbf{u} = \left(-A_{II}^{-1} A_{I\Gamma_i} \mathbf{g}, \quad \mathbf{g}\right)^T$.

Remark 5.2.5 (Singular extension). The extensions in Definition 5.2.1 and Definition 5.2.4 might give a (numerically) singular problem for subdomains that do not touch the Robin boundary. If the matrix A is singular, the solution of the system $A\mathbf{x} = \mathbf{b}$ either does not exist or it is not unique. In the first case, if a solution of the singular system does not exist, a least squares solution, e.g. computed via a *QR* decomposition, can be employed. This is feasible as the subdomain problems are small and are solved directly. In the latter case, if the solution to the singular system is not unique, one has to decide for one solution, for example by using a pseudoinverse. We do not investigate this question and our code does not check for singularity of these operators. However, we do not encounter problems in the numerical experiments, probably since, by chance, the extension matrix is never singular for our parameter settings.

We hence have defined all the necessary components to build the discrete coarse space, which is spanned by the columns of the matrix Z. The construction is equivalent to the one for the continuous setting described in Subsection 5.2.1. In the following sections, we motivate the various choices, in particular the extension operator in Definition 5.2.1 and Definition 5.2.4, and the selection strategy in Criterion 5.2.3.

5.3 How to choose the extension operator

In Definition 5.2.1 and Definition 5.2.4 in the previous sections, the extension from the boundary of a subdomain to the subdomain's interior has been defined. This definition was partially motivated in Section 5.1, where we observed that the optimal coarse functions solve the Helmholtz problem in the interior of each subdomain. However, as we do not directly work with the optimal coarse functions due to efficiency issues, we want to backup this conclusion with some further numerical experiments. For this, we divide each subdomain Ω_j into four parts: the interior part, i.e. $\{x \in \Omega_j$ such that $\pi_j(x) = 1$ and $x \notin \Gamma_D\}$, where π_j is the partition of unity function associated to Ω_j defined in Equation (3.2.1), the interface $\Gamma_j = \partial\Omega_j \setminus \partial\Omega$, the global Dirichlet boundary part $\partial\Omega_j \cap \Gamma_D$, and the remaining part that belongs to the overlap. From the results in Table 5.3.1, it is clear that the Helmholtz extension is the best choice among the extension operators considered.

Concluding, not only the values of the coarse functions in the overlap or on the interface are important, but that it is also of vital importance to choose the right extensions to the interior of the subdomains in order to get good convergence rates. From the experiments it is clear that the question, which we examine in this section, is of utter importance. We will further examine this issue when we introduce the plane wave coarse space, as here a natural extension arises, namely the pointwise evaluation of the plane waves, cf. Subsection 6.3.3.

5.4 How to choose the Dirichlet-to-Neumann coarse space functions

For indefinite systems as those arising from the FE discretization of the Helmholtz equation, in contrast to the s.p.d. case, increasing the dimension of the coarse space might lead to a deterioration of the convergence rates, cf. Section 6.2 and [53]. An incorrect coarse space might hence be fatal,

interface	overlap	interior	partition of unity	# iterations
DtN eigenfunctions	Helmholtz extension		yes	16
DtN eigenfunctions	0		no	72
DtN eigenfunctions	random		no	107
DtN eigenfunctions	Helmholtz extension	0	yes	56
DtN eigenfunctions	Helmholtz extension	random	yes	109
random	Helmholtz extension		yes	101

Table 5.3.1. *Comparison of different extension operators. On the Dirichlet boundary, the values are set to* 0*. For the remaining regions, the strategy is stated in the table. The Helmholtz extension is defined in Definition 5.2.4. The column "partition of unity" denotes whether the resulting vector on a subdomain Ω_j is multiplied by the partition of unity matrix D_j as in Line 6 of Algorithm 5.2.1. Problem 2,* 5×5 *subdomains,* $n_{\text{loc}} = 40$*,* $k = 30$*.*

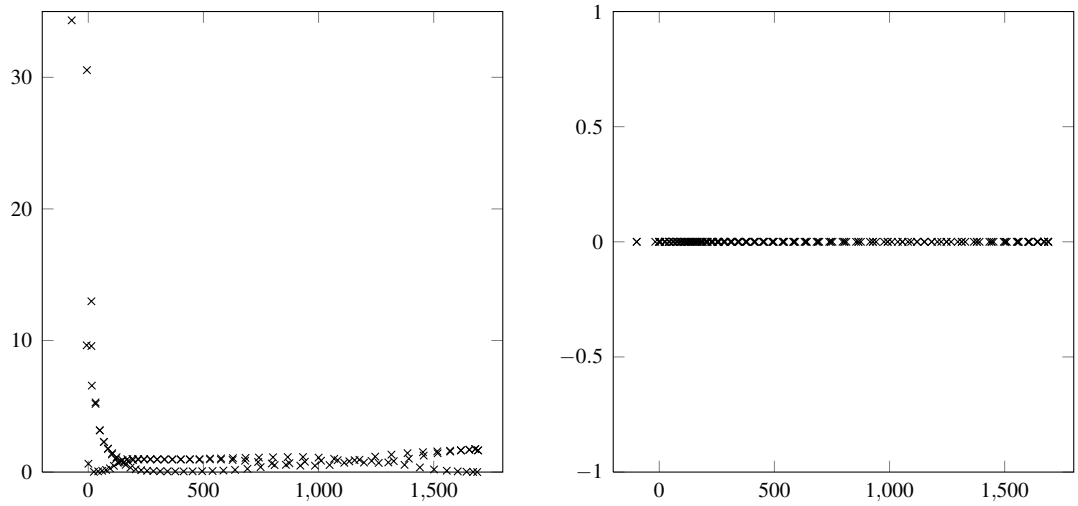
but it is difficult to decide *a priori* which and how many modes are needed. Thus an appropriate strategy for its construction is extremely important. In this section, we justify and investigate the coarse space based on the DtN operator introduced in Section 5.2 in detail and test Criterion 5.2.3.

For the tests, we take the setup described in Section 6.1 and Problem 2 from Section 1.4. Let the domain Ω be decomposed into 5×5 subdomains, $n_{\text{loc}} = 40$, $k = 30$. All experiments in this section are based on this example for ease of presentation. The conclusions have however also been verified in modified setups. Moreover, they are supported by the numerical experiments in Chapter 6.

We first examine *which* eigenfunctions of the DtN eigenproblem in Equation (5.2.6) are important. The $n = 176$ eigenvalues on the central subdomain satisfy $\operatorname{Re}(\lambda_i) \leq \operatorname{Re}(\lambda_j)$ if $i \leq j$, $\operatorname{Re}(\lambda_5) < 0 < \operatorname{Re}(\lambda_6)$, and $\operatorname{Re}(\lambda_{12}) < k < \operatorname{Re}(\lambda_{13})$. They are plotted in the complex plane both for the central subdomain and for a subdomain bordering the Robin boundary Γ_R in Figure 5.4.1. We show a few of the extensions to the interior of the subdomains of the associated eigenvectors v_i in Figure 5.4.2. From this, the first guess is that eigenfunctions associated to smaller eigenvalues are more useful. Comparing coarse spaces with 12 modes per subdomain based on the eigenvalues with the smallest real part, the smallest eigenvalues in modulus, the eigenvalues closest to the wave number $k = 30$, and the eigenvalues with the largest modulus, yields the results in the central columns of Table 5.4.1. The first alternative, which is in accordance with Criterion 5.2.3, gives the best results.

To ensure that our findings are not distorted by choosing a too small coarse space, in the last column of Table 5.4.1 the results for the same experiment with a twice as many modes per subdomain are reported. The number of iterations with the best choice is hardly influenced. Also in this setting, Criterion 5.2.3 performs best.

In the next step, we examine *how many* modes should be chosen. The more important part of this problem is that we should not choose too few modes as the convergence rates cannot be expected to depend monotonically on the coarse space size due to the indefiniteness of the system. Nevertheless,



(a) On the subdomain in the 3rd row, 1st column, bordering the Robin boundary Γ_R

(b) On the central subdomain

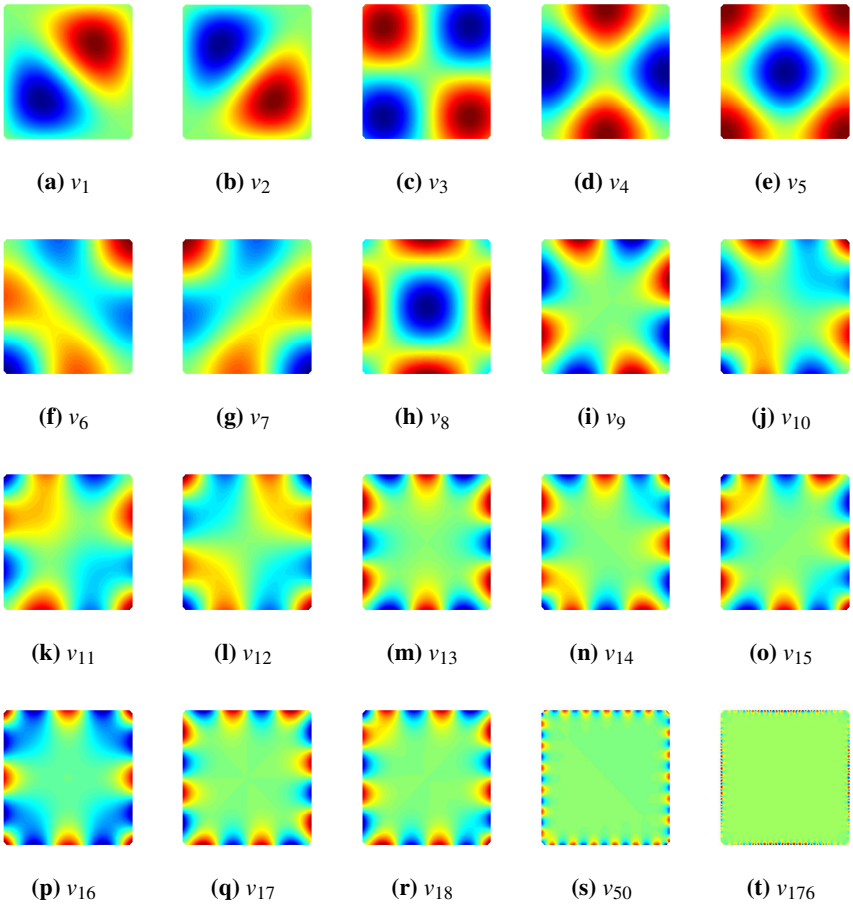
Figure 5.4.1. *Eigenvalues of the DtN eigenvalue problem in Equation (5.2.6) in the complex plane.* 5×5 *subdomains,* $n_{\text{loc}} = 40$*,* $k = 30$*.*

choosing too many modes increases the computational costs and is not desirable either. The number of modes is controlled by Criterion 5.2.3. In Figure 5.4.3 we show the resulting number of modes per subdomain. They are influenced both by the boundary conditions and the heterogeneity.

In Figure 5.4.4, we examine whether the number of modes resulting from Criterion 5.2.3 is a good estimate. It yields convergence rates that are almost independent of grid width/wave number at the cost of an increasing coarse space size. We investigate whether we can do significantly better by adding the next two eigenvectors on each subdomain to the coarse space. Here the eigenvalues are ordered by their real parts. Figure 5.4.4 shows that this is not the case; it only yields a slight improvement. Moreover, we test whether we could achieve the same behavior with a significantly

Choice	# iterations with P_B		# iterations with Q_G	
	$m_i = 12$	$m_i = 24$	$m_i = 12$	$m_i = 24$
no coarse space	115	115	115	115
$\text{Re}(\lambda)$ minimal	16	10	17	11
$\lvert\lambda\rvert$ minimal	37	26	27	17
$\lvert\lambda - k\rvert$ minimal	77	35	49	21
$\lvert\lambda\rvert$ maximal	115	115	155	145

Table 5.4.1. *Iteration numbers for different choices of DtN eigenfunctions.*



(a) v_1 **(b)** v_2 **(c)** v_3 **(d)** v_4 **(e)** v_5

(f) v_6 **(g)** v_7 **(h)** v_8 **(i)** v_9 **(j)** v_{10}

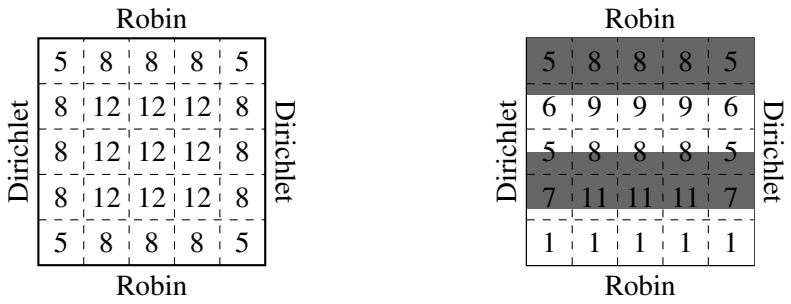
(k) v_{11} **(l)** v_{12} **(m)** v_{13} **(n)** v_{14} **(o)** v_{15}

(p) v_{16} **(q)** v_{17} **(r)** v_{18} **(s)** v_{50} **(t)** v_{176}

Figure 5.4.2. *DtN eigenfunctions for* 5×5 *subdomains,* $n_{\text{loc}} = 40$*,* $k = 30$*.*

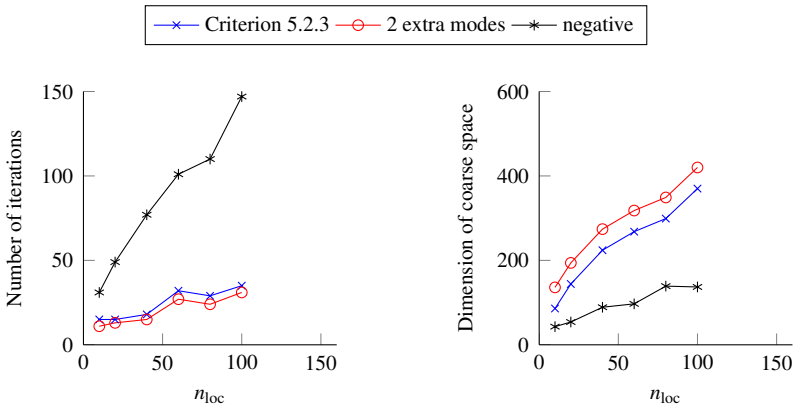
smaller coarse space. Therefore, we choose another "natural" bound, taking only eigenvectors that are associated to eigenvalues with real part smaller than 0, denoted by "negative" in the legend. For this choice, the number of iterations significantly increases with the number of grid points per subdomain n_{loc} and hence with the wave number k. While the results are qualitatively the same for both P_B and Q_G, we note that the increase in iteration numbers for choosing only the negative modes seems to be less for Q_G than for P_B.

Additionally, we test whether Criterion 5.2.3 is independent of the diameters of the subdomains. For that purpose, we take square domains $\Omega = [0,L]^2$, $L = 1,5,10$, of different sizes and choose k such that $kL = 30$ is constant, i.e. such that the number of wavelengths in both coordinate directions in the squares of different sizes is constant, while the wave number varies. For all three cases, the DtN shows exactly the same behavior, that is reported in Table 5.4.2: The number of modes that are chosen and the number of iterations do not change with L. Criterion 5.2.3 consequently provides a useful strategy.

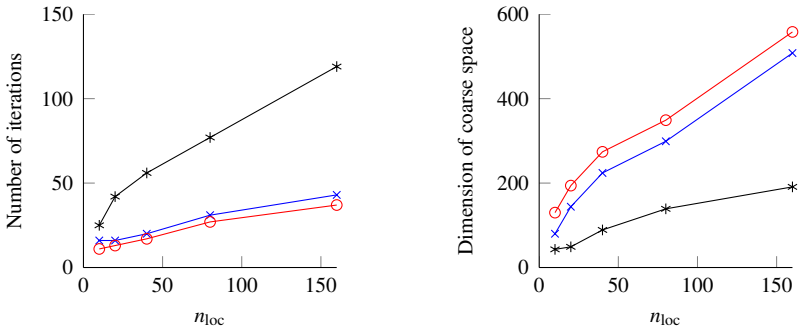


(a) Homogeneous case **(b)** Heterogeneous case

Figure 5.4.3. *Number of DtN modes per subdomain. Problem 2,* 5×5 *subdomains,* $n_{\text{loc}} = 40$, $k = 30$.



(a) For P_B



(b) For Q_G

Figure 5.4.4. *Comparison of different criteria of how many DtN modes to choose.* $k^3h^2 \approx \frac{2\pi}{10}$, *Problem 2,* 5×5 *subdomains.*

L	k	kL	# iterations with P_B	# iterations with Q_G	coarse space dimension
1	30	30	18	20	224
5	6	30	18	20	224
10	3	30	17	19	224

Table 5.4.2. *Dependence of the number of iterations with DtN coarse space on the size L of the domain $\Omega = [0,L]^2$. Ω is decomposed into 5×5 subdomains with $n_{\text{loc}} = 40$ grid points on each subdomain in each direction. The wave number k is chosen such that kL is constant.*

5.5 Summary and conclusions

We now defined the different components of the algorithm that we employ in this work for the solution of the discretized Helmholtz equation. In particular, we defined the DDM, the coarse space, the two-level preconditioner based on them, and the preconditioned GMRES method that is used as the iterative solver. In order to ease the understanding of how these different components work together, in Algorithm 5.5.1, we give the complete algorithm.

Concluding, we designed a coarse space based on the DtN operator and motivated its definition with some numerical experiments. The coarse space construction is based on the solution of local eigenvalue problems and can hence be parallelized easily. While we collected some first evidence for the validity of the approach, more extensive numerical experiments are necessary to fully test the new methods. This will be done in Chapter 6 for two-dimensional examples using a serial implementation of the method and in Chapter 7 for the three-dimensional case.

Algorithm 5.5.1 Complete algorithm: RAS with DtN coarse space

Input: Computational domain Ω and the corresponding mesh, number of subdomains N, number of elements in the overlap n_{ov}, error tolerance ε, stiffness matrix A, right-hand side vector $\mathbf{b}$

Output: Approximate solution $\tilde{\mathbf{u}}$ of the system $A\mathbf{u} = \mathbf{b}$

▷ Preprocessing, for the parallel version see Algorithm 7.1.1

1: **for** $j \leftarrow 1$ to N **do**
2: Compute the restriction operator R_j.
3: Compute the partition of unity matrix D_j.
4: Compute the local stiffness matrix A_j.
5: **end for**

▷ Construction of the coarse space

6: **for** $j \leftarrow 1$ to N **do**
7: Compute the block W_j of the coarse matrix Z as defined in Algorithm 5.2.1.
8: **end for**
9: **if** the preconditioner P_B is used **then**
10: Compute the coarse matrix $E = Z^* AZ$ from the local stiffness matrices A_j and the matrices W_j, $1 \leq j \leq N$. For a parallel implementation, use only local computations and communication without assembling the matrices A and Z, cf. Subsection 7.1.2.
11: **else if** the preconditioner Q_G is used **then**
12: Compute the coarse matrix $E = Z^* M^{-1} AZ$ from the local stiffness matrices A_j and the matrices W_j, $1 \leq j \leq N$. For a parallel implementation, use only local computations and communication without assembling the matrices A and Z, cf. Subsection 7.1.2.
13: **end if**

▷ The iterative solution of the system

14: Compute iteratively an approximate solution $\tilde{u} \approx u$ of the preconditioned system

$$P_B A u = P_B b \qquad \text{or} \qquad M^{-1} A Q_G u = M^{-1} b$$

with the GMRES method in Algorithm 3.1.1 or the restarted GMRES(m) method in Algorithm 3.1.2. For the definitions of the preconditioners P_B and Q_G see Section 4.3 and Section 4.4.

Chapter 6

Numerical results for two-dimensional problems

In this chapter, we examine numerically the two-level methods using the coarse space based on the DtN operator, which have been introduced in Chapter 4 and Chapter 5. In particular, we examine how good the coarse space based on the DtN eigenvalue problems is compared to the standard approach based on plane waves [52]. For this purpose, in this chapter, we look at two-dimensional homogeneous and heterogeneous problems. Many of the results and also large parts of the text in this chapter have been published previously in [28].

6.1 Framework and implementational details

In this section, we discuss the framework and give implementational details for the numerical experiments in this chapter. The basic method used for the numerical experiments is given in Algorithm 5.5.1. We test both two-level methods that were introduced in Chapter 4, the left preconditioner P_B and the non-singular, right preconditioner Q_G, due to two reasons. On the one hand, we were not aware of the singularity of the P_B preconditioner when first doing these tests, cf. Section 4.3. On the other hand, we refrain from only including the results for the non-singular preconditioner Q_G as the coarse operator in this case is more expensive and considerably more difficult to implement in a parallel code. Moreover, as shown in this chapter, the results for P_B and Q_G in most cases do not differ much. Even though P_B lacks the theoretical foundations, for the examples that we consider this does not seem to have any significant practical implications.

As a solver we use a GMRES method without restart, cf. Algorithm 3.1.1. This is feasible since the two-dimensional examples in this chapter are rather small and hence restarting is not needed. The termination criterion for the GMRES method is based on the error $\|u_h - u_i\|_\infty$, where u_h is the exact FE solution and u_i is the iterative solution in step i. The system is considered to be solved in step i if $\frac{\|u_h - u_i\|_\infty}{\|u_h\|_\infty} < 10^{-7}$. The size of the problems considered here allows us to use a direct solver to compute the exact discrete solution u_h and use the error as a stopping criterion. This is different from the stopping criterion for larger experiments in Chapter 7, where the residual is used instead. We

write "> 400" for the iteration count in the tables, when the maximum number of iterations, here 400, is reached before the desired tolerance. The initial iterate has pseudo-random values drawn from the standard uniform distribution on the interval $(0,1)$.

Due to the wave character of the solution, in all numerical experiments the grid has to be sufficiently fine in order for the discrete solution to be a good approximation of the continuous one. Additionally to the requirement of having a minimum number of points per wavelength, in order to avoid the pollution effect [8], not only kh, but also k^3h^2 needs to be bounded from above, cf. the discussion in Subsection 1.3.3. We choose an overlap L of two mesh elements as defined in Definition 3.2.1. If nothing else is specified a decomposition into $N = n_S \times n_S$ squares is chosen. Let n_i be the number of grid points on one side of a square subdomain. If $n_i = n_j$ for all $1 \leq i, j \leq N$, we define $n_{\text{loc}} := n_i$. In this case, the mesh is always of the type shown in Figure 1.4.2. For decompositions using Metis [86], an arbitrary triangulation is chosen. We denote the number of grid points on one side of a square domain Ω by n_{glob}.

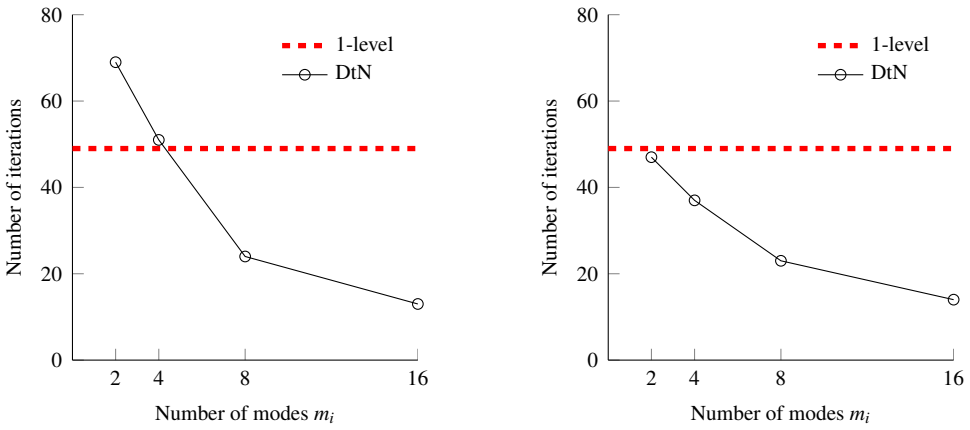
The FE part is implemented in FreeFem++ version 3.21 [75], the algebraic part in MATLAB version 7.10.0.499 (R2010a). In light of the discussion in Remark 6.3.4 later in this chapter about the influence of the *QR* factorization on the plane wave coarse space filtering, we note that for the experiments in this chapter, we use MATLAB's built-in *QR* factorization.

6.2 Influence of the Dirichlet-to-Neumann coarse space on the spectrum

As a first step to validate the quality of the proposed method, in this section we compare the convergence rates and the spectrum of the two-level methods from Algorithm 5.5.1 with DtN coarse space to those of the corresponding one-level RAS preconditioner in Equation (3.2.2). These experiments can also be seen as a demonstration of the challenges that arise when designing a coarse space for an indefinite problem: Neither convergence rates nor the spectrum necessarily improve when adding the second level, even if it is carefully designed, cf. [53].

In Figure 6.2.1, we compare the convergence rates of the one- and two-level methods for Problem 3 from Section 1.4, using the setup described in Section 6.1. For the left preconditioner P_B, using too few coarse space modes gives worse convergence rates than those of the one-level method. This problem disappears when enough modes are employed. For the right preconditioner Q_G on the other hand, for this example this problem does not occur. Even adding just a few coarse modes slightly improves the convergence rates of the method.

To understand why the two preconditioners behave differently in this respect, as a next step we look at the spectrum of the two operators for Problem 2 from Section 1.4. For the correct interpretation of the spectral information it is important to understand how the convergence rates of the GMRES method depend on the eigenvalues of the preconditioned operators. To our knowledge the relationship between the spectrum and convergence behavior is not as easy as for example for the CG method for s.p.d. matrices, cf. the discussion in Section 3.1. However, the clustering of the eigenvalues is important in this case, see [131] or Theorem 3.1.2. As the eigenvalues for the



(a) P_B **(b)** Q_G

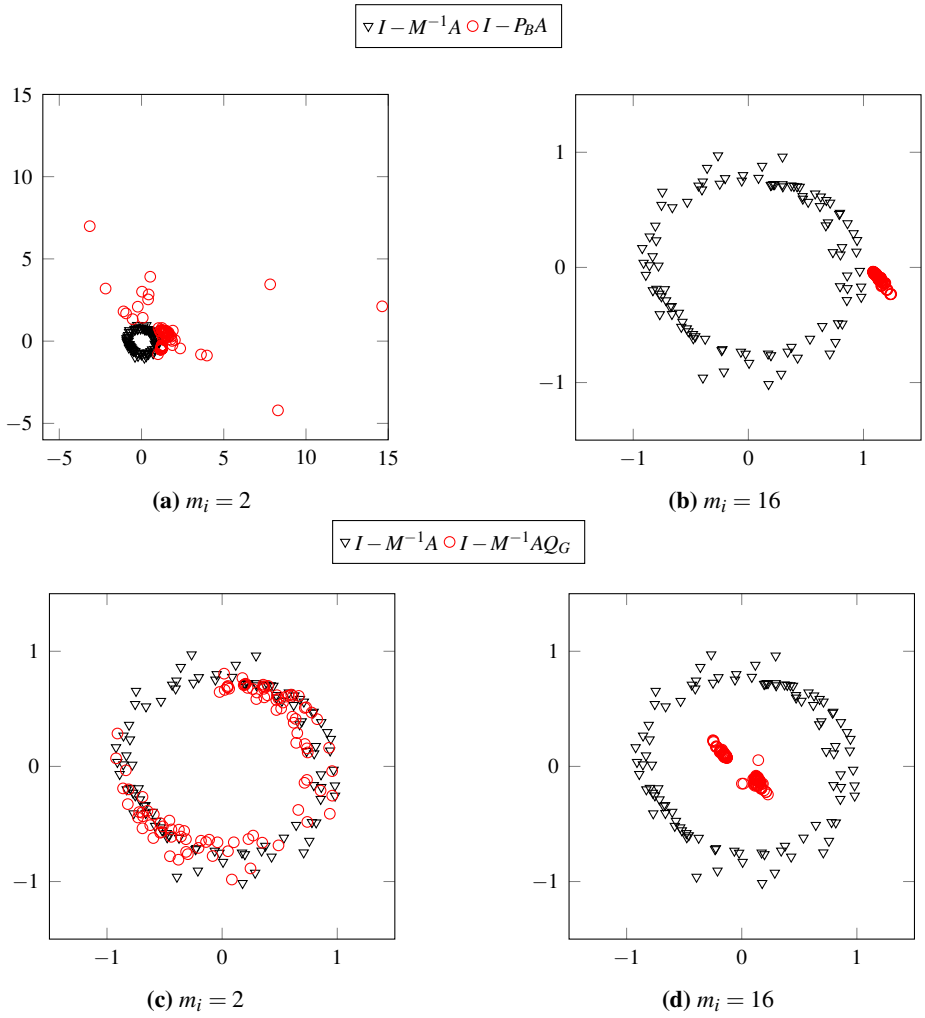
Figure 6.2.1. *Number of iterations in dependence of the number m_i of coarse modes per subdomain. Problem 3, 5×5 subdomains, $n_{\text{loc}} = 40$, $k = 29.3$.*

one-level RAS operator $I - M^{-1}A$ all lie within a circle centered at the origin of the complex plane, see Figure 6.2.2, we compare the largest eigenvalues without and with coarse space. If they decrease for the two-level method, the clustering is likely to be better. On the other hand, if they increase significantly, this is probably an indication for deterioration.

In Figure 6.2.2, we thus compare the eigenvalue distribution of the one- and the two-level methods; the largest eigenvalues of $I - M^{-1}A$, $I - P_B A$ and $I - M^{-1}AQ_G$ are plotted in the complex plane. For the BNN preconditioner P_B, if the coarse space dimension is small, there is no clear structure in the eigenvalue distribution. For the one-level matrix for $I - M^{-1}A$, they lie within a circle of radius less than one with center $(0,0)$. Adding only a few global modes has a chaotic effect, scattering the eigenvalues in the complex plane. This changes when adding more modes; the eigenvalues are then clustered near the point $(1,0)$. Contrary to that, for the preconditioner Q_G, using only a few coarse modes does not seem to have a detrimental effect. Even though being slightly shifted, the eigenvalues remain within the circle. When adding more modes, they are shifted even further into the circle, clustering close to the origin. The eigenvalue distribution in Figure 6.2.2 explains why choosing an incomplete coarse space when using the preconditioner P_B has a detrimental effect while it hardly affects the convergence rates in case Q_G is employed. A careful design of the second level is important in any case to achieve good convergence rates, but especially for the former case also its size is a crucial parameter.

6.3 Plane wave coarse space

In Section 6.2, we examined the effect that the different ways to add the DtN coarse space to the one-level RAS preconditioner have on the convergence rates. While this is a viable first step in order



(a) $m_i = 2$ **(b)** $m_i = 16$

(c) $m_i = 2$ **(d)** $m_i = 16$

Figure 6.2.2. 100 *largest eigenvalues of* $I - M^{-1}A$ *and* $I - P_BA$ *(upper row) or* $I - M^{-1}AQ_G$ *(lower row), respectively, in the complex plane. Problem 2,* 5×5 *subdomains,* $n_{\text{loc}} = 40$*,* $k = 30$*.*

to ensure that there is actually a gain in using a coarse space, the comparison is unfair, as two-level methods have the advantage of an additional global problem. For that reason, in this section, we introduce and shortly examine another coarse space, which we will use as a benchmark. It is based on plane waves, since this choice is pretty standard for iterative methods for Helmholtz problems, as already described in the literature review in Subsection 2.4.2. We are not aware of any other idea for a second level that has reached similar popularity for this class of problems.

6.3.1 Definition of the plane wave coarse space

As a first step, we define the plane wave (PW) coarse space, which we use in the following. Since there is no consensus in the literature on how to incorporate it into a DDM, we choose one possibly

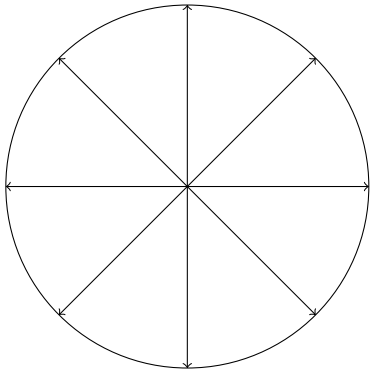


Figure 6.3.1. *Uniform discretization of the unit circle using eight directions.*

non-optimal way. We will discuss alternatives in Subsection 6.3.3. Plane waves have already been introduced in Subsection 2.4.2. A plane wave $p^\theta : D \subset \mathbb{R}^d \to \mathbb{C}$ in direction θ is a function of the form

$$p^\theta(\mathbf{x}) = \mathrm{e}^{\imath\bar{k}\theta\cdot\mathbf{x}}, \quad \mathbf{x} \in D,\ \theta \in \mathbb{R}^d,\ \|\theta\|_2 = 1. \tag{6.3.1}$$

Here, $\bar{k}$ is a constant defined as the mean value of the (possibly heterogeneous) wave number k on D. Proceeding as in the case of the DtN operator, we modify Algorithm 5.2.1. We ignore lines 1 and 2 and specify the functions $\mathbf{u}_i^l$ in Line 4 directly. Note that in contrast to DtN[1], m_i is chosen *a priori*. If nothing else is specified, $m_i = 25$ modes per subdomain are used.

Definition 6.3.1 (Plane wave coarse space)**.** Let $1 \le i \le N$. For each $1 \le l \le m_i$ choose a direction $\theta_l \in \mathbb{R}^d$, $\|\theta_l\|_2 = 1$ and let $\mathbf{g}_i^l \in \mathbb{C}^{n_{\Gamma_i}}$ be the coefficient vector of the FE approximation of p^{θ_l} defined in Equation (6.3.1) on the interface Γ_i. Let $\mathbf{u}_i^l$ be the extension of $\mathbf{g}_i^l$ in the sense of Definition 5.2.4.

It remains to be specified how the directions θ_l are chosen. In two space dimensions, we follow [52], and define the plane wave directions via a uniform discretization of the unit circle into circular sectors, see Figure 6.3.1 for an illustration.

Definition 6.3.2 (Plane wave directions in 2D)**.** The plane wave directions $\theta_l \in \mathbb{R}^2$ are defined by

$$\theta_l := \begin{pmatrix} \cos(t_l) \\ \sin(t_l) \end{pmatrix}, \quad \text{where } t_l = \frac{2\pi(l-1)}{m_i}, \quad 1 \le l \le m_i.$$

In three space dimension, the situation is a bit more complicated. In general, a basis of vectors uniformly distributed on the unit sphere is desirable, where it is however not clear what exactly "uniformly" means. For simplicity, we here follow the approach of [144, Algorithm 1].

[1] Despite the algorithm not requiring it in theory, in the parallel implementation using ARPACK, also for the DtN coarse space an estimate of the number of modes needs to be provided initially, cf. Subsection 7.1.3.

Definition 6.3.3 (Plane wave directions in 3D)**.** For $n_t \in \mathbb{N}$ construct the vectors

$$y_{j_1,j_2,j_3} = \begin{pmatrix} \tan\left(\left(2\frac{j_1}{n_t} - 1\right)\frac{\pi}{4}\right) \\ \tan\left(\left(2\frac{j_2}{n_t} - 1\right)\frac{\pi}{4}\right) \\ \tan\left(\left(2\frac{j_3}{n_t} - 1\right)\frac{\pi}{4}\right) \end{pmatrix}, \qquad 0 \leq j_i \leq n_t.$$

Then, from the $(n_t+1)^3$ vectors y_{j_1,j_2,j_3}, select $6n_t^2+2$ vectors θ_l that correspond to triplets $[j_1, j_2, j_3]$ such that at least one of the indices j_1, j_2, j_3 is equal to 0 or n_t, and normalize them to unit length.

6.3.2 Properties of the plane wave coarse space

The matrix Z based on plane waves can become rank deficient for a couple of reasons [52]. The rank deficiency of Z causes in the worst case divergence of the whole iterative scheme. To avoid this problem, we adapt the filtering of the coarse space described in [52], but apply filtering to functions defined on the entire subdomains instead of only the edges. We choose a filtering tolerance ε and do the following: Let Z have the blocks W_i. For each $1 \leq i \leq N$, perform the QR factorization of W_i, and then construct W_i^* as the union of the columns q_j of W_i for which the j-th diagonal entry of R satisfies $|R_{jj}| > \varepsilon$. This is a local procedure that is performed on each subdomain separately. We substitute Z by the matrix constructed from the W_i^*. A too small value of ε can cause the matrix Z to be still rank deficient. The authors of [52] propose to choose ε rather too large than too small, setting $\varepsilon = 10^{-2}$. We denote the method where the plane wave coarse space with filtering tolerance ε is employed by $\mathrm{PW}(\varepsilon)$. If the same number of initial modes m is chosen on all subdomains Ω_i, that is $m_i = m$ for all $1 \leq i \leq N$, then we also use $\mathrm{PW}(\varepsilon, m)$ to denote the resulting method.

Remark 6.3.4 (On the usage of QR factorization for filtering). We note that a normal QR decomposition might not always be sufficient. If the columns of the matrix M that should be decomposed are linearly dependent, then the above described filtering criterion might lead to a wrong space, i.e. one that is different from (and not only smaller than) the original, non-filtered one. Consider the example

$$M := \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} =: QR,$$

where the second and third columns of Q are not in the span of the columns of the matrix M. This problem can be avoided by using a rank-revealing QR factorization using specialized software such as [30, 54]. Moreover, different implementations of the QR factorization might lead to different coarse spaces and different results. For the numerical results in this chapter, we have used MATLAB's QR method, whereas in Chapter 7, we will use the SPQR package [30].

As an adaptive strategy for choosing the coarse space size is crucial, we here examine to what extent it is provided by the filtering procedure. We consider Problem 2 with $n_{\mathrm{loc}} = 40$, $k = 29.3$ and a decomposition into 5×5 subdomains. The dimension of the coarse space depends strongly on the number of modes per subdomain m_i that are initially chosen, even if the additional modes hardly influence the convergence rate: If we choose $m_i = 16$, the coarse space dimension is 384;

with $m_i = 32$ it is 459. However, the number of iterations is 13 versus 11 and hence hardly changes despite the larger, more expensive coarse space. Consequently, even though filtering provides some sort of adaptivity, it is very sensitive to the number of modes that are initially chosen.

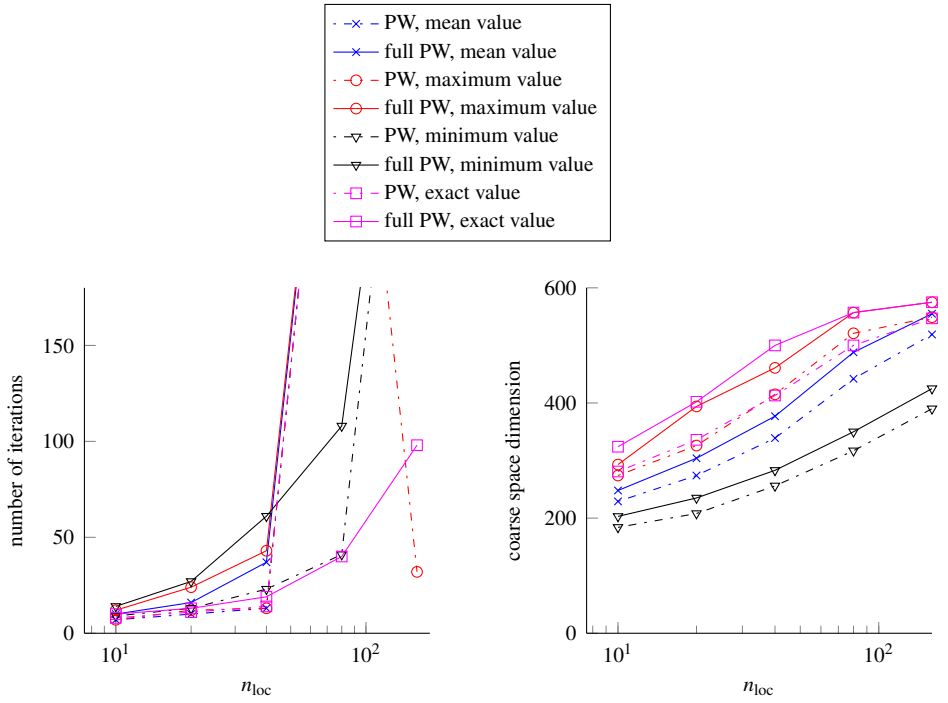
6.3.3 Discussion of alternative definitions

Even though in this work we use the plane waves as presented in the preceding section, this strategy is in no way the only possible choice. Plane waves have been employed by several researchers to enhance iterative methods for Helmholtz-like problems, cf. Chapter 2, and there is no consensus on how to incorporate them best into a DDM. This is partially related to the fact that there is a variety of different DDMs and to complicate issues further, they do not always work on the same set of degrees of freedom. For ease of presentation, we restrict the following discussion to Schwarz methods. Already for this type of DDMs, there are different possibilities known in the literature.

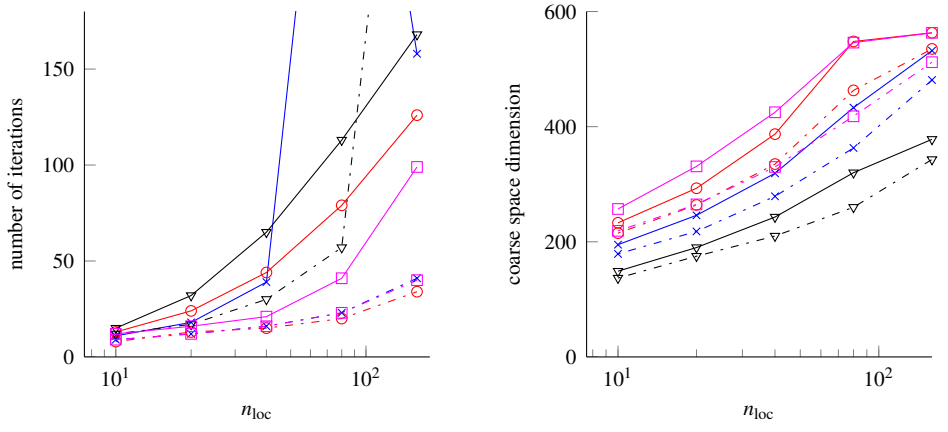
In [92], the plane waves are evaluated on subdomains that might be of a different overlap size than those used for the rest of the DDM. The resulting local subdomain functions are then multiplied by a weighting function associated to a partition of unity. Dirichlet transmission conditions are used in the DDM, which are clearly worse than Robin transmission conditions for the Helmholtz equation, cf. Figure 3.3.2. In [100], for two neighboring (non-overlapping) subdomains Ω_i and Ω_j the plane wave is evaluated on the common interface and zero Dirichlet boundary conditions are imposed on the remaining boundaries. Then either the shifted problem $-\left(\Delta - k^2\right) u = 0$ is solved with these boundary conditions or the harmonic extension $-\Delta u = 0$ is used.

As all these methods use a setting that is different from ours, we do not compare directly with neither of them. However, the approach in [92] is sufficiently similar. So we slightly adapt it in order to be easily able to compare with it. We proceed as in Subsection 6.3.1, but instead of defining $\mathbf{u}_i^l$ by Definition 6.3.1, we set it to be the evaluation of a plane wave in direction k on the whole subdomain Ω_i in direction θ_l. We call the space that is spanned by these vectors the *full* PW *space*, as the plane waves are evaluated on all grid points, whereas in Definition 6.3.1 they are only evaluated on the interface and then extended to the interior of the subdomain. For the heterogeneous case, we also consider different choices for the wave number $\overline{k}$ of the plane waves in Equation (6.3.1): We choose $\overline{k}$ to be either the mean value, the maximum value or the minimum value of the wave number k on this subdomain. Additionally, we consider the case, where $\overline{k} := k$, hence $\overline{k}$ is not a constant. The results are reported in Table 6.3.1 for the homogeneous case, and in Figure 6.3.2 for the heterogeneous case.

For the homogeneous case, we see hardly any difference between using the original version of our coarse space and the "full" version in terms of iteration numbers, except for the experiment for $n_{\text{loc}} = 80$, where conditioning problems occur only for the original version. The dimension of the coarse space with the same filtering tolerance is however smaller for the original PW definition. For the heterogeneous case, on the other hand, the differences are more significant. Here, not only the coarse space for the "full" version is always bigger than the one for the original version, but also the iteration numbers especially for larger wave numbers grow at a faster rate and are also absolutely



(a) Filtering tolerance $\varepsilon = 10^{-2}$



(b) Filtering tolerance $\varepsilon = 10^{-1}$

Figure 6.3.2. *Comparison of different ways to employ the plane waves for the heterogeneous wave guide example, Problem 2. Wave speed $c = c_2$, $\rho = 5$, 5×5 subdomains, preconditioner P_B, $k^3h^2 \approx \frac{2\pi}{10}$.*

n_{loc}	k	PW(10^{-2})		full PW(10^{-2})	
10	11.6	7	(297)	7	(321)
20	18.5	8	(352)	8	(386)
40	29.3	11	(467)	11	(517)
80	46.5	>400	(577)	15	(625)
160	73.8	24	(609)	25	(625)

Table 6.3.1. *Comparison of different ways to employ the plane waves for the homogeneous wave guide example, Problem 2.* 5×5 *subdomains, two-level preconditioner* P_B.

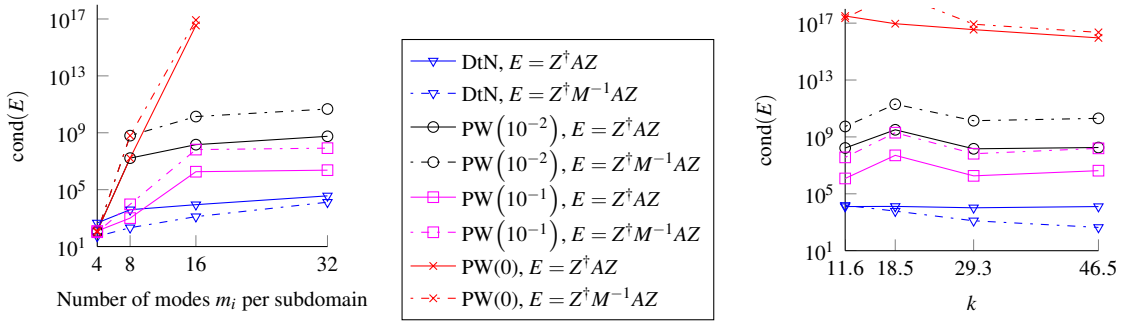
larger than the corresponding ones for our original definition. So it is very important not only to select the right functions for the coarse space, but also choose carefully how to exactly use them in order to achieve the best possible results. The problems with the "full" version might be due to two reasons: As observed in [4], for homogeneous problems, the plane waves with wave number k do not necessarily lie in the kernel of the discrete Helmholtz operator associated to the same wave number due to dispersion errors. This might have effects on the efficiency of the plane waves. On the other hand, we deal with heterogeneous problems. In this context even without dispersion error, the straightforward evaluation of the plane waves does not result in a function that lies in the kernel of the operator that we are interested in. This is also why we refrain from using a different extension operator as for example proposed in [100].

Comparing the results for different choices of $\overline{k}$ with the original, extension-based definition, the minimum value performs significantly worse than the other three choices, which yield almost the same result – except for conditioning problems in Figure 6.3.2a, which can be cured by choosing a larger filtering tolerance in Figure 6.3.2b. As the coarse space based on the mean value is the cheapest one of these three, this is our method of choice.

6.4 Conditioning of the coarse matrix

The condition number of the coarse matrix E plays an important role. If it is too large, the iterative method might stagnate. Here we investigate to what extent the different coarse spaces suffer from conditioning problems. For that purpose, we investigate both the matrix $E = Z^\dagger AZ$, which is used in the preconditioner P_B, and the matrix $E = Z^\dagger M^{-1} AZ$, which is used in the preconditioner Q_G. The matrix Z is based either on the DtN functions or on the PW ones with different filtering tolerances ε. As already outlined in Subsection 6.3.2, the plane wave coarse matrix is likely to suffer from conditioning problems if too many plane waves are used: Plane waves travelling in similar directions might be almost linearly dependent, if the wave number k is small.

For the DtN coarse space, the matrix Z is constructed from an orthonormal basis of eigenvectors defined on the interfaces of the subdomains. Their extensions to the interior of the subdomains are in general not orthogonal as the extension matrix $A_{II}^{-1} A_{I\Gamma}$ is not unitary and hence does not conserve orthogonality. Nevertheless, in the numerical experiments for Problem 2 the condition number for



(a) $n_{\text{loc}} = 40$, $k = 29.3$

(b) $m_i = 16$, $k^3h^2 \approx \frac{2\pi}{10}$

Figure 6.4.1. *Condition number of coarse matrix E. Problem 2, 5×5 subdomains.*

DtN behaves well: In Figure 6.4.1a, we examine the dependence of the condition number on the coarse space size and compare it to the one for plane waves, with and without filtering. For the DtN, the condition number of E is only mildly affected by the coarse space dimension. The same is true for plane waves with filtering, but here the upper bound is significantly larger. While for DtN the choice $E = Z^\dagger M^{-1} AZ$ has a slightly better condition number, for PW it is worse, even though the behavior in all cases is similar. This could be the reason for the convergence problems that we observe for some of the numerical experiments with Q_G in Section 6.5.

In Figure 6.4.1b, we investigate whether this behavior carries over to a broader range of wave numbers k. We choose a fixed number $m_i = 16$ of modes per subdomain. If k and h are varied such that k^3h^2 is constant, the condition number for DtN remains almost constant. So filtering of the coarse modes is not necessary here. Furthermore, the condition number is significantly lower than for the plane waves. This is again true for both versions of the coarse matrix E. Choosing an even larger filtering tolerance ε for the plane waves would eventually make the condition number of E decrease to the level of the condition number of the DtN matrix. However, this would eventually eliminate also important modes from the coarse space. To overcome these problems, it might hence be worth to either define a different filtering strategy, an automatic criterion for choosing the right plane wave directions, or use a different, less ill-conditioned coarse space.

Remark 6.4.1. Note that the conditioning problems of the coarse matrix for the PW coarse space exist independently of the possible singularity of the Helmholtz extension, cf. Remark 5.2.5. This is demonstrated e.g. in Figure 6.3.2, where also the method with the full PW coarse space does not converge in case of too many plane waves.

6.5 Numerical experiments for the wave guide problem

In this section, we investigate the two-level methods using the DtN coarse space for the wave guide problem defined in Problem 2 and compare the results to those achieved with the benchmark coarse space based on plane waves. This section is divided into two subsections, where the first one only deals with homogeneous wave numbers, whereas the second one also allows for heterogeneities.

		5×5 subdomains					10×10 subdomains				
n_{glob}	k	1-lev	DtN		PW(10^{-2})		1-lev	DtN		PW(10^{-2})	
100	18.5	80	15	(144)	8	(352)	144	18	(344)	7	(1152)
200	29.3	116	18	(224)	11	(467)	241	26	(460)	9	(1286)
400	46.5	156	29	(299)	> 400	(577)	327	51	(624)	13	(1708)
800	73.8	217	39	(508)	24	(609)	> 400	65	(936)	> 400	(2346)

(a) For P_B

n_{loc}	k	1-lev	DtN		PW(10^{-2})		PW(10^{-1})	
20	18.5	80	16	(144)	> 400	(352)	9	(293)
40	29.3	116	19	(224)	> 400	(467)	13	(382)
80	46.5	156	30	(299)	> 400	(577)	16	(505)
160	73.8	217	40	(508)	> 400	(609)	25	(597)

(b) For Q_G, 5×5 subdomains

Table 6.5.1. *Comparison of* RAS *method in Equation (3.2.2) without coarse space (*1*-lev), and with DtN and* PW *coarse spaces. Number of iterations (dimension of coarse space) for Problem 2.*

6.5.1 Performance for homogeneous wave guide problem

We study the performance of the DtN coarse space for Problem 2 with homogeneous wave number k. In Table 6.5.1, the number of iterations for different wave numbers k is shown. For both preconditioners P_B and Q_G, it increases slightly with k if k^3h^2 is constant. This could be due to the decreasing physical size Lh of the overlap, cf. Table 6.5.4. Moreover, the dimension of the DtN coarse space depends linearly on the wave number k. The number of iterations for the one-level method doubles if the number of subdomains is doubled in both directions. With the DtN coarse space, the influence of the number of subdomains is not that strong, but still present. While for the DtN coarse space, Q_G and P_B in these experiments show almost the same behavior, for the PW one, employing Q_G leads to sincere convergence problems for the standard filtering tolerance $\varepsilon = 10^{-2}$. This is probably due to the ill-conditioning of the global matrix E, cf. Section 6.4. Choosing a larger filtering tolerance $\varepsilon = 10^{-1}$ resolves this problem and leads to results similar to the balancing preconditioner case. So the preconditioner defined by Q_G seems to be even more sensitive to an ill-conditioned matrix E, cf. Figure 6.4.1.

In Table 6.5.1, the number of iterations with the PW coarse space is smaller than with the DtN one. However, it is not fair to compare these numbers as the dimensions of the coarse spaces differ significantly. Therefore, in Table 6.5.2 we compare the two methods enforcing the dimension to be the same by prescribing a fixed number of modes m_i on each subdomain also for DtN. These numbers m_i are the same on all subdomains and are computed by dividing the sizes in Table 6.5.1

			For P_B			For Q_G				
n_{loc}	k	m_i	DtN	PW(10^{-2})		DtN	PW(10^{-2})		PW(10^{-1})	
10	11.6	4	14	17	(100)	15	17	(100)	17	(100)
20	18.5	6	21	23	(150)	19	19	(150)	19	(146)
40	29.3	9	23	22	(225)	23	22	(225)	22	(225)
80	46.5	12	35	35	(296)	35	30	(296)	29	(292)
160	73.8	21	38	29	(521)	42	>400	(521)	31	(513)

(a) Number of modes m_i computed from the DtN coarse space dimension

			For P_B		For Q_G				
n_{loc}	k	m_i	DtN	PW(10^{-2})	DtN	PW(10^{-2})		PW(10^{-1})	
10	11.6	12	8	7	8	7	(288)	7	(244)
20	18.5	15	9	9	9	>400	(355)	9	(305)
40	29.3	17	13	12	13	>400	(409)	13	(373)
80	46.5	24	18	16	19	>400	(556)	16	(496)
160	73.8	25	36	24	39	>400	(609)	25	(597)

(b) Number of modes m_i computed from the PW$\left(10^{-2}\right)$ coarse space dimension

Table 6.5.2. *Comparison of number of iterations for DtN and* PW *with identical coarse space size.* 5×5 *subdomains, Problem 2.*

by the number of subdomains. With this setting, DtN and PW then yield approximately the same convergence rates for both two-level preconditioners.

In Figure 6.5.1, we examine whether the convergence rates depend on the value of the constant k^3h^2. There is no clear indication which value might be optimal, but a rather fine grid gives the highest number of iterations absolutely. This is the same for both P_B and Q_G. The coarse space dimension depends on the wave number k but is independent of the grid width h.

In Table 6.5.3, the mesh width h is kept fixed and the wave number k is varied. The coarse space dimension increases with k. The number of iterations remains only constant if k is large enough. For k small, the coarse space built using Criterion 5.2.3 is so small that the number of iterations remains rather large. Also here the behavior is similar for P_B and Q_G: While the former performs better for larger values of the wave number k, the latter is more efficient for low frequencies, with the relative difference however not exceeding 10% in terms of iteration numbers.

In Table 6.5.4, two properties of the DtN coarse space and of Criterion 5.2.3 become visible: On the one hand, for small k, only one mode per subdomain is chosen and the number of iterations is hardly influenced by the coarse space. This is not a flaw of the coarse space itself, but due to Criterion 5.2.3; choosing more modes results in a stronger impact on convergence rates. For the homogeneous case this is not a problem as cases with very small wave number k can be solved by standard methods.

		For P_B		For Q_G	
k	1-level	DtN		DtN	
5	106	88	(25)	79	(25)
10	115	68	(70)	58	(74)
15	117	61	(90)	57	(90)
30	133	31	(224)	33	(224)
45	169	36	(299)	39	(299)

Table 6.5.3. *Dependence of number of iterations (coarse space dimension) on wave number k for fixed mesh width h. Problem 2,* 5×5 *subdomains,* $n_{\text{loc}} = 120$.

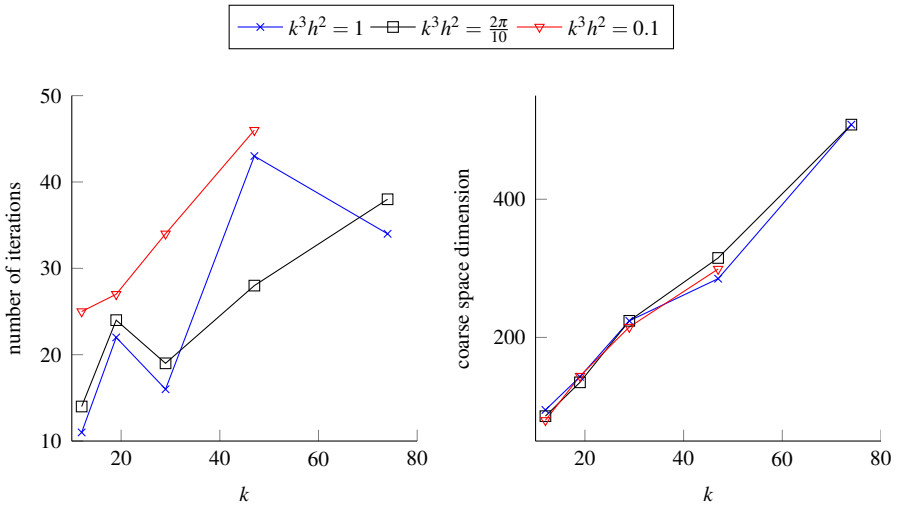
	$n_{\text{loc}} = 20, L = 2$			$n_{\text{loc}} = 80, L = 2$			$n_{\text{loc}} = 80, L = 8$		
k	1-level	DtN		1-level	DtN		1-level	DtN	
1	73	56	(25)	94	81	(25)	66	39	(25)
5	64	43	(25)	96	78	(25)	55	37	(25)
10	68	21	(74)	106	49	(74)	66	22	(74)
20	84	32	(139)	107	32	(144)	86	33	(139)

(a) For P_B

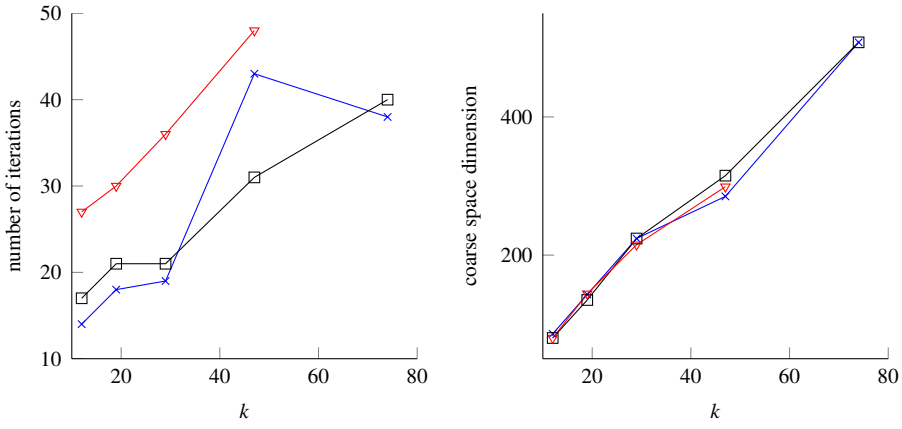
	$n_{\text{loc}} = 20, L = 2$			$n_{\text{loc}} = 80, L = 2$			$n_{\text{loc}} = 80, L = 8$		
k	1-level	DtN		1-level	DtN		1-level	DtN	
1	73	51	(25)	94	73	(25)	66	46	(25)
5	64	40	(25)	96	70	(25)	55	34	(25)
10	68	24	(74)	106	47	(74)	66	24	(74)
20	84	22	(139)	107	34	(144)	86	21	(139)

(b) For Q_G

Table 6.5.4. *Dependence of number of iterations (coarse space dimension) on overlap L / mesh width h. Problem 2,* 5×5 *subdomains.*



(a) For P_B



(b) For Q_G

Figure 6.5.1. *Number of iterations and coarse space dimension for different values of k^3h^2. Problem 2, 5×5 subdomains.*

		Number of subdomains							
n_{loc}	k	5×5		5×10		5×20		5×40	
10	11.6	15	(80)	18	(160)	22	(320)	30	(640)
20	18.5	15	(144)	16	(314)	16	(654)	19	(1334)
40	29.3	18	(224)	18	(484)	20	(1004)	22	(2044)
80	46.5	29	(299)	37	(624)	48	(1274)	66	(2574)

(a) For P_B

		Number of subdomains							
n_{loc}	k	5×5		5×10		5×20		5×40	
10	11.6	16	(80)	19	(160)	24	(320)	32	(640)
20	18.5	16	(144)	18	(314)	20	(654)	25	(1334)
40	29.3	19	(224)	21	(484)	24	(1004)	30	(2044)
80	46.5	32	(299)	43	(624)	69	(1274)		

(b) For Q_G

Table 6.5.5. *Dependence of number of iterations (coarse space dimension) on number of subdomains. DtN coarse space, Problem 2. The number of subdomains is increased in the y-direction and the domain size increases accordingly. That is, for* 5×5 *subdomains the domain* Ω *is* $[0,1]^2$*, for* 5×10 *subdomains it is* $[0,1]\times[0,2]$*, ...*

On the other hand, we study the influence of mesh refinement. If the mesh is refined twice and the overlap stays constant in terms of number of elements L, see the central columns of Table 6.5.4, the convergence rates deteriorate a lot; in the worst cases we get more than a factor 2 more iterations with about the same coarse space size. If however the physical size of the overlap Lh is constant, see the last three columns of Table 6.5.4, the number of iterations even decreases if the mesh is refined. This behavior is probably due to the transmission conditions that make the convergence rates depend on the size of the overlap [64]. Within this context, it might be worth to investigate more advanced transmission conditions, e.g. optimized ones [57]. While the two two-level preconditioners also in these experiments show a similar performance in most cases, there are some outliers, the most remarkable one for $k = 20$ in the last column, where Q_G needs only two thirds of the iterations of P_B.

In Table 6.5.5, the number of subdomains in one direction is varied, while the number of wavelengths per subdomain is kept fixed. The coarse space dimension grows approximately linearly with the number of subdomains as expected from the construction. Unfortunately, the increase in the coarse space dimension is insufficient to keep the iteration numbers constant if the number of wavelengths in the global domain grows. This might at least partially be due to the fact that the transmission conditions that we employ are non-optimal. The increase in the iteration count is worse the larger the wave number k is.

	DtN			PW(10^{-2})		
# subdomains	# it.	size	time in s	# it.	size	time in s
2×2	23	(68)	2.90×10^2	17	(96)	3.08×10^2
4×4	35	(200)	2.67×10^2	16	(368)	2.71×10^2
8×8	44	(416)	1.80×10^2	12	(1116)	5.76×10^2
16×16	57	(960)	4.78×10^2	10	(3256)	3.97×10^3
32×32	47	(2944)	3.72×10^3	8	(9208)	3.28×10^4

(a) For P_B

	DtN		PW(10^{-2})		PW(10^{-1})	
# subdomains	# it.	size	# it.	size	# it.	size
2×2	24	(68)	>400	(96)	18	(88)
4×4	31	(200)	>400	(364)	15	(320)
8×8	40	(416)	>400	(1116)	14	(924)
16×16	60	(960)	>400	(3256)	12	(2686)
32×32	48	(2944)				

(b) For Q_G

Table 6.5.6. *Strong scaling test: The problem size is fixed, but we vary the number of subdomains. We give the number of iterations (it.), the dimension of the coarse system (size) and the time needed to solve the algebraic equations (time). The global number of grid points in each direction is* 320*, the wave number* $k = 40$*. We decompose* $\Omega = [0,1]^2$ *uniformly into squares.*

In Table 6.5.6, we perform a similar test. In contrast to the previous experiment, we keep the number of wavelengths in the global domain fixed and only vary the number of subdomains in both directions. Here, the iteration numbers increase slightly even though a growing coarse space is used. For the plane waves, on the other hand, the resulting coarse space is not only up to a factor of > 3 larger than the DtN one, but also grows at a higher rate. For the case of 32×32 subdomains, the number of coarse degrees of freedom amounts to roughly 9% of the unknowns for the one-level method. This even causes the iteration numbers to decrease with respect to the case of fewer subdomains. However, each assembly of the coarse matrix and each iteration step are more costly for a large global problem. This is clear from the timings that we additionally give in Table 6.5.6a, especially for the 32×32 subdomains case. For the experiments in this chapter, a serial, non-optimized implementation of the method is used. For more results on the runtime, see Chapter 7, where a parallel implementation of the DDM is investigated for three-dimensional examples.

6.5.2 Performance for heterogeneous wave guide problem

In this section, we study some small heterogeneous test cases for Problem 2 with velocity profiles c_i, $i = 1,2,3$, defined in Figure 1.4.3. In Table 6.5.7, Table 6.5.8, and Table 6.5.9 the iteration numbers

for the different types of velocities are shown. For PW, for some cases convergence stagnates due to ill-conditioning despite the rather large filtering tolerance. Moreover, the adaptively chosen coarse space size for DtN is significantly smaller than that for PW. This also has a small effect on the convergence rates, with PW performing better. As in the homogeneous case, the coarse space size increases with the wave number. The different velocities c_i, $i = 1,2,3$, do not have much influence on the convergence rates, the only notable difference is for $\rho = 1$ and the DtN coarse space, where the iteration numbers are slightly higher for $c = c_1$.

		$\rho = 5$				$\rho = 10$			
n_{loc}	ω	DtN		PW(10^{-2})		DtN		PW(10^{-2})	
10	11.6	21	(56)	8	(222)	27	(47)	10	(189)
20	18.5	27	(83)	12	(256)	38	(67)	14	(222)
40	29.3	36	(115)	15	(321)	46	(105)	19	(277)
80	46.5	43	(180)	24	(408)	60	(149)	30	(351)
160	73.8	49	(290)	32	(477)	74	(228)	> 400	(418)

(a) For P_B

n_{loc}	ω	DtN		PW(10^{-2})		PW(10^{-1})	
10	11.6	23	(56)	8	(222)	10	(166)
20	18.5	28	(83)	12	(256)	15	(205)
40	29.3	35	(115)	> 400	(321)	20	(254)
80	46.5	41	(180)	> 400	(408)	27	(334)
160	73.8	50	(290)	> 400	(477)	> 400	(429)

(b) For Q_G, $\rho = 5$

n_{loc}	ω	DtN		PW(10^{-2})		PW(10^{-1})	
10	11.6	31	(47)	11	(189)	12	(148)
20	18.5	39	(67)	14	(222)	18	(177)
40	29.3	44	(105)	> 400	(277)	23	(225)
80	46.5	53	(149)	> 400	(351)	34	(289)
160	73.8	71	(228)	> 400	(418)	> 400	(380)

(c) For Q_G, $\rho = 10$

Table 6.5.7. *Number of iterations (coarse space dimension) for heterogeneous wave guide example, Problem 2. Wave speed $c = c_1$, 5×5 subdomains.*

		$\rho = 5$				$\rho = 10$			
n_{loc}	ω	DtN		PW(10^{-2})		DtN		PW(10^{-2})	
10	11.6	18	(69)	8	(229)	19	(69)	8	(214)
20	18.5	23	(111)	10	(274)	23	(111)	11	(263)
40	29.3	31	(159)	13	(339)	35	(159)	16	(326)
80	46.5	33	(242)	> 400	(442)	40	(236)	> 400	(414)
160	73.8	47	(388)	> 400	(519)	57	(378)	42	(494)

(a) For P_B

n_{loc}	ω	DtN		PW(10^{-2})		PW(10^{-1})	
10	11.6	21	(69)	8	(229)	10	(179)
20	18.5	27	(111)	> 400	(274)	14	(218)
40	29.3	35	(159)	> 400	(339)	12	(279)
80	46.5	38	(242)	> 400	(442)	> 400	(363)
160	73.8	53	(388)	> 400	(519)	> 400	(481)

(b) For Q_G, $\rho = 5$

n_{loc}	ω	DtN		PW(10^{-2})		PW(10^{-1})	
10	11.6	23	(69)	9	(214)	11	(169)
20	18.5	29	(111)	> 400	(263)	16	(207)
40	29.3	44	(159)	> 400	(326)	28	(263)
80	46.5	45	(236)	> 400	(414)	> 400	(346)
160	73.8	62	(378)	> 400	(494)	> 400	(455)

(c) For Q_G, $\rho = 10$

Table 6.5.8. *Number of iterations (coarse space dimension) for heterogeneous wave guide example, Problem 2. Wave speed $c = c_2$, 5×5 subdomains.*

		$\rho = 5$				$\rho = 10$			
n_{loc}	ω	DtN		PW(10^{-2})		DtN		PW(10^{-2})	
10	11.6	17	(67)	8	(234)	24	(47)	9	(185)
20	18.5	23	(92)	11	(273)	31	(64)	13	(215)
40	29.3	28	(143)	15	(340)	39	(91)	18	(262)
80	46.5	34	(217)	> 400	(439)	51	(142)	27	(335)
160	73.8	42	(336)	> 400	(522)	61	(223)	> 400	(417)

(a) For P_B

		$\rho = 5$						$\rho = 10$					
n_{loc}	ω	DtN		PW(10^{-2})		PW(10^{-1})		DtN		PW(10^{-2})		PW(10^{-1})	
10	11.6	19	(67)	8	(234)	10	(178)	27	(47)	10	(185)	11	(145)
20	18.5	24	(92)	12	(273)	13	(221)	33	(64)	13	(215)	16	(173)
40	29.3	30	(143)	> 400	(340)	18	(273)	39	(91)	> 400	(262)	22	(212)
80	46.5	37	(217)	> 400	(439)	25	(363)	47	(142)	> 400	(335)	31	(277)

(b) For Q_G

Table 6.5.9. *Number of iterations (coarse space dimension) for heterogeneous wave guide example, Problem 2. Wave speed $c = c_3$, 5×5 subdomains.*

ρ	1-level	DtN		PW(10^{-2})		PW(10^{-1})	
10^0	156	29	(299)	43	(577)	16	(505)
10^1	154	40	(236)	>400	(414)	26	(346)
10^2	173	52	(236)	>400	(388)	33	(320)
10^3	177	53	(236)	>400	(379)	35	(315)

(a) For P_B

ρ	1-level	DtN		PW(10^{-2})		PW(10^{-1})	
10^0	156	31	(299)	>400	(577)	16	(505)
10^1	154	45	(236)	>400	(414)	>400	(346)
10^2	173	59	(236)	>400	(388)	>400	(320)
10^3	177	64	(236)	>400	(379)	>400	(315)

(b) For Q_G

Table 6.5.10. *Number of iterations (coarse space dimension) for varying contrast ρ. Heterogeneous Problem 2, wave speed $c = c_2$, 5×5 subdomains, $n_{\text{loc}} = 80$, $\omega = 46.5$.*

In Table 6.5.10, we vary the contrast $\rho := k_{\max}/k_{\min}$. With increasing contrast, the convergence rates for the one-level method deteriorate. For DtN, even though the coarse space size decreases, the number of iterations grows only slightly. Only for larger contrast, the situation deteriorates. In the parts of the domain where ρ is large, the problem is very close to the Laplacian and hence almost positive definite. As we have seen in Table 6.5.4, DtN does not work well for such situations since the coarse space is too small to enhance convergence. PW does not suffer from this problem, because the coarse space size is not chosen adaptively. Here, the filtering tolerance for PW has to be larger than 10^{-2} to avoid stagnation of convergence due to ill-conditioning of the matrix E. The convergence only stagnates at a certain error; consequently visibility of this effect depends on the desired accuracy of the iterative solution. Additionally, for these experiments the conditioning problems with the PW coarse space seem to be severe; even with the larger filtering tolerance $\varepsilon = 10^{-1}$, none of the heterogeneous test cases converges to the desired tolerance for Q_G.

In Table 6.5.11, we choose the same coarse space dimension for both DtN and PW to verify that the better convergence rates for PW are due to the size of the coarse space. In contrast to the homogeneous case in Table 6.5.2, for the heterogeneous one DtN performs significantly better than PW when the number of modes chosen is the same, in particular for larger wave number k.

6.6 Extension to other problems

In this section, we consider also the other examples defined in Section 1.4 to confirm that our results are valid for a broader range of examples.

n_{loc}	ω	m_i	DtN	PW(10^{-2})	
10	11.6	3	18	20	(75)
20	18.5	5	20	24	(123)
40	29.3	7	31	40	(171)
80	46.5	10	38	55	(237)
160	73.8	16	57	89	(356)

(a) For P_B

n_{loc}	ω	m_i	DtN	PW(10^{-2})		PW(10^{-1})	
10	11.6	3	21	22	(75)	22	(75)
20	18.5	5	23	25	(123)	25	(123)
40	29.3	7	38	40	(171)	41	(163)
80	46.5	10	42	>400	(237)	45	(223)
160	73.8	16	59	>400	(356)	63	(346)

(b) For Q_G

Table 6.5.11. *Comparison of number of iterations for DtN and* PW *with identical coarse space size. Heterogeneous Problem 2, wave speed $c = c_2$, $\rho = 10$, 5×5 subdomains.*

Irregular decomposition In all the previous experiments, we have used a decomposition into square subdomains to ensure reproducibility. Here we show that this is restriction is not necessary for the method to work. In Table 6.6.1 we consider Problem 2, where both the decomposition done with Metis [86] and the triangulation are now irregular. Compared to the regular case in Table 6.5.1, the method behaves similarly. While the dimension of the coarse space increases slightly, the number of iteration is almost the same.

Free space problem Here we examine Problem 3, where non-reflecting boundary conditions are imposed on the entire boundary. The iteration numbers for different partitions are reported in

n_{glob}	k	1-level	DtN	
50	11.6	64	14	(116)
100	18.5	92	15	(168)
200	29.3	130	20	(257)
400	46.5	173	29	(381)
800	73.8	256	36	(645)

(a) For P_B

n_{glob}	k	1-level	DtN	
50	11.6	64	15	(116)
100	18.5	92	17	(168)
200	29.3	130	25	(257)
400	46.5	173	33	(381)
800	73.8	256	43	(645)

(b) For Q_G

Table 6.6.1. *Number of iterations (coarse space dimension) for an irregular domain decomposition using Metis [86]. Homogeneous Problem 2, 25 subdomains.*

		5×5 subdomains				10×10 subdomains			
		DtN		PW(10^{-2})		DtN		PW(10^{-2})	
k	n_{glob}								
18.5	100	15	(144)	8	(355)	16	(364)	7	(1152)
29.3	200	18	(224)	11	(466)	22	(460)	9	(1288)
46.5	400	26	(315)	> 400	(577)	43	(660)	12	(1712)
73.8	800	30	(514)	24	(609)	47	(956)	16	(2346)

(a) For P_B

		DtN		PW(10^{-2})		PW(10^{-1})	
k	n_{glob}						
18.5	100	15	(144)	8	(355)	9	(293)
29.3	200	18	(224)	> 400	(466)	13	(379)
46.5	400	27	(315)	> 400	(577)	16	(511)
73.8	800	33	(514)	> 400	(609)	25	(597)

(b) For Q_G: 5×5 subdomains

		DtN		PW(10^{-2})		PW(10^{-1})	
k	n_{glob}						
18.5	100	17	(364)	23	(1152)	8	(872)
29.3	200	22	(460)	> 400	(1288)	11	(1132)
46.5	400	35	(660)	> 400	(1712)	15	(1380)
73.8	800	57	(956)	> 400	(2346)	18	(1928)

(c) For Q_G: 10×10 subdomains

Table 6.6.2. *Number of iterations (coarse space dimension) for the free space problem, Problem 3.*

Table 6.6.2. The qualitative behavior is similar to the one observed for Problem 2 in Table 6.5.1, but the absolute number of iterations is lower, in particular for the one-level method.

Wedge problem We consider the wedge problem, Problem 4. The results are reported in Table 6.6.3. Also for this case, the 2-level method with the coarse space based on the DtN operator shows a good behavior. Notably, for the 60 subdomain case, the results for PW here are significantly better for Q_G than for P_B. To be able to compare with the results for the unit square, note that the number of wavelengths in the y-direction for the smallest angular frequency $\omega = 90$ corresponds to a wave number k varying between 30 and 60 for the unit square.

Marmousi problem As a last example, we look at the Marmousi problem, Problem 5. In contrast to the other experiments in this section, due to the size of the problem, the experiments for this case are done with the parallel FreeFem++ code on four nodes of the CUB cluster instead of the serial

		15 subdomains				60 subdomains			
ω	n	DtN		PW(10^{-2})		DtN		PW(10^{-2})	
90	150×250	14	(267)	13	(346)	22	(541)	11	(1038)
180	300×500	16	(514)	33	(375)	23	(1074)	22	(1426)
360	600×1000	20	(968)	73	(375)	25	(2113)	86	(1500)

(a) For P_B

ω	n	DtN		PW(10^{-2})		PW(10^{-1})	
90	150×250	14	(267)	12	(346)	12	(323)
180	300×500	15	(514)	24	(375)	24	(373)
360	600×1000	18	(968)	50	(375)	50	(375)

(b) For Q_G: 15 subdomains

ω	n	DtN		PW(10^{-2})		PW(10^{-1})	
90	150×250	21	(541)	10	(1038)	12	(877)
180	300×500	22	(1074)	15	(1426)	15	(1333)
360	600×1000	26	(2113)	42	(1500)	42	(1500)

(c) For Q_G: 60 subdomains

Table 6.6.3. *Number of iterations (coarse space dimension) for the wedge problem, Problem 4 decomposed with Metis.*

FreeFem++/MATLAB code, for details see Chapter 7. We use a decomposition into 25 subdomains with Metis [86], and give the results in Table 6.6.4.

6.7 Conclusions

In this chapter, we successfully tested the two-level method using the DtN coarse space for two-dimensional homogeneous and heterogeneous Helmholtz problems. Furthermore, we compared the DtN coarse space to one based on an established idea, using plane waves, see e.g. [52]. While the two-dimensional examples are rather small, the results are promising. For the homogeneous test cases, the DtN coarse space shows a performance similar to that of the PW coarse space, if the latter converges. However, the DtN coarse space overcomes some problems from which the PW one suffers, such as stagnation of convergence due to ill-conditioning and the need to tune its size. For heterogeneous examples, the convergence rates for the DtN coarse space are better than those for the PW one. The results are independent of the decomposition or the example chosen as the more complex problems in Section 6.6 show.

While the results in this chapter are promising, two important points remain open: On the one hand, we used a serial code for an inherently parallel method. For that reason, in Chapter 7, we will explain how to parallelize the code. On the other hand, the examples in this section are rather small

ω	$n_x \times n_y$	DtN		PW(10^{-2})	
1	1021×323	62	(25)	32	(270)
10	1021×323	29	(151)	34	(600)
20	2042×646	36	(298)	57	(625)
30	3064×968	41	(454)	78	(625)

Table 6.6.4. *Number of iterations (coarse space dimension) for the Marmousi problem, Problem 5 decomposed with Metis. n_x and n_y denote the global number of grid points in x- and y-direction, respectively. The preconditioner P_B is used.*

and restricted to two space dimensions. In order to fully explore the properties of the method, in Chapter 7 we investigate larger, three-dimensional examples.

Chapter 7

Numerical results for three-dimensional problems

In Chapter 6, we tested the two-level DDMs using the DtN coarse space on two-dimensional problems in order to understand their characteristics and their behavior. In this chapter, we investigate the performance of the coarse space also for the three-dimensional case. By testing the methods also on larger and more complicated examples, we gain further insight into its behavior. Due to the typically significantly larger size of the linear system of equations in the three-dimensional experiments, the code needs to be parallelized and executed on a cluster. Details of the parallel implementation are described in the beginning of this chapter in Section 7.1. The numerical experiments are presented in Section 7.2.

7.1 Implementation

Even though the main feature of DDMs is the possibility to parallelize them, in Chapter 6 we have worked with a serial implementation of the RAS method. This is feasible as long as the examples are relatively small. In three space dimensions, however, the problem size increases rapidly with the number of grid points per direction and hence with the wave number k. Consequently, a parallelized version of the code has to be used. In this section, we give an overview of the most important aspects when parallelizing and describe in detail the implementation of some selected operations. Subsection 7.1.2 is particularly important, as together with Section 4.5 it explains why we use the possibly singular preconditioner P_B instead of the non-singular one Q_G in this chapter.

7.1.1 Parallel implementation of the restricted additive Schwarz method

The one-level RAS preconditioner reads, see Equation (3.2.2)

$$M^{-1} = \sum_{j=1}^{N} \tilde{R}_j^T A_j^{-1} R_j,$$

where A_j is a local stiffness matrix. When solving a linear system of equations preconditioned with the RAS preconditioner, we need to apply the RAS preconditoner M^{-1} and the global stiffness matrix A to a vector, but there is no need for assembling these two matrices. In fact, in a parallel code, the global matrices should never be assembled, and also global vectors are very seldom used. Also the restriction operators R_j that appear in the definition of M^{-1} are in practice only global-to-local index mappings and appear in pairs $R_j R_i^T$, which involve only communications with neighbors. In Algorithm 7.1.1, we summarize the preprocessing necessary for the RAS method such that all the previously mentioned matrix-vector operations can be performed by local operations and communication with neighbors only. Note that everything inside the `for`-loop is localized and does not involve any global operations. The need for the various operators defined here, in particular for the matrices $\hat{A}_j$ and $R_i R_j^T$ is explained in the next paragraph.

Algorithm 7.1.1 Computation of the domain decomposition and related operators

Input: Computational domain Ω and the corresponding mesh, number of subdomains N, number of elements in the overlap n_{ov}

Output: Restriction operators R_j, partition of unity matrices D_j, local stiffness matrices A_j, $1 \leq j \leq N$

1: Compute a decomposition of the domain Ω into N non-overlapping, mesh-conforming subdomains: $\Omega = \bigcup_{j=1}^{N} \Omega'_j$.
2: **for** $j \leftarrow 1$ to N **do**
3: Add n_{ov} layers of elements to Ω'_j according to Definition 3.2.1 to construct the overlapping subdomain $\Omega_j \subset \Omega$.
4: Implement the restriction matrix $R_j \in \mathbb{R}^{|\Omega_j| \times |\Omega|}$.
5: **for** i such that $\overline{\Omega}_j \cap \overline{\Omega}_i \neq \emptyset$ **do**
6: Implement the application of $R_i R_j^T$ as the exchange of values with the neighbor Ω_i in the overlap.
7: **end for**
8: Compute the matrix $D_j \in \mathbb{R}^{|\Omega_j| \times |\Omega_j|}$ related to the partition of unity as defined in Subsection 3.2.2.
9: Compute the local stiffness matrices A_j with Robin boundary conditions as defined in Equation (3.2.3).
10: Compute the local restricted stiffness matrices $\hat{A}_j := R_j A R_j^T$ by Algorithm 7.1.2.
11: **end for**

We here explain only the multiplication with the global stiffness matrix, as it is the most involved step and all other operations can easily be performed once the operators in Algorithm 7.1.1 are at hand. For the parallel method, instead of computing $A\mathbf{u}$, we only compute the vectors $R_j A\mathbf{u}$, the restrictions of $A\mathbf{u}$ to the subdomain Ω_j, $1 \leq j \leq N$. This can be done by local computations followed

by communication with direct neighbors, as the following calculation shows:

$$R_j A\mathbf{u} = R_j A \left(\sum_{i=1}^{N} R_i^T D_i R_i \right) \mathbf{u} = \sum_{i=1}^{N} R_j A R_i^T D_i R_i \mathbf{u} = \sum_{i=1}^{N} R_j R_i^T R_i A R_i^T D_i R_i \mathbf{u}.$$

Here, the last equality holds, as by definition D_i is zero on the boundary of subdomain Ω_i and A only couples neighboring degrees of freedom. Using $R_j R_i^T = 0$ for $\overline{\Omega}_j \cap \overline{\Omega}_i = \emptyset$, we thus get

$$R_j A\mathbf{u} = \sum_{i:\overline{\Omega}_j \cap \overline{\Omega}_i \neq \emptyset} R_j R_i^T \left(R_i A R_i^T \right) D_i R_i \mathbf{u}.$$

All operations except for the multiplication with $R_j R_i^T$ can be performed locally, as $\hat{A}_i := R_i A R_i^T$ is the restriction of A to subdomain Ω_i, which can be assembled and applied completely locally, see Algorithm 7.1.2. The parallel update, represented by the term $R_i R_j^T$, requires only communication of the values in the overlap between neighboring subdomains; there is no need to use global vectors. Hence the RAS preconditioner M^{-1} can be applied to a vector using local operations and communication with neighbors only.

Algorithm 7.1.2 Computation of the restricted stiffness matrices $\hat{A}_j := R_j A R_j^T$

1: Compute Ω_j^+, the extension of Ω_j by one layer of elements according to Definition 3.2.1.
2: Compute the matrix A_j^+, the stiffness matrix yielded by the bilinear form $a(.,.)$, see Equation (1.3.2), on $\mathcal{V}_h\left(\Omega_j^+\right)$.
3: Delete the rows and columns in A_j^+ that are associated to degrees of freedom lying on elements in $\Omega_j^+ \setminus \Omega_j$. This gives the matrix $\hat{A}_j := R_j A R_j^T$.

7.1.2 Parallel coarse matrix assembly

Even though the coarse matrix E is a global matrix, which is stored and factorized on one processor only in our implementation[1], its assembly is in large parts parallelizable, for the coarse space introduced in Chapter 5. This is independent of the specific form of the coarse space, but depends solely on the following assumption, which we will use throughout this section.

Assumption 7.1.1 (Local construction)**.** *The coarse space is built of local, weighted functions: Each column of Z has the form $R_j^T D_j \mathbf{v}_j$, where $\mathbf{v}_j$ is the coefficient vector of a function $v \in \mathcal{V}_h(\Omega_j)$ and D_j is the matrix associated to a partition of unity function on Ω_j as used in Equation (3.2.2). Furthermore, the matrix D_j is zero on the degrees of freedom associated to the boundary of Ω_j.*

[1] While it is in principle possible to parallelize the coarse matrix, see e.g. [84], its nature is that of a global problem and the question how to parallelize it is out of scope in this work.

The coarse space matrix Z defined in Chapter 5, see in particular Algorithm 5.2.1, satisfies this assumption. It is not only important for the sparsity of the coarse matrix E, but it also facilitates the parallel implementation.

If $B = A$ in Equation (4.1.1), the situation is rather easy, as only communication with neighbors is necessary to assemble the coarse matrix $E = Z^\dagger AZ$. In case $B = M^{-1}A$, the situation is more complicated, cf. Section 4.5, as the computation of E in this case requires communication not only with direct neighbors but also with neighbors of neighbors. While this might be feasible in the one-dimensional case considered in Section 4.5, in three space dimensions and for a non-regular decomposition of the domain into subdomains, the assembly of the coarse matrix E is significantly more costly and its structure is less sparse. For these reasons, for the 3D experiments we exclusively consider the preconditioner P_B and hence, in the following, only explain how to assemble the matrix $E = Z^\dagger AZ$.

The coarse matrix E defined as $E = Z^\dagger AZ$ has the form $E = (E_{ij})_{i,j=1}^N$, where E_{ij} is a matrix of size $m_i \times m_j$ defined by

$$E_{ij} = \left(R_i^T W_i\right)^\dagger A \left(R_j^T W_j\right) = W_i^\dagger R_i A R_j^T W_j, \quad W_i \in \mathbb{C}^{n \times m_i}.$$

It is easy to see that $E_{ij} = 0$ if $\Omega_j \cap \Omega_i = \emptyset$. Moreover, as $\hat{A}_j = R_j A R_j^T$ is a local matrix, cf. Algorithm 7.1.2, the diagonal blocks E_{jj} can be computed completely locally. Even though the non-zero off-diagonal blocks seemingly require several communication steps, also their computation can be simplified in a way very similar to the trick used for the multiplication with the global stiffness matrix described in Subsection 7.1.1, cf. also [84]. Indeed, due to Assumption 7.1.1, the matrix W_j can be written as $D_j \tilde{W}_j$ for some matrix $\tilde{W}_j$ and with a partition of unity matrix D_j that vanishes on the boundary of Ω_j. As the global stiffness matrix A only couples neighboring degrees of freedom, we get

$$A \left(R_j^T W_j\right) = R_j^T \left(R_j A R_j^T\right) W_j = R_j^T \hat{A}_j W_j.$$

Hence the matrix $A \left(R_j^T W_j\right)$ can be computed locally on subdomain Ω_j except for the extension R_j^T of the local result to the global degrees of freedom. Consequently, the off-diagonal blocks of E can be computed as

$$E_{ij} = W_i^\dagger R_i R_j^T \hat{A}_j W_j,$$

where the only part requiring communication is the application of $R_i R_j^T$, which sends values associated to degrees of freedom in the overlap $\Omega_i \cap \Omega_j$ from the subdomain Ω_j to subdomain Ω_i, cf. Subsection 7.1.1. Summarizing, we have for the blocks of the coarse matrix E

$$E_{ij} = \begin{cases} W_i^\dagger \hat{A}_i W_i & \text{if } i = j, \\ W_i^\dagger R_i R_j^T \hat{A}_j W_j & \text{if } i \neq j \text{ and } \Omega_i \cap \Omega_j \neq \emptyset, \\ 0 & \text{else.} \end{cases}$$

The complete algorithm is given in Algorithm 7.1.3.

Algorithm 7.1.3 Assembly of the coarse matrix $E = Z^\dagger AZ$

Input: On each subdomain Ω_j, $1 \leq j \leq N$: block W_j of the coarse matrix Z, local matrix $\hat{A}_j$, restriction operator R_j

Output: coarse matrix E

▷ Local part

1: **for** $j \leftarrow 1$ to N **do**
2: Compute $T_j := \hat{A}_j W_j$ locally.
3: Compute the block E_{jj} of E by $E_{jj} := W_j^\dagger T_j$ locally.
4: **for** $i \neq j$ such that $\Omega_i \cap \Omega_j \neq \emptyset$ **do**
5: Restrict T_j to $\Omega_i \cap \Omega_j$ and send the values to Ω_i.
6: Receive the values of T_i in $\Omega_i \cap \Omega_j$ from neighbor Ω_i and expand them by zero to Ω_j. This gives the matrix $S_i := R_j R_i^T T_i$.
7: Compute $E_{ji} := W_j^\dagger S_i$.
8: **end for**
9: **end for**

▷ Global part

10: Collect the blocks E_{ij} from the subdomains Ω_j, $1 \leq j \leq N$ and build the global matrix E setting the missing blocks to 0.

7.1.3 Solution of the Dirichlet-to-Neumann eigenvalue problems

For the two-dimensional experiments, we have solved the DtN eigenproblems by using MATLAB's `eig` routine to compute *all* eigenvalues of the DtN operator. In the three-dimensional case, the DtN eigenproblem has significantly more degrees of freedom, as the total number of degrees of freedom as well as the size of the interfaces increase. Computing all the eigenvalues and eigenfunctions is hence prohibitively expensive. Instead, we use the ARPACK package [98] to compute only the eigenfunctions that we are interested in, i.e. those associated to eigenvalues whose real part is smaller than the wave number k.

This is not as straightforward as it might seem. For ARPACK, the total number of eigenvalues and -vectors that shall be computed needs to be specified *a priori*. However, we only know how many eigenvectors we need once we know all the eigenvalues smaller than k. The requirement to look at the right part of the spectrum can be resolved rather easily, as we use ARPACK in the "regular inverse mode", cf. [98, Section 3.9] and look for the eigenvalues with the smallest real part. It is not clear before the computation, how many of these eigenvalues are necessary. We hence start with an estimate $\tilde{n}_{\text{ev}}$ of the number of eigenvalues, and compute them. If the largest one is greater than k, we are done. Otherwise we repeat the *whole* eigenvalue computation aiming for a larger number of eigenvalues. This could probably be improved, e.g. by looking at a different part of the spectrum in subsequent runs.

The problem of guessing the necessary number of eigenvalues is related to the question of choosing the right number of plane waves. The most important difference is that for the DtN coarse

space, we have a reliable criterion for assessing the quality of the guess, see Criterion 5.2.3, whereas for the plane waves the filtering procedure fails to provide this, cf. the discussion in Subsection 6.3.2. Moreover, using a different strategy could reduce the costs for computing additional eigenvalues such that an initially wrong guess does not have that much influence on the runtime. In the current implementation, however, an estimate that is far away from the necessary number can significantly increase the computation time, even though the eigenvalue computation is parallelized.

7.1.4 Other computational details

The code used for the numerical experiments in this chapter is an extension of the FreeFem++ package [75] version 3.26. The additionally necessary features are either implemented inside a FreeFem++ script or in the source code of this package, when no such feature was already available or speed was crucial. Due to larger memory requirements, in the parallel code we use the restarted GMRES version, cf. Algorithm 3.1.2, where the iteration is restarted after $m = 100$ iterations. Moreover, we set the number of maximum outer GMRES cycles to 3, so that at most 300 inner GMRES iterations are performed in total. We write "$>$ 300" for the iteration count if the desired tolerance was not reached in the maximum number of iterations. The decomposition into subdomains is done using Metis [86]. To specify the mesh that was used, the global number n_{glob} of grid points in one coordinate direction is specified. As in Chapter 6, we always use a random starting iterate. Also the two-dimensional experiments in this section are preformed with these settings and the parallel code.

The experiments in this chapter, unless otherwise noted, were done on the CUB cluster at the Università della Svizzera italiana. It consists of 3×14 IBM Blades, each equipped with two quad-core Opteron (Barcelona) processors and 16 GiB memory. Some of the larger experiments were run on the Monte Rosa cluster of the Swiss National Supercomputing Centre. It is a 16 cabinet Cray XE6 system with 1496 compute nodes. Each compute node consists of two 16-core AMD Opteron 6272 2.1 GHz Interlagos processors, giving 32 cores in total per node with 32 GB of memory.

7.2 Numerical results

In this section, we examine the two-level DDM and in particular the coarse space based on the DtN operator for three-dimensional problems, using a parallelized code running on a computer cluster. The implementation has been explained in Section 7.1. As for the two-dimensional numerical experiments in Chapter 6, we first look at homogeneous problems in Subsection 7.2.1 and then also consider some heterogeneous model problems in Subsection 7.2.2.

7.2.1 Homogeneous examples

We investigate the performance of the two-level method for examples with homogeneous wave number k. As a first step, in Table 7.2.1 we compare the performance of the method for the two-dimensional wave guide problem, Problem 2, to that for the corresponding three-dimensional problem, Problem 6. We choose the same number of grid points n per direction, so that the number of degrees of freedom in 3D is roughly n times larger than in 2D. For the three-dimensional case, we

	DtN 3D ($N = 27$)		DtN 2D ($N = 9$)		DtN 2D ($N = 27$)	
k	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$
5	17	(82)	13	(23)	16	(54)
10	11	(306)	9	(49)	10	(111)
15	16	(666)	9	(74)	8	(173)
20	30	(1148)	9	(93)	16	(220)

Table 7.2.1. *Comparison between two-dimensional wave guide problem, Problem 2, and three-dimensional capacitor problem, Problem 6.* $n_{\text{glob}} = 54$ *grid points in each direction. Four nodes on* CUB *for* 3*D experiments, one node for* 2*D experiments.*

	PW(10^{-2}, 26)		PW(10^{-2}, 56)		PW(10^{-2}, 98)		DtN	
k	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$
5	12	(458)	10	(826)	12	(1131)	17	(82)
10	14	(459)	10	(826)	13	(1133)	12	(306)
15	33	(459)	22	(830)	33	(1142)	16	(666)
20	60	(460)	29	(835)	38	(1152)	30	(1148)

Table 7.2.2. *Comparison of DtN coarse space with* PW *one. Problem 6,* $N = 27$ *subdomains, four nodes on* CUB*,* $n_{\text{glob}} = 54$ *grid points in each direction.*

choose $N = 27 = 3^3$ subdomains, and for the two-dimensional case, we investigate both the case with the same number of subdomains per direction, i.e. $N = 9 = 3^2$ subdomains, and with the same total number of subdomains, i.e. $N = 27$. As in Chapter 6, the number in brackets denote the dimension of the coarse space $\mathcal{Z}$. In three space dimensions, it increases at a faster rate. This is not due to the number of subdomains, as the same is true for the two-dimensional case with 27 subdomains instead of 9. At the same time, also the number of iterations increases apparently more rapidly in 3D.

As a next step, examining again Problem 6 with a fixed grid and varying wave number k, we compare the performance of the DtN method to that of the PW one. In Table 7.2.2, the results are shown. As opposed to the two-dimensional case, we here do not observe stagnation due to ill-conditioning for the PW coarse space. However, enlarging the PW coarse space sometimes leads to an increased number of iterations. The PW coarse space size for this example depends almost exclusively on the initially chosen number of coarse modes; the wave number k and filtering hardly influence it. Moreover, only roughly half of the PW coarse functions are not filtered. As for the two-dimensional case, for a similar coarse space size also the iteration counts are comparable for this rather small example. The situation changes for a larger example, for which the results are reported in Table 7.2.3. Here, for larger wave numbers k, the method based on the PW coarse space fails to converge to the desired tolerance within 300 GMRES iterations. Increasing the number of plane

	PW(10^{-2}, 26)		PW(10^{-2}, 56)		PW(10^{-2}, 98)		PW(10^{-2}, 152)		DtN	
k	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$	# it.	$\dim(\mathcal{Z})$
10	15	(2383)	13	(3675)	98	(4666)	> 300	(5601)	30	(854)
20	53	(2386)	27	(3684)	92	(4680)	> 300	(5604)	24	(2889)
30	> 300	(2396)	> 300	(3704)	> 300	(4728)	> 300	(5643)	53	(6110)
40	> 300	(2408)	> 300	(3736)	> 300	(4775)	> 300	(5693)	161	(10533)

Table 7.2.3. *Comparison of DtN coarse space with* PW *one. Problem 6,* 144 *subdomains,* 18 *nodes on* CUB*,* $n_{\text{glob}} = 140$ *grid points in each direction.*

N	# nodes	# iterations	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
8	1	9	(127)	3.26×10^{0}	1.41×10^{2}
16	2	9	(229)	2.24×10^{0}	1.34×10^{2}
32	4	10	(362)	1.51×10^{0}	8.12×10^{1}
64	8	10	(604)	1.40×10^{0}	5.03×10^{1}
128	16	10	(952)	1.64×10^{0}	3.66×10^{1}
256	16	11	(1546)	3.57×10^{0}	7.20×10^{1}

Table 7.2.4. *Strong scaling experiment for DtN coarse space. Problem 6 with* $k = 10$*,* $n_{\text{glob}} = 40$*,* $\tilde{n}_{\text{ev}} = 50$*. For the experiments in the last line, there are less cores than processes.*

waves in this case seems to be of little use to tackle the arising problems. Even though in the last line, the largest PW coarse space has $144 \cdot 152 = 21888$ coarse modes before filtering, the dimension of the coarse space $\mathcal{Z}$ after filtering is only roughly a quarter of the original size. Hence the problem of linear dependence and ill-conditioning seems to be even more serious in the three-dimensional case. The strategy that worked well in two dimensions, namely increasing the number of plane waves in order to get a better coarse space, fails for larger wave numbers in the three-dimensional case.

N	# nodes	# iterations	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
8	1	19	(315)	5.91×10^{1}	2.70×10^{3}
16	2	14	(554)	2.29×10^{1}	1.06×10^{3}
32	4	17	(863)	1.48×10^{1}	4.84×10^{2}
64	8	12	(1321)	7.21×10^{0}	2.01×10^{2}
128	16	14	(2032)	8.28×10^{0}	1.18×10^{2}
256	16	15	(3155)	1.31×10^{1}	1.74×10^{2}

Table 7.2.5. *Second strong scaling experiment for DtN coarse space. Problem 6 with* $k = 16$*,* $n_{\text{glob}} = 60$*,* $\tilde{n}_{\text{ev}} = 100$*. For the experiments in the last line, there are less cores than processes.*

N	# nodes	n_{glob}	# iterations	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
8	1	40	10	(127)	3.64×10^0	2.77×10^2
16	2	50	11	(220)	5.13×10^0	3.38×10^2
32	4	63	12	(351)	6.55×10^0	4.49×10^2
64	8	79	17	(521)	9.09×10^0	4.75×10^2
128	16	100	21	(816)	1.23×10^1	5.50×10^2

Table 7.2.6. *Weak scaling experiments for DtN coarse space with fixed wave number k. Problem 6 with $k = 10$, $\tilde{n}_{\text{ev}} = 100$.*

In Table 7.2.4 and Table 7.2.5, we test the strong scaling of the method for fixed wave number. Here, as in the following tables, T_{solve} denotes the time for the solution of the system after all necessary matrices have been assembled and T_{total} denotes the total time. Strong scaling means that for a fixed problem size, we increase the number of subdomains. The number of processors/nodes used is increased proportionally. For a perfectly scaling problem, doubling the number of subdomains should divide the time necessary for the solution of the same system by two. From the results of Table 7.2.4, it is obvious that the scaling is not perfect. However, the results in Table 7.2.5 show almost perfect scaling both for the solution and for the total time up to 64 subdomains. For both examples, the number of iterations does not or only slightly increase, when the number of subdomains increases, as the dimension of the coarse space $\mathcal{Z}$ increases with the number of subdomains.

Another important parameter for the timing is the number $\tilde{n}_{\text{ev}}$ of eigenvalues that is computed with ARPACK initially, cf. Subsection 7.1.3. Choosing it far from the optimal value can have serious impact on the run time. As the size of the coarse system grows slower than the number of subdomains, keeping $\tilde{n}_{\text{ev}}$ constant during the scaling test results in an increasing number of unnecessary modes computed. This partially explains the performance loss, but is not the only reason, as looking at the solution time T_{solve} shows. Most likely, better results could be achieved with a more optimized implementation. In particular, parallelizing the coarse system as in [84] would be a first step to reduce the influence of the costs of the global component and to obtain better results.

In Table 7.2.6, we examine the weak scaling of the method, again for a fixed wave number k. Here, the problem size is increased proportionally to the number of processors or subdomains, respectively, employed. In the optimal case, the time should stay constant due to more resources being employed to solve the larger problem. However, when increasing the number of subdomains/processors by a factor of 16 from 8 to 128, the solution time T_{solve} increases by a factor of roughly 3.4 and the total time T_{total} by a factor of roughly 2. In Table 7.2.7, the results for larger experiments with more subdomains are given. When the number of subdomains in each coordinate direction is doubled, i.e. the total number of subdomains increases by a factor of 8, we also double the wave number k and the number of grid points in each direction n_{glob}. Similarly to the smaller example, increasing the number of subdomains by a factor of 64 from 8 to 512, the time needed for solution and setup grows. Here, the solution time T_{solve} increases by a factor of 11.0 and the total time T_{total} by a factor of 4.4.

N	# nodes	n_{glob}	# iterations	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
8	1	35	8	(130)	1.87×10^0	1.25×10^2
64	4	70	13	(545)	4.79×10^0	2.61×10^2
512	32	140	45	(1911)	2.06×10^1	5.50×10^2

Table 7.2.7. *Larger weak scaling experiments on Monte Rosa with fixed wave number k. Problem 6 with $k = 10$, $\tilde{n}_{\text{ev}} = 100$.*

N	# nodes	k	n_{glob}	# iterations	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
8	1	10.0	40	9	(127)	3.71×10^0	2.73×10^2
16	2	12.6	50	18	(571)	6.20×10^0	3.62×10^2
32	4	15.9	63	12	(853)	1.09×10^1	4.52×10^2
64	8	20.0	79	20	(1947)	2.44×10^1	5.32×10^2
128	16	25.2	100	30	(4315)	7.19×10^1	6.67×10^2

Table 7.2.8. *Weak scaling experiments with varying wave number k. k increases linearly with the number of grid points per direction. Problem 6 with $k = 16$, $\tilde{n}_{\text{ev}} = 100$.*

In Table 7.2.8, we use the same setting as before, but now vary the wave number proportionally with the grid width. For the experiments in this section, we just keep kh instead of k^3h^2 constant, when we increase the wave number k in order to be able to consider larger wave numbers. This does not account for the pollution effect, cf. Subsection 1.3.3. Here, the solution time increases much more than in Table 7.2.6; when increasing the number of subdomains/processors by a factor of 16 from 8 to 128, the solution time T_{solve} increases by a factor of roughly 19.4. This is not true for the total time T_{total}, which increases only by a factor of about 2.4. We conclude that, not surprisingly, the increasing wave number adds a further difficulty to the problem that has a strong impact on the solution time of the system. It does not affect the total time much, as it is much larger than the solution time – also due to the imperfect implementation.

In Table 7.2.9, we report the results for a larger example with more subdomains and more degrees of freedom. Here, the coarse space is huge for the last line and the method needs the maximum of 300 GMRES iterations in order to converge. Due to the large coarse space, both the setup and the

N	# nodes	k	n_{glob}	# iterations	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
8	1	10.0	35	8	(130)	1.87×10^0	1.25×10^2
64	4	20.0	70	26	(1975)	2.91×10^1	3.14×10^2
512	32	40.0	140	300	(20239)	3.69×10^3	4.40×10^3

Table 7.2.9. *Larger weak scaling on Monte Rosa with varying wave number k. k increases linearly with the number of grid points per direction. $\tilde{n}_{\text{ev}} = 100$.*

ω	n_{glob}	# iterations	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
10	32	11	(616)	1.93×10^0	6.22×10^1
20	64	63	(2208)	1.46×10^2	1.22×10^3
30	96	196	(4607)	3.27×10^3	2.00×10^4

Table 7.2.10. *Heterogeneous layer problem, Problem 7, using a finer grid. $N = 32$ subdomains, four nodes on* CUB*, DtN coarse space with $\tilde{n}_{\text{ev}} = 100$.*

solution takes a lot of time, as the coarse system is solved using the direct solver UMFPACK. This time could probably be reduced using a better, parallel solver for the coarse space.

7.2.2 Heterogeneous examples

In this section, we examine two heterogeneous examples taken from [41, Chapter 7]. We start with Problem 7, which has been defined in Section 1.4. As in [41], for the results in Table 7.2.11, we choose $\omega h = 0.625$. Similar to the homogeneous experiments, the DtN coarse space size grows with increasing wave number. As opposed to the PW coarse space, the method however converges for larger wave numbers, even though the iteration counts increase with the wave number k. The resolution $\omega h = 0.625$ chosen for the previous experiments yields less than 7 points per wavelength in the layer with smallest velocity c for Problem 7. Therefore, in Table 7.2.10, we repeat the experiment with twice as many grid points per direction. However, the low resolution does not seem to be the reason for either the size of the coarse space or the deteriorating convergence, as the finer grid does not resolve any of problems noted above. In Table 7.2.12, finally the results for the wedge example from [41], Problem 8, are shown. In all these experiments, it becomes clear that the method has severe convergence problems for these cases. The problems are probably more severe than in the two-dimensional case, as the effect of the transmission conditions is larger. However, also the plane waves do not offer a better alternative for the heterogeneous examples.

		DtN, $\tilde{n}_{\text{ev}} = 100$				PW(10^{-2}, 56)				PW(10^{-2}, 98)	
ω	n_{glob}	# it.	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s	# it.	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s	# it.	$\dim(\mathcal{Z})$
10	16	9	(369)	7.99×10^{-1}	2.14×10^{1}	17	(500)	1.70×10^{0}	1.25×10^{1}	45	(699)
20	32	35	(1203)	3.02×10^{1}	2.06×10^{2}	68	(522)	1.41×10^{1}	7.81×10^{1}	74	(730)
30	48	39	(2623)	1.57×10^{2}	1.66×10^{3}	> 300	(553)	1.84×10^{2}	3.33×10^{2}	> 300	(766)
40	64	143	(4620)	2.09×10^{3}	1.26×10^{4}	> 300	(577)	4.55×10^{2}	8.39×10^{2}	> 300	(811)

Table 7.2.11. *Heterogeneous layer problem, Problem 7. $N = 16$ subdomains, two nodes on* CUB.

		DtN, $\tilde{n}_{\text{ev}} = 100$				PW(10^{-2}, 56)			
ω	n_{glob}	# it.	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s	# it.	$\dim(\mathcal{Z})$	T_{solve} in s	T_{total} in s
10	16	10	(350)	7.54×10^{-1}	4.02×10^{1}	8	(499)	1.11×10^{0}	2.39×10^{1}
20	32	26	(1138)	2.20×10^{1}	2.69×10^{2}	50	(523)	1.07×10^{1}	1.45×10^{2}
30	48	50	(2448)	1.95×10^{2}	1.97×10^{3}	> 300	(550)	1.91×10^{2}	5.05×10^{2}
40	64	98	(4209)	1.19×10^{3}	1.23×10^{4}	> 300	(573)	4.57×10^{2}	1.21×10^{3}

Table 7.2.12. *Heterogeneous wedge problem, Problem 8. $N = 16$ subdomains, two nodes on* CUB.

7.3 Conclusions

In this chapter, we extended the two-dimensional numerical experiments of Chapter 6 to the three-dimensional case. In order to solve larger problems, the serial code, which was used in Chapter 6 was parallelized. The resulting FreeFem++ code was run on parallel clusters, using the additional computational power and memory to compute larger three-dimensional examples.

The experiments in this chapter have shown that the three-dimensional case is more difficult than the two-dimensional one. This does not only hold true for the implementation, as a much more advanced parallel code was necessary in order to tackle problems with a larger wave number, but applies also to the methods. However, the two-level DDM using the DtN coarse space converged reliably in all the experiments. In contrast to that, for the PW coarse space, the convergence behavior of the method depends critically on the initial number of plane waves chosen. A too small or a too large estimate causes the method to fail to converge in the maximum number of iterations. For examples with a rather large wave number, we even observed that increasing the number of plane waves was not sufficient to improve the convergence rates, as most of the additional waves were filtered out. Consequently, plane waves seem to reach their limit at a point, where the DtN based method still converges in a significantly smaller number of iterations. The better behavior of the DtN coarse space comes at the cost of the additional time spent on the solution of the local DtN eigenvalue problems. This additional cost might be decreased by improving our naive implementation of the eigenvalue solver. As for the two-dimensional examples in Chapter 6, *if* the PW method converges, the iteration counts for the two coarse spaces are comparable for the homogeneous examples and better for the DtN coarse space for the heterogeneous examples.

While the DtN coarse space thus overcomes some of the problems of its competitor, two points that could be further improved became clear in this chapter. First, with increasing wave number k the dimension of the coarse space grows significantly. Even though the global system is not dense, but is composed of blocks associated to overlapping subdomains, its solution with a direct method becomes expensive. Therefore, an important next step to improve the scalability of the method would be the parallelization of the coarse system, cf. [84].

Second, the number of iterations increases with the wave number k and the number of subdomains *despite* the fact that a larger coarse space is used. Consequently, either Criterion 5.2.3 does not select enough modes or the DtN eigenvectors do not include all the modes that are necessary. However, it is debatable whether further enriching and hence enlarging the coarse space is the best approach. As Fourier analysis in Section 3.3 showed, the transmission conditions used in our experiments suffer from problems when the number of subdomains increases. In the current method, the coarse space thus does not only provide a global component, but also has to tackle all the additional problems that arise for higher frequencies and more subdomains. In our opinion, at least partially removing these problems with the transmission conditions leads to better convergence rates without the costs of an even larger global problem. Optimized Schwarz methods [57] might for example be a good option, even though to our knowledge they have mainly been investigated for model problems with only a few subdomains.

Discussion and conclusions

For large wave numbers, FE discretizations of the Helmholtz equation lead to very large, sparse, non-Hermitian, highly indefinite, and ill-conditioned systems of linear equations. These properties represent a major challenge for its numerical solution. While standard direct methods in two space dimensions might be a feasible option, their use for three-dimensional problems at high frequencies is prohibitive, as memory requirements increase substantially. Standard iterative methods, on the other hand, suffer from slow convergence or even divergence.

In this thesis, we introduced and tested a two-level DDM especially tailored for the iterative solution of the heterogeneous Helmholtz equation. The main new ingredient is the coarse space, whose construction is based on local eigenproblems involving the DtN operator. Our coarse space inherently respects variations in the wave number, making it possible to treat heterogeneous Helmholtz problems. Moreover, it does not suffer from ill-conditioning, in contrast to the standard approach based on plane waves. This is important, as ill-conditioning can cause the iterative solver to stagnate. The resulting method has been tested successfully on two- and three-dimensional problems.

We tackled different aspects of the problem of constructing a two-level method for the difficult Helmholtz problem. In a first step, we investigated different ways to incorporate a second level into a one-level method. Here, the fact that the Helmholtz system is indefinite and non-Hermitian causes difficulties. We examined two desirable properties: The two-level preconditioner should be non-singular and it should never be worse than the one-level preconditioner.

The non-singularity is important to make the preconditioned system uniquely solvable and to guarantee convergence of the GMRES method. While we showed that the preconditioner based on the balancing method [42] leads to a singular, underdetermined system, the preconditioner proposed in [74] is provably invertible. This comes at the cost of a more difficult to assemble in parallel and more densely populated coarse system. The question whether it is possible to define a non-singular variation of the balancing preconditioner while preserving the nice properties of the resulting coarse system remained open. However, our numerical experiments showed that both preconditioners perform reasonably well in practice despite the possible singularity of the balancing preconditioner.

The other important question that we examined in this context is under which conditions adding a second level to the one-level preconditioner can be guaranteed not to deteriorate the convergence rates. Our analysis showed that linear combinations of eigenvectors associated to eigenvalues with different signs cause problems in the coarse space if it is incomplete, possibly making convergence rates worse than those for the one-level method. As global spectral information in practice is too

costly to obtain, this theoretical insight only partially helps when defining the coarse space. Being aware of this problem is however a key to understand the behavior of the method.

In a second step, we introduced the DtN coarse space and motivated it with several numerical tests. The construction has two parts. While the most important step is the choice of the right functions on the interfaces of the subdomains, choosing the correct extension operator to the interior of the subdomains is also crucial in order to get good convergence results. We discussed several alternatives and concluded that extending with the original Helmholtz operator gives the best results. The question how to define the two-level preconditioner in such a way that it always improves convergence compared to the one-level method remained open for general matrices. However, for the concrete DtN coarse space, we provided a criterion to choose the necessary coarse space modes such that convergence problems did not occur for our test cases. Our criterion ensures that all important modes are present, refraining from the need of tuning the dimension manually and guaranteeing good convergence rates.

In a last step, we tested the resulting method for homogeneous and heterogeneous problems in two and three space dimensions. In order to assess the quality of the coarse space, we adapted the popular plane wave coarse space to our setting. Here again the question how to choose the extension operator is crucial. Our experiments showed that pointwise evaluation of the plane waves, as proposed for example in [92], at least in our setting leads to worse convergence and to more conditioning problems than using the same extension as for the DtN coarse space.

In the two-dimensional experiments, the DtN and the PW coarse spaces performed similarly for homogeneous problems if they had approximately the same size. The main advantage of the DtN coarse space for these problems is the fully automatic construction without the need to tune a critical parameter, the coarse space size. Moreover, the DtN coarse space does not suffer from the serious conditioning problems the PW coarse space has and converges reliably. Adding to these advantages, the DtN coarse space performs better for heterogeneous problems.

Three-dimensional problems are even more difficult to solve numerically. In order to be able to examine also larger examples, we implemented the method in parallel for the 3D tests. The method employing the DtN coarse space converged reliably in all the experiments. In contrast to that, plane waves were not able to guarantee convergence. In addition to the issues already observed for the two-dimensional case, here for some examples with larger wave number a further problem arose. Even increasing the number of plane waves was not sufficient to provide an iteration count comparable to the one with DtN coarse space. The reason for this was probably that most of the additional plane waves were filtered out. As for the two-dimensional examples, *if* the PW method converged, the iteration counts for the two coarse spaces were comparable for the homogeneous examples and better for the DtN coarse space for the heterogeneous examples. Hence, even though its performance gets worse for larger wave numbers, the DtN based method converges well compared to its competitor also for three-dimensional examples and overcomes many of the problems from which the plane wave based approach suffers.

While we have thus introduced and successfully tested a new two-level DDM for the difficult heterogeneous Helmholtz equation, some problems remain unsolved. In the three-dimensional case, while our coarse space performs better and more reliably than the one based on plane waves, the

size and the costs of the global problem grow with the wave number k. On the one hand, this means that more eigenvectors of the DtN eigenvalue problems on the subdomains need to be computed. Even though this computation is completely parallel, the runtime of the algorithm could benefit from improving the strategies to solve the local eigenproblems. On the other hand, in the current implementation, the coarse system is assembled and solved by a direct method on one processor only. At some point we reach the memory limits either of this processor or of the employed direct solver. The parallelization of the coarse system and the application of iterative solution techniques are hence important next steps.

It is known that the first order approximation of the Sommerfeld radiation condition, which we have used in this thesis, does not provide the best transmission conditions for Helmholtz problems. Consequently, in our method, almost all the responsibility for handling the difficulties due to an increasing wave number or an increasing number of subdomains lies with the coarse space. To lighten the burden on the coarse space, it would be beneficial to also examine different transmission conditions. By that, better results could probably be achieved without an exploding global problem size. Simultaneously, one should also try to improve the coarse space by adapting it to the different transmission conditions.

While we have done extensive numerical experiments in order to test our approach, a theoretical foundation is missing. A convergence theory of two-level DDMs for the Helmholtz equation, and in particular of the method that we proposed, would be an important step in order to improve their design. However, despite some attempts [101], such a theory is still in its infancy. Especially interesting for the work in this thesis would be understanding rigorously why the DtN eigenfunctions are good coarse space functions and how one could possibly improve them.

Glossary

Symbols

$\mathbb{R}$	field of real numbers. 3
$\mathbb{C}$	field of complex numbers. 3
$\mathbb{N}$	field of natural numbers. 3
$\mathbb{N}_0$	field of natural numbers plus zero. 3
$\lvert\alpha\rvert$	absolute value of α in $\mathbb{R}$ or $\mathbb{C}$. 3
$\overline{\alpha}$	conjugation of α in $\mathbb{R}$ or $\mathbb{C}$. 3
δ_{ij}	Kronecker delta. 5
$O(.)$	big-O notation. 5
$o(.)$	little-o notation. 5
$\lambda_{\min}(A)$	smallest (non-zero) eigenvalue of a matrix A by modulus. 5
$\lambda_{\max}(A)$	biggest eigenvalue of a matrix A by modulus. 5
$\kappa(A)$	condition number of a matrix A. 5
$\mathring{A}$	interior of a set A. 3
$\overline{A}$	closure of a set A. 3
∂A	boundary of a set A. 3
$\langle .,. \rangle$	Euclidean scalar product in $\mathbb{R}^n$ or $\mathbb{C}^n$. 3
supp(f)	support of a function f. 27
$\partial_i f(x)$ or $\frac{\partial}{\partial x_i} f(x)$	partial derivative of f in direction of the i-th unit vector. 3
$\partial_v f$ or $\frac{\partial}{\partial v} f$	directional derivative of f in direction $v \in \mathbb{R}^n$. 3
$C^m\left(\overline{\Omega}, Y\right)$ or $C^m\left(\overline{\Omega}\right)$	space of m times continuously differentiable functions. 3
$C^\infty(\Omega)$	space of inifinitely continuously differentiable functions. 4
$L^p(\Omega)$	Lebesgue space of order p. 4
$L^\infty(\Omega)$	Lebesgue space of essentially bounded, measurable functions. 4
$H^{m,p}(\Omega)$	Sobolev space of m times weakly differentiable functions in L^p. 4
$\lVert f \rVert_{H^{m,p}(\Omega)}$	norm of the function f in the Sobolev space $H^{m,p}(\Omega)$. 4
$H^m(\Omega)$	$= H^{m,2}(\Omega)$: Sobolev space of m times weakly differentiable functions in $L^2(\Omega)$. 4

$(.,.)_{H^m(\Omega)}$	inner product on $H^m(\Omega)$. 4
$H_0^{m,p}(\Omega)$	Sobolev space of functions in $H^{m,p}(\Omega)$ with zero boundary values. 4
k	wave number. 5
ω	angular frequency. 6
c	wave speed. 6
n	number of degrees of freedom in Ω. 27
n_j	number of degrees of freedom in subdomain Ω_j. 27
R_j	$\in \mathbb{R}^{n_j \times n}$, restriction matrix from Ω to Ω_j. 29
D_j	$\in \mathbb{R}^{n_j \times n_j}$, diagonal matrix corresponding to a partition of unity function on Ω_j. 29
A_j	$\in \mathbb{R}^{n_j \times n_j}$, local stiffness matrix on Ω_j. 29
M^{-1}	one-level RAS preconditioner. 29
m_j	number of coarse modes on subdomain Ω_j. 61
Z	matrix whose columns span the coarse space. 62

Acronyms

BNN	balancing Neumann-Neumann. 45
CG	conjugate gradient. 24
DtN	Dirichlet-to-Neumann. 2, 19
DDM	domain decomposition method. 2, 15
FE	finite element. 1, 3
FETI	finite element tearing and interconnecting. 2, 21
FETI-DP	dual-primal FETI. 21
FETI-H	FETI for Helmholtz. 21
FETI-DPH	FETI-DP for Helmholtz. 21
GMRES	generalized minimal residual. 16
PDE	partial differential equation. 5
PML	perfectly matched layer. 2, 7
PW	plane wave. 74
RAS	restricted additive Schwarz. 2, 23
s.p.d.	symmetric positive definite. 2, 15
w.l.o.g.	without loss of generality. 43

Bibliography

[1] T. Airaksinen, E. Heikkola, A. Pennanen, and J. Toivanen. An algebraic multigrid based shifted-Laplacian preconditioner for the Helmholtz equation. *J. Comput. Phys.*, 226(1): 1196–1210, 2007.

[2] H. W. Alt. *Lineare Funktionalanalysis: Eine anwendungsorientierte Einführung*. Springer Lehrbuch. Springer, 2002.

[3] M. Amara, R. Djellouli, and C. Farhat. Convergence analysis of a discontinuous Galerkin method with plane waves and Lagrange multipliers for the solution of Helmholtz problems. *SIAM J. Numer. Anal.*, 47(2):1038–1066, 2009.

[4] R. Aubry, S. Dey, and R. Löhner. Iterative solution applied to the Helmholtz equation: Complex deflation on unstructured grids. *Comput. Methods Appl. Mech. Engrg.*, 2012.

[5] I. Babuška, F. Ihlenburg, E. T. Paik, and S. A. Sauter. A generalized finite element method for solving the Helmholtz equation in two dimensions with minimal pollution. *Comput. Methods Appl. Mech. Engrg.*, 128(3-4):325–359, 1995.

[6] I. Babuška, F. Ihlenburg, T. Strouboulis, and S. Gangaraj. A posteriori error estimation for finite element solutions of Helmholtz' equation. Part I: The quality of local indicators and estimators. *Internat. J. Numer. Methods Engrg.*, 40(18):3443–3462, 1997.

[7] I. Babuška, F. Ihlenburg, T. Strouboulis, and S. Gangaraj. A posteriori error estimation for finite element solutions of Helmholtz' equation. Part II: estimation of the pollution error. *Internat. J. Numer. Methods Engrg.*, 40(21):3883–3900, 1997.

[8] I. M. Babuška and S. A. Sauter. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Rev.*, 42(3):451–484, 2000.

[9] A. Bayliss and E. Turkel. Radiation boundary conditions for wave-like equations. *Comm. Pure Appl. Math.*, 33(6):707–725, 1980.

[10] A. Bayliss, C. I. Goldstein, and E. Turkel. An iterative method for the Helmholtz equation. *J. Comput. Phys.*, 49(3):443–457, 1983.

[11] M. Bebendorf. *Hierarchical matrices*. Springer, 2008.

[12] J. P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.

[13] M. Bollhöfer, M. J. Grote, and O. Schenk. Algebraic multilevel preconditioner for the Helmholtz equation in heterogeneous media. *SIAM J. Sci. Comput.*, 31(5):3781–3805, 2009.

[14] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Eng. Anal. Bound. Elem.*, 27(5):405–422, 2003.

[15] Y. Boubendir, X. Antoine, and C. Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. *J. Comput. Phys.*, 231(2):262 – 280, 2012.

[16] D. Braess. *Finite elements*. Cambridge University Press, 2007.

[17] A. Brandt and S. Ta'asan. Multigrid method for nearly singular and slightly indefinite problems. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods II*, volume 1228 of *Lecture Notes in Mathematics*, pages 99–121. Springer Berlin Heidelberg, 1986.

[18] A. Brandt and I. Livshits. Wave-ray multigrid method for standing wave equations. *Electron. Trans. Numer. Anal.*, 6(162-181):91, 1997.

[19] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer, 2008.

[20] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial: Second Edition*. Society for Industrial and Applied Mathematics, 2000.

[21] A. Buffa and P. Monk. Error estimates for the ultra weak variational formulation of the Helmholtz equation. *Math. Model. Numer. Anal.*, 42(06):925–940, 2008.

[22] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21(2):792–797 (electronic), 1999.

[23] X.-C. Cai, M. A. Casarin, F. W. Elliott Jr., and O. B. Widlund. Overlapping Schwarz algorithms for solving Helmholtz's equation. *Contemp. Math.*, 218:391–399, 1998.

[24] H. Calandra, S. Gratton, X. Pinel, and X. Vasseur. An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media. *Numer. Linear Algebra Appl.*, 20(4):663–688, 2013.

[25] O. Cessenat and B. Desprès. Application of an ultra weak variational formulation of elliptic PDEs to the two-dimensional Helmholtz problem. *SIAM J. Numer. Anal.*, 35(1):255–299, 1998.

[26] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 40. Cambridge University Press, 2002.

[27] F. Collino, S. Ghanemi, and P. Joly. Domain decomposition method for harmonic wave propagation: a general presentation. *Comput. Methods Appl. Mech. Engrg.*, 184(2–4):171 – 211, 2000.

[28] L. Conen, V. Dolean, R. Krause, and F. Nataf. A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator. *J. Comput. Appl. Math.*, 271(0):83 – 99, 2014.

[29] S. Cools and W. Vanroose. Local Fourier analysis of the complex shifted Laplacian preconditioner for Helmholtz problems. *Numer. Linear Algebra Appl.*, 20(4):575–597, 2013.

[30] T. A. Davis. Multifrontal multithreaded rank-revealing sparse QR factorization. *University of Florida, Tech. Rep*, 2009.

[31] A. Deraemaeker, I. Babuška, and P. Bouillard. Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions. *Internat. J. Numer. Methods Engrg.*, 46(4):471–499, 1999.

[32] B. Després. *Méthodes de décomposition de domaine pour les problèmes de propagation d'ondes en régime harmonique*. PhD thesis, Université de Paris IX (Dauphine), Paris, 1991.

[33] V. Dolean, F. Nataf, R. Scheichl, and N. Spillane. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet–to–Neumann maps. *Comput. Methods Appl. Math.*, 12(4):391–414, 2012.

[34] O. Dubois and M. J. Gander. Convergence behavior of a two-level optimized Schwarz preconditioner. *Domain Decomposition Methods in Science and Engineering XVIII*, pages 177–184, 2009.

[35] O. Dubois and M. J. Gander. Convergence behavior of a two-level optimized Schwarz preconditioner. In *Domain Decomposition Methods in Science and Engineering XVIII*, volume 70 of *Lecture Notes in Computational Science and Engineering*, pages 177–184. Springer Berlin Heidelberg, 2009.

[36] O. Dubois, M. J. Gander, S. Loisel, A. St-Cyr, and D. B. Szyld. The optimized Schwarz method with a coarse grid correction. *SIAM J. Sci. Comput.*, 34(1):A421–A458, 2012.

[37] H. C. Elman, O. G. Ernst, and D. P. O'Leary. A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. *SIAM J. Sci. Comput.*, 23(4):1291–1315, 2002.

[38] B. Engquist and A. Majda. Radiation boundary conditions for acoustic and elastic wave calculations. *Comm. Pure Appl. Math.*, 32(3):313–357, 1979.

[39] B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: hierarchical matrix representation. *Comm. Pure Appl. Math.*, 64(5):697–735, 2011.

[40] B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. *Multiscale Model. Simul.*, 9(2):686–710, 2011.

[41] Y. A. Erlangga. *A robust and efficient iterative method for the numerical solution of the Helmholtz equation*. PhD thesis, Technische Universiteit Delf, 2005.

[42] Y. A. Erlangga and R. Nabben. Deflation and balancing preconditioners for Krylov subspace methods applied to nonsymmetric matrices. *SIAM J. Matrix Anal. Appl.*, 30(2):684–699, 2008.

[43] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math.*, 50(3-4):409–425, 2004.

[44] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.*, 27(4):1471–1492, 2006.

[45] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee. Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Appl. Numer. Math.*, 56(5):648–666, 2006.

[46] O. G. Ernst and M. J. Gander. Why it is difficult to solve Helmholtz problems with classical iterative methods. In I. Graham, T. Hou, O. Lakkis, and R. Scheichl, editors, *Numerical Analysis of Multiscale Problems*, volume 83 of *Lecture Notes in Computational Science and Engineering*, pages 325–363. Springer Berlin Heidelberg, 2012.

[47] O. G. Ernst and M. J. Gander. Multigrid methods for Helmholtz problems: A convergent scheme in 1D using standard components. In *Proceedings of the RICAM workshop on Wave Propagation, in print*, 2013.

[48] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Internat. J. Numer. Methods Engrg.*, 32(6):1205–1227, 1991.

[49] C. Farhat, A. Macedo, and M. Lesoinne. A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. *Numer. Math.*, 85(2): 283–308, 2000.

[50] C. Farhat, I. Harari, and L. P. Franca. The discontinuous enrichment method. *Comput. Methods Appl. Mech. Engrg.*, 190(48):6455–6479, 2001.

[51] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual–primal unified FETI method - part I: A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7):1523–1544, 2001.

[52] C. Farhat, P. Avery, R. Tezaur, and L. Jing. FETI-DPH: a dual-primal domain decomposition method for acoustic scattering. *J. Comput. Acoust.*, 13(3):499–524, 2005.

[53] J. Fish and Y. Qu. Global-basis two-level method for indefinite systems. Part 1: convergence studies. *Internat. J. Numer. Methods Engrg.*, 49(3):439–460, 2000.

[54] L. V. Foster and T. A. Davis. Algorithm 933: Reliable calculation of numerical rank, null space bases, pseudoinverse solutions, and basic solutions using SuiteSparseQR. *ACM Trans. Math. Softw.*, 40(1):7:1–7:23, October 2013.

[55] L. P. Franca and A. P. Macedo. A two-level finite element method and its application to the Helmholtz equation. *Internat. J. Numer. Methods Engrg.*, 43(1):23–32, 1998.

[56] J. Galvis and Y. Efendiev. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Model. Simul.*, 8(4):1461–1483, 2010.

[57] M. J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

[58] M. J. Gander and F. Nataf. AILU for Helmholtz problems: a new preconditioner based on the analytic parabolic factorization. *J. Comput. Acoust.*, 9(04):1499–1506, 2001.

[59] M. J. Gander and F. Nataf. An incomplete LU preconditioner for problems in acoustics. *J. Comput. Acoust.*, 13(03):455–476, 2005.

[60] M. J. Gander and H. Zhang. Domain decomposition methods for the Helmholtz equation: a numerical investigation. In *Domain Decomposition Methods in Science and Engineering XX*, Lecture Notes in Computational Science and Engineering, San Diego, 2012. Springer.

[61] M. J. Gander and H. Zhang. Optimized Schwarz methods with overlap for the Helmholtz equation. In J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, editors, *Domain Decomposition Methods in Science and Engineering XXI*, volume 98 of *Lecture Notes in Computational Science and Engineering*, pages 207–215. Springer International Publishing, 2014.

[62] M. J. Gander, L. Halpern, and F. Nataf. Optimized Schwarz methods. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan*, pages 15–28, Bergen, 2001. Domain Decomposition Press.

[63] M. J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.

[64] M. J. Gander, L. Halpern, and F. Magoulès. An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. *Internat. J. Numer. Methods Fluids*, 55(2):163–175, 2007.

[65] M. J. Gander, I. G. Graham, and E. A. Spence. How should one choose the shift for the Laplacian to be a good preconditioner for the Helmholtz equation? Preprint, January 2014.

[66] M. B. V. Gijzen, Y. A. Erlangga, and C. Vuik. Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian. *SIAM J. Sci. Comput.*, 2007.

[67] D. Givoli. Non-reflecting boundary conditions. *J. Comput. Phys.*, 94(1):1 – 29, 1991.

[68] D. Givoli. High-order local non-reflecting boundary conditions: a review. *Wave Motion*, 39 (4):319 – 326, 2004.

[69] W. Hackbusch. A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. *Computing*, 62(2):89–108, 1999.

[70] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer Series in Computational Mathematics. Springer, 2003.

[71] I. Harari. A survey of finite element methods for time-harmonic acoustics. *Comput. Methods Appl. Mech. Engrg.*, 195(13-16):1594–1607, 2006.

[72] I. Harari and T. J. Hughes. Finite element methods for the Helmholtz equation in an exterior domain: Model problems. *Comput. Methods Appl. Mech. Engrg.*, 87(1):59–96, 1991.

[73] I. Harari and E. Turkel. Accurate finite difference methods for time-harmonic wave propagation. *J. Comput. Phys.*, 119(2):252–270, 1995.

[74] P. Havé, R. Masson, F. Nataf, M. Szydlarski, H. Xiang, and T. Zhao. Algebraic domain decomposition methods for highly heterogeneous problems. *SIAM J. Sci. Comput.*, 35(3): C284–C302, 2013.

[75] F. Hecht, O. Pironneau, A. Le Hyaric, and K. Ohtsuka. *FreeFem++*. Université Pierre et Marie Curie, 2007. `http://www.freefem.org/ff++/ftp/freefem++doc.pdf`.

[76] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

[77] R. Hiptmair, A. Moiola, and I. Perugia. Plane wave discontinuous Galerkin methods for the 2D Helmholtz equation: analysis of the p-version. *SIAM J. Numer. Anal.*, 49(1):264–284, 2011.

[78] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1990.

[79] F. Ihlenburg. *Finite element analysis of acoustic scattering*, volume 132. Springer, 1998.

[80] F. Ihlenburg and I. Babuška. Finite element solution of the Helmholtz equation with high wave number. Part I: The h-version of the FEM. *Computers Mathematics Appl.*, 30(9):9–37, 1995.

[81] F. Ihlenburg and I. Babuška. Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation. *Internat. J. Numer. Methods Engrg.*, 38(22): 3745–3774, 1995.

[82] F. Ihlenburg and I. Babuška. Finite element solution of the Helmholtz equation with high wave number Part II: The h-p version of the FEM. *SIAM J. Numer. Anal.*, 34:315–358, 1997.

[83] S. G. Johnson. Notes on perfectly matched layers (PMLs). *Lecture notes, Massachusetts Institute of Technology, Massachusetts*, 2008.

[84] P. Jolivet, F. Hecht, F. Nataf, and C. Prud'homme. Scalable domain decomposition preconditioners for heterogeneous elliptic problems. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 80:1–80:11, New York, NY, USA, 2013. ACM.

[85] T. B. Jönsthövel, M. B. Van Gijzen, C. Vuik, and A. Scarpas. On the use of rigid body modes in the deflated preconditioned conjugate gradient method. *SIAM J. Sci. Comput.*, 35(1): B207–B225, 2013.

[86] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[87] R. Kechroud, A. Soulaimani, Y. Saad, and S. Gowda. Preconditioning techniques for the solution of the Helmholtz equation by the finite element method. *Math. Comput. Simulation*, 65(4-5):303 – 321, 2004. Wave Phenomena in Physics and Engineering: New Models, Algorithms, and Applications.

[88] J. B. Keller and D. Givoli. Exact non-reflecting boundary conditions. *J. Comput. Phys.*, 82(1): 172–192, 1989.

[89] C. T. Kelley. *Iterative methods for linear and nonlinear equations*. Society for Industrial Mathematics, 1995.

[90] S. Kim and S. Kim. Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media. *SIAM J. Sci. Comput.*, 24(2):684–701, 2002.

[91] J.-H. Kimn and M. Sarkis. OBDD: Overlapping balancing domain decomposition methods and generalizations to the Helmholtz equation. *Domain Decomposition Methods in Science and Engineering XVI*, pages 317–324, 2007.

[92] J.-H. Kimn and M. Sarkis. Restricted overlapping balancing domain decomposition methods and restricted coarse problems for the Helmholtz problem. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1507–1514, 2007.

[93] J.-H. Kimn and M. Sarkis. Shifted Laplacian RAS solvers for the Helmholtz equation. In *Proceedings of the 20th International Conference on Domain Decomposition Methods, San Diego, USA*, pages 2047–2078, 2011.

[94] A. Klawonn and O. Rheinbach. Inexact FETI-DP methods. *Internat. J. Numer. Methods Engrg.*, 69(2):284–307, 2007.

[95] O. Laghrouche, P. Bettess, and R. J. Astley. Modelling of short wave diffraction problems using approximating systems of plane waves. *Internat. J. Numer. Methods Engrg.*, 54(10): 1501–1533, 2002.

[96] A. L. Laird and M. Giles. Preconditioned iterative solution of the 2D Helmholtz equation. Report NA 02-12, Comp. Lab, 2002.

[97] B. Lee, T. Manteuffel, S. McCormick, and J. Ruge. First-order system least-squares for the Helmholtz equation. *SIAM J. Sci. Comput.*, 21(5):1927–1949, 2000.

[98] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. SIAM, 1998.

[99] S. K. Lele. Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.*, 103(1):16–42, 1992.

[100] A. Leong. Extension of two-level Schwarz preconditioners to symmetric indefinite problems. Master's thesis, New York University, Courant Institute of Mathematical Sciences, 2008.

[101] J. Li and X. Tu. Convergence analysis of a balancing domain decomposition method for solving a class of indefinite linear systems. *Numer. Linear Algebra Appl.*, 16(9):745–773, 2009.

[102] I. Livshits. A scalable multigrid method for solving indefinite Helmholtz equations with constant wave numbers. *Numer. Linear Algebra Appl.*, 21(2):177–193, 2014.

[103] I. Livshits. An algebraic multigrid wave-ray algorithm to solve eigenvalue problems for the Helmholtz operator. *Numer. Linear Algebra Appl.*, 11(2-3):229–239, 2004.

[104] R. E. Lynch and J. R. Rice. A high-order difference method for differential equations. *Math. Comp.*, 34(150):333–372, 1980.

[105] M. Magolu Monga Made. Incomplete factorization-based preconditionings for solving the Helmholtz equation. *Internat. J. Numer. Methods Engrg.*, 50(5):1077–1101, 2001.

[106] J. Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3):233–241, 1993.

[107] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10(7):639–659, 2003.

[108] Marmousi problem. Homepage of the "Workshop on computation of multi-valued traveltimes", September 1996. URL `http://www.caam.rice.edu/~benamou/traveltimes.html`.

[109] J. M. Melenk and S. A. Sauter. Wavenumber explicit convergence analysis for Galerkin discretizations of the Helmholtz equation. *SIAM J. Numer. Anal.*, 49(3):1210–1243, 2011.

[110] J. M. Melenk and I. Babuška. The partition of unity finite element method: basic theory and applications. *Comput. Methods Appl. Mech. Engrg.*, 139(1):289–314, 1996.

[111] A. Moiola and E. A. Spence. Is the Helmholtz equation really sign-indefinite? *SIAM Rev.*, 56 (2):274–312, 2014.

[112] R. Nabben and C. Vuik. A comparison of deflation and coarse grid correction applied to porous media flow. *SIAM J. Numer. Anal.*, pages 1631–1647, 2005.

[113] R. Nabben and C. Vuik. A comparison of deflation and the balancing preconditioner. *SIAM J. Sci. Comput.*, 27(5):1742–1759, 2006.

[114] R. Nabben and C. Vuik. A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners. *Numer. Linear Algebra Appl.*, 15(4):355–372, 2008.

[115] R. Nabben, J. Tang, and C. Vuik. Deflation acceleration for domain decomposition preconditioners. In P. Wesseling, C. Oosterlee, and P. Hemker, editors, *Proceedings of the 8th European Multigrid Conference September 27-30, (2005) Scheveningen The Hague, The Netherlands*, Delft, 2006.

[116] F. Nataf, H. Xiang, V. Dolean, and N. Spillane. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.*, 33(4):1623–1642, 2011.

[117] Y. Notay. Aggregation-based algebraic multilevel preconditioning. *SIAM J. Matrix Anal. Appl.*, 27(4):998–1018, 2006.

[118] Y. Notay. An aggregation-based algebraic multigrid method. *Electron. Trans. Numer. Anal.*, 37:123–146, 2010.

[119] L. N. Olson and J. B. Schroder. Smoothed aggregation for Helmholtz problems. *Numer. Linear Algebra Appl.*, 17(2-3):361–386, 2010.

[120] D. Osei-Kuffuor and Y. Saad. Preconditioning Helmholtz linear systems. *Appl. Numer. Math.*, 60(4):420–431, 2010.

[121] R. E. Plessix and W. A. Mulder. Separation-of-variables as a preconditioner for an iterative Helmholtz solver. *Appl. Numer. Math.*, 44(3):385–400, 2003.

[122] J. Poulson, B. Engquist, S. Li, and L. Ying. A parallel sweeping preconditioner for heterogeneous 3D Helmholtz equations. *SIAM J. Sci. Comput.*, 35(3):C194–C212, 2013.

[123] J. L. Poulson. *Fast Parallel Solution of Heterogeneous 3D Time-harmonic Wave Equations*. PhD thesis, University of Texas at Austin, December 2012.

[124] Y. Qu and J. Fish. Global-basis two-level method for indefinite systems. Part 2: Computational issues. *Internat. J. Numer. Methods Engrg.*, 49(3):461–478, 2000.

[125] Y. Qu and J. Fish. Multifrontal incomplete factorization for indefinite and complex symmetric systems. *Internat. J. Numer. Methods Engrg.*, 53(6):1433–1459, 2002.

[126] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, USA, 1999.

[127] A. Quarteroni, A. M. Quarteroni, and A. Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Verlag, 2008.

[128] L. Reichel and Q. Ye. Breakdown-free GMRES for singular systems. *SIAM J. Matrix Anal. Appl.*, 26(4):1001–1021, 2005.

[129] C. D. Riyanti, A. Kononov, Y. A. Erlangga, C. Vuik, C. W. Oosterlee, R. E. Plessix, and W. A. Mulder. A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. *Journal of Computational physics*, 224(1):431–448, 2007.

[130] V. S. Ryaben'kii and S. V. Tsynkov. Artificial boundary conditions for the numerical solution of external viscous flow problems. *SIAM J. Numer. Anal.*, 32:1355–1389, October 1995.

[131] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. and Stat. Comput.*, 7(3):856–869, 1986.

[132] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2003.

[133] S. Sauter and C. Schwab. *Randelementmethoden: Analyse, Numerik und Implementierung schneller Algorithmen*. Springer DE, 2004.

[134] A. Schädle and L. Zschiedrich. Additive Schwarz method for scattering problems using the PML method at interfaces. *Domain Decomposition Methods in Science and Engineering XVI*, pages 205–212, 2007.

[135] A. H. Sheikh, D. Lahaye, and C. Vuik. On the convergence of shifted Laplace preconditioner combined with multilevel deflation. *Numerical Linear Algebra with Applications*, 20(4): 645–662, 2013.

[136] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[137] I. Singer and E. Turkel. High-order finite difference methods for the Helmholtz equation. *Comput. Methods Appl. Mech. Engrg.*, 163(1):343–358, 1998.

[138] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 2004.

[139] A. Sommerfeld. *Partial differential equations in physics*. Academic Press, 1964.

[140] A. St-Cyr, M. J. Gander, and S. J. Thomas. Optimized multiplicative, additive, and restricted additive Schwarz preconditioning. *SIAM J. Sci. Comput.*, 29(6), 2008.

[141] O. Steinbach. *Numerical approximation methods for elliptic boundary value problems: Finite and boundary elements*, volume 99. Springerverlag New York, 2008.

[142] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, Second Edition*. Society for Industrial and Applied Mathematics, 2004.

[143] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *J. Sci. Comput.*, 39(3): 340–370, 2009.

[144] R. Tezaur and C. Farhat. Three-dimensional discontinuous Galerkin elements with plane waves and Lagrange multipliers for the solution of mid-frequency Helmholtz problems. *Internat. J. Numer. Methods Engrg.*, 66(5):796–815, 2006.

[145] J. W. Thomas. *Numerical partial differential equations: finite difference methods*, volume 1. Springer Verlag, 1995.

[146] A. Toselli. Some results on overlapping Schwarz methods for the Helmholtz equation employing perfectly matched layers. Technical Report 765, Courant Institute of Mathematical Sciences, New York University, New York, 1998.

[147] A. Toselli and O. B. Widlund. *Domain decomposition methods–algorithms and theory*. Springer Verlag, 2005.

[148] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

[149] S. V. Tsynkov, E. Turkel, and S. Abarbanel. External flow computations using global boundary conditions. *AIAA journal*, 34(4):700–706, 1996.

[150] N. Umetani, S. P. MacLachlan, and C. W. Oosterlee. A multigrid-based shifted Laplacian preconditioner for a fourth-order Helmholtz discretization. *Numer. Linear Algebra Appl.*, 16 (8):603–626, 2009.

[151] O. B. Widlund. The development of coarse spaces for domain decomposition algorithms. *Domain Decomposition Methods in Science and Engineering XVIII*, pages 241–248, 2009.

[152] L. C. Wrobel. *The Boundary Element Method, Applications in Thermo-Fluids and Acoustics*, volume 1. Wiley, 2002.