A robust and efficient iterative method for the numerical solution of the Helmholtz equation

Yogi Ahmad ERLANGGA

A robust and efficient iterative method for the numerical solution of the Helmholtz equation

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties, in het openbaar te verdedigen op
maandag 22 december 2005 om 10:30 uur

door

Yogi Ahmad ERLANGGA
ingenieur luchtvaart- en ruimtevaarttechniek, Technische Universiteit Delft

geboren te Tasikmalaya, Indonesië.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are concerned with the numerical solution of wave problems in two and three dimensions. The wave problems to be considered are modeled by means of the *Helmholtz equation*, which represents time-harmonic wave propagation in the frequency domain.

The Helmholtz equation finds its applications in many fields of science and technology. For example, the Helmholtz equation has been used to investigate acoustic phenomena in aeronautics [64] and underwater acoustics [41, 4]. The Helmholtz equation is also solved in electromagnetic applications, e.g. in photolithography [96]. Due to recently increased interest in a more efficient solver for migration in 3D geophysical applications the use of the Helmholtz equation is also investigated in that field [75, 80].

In this thesis we will focus on the application of the Helmholtz equation in geophysics. The method which is proposed and explained in this thesis can, however, be used for any class of problems related to the Helmholtz equation.

1.1 Motivation: seismic applications

In geophysical applications, seismic information on the earth's subsurface structures is very important. In the petroleum industry, e.g., accurate seismic information for such a structure can help in determining possible oil reservoirs in subsurface layers. This information, or in particular *an image of the earth's subsurface*, is gathered by measuring the times required for a seismic wave to return to the surface after reflection from the interfaces of layers of different local physical properties. The seismic wave is usually generated by shots of known frequencies, placed on the earth's surface, and the returning wave is recorded by instruments located along the earth's surface. Variations in the reflection times from place to place on the surface usually indicate structural features of the strata up to 6,000 m below the earth's surface. One technique, which is popular nowadays, to *postprocess* the seismic data and construct the image of the earth's subsurface is *migration*.

Migration is a technique to focus the seismic waves such that the exact information of the reflector/secondary sources on the subsurface is correctly known. A modern migration technique is based on the wave equation, originally proposed by Claerbout in the early 1970s, which is based on the finite-difference approach [16].

To keep the computational cost acceptable, the wave equation is usually replaced by a one-way or paraxial approximation [10, 17, 25, 28, 60]. This approximation is in most cases valid for not too large velocity contrasts and not too wide angles of incidence. With the continuing increase in computer power, it may be worthwhile to develop finite-difference two-way or full wave-equation migration techniques, without making the approximations needed for ray-based or one-way migration methods [107].

Nowadays, for the imaging of seismic data, the industry is gradually moving from 1D models based on ray-based techniques to 2D/3D finite-difference wave-equation migration. Ray-based methods are difficult to use or may even fail in complex earth models and in the presence of large velocity contrasts. Wave-equation migration can better handle these situations.

In two-dimensional space, two-way wave-equation migration can be carried out efficiently by working in the frequency domain. In that case, the linear system arising from the discretization of the two-way wave equation is solved once with a direct solution method for each frequency. The result can be used for the computation of all the wave fields for all shots and also for the back-propagated receiver wave fields. The latter correspond to the reverse-time wave fields in the time domain [72, 75, 80]. This makes the 2D method an order of magnitude faster than its time-domain counterpart when many shots need to be processed.

Time-domain reverse-time migration requires the storage of the forward and time-reversed wave fields at time intervals to avoid aliasing. These wave fields are correlated to obtain a partial migration image for each shot. Adding over all shots provides the desired result. In the frequency domain, only one forward and one back-propagated wave field need to be stored. They are simply multiplied to obtain a partial migration image. The summation over shots and frequencies produces the final migration image. In this way, a substantial reduction of storage requirement is obtained.

Because direct solvers are computationally out of reach in 3D, a suitable iterative method for the two-way wave equation is needed. This iterative method, however, must be applied for each shot and each back-propagated wave field computation. Differently, a direct method is employed only once to compute an LU-decomposition of the linear system. Once this costly step has been carried out, the computation of wave fields for all shots and receiver wave fields can be carried at a small computational cost [72]. This attractive feature, that makes the frequency-domain approach so efficient in 2D is lost when we use an iterative method.

If we ignore storage requirements and only consider computational time, a frequency-domain formulation can only compete with the time-domain approach if the work involved in the iterations times the number of frequencies is significantly less than the work needed for doing all the time steps in the time-domain method [36]. We will return to this issue in Section 1.5.

From its formulation, the time-harmonic Helmholtz equation looks easy and straightforward to solve as it can be considered as the Poisson equation with zeroth order perturbation. Many efficient numerical methods have been developed for the Poisson equation. This extra term, however, appears in the Helmholtz equation with the wrong sign. Therefore, the Helmholtz equation does not inherit the same nice properties the Poisson equation has. This perturbation is actually the source of complications when one tries to numerically solve the Helmholtz equation.

In the last three decades attempts to iteratively solve the Helmholtz equation have been made by many authors. The paper of Bayliss, Goldstein and Turkel [12], that appeared in the early 1980s, can be considered as the first publication which shows efficient implementation of an iterative method (i.e. conjugate gradients) on the Helmholtz equation. The follow-up paper by Gozani, Nochshon and Turkel [51] includes multigrid as a preconditioner in the conjugate gradient algorithm. Because the methods are not of highest efficiency for high wavenumber (or frequency) problems, many contributions have been made since. Work in [12] and [51], however, gives cause for optimism that a well-designed iterative method can be used to solve the Helmholtz equation with a greater efficiency than a traditional direct solver, especially in three-dimensional space.

Ideally, the performance of an iterative method for the Helmholtz equation should be independent of both the grid size and the wavenumber. As independence of the grid size can sometimes be easily achieved, the dependence on the wavenumber is the difficult part to tackle. Iterative methods typically suffer from efficiency degradation when the wavenumber is increased. The research in this thesis is geared towards an iterative method whose performance is independent of grid size and (only mildly) of the wavenumber.

1.2 The Helmholtz equation

In this section we derive the Helmholtz equation, which is used in the frequency-domain wave equation migration. The discussion is given for fluids and solids, the media which are present in the earth subsurface. For fluids, the Helmholtz equation can be obtained from the Euler equations, after applying a linearization; see [29, 49]. We, however, show a different approach to derive the Helmholtz equation, as used in [16]. This approach is unified approach, that can be applied to fluids and solids. Furthermore, the relationship between the wave equation for fluids and solids is clearly seen.

The basic equations used for the derivation of the wave equation are the equation of motion (governed by Newton's second law) and the Hooke law, which relates particle velocity in space and pressure in time. We first discuss the wave equation for fluids.

Consider an infinitesimally small fluid element with a volume δV in a domain $\Omega = \mathbb{R}^3$, as depicted in Figure 1.1. By assuming zero viscosity, the spatial

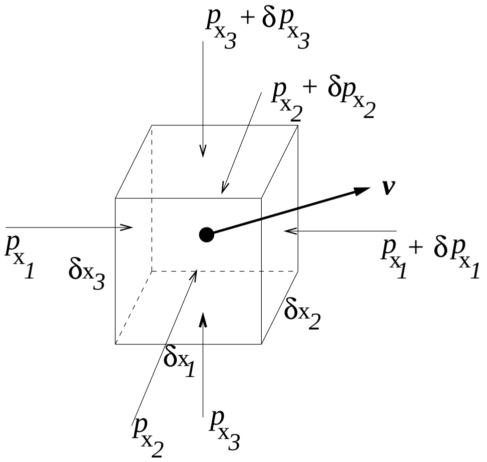
$p_{x_3} + \delta p_{x_3}$
$p_{x_2} + \delta p_{x_2}$
v
p_{x_1}
δx_3
$p_{x_1} + \delta p_{x_1}$
δx_2
δx_1
p_{x_2}
p_{x_3}

Figure 1.1: Infinitesimally small fluid particle

variation of the pressure $p = p(\boldsymbol{x}, t)$ on this element will generate a force F according to Newton's second law:

$$\delta F = \delta m \frac{\partial \boldsymbol{v}}{\partial t}, \tag{1.1}$$

with δm the element mass, $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x}, t)$ the particle velocity, $\boldsymbol{x} = (x_1, x_2, x_3)$ and $\delta F = -(\frac{\partial p}{\partial x_1}, \frac{\partial p}{\partial x_2}, \frac{\partial p}{\partial x_3})\delta V = -\nabla p \delta V$. The operator ∇ is the gradient operator. Hence,

$$\nabla p = -\frac{\delta m}{\delta V}\frac{\partial \boldsymbol{v}}{\partial t} = -\rho_0 \frac{\partial \boldsymbol{v}}{\partial t}, \tag{1.2}$$

with ρ_0 the static density.

For fluids, Hooke's law reads

$$dp = -K \frac{d\boldsymbol{v}}{\delta V}, \tag{1.3}$$

with K the compression modulus of the media, with "$d\boldsymbol{v}$" indicating the change of the velocity with respect to time. For small spatial variations,

$$\begin{aligned} \frac{dV}{\delta V} &= \frac{dx_1}{\delta x_1} + \frac{dx_2}{\delta x_2} + \frac{dx_3}{\delta x_3} \\ &= \frac{\left((v_1 dt)_{x_1+\delta x_1} - (v_1 dt)_{x_1}\right)\delta x_1}{\delta x_1} + \frac{\left((v_2 dt)_{x_2+\delta x_2} - (v_2 dt)_{x_2}\right)\delta x_2}{\delta x_2} \\ &+ \frac{\left((v_3 dt)_{x_3+\delta x_3} - (v_3 dt)_{x_3}\right)\delta x_3}{\delta x_3} \\ &= \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3}\right) dt \\ &= (\nabla \cdot \boldsymbol{v}) dt. \end{aligned} \tag{1.4}$$

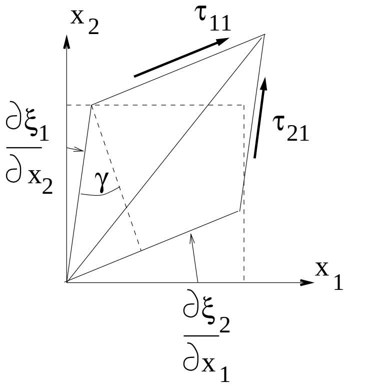

Figure 1.2: Solid element under stress forces

By using (1.4) and because $dp = (\partial p/\partial t)dt$, Hooke's law for fluids now reads

$$\nabla \cdot \boldsymbol{v} = -\frac{1}{K}\frac{\partial p}{\partial t}. \tag{1.5}$$

Applying the gradient operator to (1.2) leads to

$$\nabla \cdot \left(-\frac{1}{\rho_0}\nabla p\right) = \frac{\partial}{\partial t}\nabla \cdot \boldsymbol{v}. \tag{1.6}$$

Substitution of (1.5) into (1.6) and assuming that the gradient of the density ρ_0 is infinitesimally small result in

$$\Delta p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}, \tag{1.7}$$

which is the wave equation for fluids in terms of pressure, with $\Delta \equiv \nabla^2$ the Laplace operator, and $c = \sqrt{K/\rho_0}$ the propagation speed of compressional waves in fluids.

For isotropic solids with density ρ, the equation of motion is derived under the following hypothesis. Under forces, an element is deformed whose deformation is assumed to be linear; see Figure 1.2. The displacement ξ_i, $i = 1, \ldots, 3$, due to the stress per unit area τ is given by

$$\rho\frac{\partial^2 \xi_i}{\partial t^2} = \sum_{j=1}^{3}\frac{\partial \tau_{ij}}{\partial x_i}, \qquad i = 1, \ldots, 3. \tag{1.8}$$

Hooke's law can be represented by the generalized stress-strain relationship as follows:

$$\tau_{ij} = \tau_{ji} = 2\mu\epsilon_{ij} + \delta_{ij}\lambda\nabla \cdot \boldsymbol{\xi}, \tag{1.9}$$

with $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$, $\epsilon_{ij} = \frac{1}{2}(\partial\xi_j/\partial x_i + \partial\xi_i/\partial x_j)$ the strain, μ the shear modulus, λ the Lamé constant and δ_{ij} the Kronecker symbol satisfying $\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ij} = 1$ for $i = j$. Substitution of (1.9) into (1.8) results in

$$\rho\frac{\partial^2\xi_1}{\partial t^2} = \mu\nabla^2\xi_1 + (\mu+\lambda)\frac{\partial}{\partial x_1}(\nabla\cdot\boldsymbol{\xi}) \tag{1.10}$$

$$\rho\frac{\partial^2\xi_2}{\partial t^2} = \mu\nabla^2\xi_2 + (\mu+\lambda)\frac{\partial}{\partial x_2}(\nabla\cdot\boldsymbol{\xi}) \tag{1.11}$$

$$\rho\frac{\partial^2\xi_1}{\partial t^2} = \mu\nabla^2\xi_3 + (\mu+\lambda)\frac{\partial}{\partial x_3}(\nabla\cdot\boldsymbol{\xi}), \tag{1.12}$$

or

$$\rho\frac{\partial^2\boldsymbol{\xi}}{\partial t^2} = \mu\nabla^2\boldsymbol{\xi} + (\mu+\lambda)\nabla(\nabla\cdot\boldsymbol{\xi}). \tag{1.13}$$

The displacement $\boldsymbol{\xi}$ can also be expressed by a scalar potential Φ and a vector potential H such that

$$\boldsymbol{\xi} = \nabla\Psi + \nabla\times H, \tag{1.14}$$

with an assumption that $\nabla\cdot H = 0$. By substituting the above relation into (1.13) one finds that

$$\nabla\left((2\mu+\lambda)\nabla^2\Psi - \rho\frac{\partial^2\Psi}{\partial t^2}\right) + \nabla\times\left(\mu\nabla^2 H - \rho\frac{\partial^2 H}{\partial t^2}\right) = 0, \tag{1.15}$$

after making use of relations $\nabla^2(\nabla\times H) = \nabla\times(\nabla^2 H)$ and $\nabla\cdot(\nabla\times H) = 0$. The above relation is satisfied if

$$\nabla^2\Psi \equiv \Delta\Psi = \frac{1}{c_1^2}\frac{\partial^2\Psi}{\partial t^2}, \tag{1.16}$$

$$\nabla^2 H \equiv \Delta H = \frac{1}{c_2^2}\frac{\partial^2 H}{\partial t^2}, \tag{1.17}$$

where $c_1^2 = (2\mu+\lambda)/\rho$ and $c_2^2 = \mu/\rho$. The first equation is related to compressional waves, while the second equation corresponds to shear/translational waves.

For compressional waves the potential Ψ can be redefined as

$$\psi = -\rho\frac{\partial^2\Psi}{\partial t^2}, \tag{1.18}$$

which satisfies the wave equation

$$\Delta\psi = \frac{1}{c_1^2}\frac{\partial^2\psi}{\partial t^2}. \tag{1.19}$$

From (1.14), with $H = 0$ we then have

$$\boldsymbol{\xi} = \nabla\Psi \Rightarrow \frac{\partial^2\boldsymbol{\xi}}{\partial t^2} = \frac{\partial^2}{\partial t^2}\nabla\Phi \Rightarrow \rho\frac{\partial\boldsymbol{v}}{\partial t} = -\nabla\psi, \tag{1.20}$$

with $\boldsymbol{v} = \partial\boldsymbol{\xi}/\partial t$, the particle velocity. Furthermore, one can easily verify that

$$\nabla \cdot \boldsymbol{v} = -\frac{1}{2\mu + \lambda}\frac{\partial \psi}{\partial t}. \tag{1.21}$$

Equations (1.20) and (1.20) are equations which relate velocity v to the potential ψ. The role of these equations is similar to (1.2) and (1.5) which relate the velocity v to the pressure p. Therefore, the potential ψ in solids and the pressure p may be equated, with $2\mu + \lambda$ being the compressional modulus in the cases with solids.

In the discussion to follow, we will use the wave equation for fluids (1.7).

We are concerned with time-harmonic waves of the time-dependent pressure of the form

$$p(\boldsymbol{x}, t) = u(\boldsymbol{x}) \exp(-\hat{j}\omega_w t), \tag{1.22}$$

where $\omega_w > 0$ and $\hat{j} = \sqrt{-1}$ denote the angular frequency and the imaginary unit, respectively. Substituting (1.22) into (1.7) we obtain

$$-\Delta u(\boldsymbol{x}) - k^2(\boldsymbol{x})u(\boldsymbol{x}) =: \mathcal{A}u(\boldsymbol{x}) = 0, \tag{1.23}$$

with $\mathcal{A}$ the Helmholtz operator. In (1.23) k is the wave number, and $k(\boldsymbol{x}) = \omega_w/c(\boldsymbol{x})$. Because $\omega_w = 2\pi f$, where f is the wave frequency, we find also that $k = 2\pi/\lambda_w$, where $\lambda_w = c/f$ is defined as the wavelength. Equation (1.23) is known as the *Helmholtz equation* for the pressure.

Remark 1.2.1 *The Helmholtz equation can also be expressed for the velocity using the velocity potential relation* $\boldsymbol{v} = 1/\rho_0 \nabla\Phi$ *and* $p = -\partial_t \Phi$*, where* $\Phi = \Phi(\boldsymbol{x}, t)$ *is the velocity potential. If this relation is substituted into (1.7) one can verify that this relation also satisfies the wave equation. Introducing time-harmonic waves with* $\Phi(\boldsymbol{x}, t) = \phi(\boldsymbol{x}) \exp(-\hat{j}\omega_w t)$*,* $\omega_w > 0$ *we find the Helmholtz equation in velocity:*

$$-\Delta\phi(\boldsymbol{x}) - k^2(\boldsymbol{x})\phi(\boldsymbol{x}) = 0. \tag{1.24}$$

This equation and (1.23) are identical, and one can transform the solution from one into the other by using the velocity potential relation.

Remark 1.2.2 *By introducing a source term, and furthermore assuming that this source term is also time-harmonic, a more general formulation of the Helmholtz equation is obtained, namely*

$$\mathcal{A}u(\boldsymbol{x}) := -\Delta u(\boldsymbol{x}) - k^2(\boldsymbol{x})u(\boldsymbol{x}) = g(\boldsymbol{x}), \tag{1.25}$$

where $g(\boldsymbol{x})$ *is the source term.*

Remark 1.2.3 *Sometimes we are also interested in a barely attenuative medium. For this type of problem, the Helmholtz equation is written as*

$$\mathcal{A}u := -\Delta u(\boldsymbol{x}) - (1 - \alpha\hat{j})k^2(\boldsymbol{x})u(\boldsymbol{x}) = g(\boldsymbol{x}), \tag{1.26}$$

with $0 \leq \alpha \ll 1$ *indicating the fraction of damping in the medium. In geophysical applications, that are of our main interest, this damping can be set up to* 5%*;* $(\alpha = 0.05)$.

The "damped" Helmholtz equation expressed by (1.26) is more general than (1.23) and (1.25). Therefore, due to its generality we prefer to use the Helmholtz equation (1.26) throughout the presentation. Whenever necessary, the "standard" Helmholtz equation can be recovered from (1.26) by setting $\alpha = 0$.

Remark 1.2.4 Scaling. *Assume that the physical domain has a characteristic length* $\bar{l}$ *and consider a nondimensional domain* $[0,1]^3$*. The nondimensional length is determined as* $\widetilde{x}_1 = x_1/\bar{l}$*,* $\widetilde{x}_2 = x_2/\bar{l}$ *and* $\widetilde{x}_3 = x_3/\bar{l}$*. Thus,*

$$\frac{\partial}{\partial x_1} = \frac{1}{\bar{l}}\frac{\partial}{\partial \widetilde{x}_1}, \quad \frac{\partial^2}{\partial x_1^2} = \frac{1}{\bar{l}^2}\frac{\partial^2}{\partial \widetilde{x}_1^2}, \quad \textit{and so on.}$$

Substituting these relations into (1.26) results in

$$-\widetilde{\Delta} u(\widetilde{\boldsymbol{x}}) - (1 - \alpha\hat{j})\overline{k}^2(\boldsymbol{x})u(\widetilde{\boldsymbol{x}}) = g(\widetilde{\boldsymbol{x}}), \quad \widetilde{\boldsymbol{x}} = (\widetilde{x}_1, \widetilde{x}_2, \widetilde{x}_2), \quad \textit{in } \Omega = (0,1)^3,$$

with the wavenumber in the non-dimensional domain, denoted by $\overline{k}$*, can be related to the physical quantities in the physical domain by the relation*

$$\overline{k} = 2\pi f \bar{l}/c. \tag{1.27}$$

In the discussion to follow we will use the notation k *for wavenumbers regardless of the domain we are considering. However, the meaning should be clear from the context. If we consider a unit domain, the wavenumber should be dimensionless.*

1.3 Boundary conditions

For a partial differential equation, of the form (1.23), (1.25) or (1.26), proper boundary conditions are required. These boundary conditions should be chosen such that the resulting problem is well-posed. In general, one can identify two types of boundary conditions. The first type is a boundary condition for waves propagating to infinite distance. The second type is related to waves scattered by obstacles in a medium.

A boundary condition at infinity can be derived by considering the physical situation at infinity [49]. This situation can be viewed directly from the Helmholtz equation (1.23) if we consider a domain Ω with a homogeneous medium. Assume spherical symmetric waves propagating from a source or a scatterer in the domain Ω. In most cases, close to the source/scatterer this assumption is easily violated; there the waves are arbitrary and more complex than just spherical. We assume, however, that at infinity these complex waves are disentangled and become spherical. Under this assumption, the problem can be evaluated in a spherical coordinate system. In the spherical coordinate system, the Helmholtz equation is transformed into

$$-(ru)'' - k^2(ru) = 0,$$

with a general solution of the form

$$u(r) = A\frac{\cos(kr)}{r} + B\frac{\sin(kr)}{r}. \tag{1.28}$$

Combining (1.28) with (1.22) we find that

$$p(r,t) = A^*\frac{\exp(\hat{\jmath}(kr - \omega_w t))}{r} + B^*\frac{\exp(-\hat{\jmath}(kr + \omega_w t))}{r}. \tag{1.29}$$

By considering surfaces of constant phase it is easily deduced that the first term at the right hand side describes waves propagating away from the source/scatterer, as $\omega_w t$ increases with time, while the second term represents waves propagating inwards from infinity. If a region Ω bounded by a spherical surface $\Gamma = \partial\Omega$ contains a scatterer, the second term in (1.29) cannot be a solution, as physics only allows waves propagating away from the scatterer. The solution, therefore, consists only of the first term in (1.29). Since the first term contains a factor r^{-1}, the amplitude of the wave must disappear at infinity.

The vanishing criterion ensuring $u(r) \to 0$ as $r \to \infty$ is provided by the radiation condition. For $k > 0$, if we differentiate (1.29), after dropping the second right hand term, the following relation results:

$$r(u' - \hat{\jmath}ku) = -u.$$

For $r \to \infty$, one finds that $u \to 0$. By this we can reformulate the problem as

$$\lim_{r\to\infty}\left(u' - \hat{\jmath}ku\right) \sim o(r^{-1}); \;\; \lim_{r\to\infty} ru \sim O(1). \tag{1.30}$$

In the above relations, symbols "o" and "O" are the Landau's symbols, defined as

$$\begin{aligned} f_1(x) \sim O(f_2(x)) \quad &\Longrightarrow \quad \frac{f_1(x)}{f_2(x)} \to C, \\ f_1(x) \sim o(f_2(x)) \quad &\Longrightarrow \quad \frac{f_1(x)}{f_2(x)} \to 0, \end{aligned}$$

with C being a constant. The first condition in (1.30) is the *Sommerfeld radiation* condition, whereas the second condition is the *finiteness* condition.

Remark 1.3.1 *A different terminology is also used for the radiation condition. For instance it is sometimes called "non-reflecting" or "absorbing" boundary condition. We will use them interchangeably in this thesis.*

The radiation condition (1.30) is only satisfied at infinity. In a numerical model, this condition is approximately satisfied, as an infinite domain is always truncated to a finite region. Therefore, a numerical radiation condition is required for boundaries at finite distance, which mimics the radiation condition at infinity. This will be addressed in Chapter 2.

In the scattering situation, additional boundary conditions inside Ω representing the presence of the obstacles should be added. Consider an impenetrable obstacle D in Ω with boundary $\Gamma_s = \partial D$. We distinguish two types of boundary conditions commonly used for scattering problems. In the case of *sound-soft* obstacles, the velocity of the total wave (i.e. the sum of the incoming wave ϕ^i and the scattered wave ϕ^s) vanishes at Γ_s, i.e. $\phi = 0$ on Γ_s. This is known as the *Dirichlet* boundary condition. For the case of *sound-hard* obstacles, the condition on Γ_s leads to a *Neumann* condition: $\partial\phi/\partial n = 0$, implying a condition of a vanishing normal velocity on the scatterer surface.

1.4 Direct methods and iterative methods

Numerical approximations based on finite differences/volumes/elements require the solution of a sparse, large linear system of equations. There are in general two classes of methods to solve a linear system: direct methods and iterative methods.

Direct methods are basically derived from Gaussian elimination. They are well-known for their robustness for general problems. However, Gaussian elimination is not favorable for sparse linear systems. During the elimination process, zero elements in the matrix may be filled by non-zero entries. This is called fill-in and gives rise to two complications: (i) extra computer storage required to store the additional non-zero entries, (ii) extra computational work during the elimination process.

In many problems, a reordering strategy or pre-processing (e.g. nested dissection) technique can be used to modify the structure of the matrix such that Gaussian elimination can be performed more efficiently (see e.g. [48]). Nevertheless, it is also possible that the pre-processing adds some more computational work.

Iterative methods, on the other hand, rely primarily on matrix-vector multiplications. An example of a simple iterative method is the Jacobi iteration. In the Jacobi iteration, the solution of linear system is iteratively obtained from a recursion consisting of one matrix-vector multiplication, starting with a given initial solution. A matrix-vector multiplication is a cheap process, requiring only $O(N)$ arithmetic operations per iteration with N the number of unknowns. If the method converges after a finite small number of iterations, the method would be efficient.

Iterative methods, however, are not always guaranteed to have fast convergence. In fact there are many situations for iterative methods to diverge. In such cases, iterative methods do not offer any advantage as compared to direct methods.

Table 1.1 compares the typical complexities of different solvers for a banded linear system obtained from a finite difference discretization of the 2D Poisson equation. The size of the linear system to solve is $N \times N$, where N is the number of unknowns. The iterative methods converge to a tolerance ϵ. Conjugate gradients and multigrid, two advanced iterative methods, are later discussed in this thesis.

Table 1.1: Some two-dimensional Poisson solvers with arithmetic complexity [92]. N is the number of unknowns

Method	Complexity
Gauss elimination	$O(N^2)$
Jacobi/Gauss-Seidel iteration	$O(N^2 \log \epsilon)$
Successive over relaxation (SOR)	$O(N^{3/2} \log \epsilon)$
Conjugate gradient	$O(N^{3/2} \log \epsilon)$
Nested dissection [48]	$O(N^{3/2})$
Alternating-direction iteration (ADI) [101]	$O(N \log N \log \epsilon)$
Multigrid (iterative)	$O(N \log \epsilon)$
Multigrid (FMG)	$O(N)$

For a sparse, large linear system, direct methods become impractical. This is typical for problems in three-dimensional space. The complexity of the method becomes $O(N^3)$ and storage requirement can be up to $O(N^2)$ due to fill-in. Therefore, for three-dimensional problems iterative methods are the method of choice. In the case of slow convergence or even divergence remedies based on theory often exist.

The Helmholtz equation is one of the problems for which iterative methods typically result in an extremely slow convergence. For reasons explained systematically in Chapters 2 and 3, the Helmholtz equation is inherently unfavorable for iterative methods. However, with a proper remedy, e.g. with a good preconditioner (to be discussed in Chapters 4 and 5), an efficient iterative method can be designed.

Conjugate gradients and conjugate-gradient-like methods have been used in [12], [51], [47] and [64] to solve the Helmholtz equation. ADI is used in [32]. Multigrid implementations are found in [22], [100], [65], [33] and [63].

1.5 Time-domain vs. frequency-domain solver

As mentioned in Section 1.1, to make a frequency-domain solver competitive with the time-domain approach for geophysical applications, we require that the work for the iterations times the number of frequencies is significantly less than the work needed for doing all the time steps in the time-domain method. In this section we elaborate on this subject, based on a complexity analysis [36].

The basis for the performance prediction is the well-known time-domain approach complexity. With n_s the number of shots and n_t the number of time steps, the time-domain complexity is $n_s\, n_t\, O(N)$, with N again the number of unknowns. Stability requirements usually dictate $n_t = O(N^{1/d})$, with $d = 1, \ldots, 3$ the dimension. So, the overall complexity of the time-domain method is $n_s\, O(N^{1+1/d})$ for n_s shots.

Suppose that there exists in frequency domain an efficient iterative method for the Helmholtz equation with a complexity of order $O(N)$ per iteration step. With n_f the number of frequencies, the overall complexity of the iterative solver is $n_s\, n_f\, n_{it}\, O(N)$, with n_{it} the number of iteration. The choice of the grid spacing is determined by the maximum wavelength that can be accurately represented by the discretization. This implies that $n_f = O(N^{1/d})$ if the maximum frequency in the data is adapted to the grid spacing, whereas n_f is independent of N as soon as the grid is finer than required to accurately represent the highest frequency in the data. In practice, one generally operates in the first regime where $n_f = O(N^{1/d})$. With these considerations, the complexity of the iterative frequency-domain solver is $n_s\, n_{it}\, O(N^{1+1/d})$. From this estimate, the optimal iterative frequency-domain solver would have the same complexity as the time-domain solver if $n_{it} = O(1)$.

For a comparison between two solvers, however, the actual multiplicative constants in the cost estimates also play an important role. The complexity of the time-domain solver is $C_t\, n_s\, N^{1+1/d}$ and of the iterative frequency-domain solver $C_f\, n_s\, n_{it}\, N^{1+1/d}$, with constants C_t and C_f. If one constant is several orders of magnitude smaller than the other, a cost comparison of the methods cannot be based on considerations that involve $O(n)$, $O(n_f)$, etc., only. This is observed in 2D migration problems presented in [74], where nested dissection is used in the frequency domain. From complexity analysis, the time-domain approach shows a better asymptotic behavior than the frequency domain approach. Actual computations reveal, however, that the frequency-domain method is about one order of magnitude faster than the time-domain approach. The multiplicative constant for the frequency-domain direct solver is apparently far smaller than the one for the time-domain solver. The reason is that the number of frequencies needed is smaller than the number of time steps imposed by the stability condition of the time-domain solver [75]. This result gives an indication that if an iterative method with complexity of $O(N)$ can be made convergent after an $O(1)$ number of iterations, the method can be competitive. As in 3D direct methods become out of question, the importance of an efficient iterative method becomes more obvious.

1.6 Scope of the thesis

This thesis deals with numerical methods for solving the Helmholtz equation in two and three dimensions, with the following characteristics:

(i) the property of the medium (in terms of the local speed of sound, c) can vary; i.e, heterogeneity is present in the medium,

(ii) the wavenumber k can be very large, e.g. ~ 1000 in a unit square domain.

We focus on iterative methods within the classes of Krylov subspace and multigrid methods. As already mentioned, we aim at a method whose performance is independent of grid size and wavenumber. By using a proper preconditioner, the performance of the method can be made grid-independent. The method presented in this thesis shows linear dependence on the wavenumber, with only small constant.

Furthermore, we focus on linear systems obtained from finite difference discretizations. This is, however, *not* a limitation of the iterative method discussed here. The method can handle different types of discretizations without losing its efficiency and robustness.

Numerical results presented in this thesis are based on problems arising from geophysical applications with extremely high wavenumbers. But, the method developed in this thesis can also be used in other applications governed by the Helmholtz equation.

1.7 Outline of the thesis

The outline of the thesis is as follows.

- In Chapter 2, discretization methods for the Helmholtz equation are discussed. The discussion emphasizes finite differences suitable for geophysical applications.

- In Chapter 3 iterative methods are introduced for solving the resulting linear systems. These include the basic iterative (Jacobi and Gauss-Seidel) and Krylov subspace methods, which are the important and principal methods in our solver. Some preconditioners which are relevant for Krylov subspace methods are also discussed.

- Chapter 4 is devoted to a novel class of preconditioners for the Helmholtz equation, called *complex shifted Laplace* preconditioners, which is an important ingredient in the method presented in this thesis and introduced by us in a series of papers [37, 38, 40]. Discussions include the spectral properties which are helpful to evaluate the convergence of the Krylov subspace methods.

- In Chapter 5 the multigrid method is introduced. In particular, we discuss the use of multigrid to approximately invert the complex shifted-Laplace operator.

- In Chapter 6 a multigrid-based preconditioner especially designed for the Helmholtz equation is introduced and discussed. Numerical tests for two-dimensional problems are presented.

- Chapter 7 discusses the extension of the method to three dimensions and shows results for three-dimensional problems.

- In Chapter 8 we draw some conclusions and give remarks and suggestions for future work.

1.8 About numerical implementations

In this thesis computations are performed for a variety of model problems. In the early stages of the research a not-the-fastest computer was used to produce the convergence results. As a new, faster computer has replaced the old one, the numerical results are updated to have more impressing CPU times. Therefore, one finds that some results here differ with the results that appeared in early publications, e.g. in [38].

To reproduce the numerical results presented in this thesis, we have used a LINUX machine with one Intel Pentium 4, 2.40 GHz CPU. This machine is equipped with 2 GByte of RAM and 512 KByte of cache memory. For small problems, MATLAB has been used for coding. GNU's Fortran 77 compiler is used mainly for large problems.

Chapter 2

Finite differences for the Helmholtz equation

The standard procedure for obtaining a numerical solution of any partial differential equation (PDE) is first replacing the equation by its discrete formulation. For the Helmholtz equation, there are many methods to realize this, including the boundary element method (BEM) and the finite element method (FEM). BEM, which typically results in a small but full linear system after discretization, will not be discussed here. References [57], [58], [59], [7], [8], [9] and [31], amongst others, discuss FEM for the Helmholtz equation. We focus on finite difference methods (FDM). All numerical results presented in this thesis are obtained from a finite difference approximation of the Helmholtz equation. Furthermore, structured equidistant grids are employed.

A general discussion on finite difference methods for partial differential equations can be found, e.g. in [88]. The development of a high order FDM is given, e.g. in [68], [13], [66] and [90]. There exist different finite difference approximations for the Helmholtz equation (see e.g. [54], [85] and [61]). Here we consider two of them: the pointwise representation and a high order representation. The pointwise representation is usually related to the so-called five-point stencil. Dispersion and anisotropy properties of the two representations are briefly discussed in Section 2.3.

In geophysical applications finite differences are widely used. One practical reason is that the domains used in these applications are often simple (e.g. rectangular or box shape) and can be well fitted by regular finite difference meshes. For Helmholtz problems with a complex geometry arising from scattering problems, finite elements are often the method of choice.

For ease of presentation, we consider the “damped” Helmholtz equation (1.26) in $\Omega \subset \mathbb{R}^2$. Extension to three-dimensional space is straightforward.

2.1 Second order finite difference approximation

Let the sufficiently smooth domain Ω be discretized by an equidistant grid with the grid size h. The discretized domain is denoted by Ω_h. We compute approximate solutions of the Helmholtz equation on Ω_h. Let us consider the solution at the grid point $\boldsymbol{x} = (x_1, x_2) = \{(i_1h, i_2h) | i_1, i_2 = 1, 2, \ldots, \sqrt{N}\}$ in Ω_h, with N the number of unknowns. We introduce the standard lexicographical numbering and denote the approximate solution $u(\boldsymbol{x}) = u(i_1h, i_2h)$ as u_{i_1,i_2} and those at the four neighboring points with $u_{i_1-1,i_2}, u_{i_1+1,i_2}, u_{i_1,i_2-1}, u_{i_1,i_2+1}$ (see Figure 2.1). By using the approximation

$$\begin{aligned}\frac{\partial^2 u}{\partial x_1^2} &\approx \frac{1}{h^2}\left(u_{i_1+1,i_2} - 2u_{i_1,i_2} + u_{i_1-1,i_2}\right) \quad \text{and} \\ \frac{\partial^2 u}{\partial x_2^2} &\approx \frac{1}{h^2}\left(u_{i_1,i_2+1} - 2u_{i_1,i_2} + u_{i_1,i_2-1}\right),\end{aligned}$$

the Helmholtz equation

$$\mathcal{A}u := -\Delta u - (1 - \alpha\hat{j})k^2 u = g, \quad \text{in } \Omega, \tag{2.1}$$

can now be approximated in Ω_h by the equation

$$\begin{aligned}-\frac{1}{h^2}&\left(u_{i_1,i_2-1} + u_{i_1-1,i_2} - 4u_{i_1,i_2} + u_{i_1+1,i_2} + u_{i_1,i_2+1}\right) \\ &- (1-\alpha\hat{j})k^2 u_{i_1,i_2} = g_{i_1,i_2}, \quad \text{for } i_1, i_2 = 1, 2, \ldots, n,\end{aligned}$$

which can be written in stencil notation as

$$A_{h,5p}u_h \mathrel{\hat{=}} -\frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & -4 + (1-\alpha\hat{j})k^2h^2 & 1 \\ & 1 & \end{bmatrix} u_h = g_h, \tag{2.2}$$

with $h = 1/\sqrt{N}$, the grid size.

In the above equation we approximate the second order differential term by the central difference scheme. For smooth solutions in uniform grids, this approximation is of $O(h^2)$ accuracy which is sufficient for many applications. One can easily prove that on non-uniform grids, this approximation is only accurate up to $O(h)$.

Extension to the 3D Helmholtz equation. In 3D, the discretized Helmholtz operator can be written in stencil notation as

$$\begin{aligned}&A_{h,7p} \mathrel{\hat{=}} \\ &-\frac{1}{h^2}\left(\begin{bmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{bmatrix}_{i_3-1} \begin{bmatrix} & 1 & \\ 1 & -6 + (1-\alpha\hat{j})k^2h^2 & 1 \\ & 1 & \end{bmatrix}_{i_3} \begin{bmatrix} & 0 & \\ 0 & 1 & 0 \\ & 0 & \end{bmatrix}_{i_3+1}\right).\end{aligned} \tag{2.3}$$

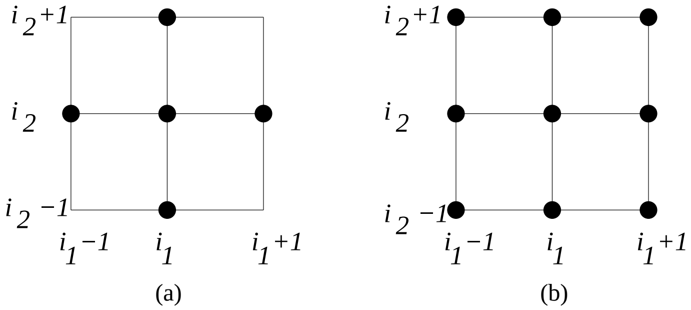

Figure 2.1: 2D finite difference stencil: (a) 5-point stencil, (b) 9-point stencil

2.2 High order finite difference approximation

A higher order accurate finite difference stencil can be obtained for the Helmholtz equation in different ways. One way to construct this high order finite difference stencil is by using Padé approximation.

Consider the Taylor expansion of a second order derivative

$$D_{x_1x_1}u = \partial_{x_1x_1}u + \frac{1}{12}h^2\partial_{x_1}^{(4)}u + O(h^4), \tag{2.4}$$

where D is the a discrete operator. The fourth order derivative can be approximated as follows:

$$\partial_{x_1}^{(4)}u = \partial_{x_1x_1}(\partial_{x_1x_1}u) \approx D_{x_1x_1}(\partial_{x_1x_1}u) + O(h^2). \tag{2.5}$$

Substituting (2.5) into (2.4) we have

$$\partial_{x_1x_1}u = \left(1 + \frac{1}{12}h^2 D_{x_1x_1}\right)^{-1} D_{x_1x_1}u + O(h^4). \tag{2.6}$$

Substituting (2.6) and also the similar expression for $\partial_{x_2x_2}$ into (2.1) we finally obtain

$$-\left(\frac{D_{x_1x_1}}{1+\frac{1}{12}h^2D_{x_1x_1}} + \frac{D_{x_2x_2}}{1+\frac{1}{12}h^2D_{x_2x_2}} + (1-\alpha\hat{j})k^2\right)u = g, \tag{2.7}$$

or

$$-(D_1 + D_2)u - (1 + K_1 + K_2)(1 - \alpha\hat{j})k^2u = (1 + K_1 + K_2)g. \tag{2.8}$$

where

$$
\begin{aligned}
D_1 &= D_{x_1x_1} + D_{x_2x_2} && (2.9)\\
D_2 &= \frac{h^2}{6} D_{x_1x_1} D_{x_2x_2} && (2.10)\\
K_1 &= \frac{h^2}{12}(D_{x_1x_1} + D_{x_2x_2}) && (2.11)\\
K_2 &= \frac{h^4}{144} D_{x_1x_1} D_{x_2x_2} && (2.12)
\end{aligned}
$$

Equation (2.8) is called the point-wise Padé approximation of the Helmholtz equation. Harari and Turkel [54] (and also Singer and Turkel in [85]) propose a generalization to this approximation by introducing a free parameter $\gamma \in \mathbb{R}$. This parameter is tuned such that the dispersion or the anisotropy is minimal.

The following 9-point stencil is due to Harari and Turkel [54], for $\alpha = 0$:

$$
A_{h,9p} \overset{\wedge}{=} -\frac{1}{h^2} \begin{bmatrix} \frac{1}{6} + \gamma\frac{(kh)^2}{144} & \frac{2}{3} + \frac{(6-\gamma)(kh)^2}{72} & \frac{1}{6} + \gamma\frac{(kh)^2}{144} \\ \frac{2}{3} + \frac{(6-\gamma)(kh)^2}{72} & -\frac{10}{3} + \frac{(24+\gamma)(kh)^2}{36} & \frac{2}{3} + \frac{(6-\gamma)(kh)^2}{72} \\ \frac{1}{6} + \gamma\frac{(kh)^2}{144} & \frac{2}{3} + \frac{(6-\gamma)(kh)^2}{72} & \frac{1}{6} + \gamma\frac{(kh)^2}{144} \end{bmatrix}. \quad (2.13)
$$

It can be shown that this scheme is consistent for $kh \to 0$ and $O(h^4)$ accurate for constant wavenumber and uniform grids. The scheme, however, is only $O(h^3)$ for nonconstant wavenumber and non-uniform grids.

Remark 2.2.1 *The Padé approximation for the 3D Helmholtz equation is given in Appendix A.*

Remark 2.2.2 *A non standard 9-point finite difference approximation is also proposed by [61]. The approach is based on a splitting of the Laplacian term into two parts: one in grid line direction and one in cell-diagonal direction, with some weights. It can be shown that the dispersion and anisotropy can also be minimized in this case.*

2.3 Dispersion and anisotropy

In order to illustrate the dispersion and anisotropy properties of a discretization scheme, we define two parameters which characterize wave propagations [54]:

$$
c_p = \frac{\omega_w}{k}, \quad c_g = \frac{\partial \omega_w}{\partial k}, \quad (2.14)
$$

which are called the phase velocity and the group velocity, respectively. For homogeneous and isotropic continuous media, $c_p = c_g = c_0$; i.e. the media is non-dispersive.

In a discrete formulation the non-dispersive property no longer holds. Instead, we have that

$$
c_p^h = \frac{\omega_w}{k^h} = \frac{k}{k^h} c_0, \;\; c_g^h = \frac{\partial \omega_w}{\partial k^h} = \left(\frac{\partial k^h}{\partial k}\right)^{-1} c_0, \quad (2.15)
$$

where $k^h = f(kh)$, depending on the discretization. For the 5-point stencil, we have that [54]:

$$\frac{c_p^h}{c_0} = \frac{kh}{\arccos(1-(kh)^2/2)}, \tag{2.16}$$

$$\frac{c_g^h}{c_0} = \sqrt{1-(kh)^2/4}. \tag{2.17}$$

Thus, for the 5-point stencil the phase and group velocity are slower than the speed of sound c_0.

For the 9-point stencil (2.13), the phase and group velocity along the grid lines are determined respectively as:

$$\frac{c_p^h}{c_0} = \frac{kh}{\arccos((1-5(kh)^2/12)/(1+(kh)^2/12))}, \tag{2.18}$$

$$\frac{c_g^h}{c_0} = \sqrt{1-(kh)^2/6}\,(1+(kh)^2/12). \tag{2.19}$$

Along the grid line, for a resolution up to a minimum of 4 grid points per wavelength (i.e. $kh < 1.57$), the 9-point stencil is less dispersive compared to the 5-point stencil. In fact, the error of the 9-point stencil falls within 7 % compared to non-dispersion solution. The 5-point stencil, however, is still accurate if more than 10 grid points per wavelength are used (or, equivalently, if $kh < 0.625$).

Similarly, one can derive the phase and group velocity of discrete cases for plane waves oriented at an angle θ with respect to the grid lines. An extreme case occurs when the wave is in the direction of the cell's diagonal ($\theta = \pi/4$). For the 5-point stencil we then have

$$\frac{c_p^h}{c_0} = \frac{kh}{\sqrt{2}\arccos(1-(kh)^2/4)}, \tag{2.20}$$

$$\frac{c_g^h}{c_0} = \sqrt{1-(kh)^2/8}, \tag{2.21}$$

and for the 9-point stencil, the phase and group velocities can be determined from the relation

$$k^h h = \sqrt{2}\arccos\left(\frac{6\sqrt{1+(1-\gamma)(kh)^4/144}-(4+(6-\gamma)(kh)^2/12)}{2+\gamma(kh)^2/12}\right). \tag{2.22}$$

In both cases, the phase and group velocity are slower than the speed of sound c_0. For the 5-point stencil, however, the scheme is less dispersive for waves oriented in the direction of cell diagonals compared to waves in the direction of grid lines. The value $\gamma = 2/5$ results in a 9-point discretization stencil with minimal dispersion in the direction of cell diagonals. If $\gamma = 14/5$ the difference between dispersion in the direction of cell diagonals and grid lines is minimal. Thus, $\gamma = 14/5$ leads to a 9-point stencil with minimal anisotropy. We refer the reader to [54] for a more detailed discussion on this subject.

Remark 2.3.1 *For the second order scheme, however, the quantity kh is not sufficient to determine the accuracy of the numerical solution of the Helmholtz equation [13]. The quantity k^3h^2 is also found important in determining the accuracy in the L^2 norm, especially at high wavenumbers.*

Remark 2.3.2 *In our numerical tests we often use the five-point finite difference stencil on the minimum allowable grid resolution $kh = 0.625$. Table 2.1 displays the number of grid points for $kh = 0.625$.*

Table 2.1: Number of grid points employed, related to the wavenumber, so that $kh = 0.625$.

k:	10	20	30	40	50	80	100	150	200	500	600
$N(=1/h)$:	8	16	32	64	80	128	160	240	320	800	960

2.4 Numerical boundary conditions

In Chapter 1 we have formulated a condition for the radiation boundary. This condition, however, is only satisfied in an infinite domain. For computational purposes extending the domain to infinity is not practical due to, e.g., restricted hardware resources. Therefore, we often truncate the domain in such away that a physically and computationally acceptable compromise is reached. In this truncated domain, the non-reflecting boundary condition given in (1.30) is, however, no longer valid.

In [70] and [67] *non*-local boundary conditions are proposed to mimic the radiation condition at infinity. Despite their accuracy for any direction of wave incidence, the inclusion of this type of boundary condition in the discretization is not practical because of the non-locality. Enquist et al in [34] proposed *local* boundary conditions for a truncated domain; see also [26], [27]. Different types of local boundary conditions have also been proposed elsewhere; see, e.g., in [56], [50] and the references therein. In this thesis, we limit ourselves to the local boundary conditions proposed in [34], either in the first order or the second order formulation.

In our case we consider a rectangular domain which is typically used in geophysics. We distinguish on a boundary Γ: faces, edges and corners. For each part of Γ we have the following boundary conditions (see [26, 34, 55, 11]):

Faces:

$$\mathcal{B}_2 u|_{face} := \pm\frac{\partial u}{\partial x_i} - \hat{j}ku - \frac{\hat{j}}{2k}\sum_{j=1,j\neq i}^{3}\frac{\partial^2 u}{\partial x_j^2} = 0, \quad i = 1,\ldots,3. \tag{2.23}$$

Here x_i is the coordinate perpendicular to the face.

Edges:

$$\mathcal{B}_2 u|_{edge} := -\frac{3}{2}k^2 u - \hat{j}k \sum_{j=1, j\neq i}^{3} \left(\pm \frac{\partial u}{\partial x_j} \right) - \frac{1}{2}\frac{\partial^2 u}{\partial x_i^2} = 0, \quad i = 1, \ldots, 3, \tag{2.24}$$

with x_i the coordinate parallel to the edge.
Corners:

$$\mathcal{B}_2 u|_{corner} := -2\hat{j}ku + \sum_{i=1}^{3} \pm \frac{\partial u}{\partial x_i} = 0. \tag{2.25}$$

In (2.23)–(2.25) the ± sign is determined such that for out going waves the non-reflecting condition is satisfied.

For the second order derivatives, to retain the same order of accuracy as in the interior the same discretization is used as for the Helmholtz operator.

In Chapters 4 and 5 we implement the first order boundary condition on the faces described by [34]

$$\mathcal{B}_1 u|_{face} := \left(\frac{\partial}{\partial \eta} - \hat{j}k \right) u = 0, \tag{2.26}$$

with η the outward normal component to the boundary. This boundary condition (2.26) is straightforward to discretize (either with one-sided or central schemes), but it is not accurate for inclined outgoing waves. In Chapter 6 boundary conditions of the form (2.23)–(2.25) are implemented.

Remark 2.4.1 *In practice the use of the second order radiation conditions sometimes is not sufficient to reduce reflections. In the last decade a new and accurate boundary condition for wave equations, called the perfectly matched layer (PML) has attracted many researchers. PML is first introduced in Berenger's two papers [14] and [15]. Contributions to the well-posedness of PML are given by Abarbanel et al in [1] (and also [2]). Applications of PML in wave problems can be found, e.g. in [95] and [93]. In our research, the use of PML is not investigated in detail.*

Instead, we will use in some of numerical tests (in Chapter 6) a classical method similar to PML, which is dubbed as "sponge" layers (or absorption layers). These layers are added around the physical domain whose functions is to damp the out going waves. Therefore, the "damped" Helmholtz equation is used in this layer. On the boundaries of the absorption layers, the first order Sommerfeld radiation condition is imposed.

2.5 The properties of the linear system

Discretization of the Helmholtz equation (2.1) and boundary conditions (2.23)–(2.25) (or (2.26)) leads to a linear system

$$A_h u_h = g_h, \; A_h \in \mathbb{C}^{N\times N}, \; u_h, g_h \in \mathbb{C}^N. \tag{2.27}$$

A_h is complex-valued because of the inclusion of radiation boundary conditions. The matrix A_h is *indefinite* for sufficiently large wavenumbers k. Here, the term "indefinite" means that the real part of eigenvalues of A_h has both positive and negative signs. A_h is also symmetric, but non-Hermitian, i.e. $A_h^* \neq A_h$. So, A_h is not self-adjoint. Furthermore, A_h is extremely ill-conditioned.

The two properties: indefiniteness and ill-conditioning are of highest important if we consider an iterative method as a solver for (2.27). Without being too detailed here, standard iterative methods mentioned in Table 1.1 simply do not work well for (2.27).

2.6 Model problems

Throughout this thesis we present numerical solutions for different problems in 2D (and 3D). Some problems are used only for a preliminary evaluation of the numerical performance of the method, for example the "closed-off" and the three-layer models used in Chapter 4. Other problems are designed to mimic 2D geophysical applications. We use the latter as our model problems (MP) which are frequently used throughout this thesis. The model problems (MP) represent an increasing level of difficulty which is suitable to test the robustness and efficiency of the iterative methods proposed in this thesis. Problems in three-dimensional space are introduced separately in Chapter 7.

MP1. 2D constant wave number.

For $\alpha \in \mathbb{R}$ and $k = constant$, find $u \in \mathbb{C}^N$ satisfying:

$$\begin{cases} -\Delta u - (1-\alpha)k^2 u = g, \quad \text{in } \Omega = (0,1)^2 \\ g = \delta(x_1 - \frac{1}{2}, x_2 - \frac{1}{2}), \quad x_1 = (0,1), x_2 = (0,1) \\ \textit{with the first or second order radiation conditions on } \Gamma \equiv \partial\Omega. \end{cases} \tag{2.28}$$

Here δ is the Dirac delta function:

$$\delta(x_1, x_2) = \begin{cases} 1, & x_1 = 0, x_2 = 0 \\ 0, & \text{elsewhere.} \end{cases} \tag{2.29}$$

A typical real part of numerical solution of MP1 is shown in Figure 2.2, for $k = 50$, $\alpha = 0$, obtained by the 5-point and 9-point stencil ($\gamma = 2/5$, minimal dispersion) with $h = 1/80$. Here for the 5-point stencil we have used the second order radiation conditions. In case of 9-point stencil extra damping layers are used, surrounding the physical domain. Unphysical reflections at the boundaries are not present due to the boundary treatment.

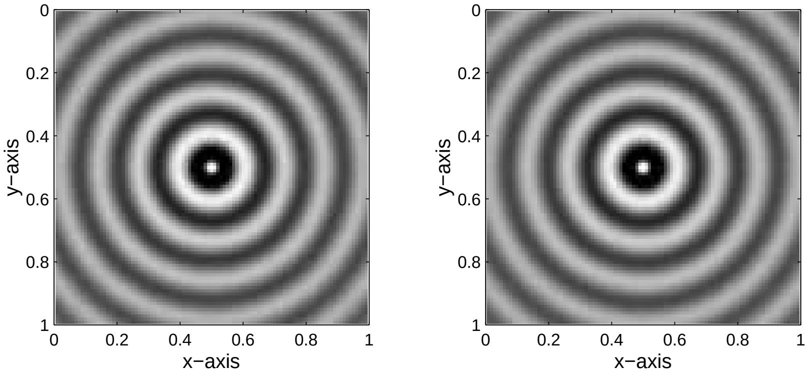

Figure 2.2: Numerical solution (real part) at $k = 50$, $\alpha = 0$ for the model problem MP1 with k constant. Left: 5-point stencil with 2nd order radiation boundary conditions. Right: 9-point stencil ($\gamma = 2/5$) and 1st order radiation boundary conditions imposed on the damping layer

MP2. 2D wedge problem.

This is a problem which mimics three layers with a simple heterogeneity, taken from [81].

For $\alpha \in \mathbb{R}$, find $u \in \mathbb{C}^N$ satisfying:

$$\begin{cases} -\Delta u - (1-\alpha)k^2(\boldsymbol{x})u = g, \quad \text{in } \Omega = (0,600) \times (0,1000) \text{ meter}^2 \\ g = \delta(x_1 - 300, x_2), \qquad x_1 = (0,600), x_2 = (0,1000) \\ \textit{with the first or second order radiation conditions on } \Gamma \equiv \partial\Omega. \end{cases} \tag{2.30}$$

So, $k = k(x_1, x_2) = 2\pi f / c(x_1, x_2)$ is given as in Figure 2.3a in terms of the local speed of sound, c. The real parts of the solutions for $f = 30$, $h = 2.6$ m and with $\alpha = 0$ are shown in Figure 2.3b and 2.3c. In Figure 2.3b the 5-point stencil is used with the second order radiation conditions. In Figure 2.3c the 5-point stencil is used with sponge layers added. At the boundaries of the sponge layers, the first order Sommerfeld radiation condition is imposed.

MP3. 2D Marmousi problem.

This problem mimics the earth's subsurface layer with a complex heterogeneity [18, 81]. It represents a part of the full Marmousi problem, a relevant and difficult test in geophysics.

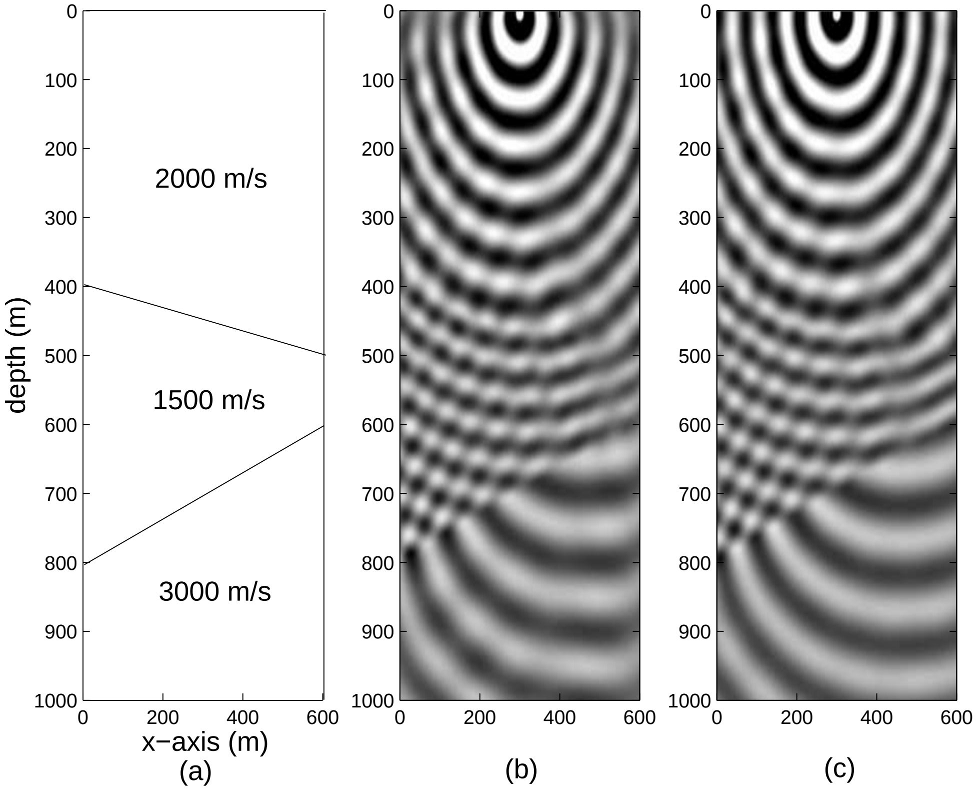

Figure 2.3: MP2. Wedge problem for $f = 30$ Hz, $\alpha = 0$ (no damping): (a) Problem geometry with velocity profile indicated, (b) Real part of numerical solution: 5-point stencil, second order radiation condition (c) Real part of numerical solution: 5-point stencil, sponge layers, first order Sommerfled radiation condition

For $\alpha \in \mathbb{R}$, find $u \in \mathbb{C}^N$ satisfying:

$$\begin{cases} -\Delta u - (1-\alpha)k^2(\boldsymbol{x})u = g, \quad \text{in } \Omega = (0,6000) \times (0,1600) \text{ meter}^2 \\ g = \delta(x_1 - 3000, x_2), \quad x_1 = (0,6000), x_2 = (0,1600) \\ \textit{with the first or second order radiation conditions on } \Gamma \equiv \partial\Omega. \end{cases} \tag{2.31}$$

Here $k = k(x_1, x_2) = 2\pi f/c(x_1, x_2)$ is given as in Figure 2.4a in terms of the local speed of sound, c. The real parts of the solutions for $f = 20$ Hz and $\alpha = 0$ (no damping) and $\alpha = 0.025$ (with damping) are shown in Figures 2.4 and 2.5, respectively. There we compare two discretizations: the 5-point (with the second order radiation condition) and the 9-point stencil. For the latter, absorption layers are added to reduce spurious reflections from the boundaries.

For both cases, we set $h = 4$ m.

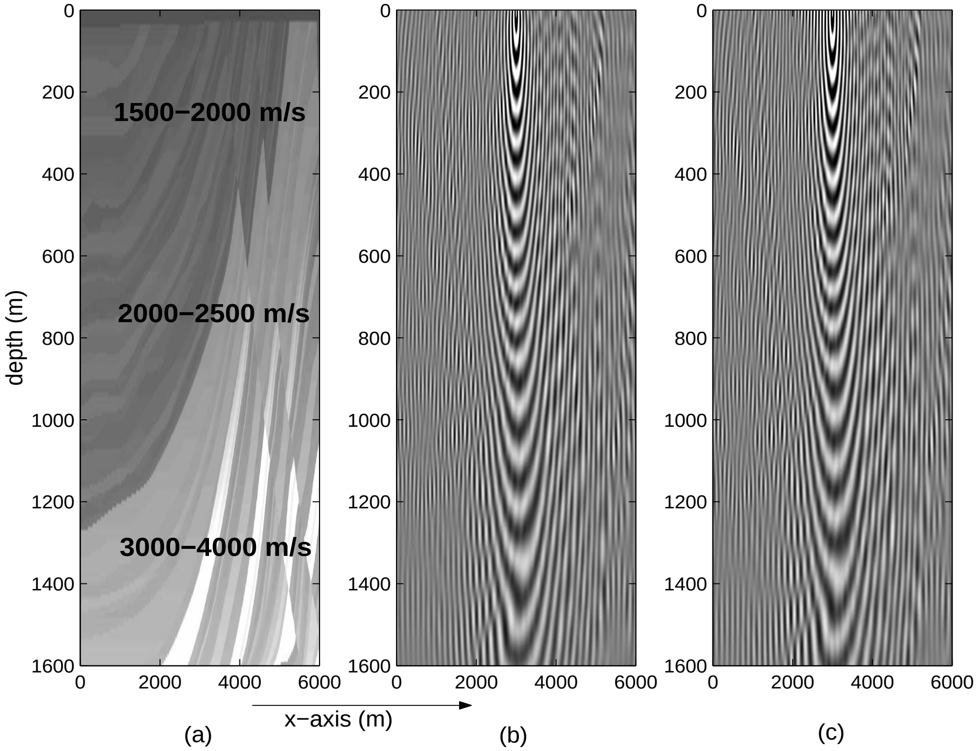

Figure 2.4: MP3. Marmousi problem (not to scale). a) Velocity distribution in meter/s, b) real part of the solution for $f = 20$ Hz, 5-point stencil, c) real part of the solution for $f = 20$ Hz, 9-point stencil ($\gamma = 2/5$) with absorption layers. In both cases, $\alpha = 0$

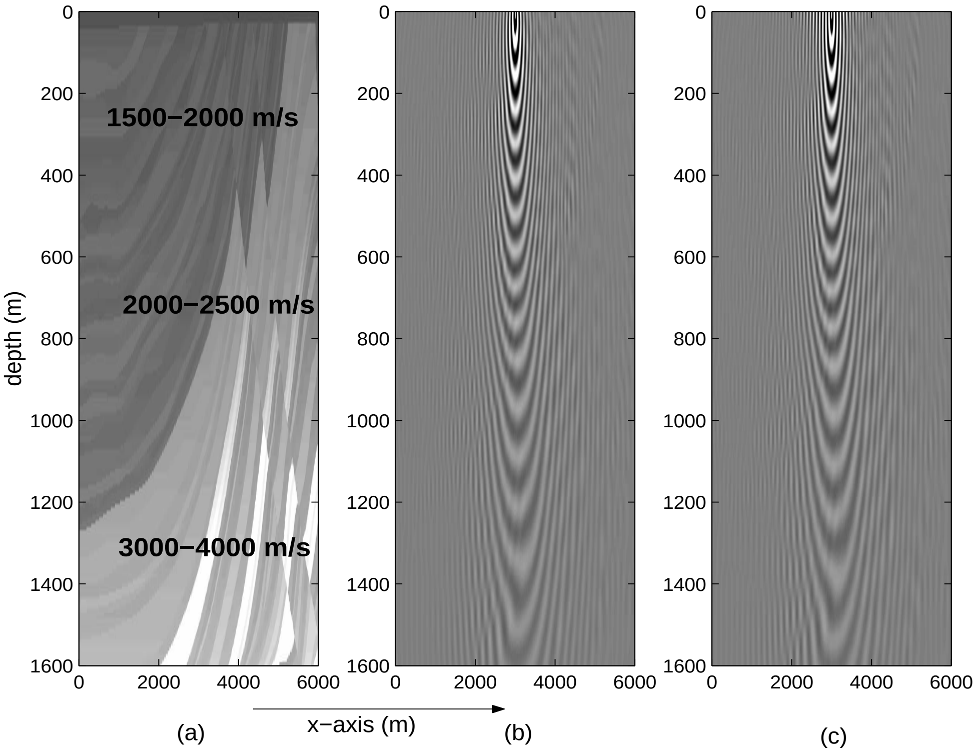

Figure 2.5: MP3. Marmousi problem (not to scale). a) Velocity distribution in meter/s, b) real part of the solution for $f = 20$ Hz, 5-point stencil c) real part of the solution for $f = 20$ Hz, 9-point stencil ($\gamma = 2/5$). In both cases, $\alpha = 0.025$.

Chapter 3

Krylov subspace iterative methods

In this chapter some iterative methods for solving the linear system

$$A_h u_h = g_h. \tag{3.1}$$

are reviewed. Unless mentioned otherwise, A_h is a square, complex, non-symmetric matrix.

We begin our discussion by introducing standard iterative methods. Section 3.2 is devoted to Krylov subspace iterative methods. We give some preliminary numerical results for the Helmholtz equation in Section 3.7 for the so-called *unpreconditioned* Krylov subspace methods. To accelerate the convergence preconditioning is presented in Section 3.8.

In this chapter we omit the subscript "$_h$" from matrices and vectors for convenience.

3.1 Basic iterative methods

Let us consider a splitting

$$A = F - G, \quad F, G \in \mathbb{C}^{N \times N},$$

where N is the number of unknowns. After substitution into (3.1), we have

$$(F - G)u = g \iff Fu = g + Gu. \tag{3.2}$$

For u^{j-1} the approximate solution after the $(j-1)$-th iteration, the new approximation u^j can be computed as

$$Fu^j = g + Gu^{j-1} \Longrightarrow u^j = F^{-1}(g + Gu^{j-1}). \tag{3.3}$$

Thus,

$$\begin{aligned} u^j &= F^{-1}g + (I - F^{-1}A)u^{j-1} \\ &= u^{j-1} + F^{-1}r^{j-1}, \end{aligned} \tag{3.4}$$

with $r^{j-1} := g - Au^{j-1}$ the residual after the $(j-1)$-th iteration, and I the identity matrix. Equation (3.4) is called the *basic iterative* method.

The basic iteration is distinguished by the way the splitting is chosen. If the splitting is defined by $A = D - E$, where $D = \text{diag}(A)$, the *Jacobi* iteration results, namely

$$u^j = u^{j-1} + D^{-1}r^{j-1}. \tag{3.5}$$

The *Gauss-Seidel* iteration is obtained from the splitting $A = L - U$, where L and U are lower and upper triangular matrices, respectively. This iteration is written as

$$u^j = u^{j-1} + L^{-1}r^{j-1}. \tag{3.6}$$

3.2 Krylov subspace methods

The Krylov subspace iteration methods are developed based on a construction of consecutive iterants in a Krylov subspace, i.e. a subspace of the form

$$\mathcal{K}^j(A, r^0) = \text{span}\left\{r^0, Ar^0, A^2r^0, \ldots, A^{j-1}r^0\right\}, \tag{3.7}$$

where $r^0 := g - Au^0$ is the initial residual, with u^0 the initial solution. The dimension of $\mathcal{K}^j$ is equal to j and increases by one at each step of the approximation process.

The idea of Krylov subspace methods can be outlined as follows. For an initial solution u^0, approximations u^j to the solution u are computed every step by iterants u^j of the form

$$u^j \in u^0 + \mathcal{K}^j(A, r^0), \quad j > 1. \tag{3.8}$$

The Krylov subspace $\mathcal{K}^j$ is constructed by the basis $v^1, v^2, \ldots, v^j$, where

$$V^j = [v^1, v^2, \ldots, v^j] \in \mathcal{K}^j. \tag{3.9}$$

With residual $r^j = g - Au^j$, (3.7) gives an expression for the residual at the j-th step

$$r^j = r^0 - AV^jy^j, \tag{3.10}$$

where $y^j \in \mathbb{C}^N$ and $u^j = u^0 + V^jy^j$. From (3.10) we observe that Krylov subspace methods rely on constructing the basis of V^j and the vector y^j. In general we identify two methods that can be used for constructing the basis of

V^j: Arnoldi's method and Lanczos' method. The vector y^j can be constructed by a residual projection or by a residual norm minimization method.

In the following subsections, we present some Krylov subspace algorithms, that are used in numerical simulations in this thesis. In particular, we discuss the Conjugate Gradient (CG) (see e.g. [83]), GMRES [84] and Bi-CGSTAB [97] algorithms. CG can be used to derive CGNR (which can be used for the Helmholtz equation) and as a basis for COCG [98], a CG-like method for complex, symmetric matrices.

Some authors, e.g. [47] have also used QMR [46] and its symmetric version SQMR [44] for the Helmholtz equation.

3.3 Conjugate gradient

Here we assume the matrix A to be symmetric, positive definite (SPD). In CG one constructs a vector $u^j \in \mathcal{K}^j(A, r^0)$ such that $\|u - u^j\|_A$ is minimal, where $\|u\|_A = (Au, u)^{\frac{1}{2}}$. For this purpose, the vector u^{j+1} is expressed as

$$u^{j+1} = u^j + \alpha^j p^j, \tag{3.11}$$

where p^j is the search direction. The residual vectors satisfy the recurrence

$$r^{j+1} = r^j - \alpha^j A p^j. \tag{3.12}$$

For all r^j's to be orthogonal, the necessary condition is that $(r^{j+1}, r^j) = 0$. Here (a, b) denotes the standard Hermitian inner product $a^* b$, which reduces to the transposed inner product $a^T b$ if $a, b \in \mathbb{R}^N$. Thus,

$$(r^{j+1}, r^j) = 0 \to (r^j - \alpha^j A p^j, r^j) = 0, \tag{3.13}$$

which gives

$$\alpha^j = \frac{(r^j, r^j)}{(Ap^j, r^j)}. \tag{3.14}$$

Since the next search direction p^{j+1} is a linear combination of r^{j+1} and p^j, i.e.,

$$p^{j+1} = r^{j+1} + \beta^j p^j, \tag{3.15}$$

the denominator in (3.14) can be written as $(Ap^j, p^j - \beta^{j-1} p^{j-1}) = (Ap^j, p^j)$, since $(Ap^j, p^{j-1}) = 0$. Also because $(Ap^{j+1}, p^j) = 0$, we find that

$$\beta^j = \frac{r^{j+1}, r^{j+1}}{r^j, r^j}. \tag{3.16}$$

The CG algorithm is summarized as follows:

Algorithm 3.1. Conjugate gradient, CG

```
1. Set the initial guess: $u^0$. Compute $r^0 = g - Au^0$. Set $p^0 = r^0$.
2. Do $j = 0, 1, \ldots$
3.     $\alpha^j = (r^j, r^j)/(Ap^j, p^j)$.
4.     $u^{j+1} = u^j + \alpha^j p^j$. If accurate then quit.
5.     $r^{j+1} = r^j - \alpha^j Ap^j$.
6.     $\beta^j = (r^{j+1}, r^{j+1})/(r^j, r^j)$.
7.     $p^{j+1} = r^{j+1} + \beta^j p^j$.
8. Enddo
```

It can be shown that this algorithm minimizes $\|u - u^j\|_A$ over $\mathcal{K}^j(A, r^0)$.

Algorithm 3.1 has the nice properties that it requires only short recurrences, one matrix/vector multiplication and a few vector updates. However, for general matrices the algorithm may not converge, because the orthogonality condition cannot be satisfied. In case the definiteness of A (like for the Helmholtz equation) is not guaranteed, the product $(V^j)^* AV^j =: T^j$ is possibly singular or nearly singular. Since $x^j = V^j T^j \beta e_1$, for T^j nearly/possibly singular, x^j is poorly determined then, see [79].

One simple remedy for the indefiniteness is to apply the CG algorithm to the normal equations A^*A. In this case one considers the linear system

$$A^*Au = A^*g. \tag{3.17}$$

For $u \neq 0$, $u^*A^*Au = (Ax, Ax) > 0$. Hence, the product A^*A is positive definite. Furthermore, if $A \in \mathbb{C}^{N \times N}$, the product A^*A is Hermitian, because $(A^*A)^* = A^*A$. Therefore, Algorithm 3.1 can be applied to the linear system (3.17).

Direct implication of Algorithm 3.1 applied on (3.17) results in the following algorithm, called CGNR.

Algorithm 3.2. CGNR

```
1. Set the initial guess: $u^0$. Compute $r^0 = g - Au^0$.
2. Compute $z^0 = A^*r^0$. Set $p^0 = r^0$.
3. Do $j = 0, 1, \ldots$
4.     $w^j = Ap^j$.
5.     $\alpha^j = (z^j, z^j)/(Ap^j, Ap^j)$.
6.     $u^{j+1} = u^j + \alpha^j p^j$. If accurate then quit.
7.     $r^{j+1} = r^j - \alpha^j Ap^j$.
8.     $z^{j+1} = A^*r^{j+1}$.
9.     $\beta^j = (z^{j+1}, z^{j+1})/(z^j, z^j)$.
10.    $p^{j+1} = z^{j+1} + \beta^j p^j$.
11. Enddo
```

In general, Algorithm 3.2. is twice as expensive as Algorithm 3.1. Furthermore, A^* is sometimes not readily available, meaning one has to compute A^* each time it is required. In this situation, it is better to have an algorithm for general matrices where A^* needs not necessarily be computed explicitly. This is one of the two main reasons why Bi-CGSTAB (discussed in Section 3.6) is more preferable for our problems. The other reason will be explained shortly when we discuss the convergence of CG iterations.

For algorithms with optimality properties, like CG (and also GMRES), convergence estimates can be derived. This property is essential, especially in Chapter 4, when we are considering the convergence behavior of Krylov subspace iterations for the Helmholtz equation. Here, we present one of the convergence estimates of CG. We refer the reader to, e.g., [83] for more details.

Given $\kappa = \lambda_{\max}/\lambda_{\min}$, the l_2 condition number of an SPD matrix A, where $\lambda_{\max}$ and $\lambda_{\min}$ represent the maximum and the minimum eigenvalues of A, respectively. The error between the exact solution u_h and the approximate solution after the j-th CG iteration can be bounded by the following upper bound:

$$\|u - u^j\|_A \leq 2 \left[\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right]^j \|u - u^0\|_A. \tag{3.18}$$

It is clear that CG converges faster for a matrix A with a smaller condition number κ.

For the normal equations and A an SPD matrix, $\kappa(A^*A) = \kappa^2(A)$. Thus one can immediately conclude that CGNR (Algorithm 3.2) is far less efficient than CG. (This conclusion is generally not true for a non SPD matrix. For A indefinite, CG may diverge, whereas CGNR will still converge, even though the convergence is very slow.)

3.4 Conjugate orthogonal-conjugate gradient

The next short recurrence algorithm which requires only one matrix/vector multiplication can also be derived for a complex, symmetric but *non-Hermitian* matrix A by replacing the orthogonality condition (3.13) by the so-called conjugate orthogonality condition [98]:

$$(\overline{r}^j, r^i) = 0 \quad \text{if } j \neq i. \tag{3.19}$$

Since $A = A^T$ we have that $(\overline{u}, Ay) = (\overline{A}^T\overline{u}, y) = (\overline{Au}, y) = (\overline{y}, Au)$. One can prove that the vectors $\{r^0, \ldots, r^j)$ form a basis for $\mathcal{K}^{j+1}(A, r^0)$. Furthermore, the vectors r^j are conjugate orthogonal.

Based on r^j, the solution u^{j+1} can be constructed in $\mathcal{K}^{j+1}(A, r^0)$ that satisfies the condition that the residual $r^{j+1} = g - Au^{j+1}$ is conjugate orthogonal to $\mathcal{K}^{j+1}(A, r^0)$. This results in an algorithm which is similar to CG, but with the (Hermitian) inner product being replaced by the standard inner product. The algorithm, called conjugate orthogonal-conjugate gradient (COCG) is coded as follows:

Algorithm 3.3. Conjugate orthogonal-conjugate gradient (COCG)

```
1. Set the initial guess: $u^0$. Compute $r^0 = g - Au^0$. Set $p^0 = r^0$.
2. Do $j = 0, 1, \ldots$
```

```
        $\alpha^j = (\overline{r}^j, r^j)/(\overline{Ap^j}, p^j)$.
        $u^{j+1} = u^j + \alpha^j p^j$. If accurate then quit.
        $r^{j+1} = r^j - \alpha^j Ap^j$.
        $\beta^j = (\overline{r}^{j+1}, r^{j+1})/(\overline{r}^j, r^j)$.
        $p^{j+1} = r^{j+1} + \beta^j p^j$.
    Enddo
```

Algorithm 3.3, however, has several drawbacks. First of all, the algorithm is no longer optimal. This is because the error $\|u - u^j\|_A$ at every step is not minimal. The error, therefore, is not monotonically decreasing. In many cases, the lack of optimality condition for COCG is characterized by an erratic convergence behavior. Residual smoothing [110] can be used to obtain a rather monotonic decreasing convergence.

Secondly, the COCG algorithm may suffer from two kinds of breakdown. The first breakdown is related to the *quasi null-space* formed in the algorithm. The quasi null-space is characterized by the situation that for $r^j \neq 0$, $(\overline{r}^j, r^j) = 0$. This breakdown can be overcome by restarting the process or by switching to another algorithm. The second breakdown is due to the rank deficiency of the linear system for computing the coefficients y^j. This type of breakdown is, however, uncured.

Thirdly, the condition $A = A^T$ is often too restrictive for preconditioning. We postpone the discussion of this issue until Section 3.8, where we introduce preconditioners to enhance the convergence of Krylov subspace methods.

3.5 Generalized Minimal Residual, GMRES

The GMRES method is an iterative method for nonsymmetric matrices, which minimizes the residual norm over the Krylov subspace. The GMRES algorithm is given as follows (see Saad [83] and Saad and Schultz [84]).

Algorithm 3.4. Generalized Minimum Residual, GMRES

```
1. Choose $u^0$. Compute $r^0 = g - Au^0$, $\beta := \|r^0\|_2$ and $v^1 := r^0/\beta$
2. For $j = 1, 2, \ldots, m$ do:
3.     Compute $w^j := Av^j$
4.     For $i = 1, 2, \ldots, m$ do:
5.         $h_{i,j} := (w^j, v^i)$, $w^j := w^j - h_{i,j}v^i$
6.     Enddo
7.     $h_{j+1,j} = \|w^j\|_2$.
8.     $v^{j+1} = w^j / h_{j+1,j}$
9. Enddo
10. Compute $y^m$: the minimizer of $\|\beta e_1 - \bar{H}_m y\|_2$ and $u^m = u^0 + V^m y^m$
```

Line 2 to 9 represent the Arnoldi algorithm for orthogonalization. In line 10, we define a minimalization process by solving a least squares problem

$$J(y) = \|g - Au\|_2, \tag{3.20}$$

where

$$u = u^0 + V^k y \tag{3.21}$$

is any vector in $\mathcal{K}^k$. Except for line 10, the GMRES algorithm is almost identical with the Full Orthogonalization Method (FOM) [82]. (In FOM, we compute $y^m = H_m^{-1}(\beta e_1)$, where e_1 is the first unit vector.) Inserting the expression of u (3.21) to (3.20) and making use of the following property:

$$AV^j = V^{j+1}\bar{H}_j \tag{3.22}$$

(see Preposition 6.5 in [83]), we arrive at the following result:

$$J(y) = \|\beta e_1 - \bar{H}_m y\|_2, \tag{3.23}$$

which is subjected to minimization. In (3.23), $\beta = \|r^0\|$, and $e_1 \equiv \{1, 0, \cdots, 0\}^T$.

The GMRES algorithm may break down if at iteration step j, $h_{j+1,j} = 0$ (see line 8). However, this situation implies that residual vector is zero and therefore, the algorithm gives the exact solution at this step. Hence, examination of value $h_{j+1,j}$ in algorithm step 7 becomes important. For the convergence of GMRES, the following theorem is useful.

Theorem 3.5.1 Convergence of GMRES [84]. *Suppose that A is diagonalizable so that $A = X\Lambda X$, where $\Lambda = diag\{\lambda_1, \ldots, \lambda_N\}$ with positive real parts. Assume that all eigenvalues are enclosed by a circle centered at $C_c > 0$ and having radius $R_c < C_c$ (so the circle does not enclose the origin). Then the residual norm at the j-th GMRES iteration satisfies the inequality*

$$\|r^j\|_2 \leq \left(\frac{R_c}{C_c}\right)^j \kappa(X)\|r^0\|_2. \tag{3.24}$$

If iteration number m is large, the GMRES algorithm becomes impractical because of memory and computational requirements. This is understandable from the fact that during the Arnoldi steps the number of vectors requiring storage increases. To remedy this problem, the algorithm can be restarted. The restarted GMRES follows the idea of the original GMRES except that after the l-th step, the algorithm is repeated by setting $l = 0$.

The restarted GMRES algorithm may, however, lead to difficulties if A is not positive definite [83]. For such a type of matrix, the algorithm can stagnate, which is not the case for the non-restarted GMRES algorithm (that is guaranteed to converge in at most N steps [84]).

3.6 Bi-CGSTAB

For non-symmetric matrices an iterative method can be developed based on the non-symmetric Lanczos algorithm. The basis of this algorithm is the construction of two biorthogonal bases for two Krylov subspaces: $\mathcal{K}(A, r^0)$ and $\mathcal{L}(A^*, \widehat{r}^0)$.

As shown by Fletcher [42], the resulting algorithm also has short recurrences. The algorithm, called BiCG, solves not only the original linear system $Au = g$, but also the dual system $A^*\widehat{u} = g$, which is usually not required. Sonneveld [87] makes the observation that throughout the BiCG process until convergence, only $\{r^j\}$ is exploited. He further notices that the vectors $\widehat{r}^j$ can be constructed from BiCG polynomials $\phi(A)$. They can be recovered during the BiCG process, by the relation $\widehat{r}^j = \phi_j^2(A)r^0$. This results in an algorithm, called CGS, which does not require multiplication by A^*. CGS, however, suffers from an irregular convergence.

A smoothly converging variant of CGS is proposed by van der Vorst [97] by introducing another polynomial relation for $\widehat{r}^j$ of the form $\widehat{r}^j = \phi_j(A)\psi_j(A)r^0$. The residual polynomial ϕ_j is associated with the BiCG iteration while ψ_j is another polynomial determined from a simple recurrence in order to stabilize the convergence behavior. The Bi-CGSTAB algorithm is then written as follows:

Algorithm 3.5. Bi-CGSTAB

```
1.  Compute $r^0 := g - Au^0$; $\widetilde{r}$ arbitrary.
2.  $p^0 := r^0$
3.  For $j = 1, 2, \cdots$, *until convergence* do:
4.      $\alpha^j := (r^j, \widetilde{r}/(Ap^j, \widetilde{r})$
5.      $s^j := r^j - \alpha^j Ap^j$
6.      $\omega^j := (As^j, s^j)/(As^j, As^j)$
7.      $u^{j+1} := u^j + \alpha^j p^j + \omega^j s^j$
8.      $r^{j+1} := s^j - \omega^j As^j$
9.      $\beta^j := (r^{j+1}, \widetilde{r})/(r^j, \widetilde{r}) \times \alpha^j/\omega^j$
10.     $p^{j+1} := r^{j+1} + \beta^j(p^j - \omega^j Ap^j)$
11. Enddo
```

Bi-CGSTAB is an attractive alternative to CGS. However, if parameter ω_j in step 6 gets very close to zero the algorithm may stagnate or break down. Numerical experiments confirm that this is likely to happen if A is real and has complex eigenvalues with an imaginary part larger than the real part. In such a case, one may expect that the situation where ω is very close to zero can be better handled by the minimum-residual polynomial $\psi(A)$ of higher order. Improvements to Bi-CGSTAB have been proposed, e.g., by Gutknecht [52] and Sleijpen and Fokkema [86]. Gutknecht [52] proposed the use of a second order polynomial for $\psi(A)$ in his BiCGSTAB2 algorithm. However, it becomes clear from experiments that for the same situation in which BiCGSTAB breaks down, BiCGSTAB2 stagnates or also breaks down. Sleijpen and Fokkema [86] proposed a modification by forming a general l-th order polynomial for $\psi(A)$, resulting in the BiCGSTAB(l) algorithm. For $l = 1$, the algorithm resembles van der Vorst's BiCGSTAB.

Numerical experiments reveal that BiCGSTAB(l) improves BiCGSTAB in the case of stagnation or breakdown. It is also known that a large l can be chosen to improve the convergence. In general, BiCGSTAB(l) is superior to BiCGSTAB or BiCGSTAB2, but it is also more costly.

3.7 Numerical example

Example 3.1. A first numerical result is shown in Table 3.1, which compares the convergence of several Krylov subspace methods applied directly to Model Problem 1 (constant k). We set $\alpha = 0$; so, we consider the undamped Helmholtz equation. Here we use the first order radiation boundary conditions (2.26) on Γ. The problem is discretized by the 5-point finite difference stencil.

Starting with the initial guess $u^0 = 0$, the iteration is terminated at the j-th iteration if the following condition is satisfied:

$$\|g - Au^j\|/\|g\| \leq 10^{-7}. \tag{3.25}$$

(Unless mentioned otherwise, we use (3.25) as the termination criterion in this thesis). We set $kh = 0.625$, which is equivalent to choosing ∼10 grid points per wavelength (see Table 2.1).

From Table 3.1, the performance of all algorithms decreases as k increases, and COCG seems to be the method of choice (short recurrence and with comparable convergence as GMRES). Convergence history for $k = 30$ is displayed in Figure 3.1. An erratic convergence behavior is observed for COCG. Superlinear convergence is clearly seen for GMRES and CGNR. A long, slow converging part of CGNR convergence is due to many small eigenvalues of A^*A.

Table 3.1: Number of iterations of several Krylov subspace methods to reach convergence for Model Problem 1. $kh = 0.625$ (∼ 10 grid points per wavelength)

k	CGNR	COCG	GMRES	Bi-CGSTAB
10	49	33	32	35
20	208	80	79	145
30	492	154	143	402
40	943	255	241	816

Figure 3.2 shows the effect of increasing the number of grid points per wavelength on the number of iterations for $k = 10$. For all algorithms, the convergence slows down if more accurate solutions are needed. Among them, CGNR is the most sensitive algorithm. This is somewhat disappointing; as we tend to increase the accuracy of the solution, the performance of the iterative methods without any enhancement degrades.

3.8 Preconditioning

Even though the methods are convergent for the numerical examples given in Section 3.7, the convergence is too slow. As we only considered very simple problems (with small k), one may imagine that the methods are inefficient if a

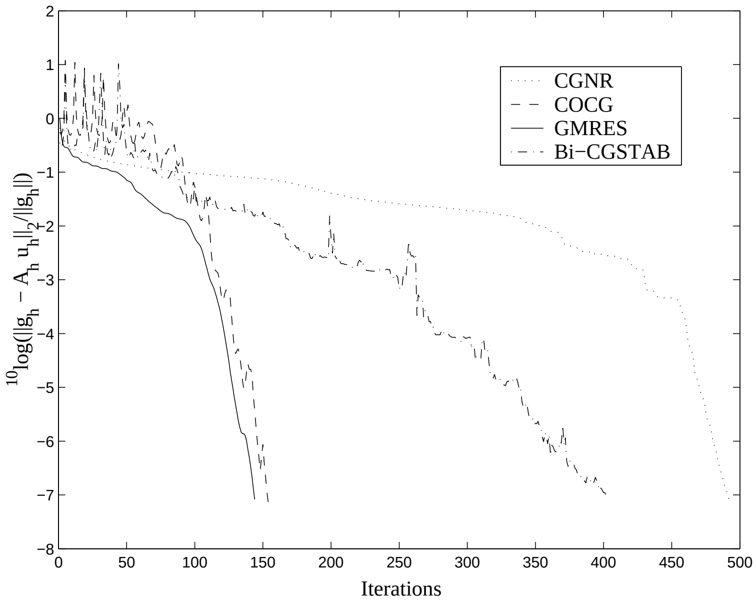

Figure 3.1: Convergence history of CGNR, COCG, GMRES and Bi-CGSTAB for Model Problem 1 with $k = 30$, $\alpha = 0$

problem with, say $k = 500$, is to be solved. In particular for GMRES, many iterations means a large memory required to store all orthogonal vectors. A small restart parameter does not bring any improvement to the convergence of GMRES, because it leads to many more iterations.

As already discussed in Section 3.3, the convergence of the CG iterations (3.18) is closely related to the condition number κ. The purpose of preconditioning is to improve the condition of the system to be solved.

Suppose there exists a matrix M such that its inverse is easily computed or approximated. By preconditioning we solve the following *equivalent linear system*:

$$M^{-1}Au = M^{-1}g. \tag{3.26}$$

A preconditioned Krylov subspace method can therefore be defined as find the approximate solution u by constructing iterants u^j such that

$$u^j \in u^0 + \mathcal{K}^j(M^{-1}A, r^0), \quad j > 1 \tag{3.27}$$

One important aspect from (3.26) is that if CG is applied to (3.26), its convergence depends on the condition number of $M^{-1}A$, and not on that of A. Therefore, in order to have an improved convergence the product $M^{-1}A$ must have a smaller condition number compared to A, i.e. $\kappa(M^{-1}A) < \kappa(A)$. Otherwise, no benefit can be gained from preconditioning.

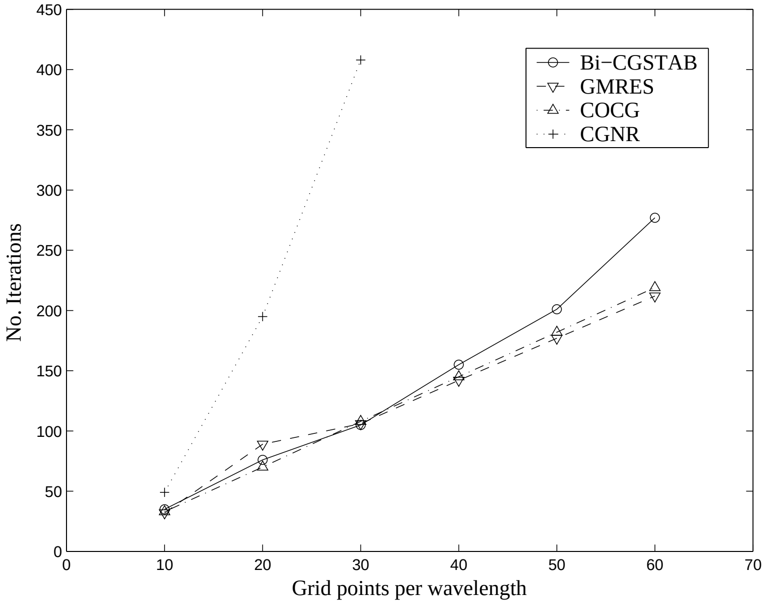

Figure 3.2: Performance of CGNR, COCG, GMRES and Bi-CGSTAB for Model Problem 1 with respect to the increase of the number of grid points per wavelength, for $k = 10$

Remark 3.8.1 *Besides formula (3.26), which is called left preconditioning, we also identify right preconditioning for an equivalent linear system:*

$$AM^{-1}\widetilde{u} = g, \ \textit{where } \widetilde{u} = Mu. \tag{3.28}$$

Furthermore, if $M = M_1M_2$, we can write an equivalent system

$$M_1^{-1}AM_2^{-1}\widetilde{u} = M^{-1}g, \ \textit{where } \widetilde{u} = Mu. \tag{3.29}$$

We call this a split preconditioner.

Remark 3.8.2 *The spectra of $M^{-1}A$, AM^{-1} and $M_1^{-1}AM_2^{-1}$ are identical. In the following chapters, we will see that it is easier to use left preconditioning when we theoretically analyze the spectral properties of the preconditioned systems. In practice, we perform right preconditioning as it turns out to give cheaper algorithms.*

3.8.1 Incomplete LU decomposition

Suppose that $M^{-1} = A^{-1}$, then $\kappa(M^{-1}A) = 1$. But, computing A^{-1} is as expensive as solving the original problem. One frequently used preconditioner for A can be obtained by approximately decomposing A into incomplete LU

factors [83], where L and U are lower and upper triangular matrices, respectively. This is achieved by applying an incomplete LU factorization to A. The degree of approximation of LU depends on the number of fill-in elements allowed in the LU factors. The simplest one is the so-called ILU(0), wherein the same non-zero structure as A is retained in ILU.

A more accurate approximation can be obtained by increasing the level of fill-in. Two scenarios exist. The first is more structure-oriented, and is done by adding more off-diagonals in the LU factors. We denote this as ILU(n_{lev}), which $n_{lev} > 0$ is a reasonably small integer, indicating the level of fill-in. This scenario will result in a structured linear system for L and U. The second scenario is related to a drop tolerance of fill-in. So, this scenario is more value-oriented. If during an LU factorization the value of an element falls below a prescribed tolerance ϵ, which is small, this element is set to zero. We denote this incomplete LU decomposition as ILU(ϵ), with ϵ the drop tolerance. An ILU(ϵ) process often leads to unstructured L and U matrices. We refer to [83] for a more detailed discussion on incomplete LU preconditioners.

The inclusion of a preconditioner in the CGNR, COCG, GMRES and Bi-CSGTAB algorithms is straightforward.

Example 3.2. Tables 3.2 and 3.3 show convergence results for Model Problem 1 as in Section 3.7 Example 3.1, obtained after the inclusion of a preconditioner in the COCG, GMRES and Bi-CGSTAB algorithms. (Here we do not compute the solution by CGNR). The preconditioners are ILU(0) and ILU(ϵ) with $\epsilon = 0.01$ (thus, denoted by ILU(0.01)). We take $kh = 0.625$. Compared to the unpreconditioned case, the use of ILU(0) and ILU(0.01) as preconditioners accelerates the convergence significantly, with more fill-in, in general, giving faster convergence for small wavenumber. For large wavenumber, for example $k = 100$, all methods show a slow convergence, with ILU(0.01) becoming ineffective as compared to ILU(0). Since A is not an M-matrix, an ILU factorization applied to A is not a stable process. This may lead to an inaccurate LU approximation of A, and, therefore, lead to a bad preconditioner for A.

Different from the case without preconditioners, COCG turns out to converge slower than the other methods, and does not converge (after 10,000 iterations, Table 3.3) for high wavenumbers if ILU(0.01) is used as preconditioner. This is because ILU(0.01) does not lead to a symmetric preconditioner. Thus, the preconditioned linear system is not guaranteed to be symmetric. GMRES and Bi-CGSTAB result in a comparable convergence in terms of the number of iterations. As Bi-CGSTAB requires two matrix-vector products and two preconditioner solves, the overall performance of Bi-CGSTAB is not better than GMRES. GMRES may, however, suffer from storage problems if larger problems (for an increasing k, as indicated by "–" in Table 3.2 for $k = 50$ and 100) have to be solved, which tend to require too many iterations to reach convergence.

In Table 3.3 we measure the number of nonzero elements in the L and U matrices for ILU(0.01). For increasing k, and thus N, the number of nonzero elements becomes unacceptably large. ILU(0), which results in far less nonzero elements (similar to the nonzero elements of A) in the L and U matrices compared to ILU(0.01) seems to be a more effective preconditioner for the Helmholtz equation in this case.

Table 3.2: Number of iterations of several preconditioned Krylov subspace methods for Model Problem 1. The preconditioner is ILU(0). $kh = 0.625$ ($\sim$ 10 grid points per wavelength)

k	COCG	GMRES	Bi-CGSTAB
10	25	24	22
20	73	63	102
30	164	123	182
40	267	199	296
50	425	–	335
100	>2000[a]	–	672

[a] $\|r\|_2/\|b\|_2 = 6.2 \cdot 10^{-3}$

Table 3.3: Number of iterations of several preconditioned Krylov subspace methods for Model Problem 1. The preconditioner is ILU(0.01). $kh = 0.625$ ($\sim$ 10 grid points per wavelength). COCG stagnates for $k \geq 20$

k	$nz(A)$	$nz(L)$	$nz(U)$	COCG	GMRES	Bi-CGSTAB
10	1065	2023	2008	30	10	6
20	4681	10430	10224	>2000[a]	18	11
30	10857	25355	24806	–	31	23
40	19593	46762	45700	–	48	53
50	30889	74711	73129	–	64	82
100	125769	316167	309097	–	> 150[b]	>2000[c]

[a] $\|r\|_2/\|b\|_2 = 1.2 \cdot 10^{-1}$
[b] $\|r\|_2/\|b\|_2 = 6.7 \cdot 10^{-2}$
[c] $\|r\|_2/\|b\|_2 = 1.3 \cdot 10^{0}$

Example 3.3. The dependence of the number of iterations on the gridsize h in case of preconditioning with ILU(0) and ILU(0.01) for Model Problem 1 is displayed in Figure 3.3 for GMRES and Bi-CGSTAB. With ILU(0), we again observe a strong dependence of the convergence on the gridsize h. With ILU(0.01) this is much less pronounced.

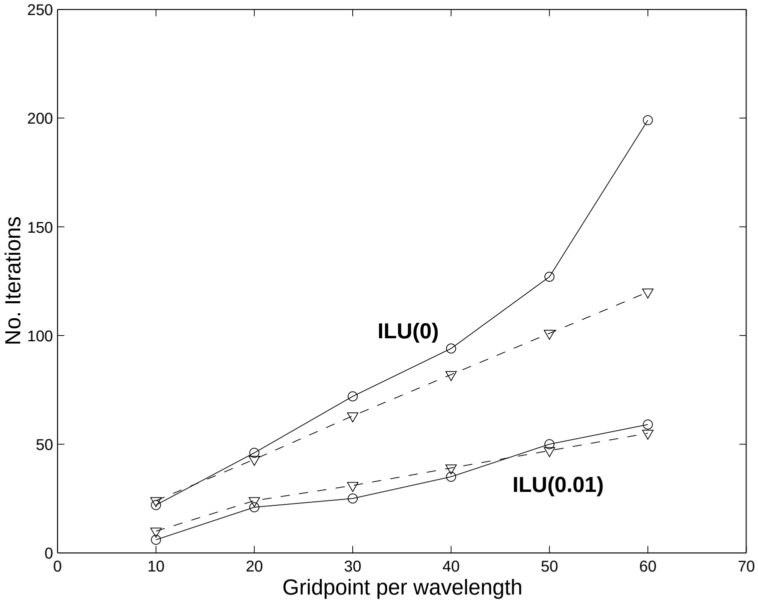

Figure 3.3: Performance (in number of iterations) of GMRES (dashed line) and Bi-CGSTAB (solid line) for Model Problem 1 with respect to an increasing number of grid points per wavelength. The upper part is for preconditioning with ILU(0). The lower part is for ILU(0.01).

3.8.2 Incomplete factorization-based preconditioner

An ILU factorization may not be stable if A is not an M-matrix, which is the case for the discrete Helmholtz equation. This is particularly observed from numerical results in Section 3.8.1 with ILU(0.01) as preconditioner. In the remainder of this chapter we discuss some preconditioners which are suitable for the Helmholtz equation. Therefore, A should now be considered as a matrix which is obtained from a discretization of the Helmholtz equation.

For the indefinite Helmholtz equation, various types of preconditioners have been proposed. Evaluations of incomplete Cholesky factorization applied to "less-indefinite" Helmholtz matrices are given in [69]. Reference [47] shows that a parabolic factorization of the Helmholtz equation on a semi-discrete level can be used as a preconditioner. Reference [81] proposes a separation-of-variables technique to approximately factorize the linear system. The factorization is exact for constant wavenumbers, but becomes somewhat cumbersome in the presence of heterogeneities, that may even lead to divergence of the iterative methods.

In [69] a modification of A is proposed such that ILU factorizations can be constructed safely. For the approximation of A^{-1}, denoted by M_I^{-1}, a constraint is set such that the preconditioned system AM_I^{-1} is definite or "less indefinite" than the original system. Here the term "indefiniteness" is related to the real part of spectrum of the given linear system: one demands that Re $\left(\sigma(AM_I^{-1})\right) > 0$ (or Re $\left(\sigma(AM_I^{-1})\right) < 0$).

For $A \in \mathbb{C}^{N\times N}$, a matrix $\widetilde{A}$ can be extracted from A where the real part of $\widetilde{A}$ is a non-singular symmetric M-matrix [101]. In the situation of interest here (i.e., the Helmholtz equation discretized by the 5-point stencil), if one introduces a parameter $\gamma \geq 1$ and defines

$$\mathrm{Re}(\widetilde{a}_{i,j}) = \begin{cases} \mathrm{Re}(a_{i,j}) & \text{if } i \neq j, \\ \mathrm{Re}(a_{i,j}) - \gamma \min\{0, \mathrm{Re}((Ae)_i)\} & \text{if } i = j, \end{cases}$$

it can be proven that $\mathrm{Re}(\widetilde{A})$ is a non-singular symmetric M-matrix [69]. Then, $\mathrm{Re}(\widetilde{A})$ can be considered as a real perturbation of $\mathrm{Re}(A)$. Since $\mathrm{Re}(\widetilde{A})$ is a symmetric M-matrix, ILU can be applied safely. For the imaginary part, one simply sets

$$\mathrm{Im}(\widetilde{a}_{ij}) = \mathrm{Im}(a_{ij}), \ \forall i, j.$$

In [69] several possible strategies for this preconditioner are proposed. In this thesis, we only describe one of them, namely

$$M_I \equiv \widetilde{A} = A_0 + \hat{j}\mathrm{Im}(A), \ \ A_0 = \mathrm{Re}(A) + Q, \tag{3.30}$$

with

$$q_{ii} = -\min\{0, \mathrm{Re}((Ae)_i)\}. \tag{3.31}$$

We use (3.30)–(3.31) in Chapter 5, in combination with ILU(n_{lev}), $n_{lev} = 0, 1$. Reference [69] evaluates the preconditioner for a finite element discretization of the Helmholtz equation. In principle, one could also use a multigrid method for approximating M_I^{-1}, but since M_I is constructed based on matrix modifications, the multigrid method should preferably be an algebraic multigrid method (see, for example, [92]).

3.8.3 Separation-of-variables preconditioner

It is well known that a separation of variables technique can be used to analytically solve the Laplace equation with special boundary conditions. The zeroth order term $k^2(x_1, x_2)u$ prevents the use of this technique in the same way for the Helmholtz operator. An approximation can, however, be made in the separation of variables context. This approximation can then be used as a preconditioner for the Helmholtz equation.

For $k^2(x_1, x_2)$ an arbitrary twice integrable function, the following decomposition can be made,

$$k^2(x_1, x_2) = k^2_{x_1}(x_1) + k^2_{x_2}(y) + \tilde{k}^2(x_1, x_2), \text{ in } \Omega = [x_{1_a}, x_{1_b}] \times [x_{2_a}, x_{2_b}], \tag{3.32}$$

satisfying the conditions

$$\int_{x_{1_a}}^{x_{1_b}} \tilde{k}^2(x_1, x_2)dx_1 = 0, \forall x_2, \quad \int_{x_{2_a}}^{x_{2_b}} \tilde{k}^2(x_1, x_2)dx_2 = 0, \forall x_1.$$

It can be proven that the decomposition (3.32) is unique [81]. Denoting by K, a matrix representation of the zero-th order term, and L_Δ, the Laplace term, matrix A can be written as

$$A = L_\Delta - K^2 = X_1 + X_2 - K^2,$$

where

$$X_1 = I_{x_2} \otimes A_{x_1}, \;\; X_2 = A_{x_2} \otimes I_{x_1}, \text{ and } K^2 = I_{x_2} \otimes K_{x_1}^2 + K_{x_2}^2 \otimes I_{x_1} + \widetilde{K^2},$$

with $\otimes$ the Kronecker product, I_{x_1}, I_{x_2} identity matrices and $K_{x_1}^2, K_{x_2}^2, \widetilde{K^2}$ diagonal matrices related to (3.32).

It is $\tilde{k}$ in (3.32) which prevents a complete decomposition of A. If we neglect this term, K^2 can be decomposed in the same way as L_Δ. This results in the following separated variables formulation

$$\widehat{A} := X_1 + X_2 - \widehat{K^2} = I_{x_2} \otimes (A_{x_1} - K_{x_1}^2) + (A_{x_2} - K_{x_2}^2) \otimes I_{x_1}, \tag{3.33}$$

where $\widehat{A}$ approximates A up to the term $\widetilde{K^2}$. If wavenumber k is constant then decomposition (3.32) is exact. As $\widehat{A}$ can be further decomposed into a block tridiagonal matrix, it is motivating to use $\widehat{A}$ as a preconditioner for A. We denote this preconditioner throughout this paper by $M_{SV} := \widehat{A}$.

The construction of a block tridiagonal decomposition of M_{SV} involves the singular value decomposition in one direction, e.g. in the x_1-direction. We refer to [81] for more details.

Chapter 4

Shifted Laplace preconditioner

We have seen in Chapter 3 that standard Krylov subspace methods converge too slowly for the 2D Helmholtz equation. Furthermore, the convergence was strongly dependent on the gridsize. Preconditioning based on an incomplete LU factorization of the discrete Helmholtz equation does not effectively improve the convergence for general cases. ILU(0.01) requires an unacceptably large storage, but is still not effective in improving the convergence. ILU(0), on the other hand, only requires a moderate storage. Its convergence is, however, not impressive and sensitive to the gridsize.

In the last three decades efficient preconditioners for indefinite linear systems have attracted attention of many researchers. The linear algebra theory for indefinite linear systems, however, is not as well-developed as the theory for definite linear systems.

For definite, elliptic boundary value problems, theoretical guidelines for preconditioning are provided in [71] which justify the usual suggestion of choosing the leading parts of an elliptic operator as the preconditioner. Reference [5] analyzes preconditioners for an M-matrix, with the rate of convergence of the conjugate gradient method given in [6]. For indefinite linear systems, theoretical analysis of preconditioners can be found in [108]. Implementation of incomplete factorization and block SSOR as preconditioners for indefinite problems is discussed in [45].

In [12],[51] and [64] preconditioners based on definite elliptic operators are used in the context of Krylov subspace methods accelerated by multigrid.

Here we introduce a special class of preconditioners for the Helmholtz equation [37, 38, 102] to effectively improve the convergence of Krylov subspace methods. This class of preconditioners is constructed by discretization of the following operator

$$\mathcal{M}_{(\beta_1,\beta_2)} := -\Delta - (\beta_1 - \hat{j}\beta_2)k^2, \quad \beta_1, \beta_2 \in \mathbb{R}, \tag{4.1}$$

which is called “shifted Laplace operator”. The preconditioners used in [12] and [64] also belong to this class of preconditioners. They can be recovered from (4.1) by setting $(\beta_1, \beta_2) = (0, 0)$ and $(\beta_1, \beta_2) = (-1, 0)$, respectively.

In this chapter we discuss the shifted Laplace preconditioner in detail. To get insight in this class of preconditioners we first look at a 1D analysis for the continuous operator with a “real” shift as used in [12] and [64], followed by a purely “imaginary” shift (introduced in [37, 38]). Then, spectral analysis on the discrete level is given in Section 4.2 under the restriction that the discrete formulation of $\mathcal{M}_{(\beta_1,\beta_2)}$ (4.1) should result in a (complex) symmetric, positive definite matrix (CSPD). A convergence analysis which is based on the convergence rate of GMRES is derived in Section 4.3. We then show in Section 4.4 that the convergence is gridsize-independent. Some preliminary numerical results are presented in Section 4.5, to show the effectiveness of this class of preconditioners for a simple heterogeneous Helmholtz problem.

4.1 1D analysis for constant wavenumbers

In this section some analysis which motivates the development of the shifted Laplace preconditioners is given. For simplicity and clarity we provide the analysis for the “undamped”, one-dimensional Helmholtz equation in this section. Thus, $\alpha = 0$ in (1.26).

4.1.1 Real shift

We consider here a 1D Helmholtz equation in a unit domain $\Omega = (0, 1)$:

$$-\frac{d^2u}{dx_1^2} - k^2u = 0, \tag{4.2}$$

with Dirichlet boundary conditions $u(0) = u(1) = 0$. For simplicity we only consider problems with constant k over Ω. A non-trivial solution for the related *continuous* eigenvalue problem

$$-\left(\frac{d^2}{dx_1^2} + k^2\right)\widetilde{u} = \lambda\widetilde{u}. \tag{4.3}$$

is a general solution of the form $\widetilde{u} = \sin(ax_1)$. This solution satisfies the conditions at $x_1 = 0$ and $x_1 = 1$. By substituting this solution in (4.3) we arrive at the following relation:

$$(k_{\ell_1}^2 - k^2)\sin(\pi\ell_1x_1) = \lambda\sin(\pi\ell_1x_1) \to \lambda^{\ell_1} = k_{\ell_1}^2 - k^2, \tag{4.4}$$

where $k_{\ell_1} = \pi\ell_1$, $\ell_1 \in \mathbb{N}/\{0\}$. Thus, for large wavenumbers k the eigenvalues change sign, indicating the indefiniteness of the problem.

In 1D the preconditioning operator (4.1) reads

$$\mathcal{M}_{\beta_1} := -\frac{d^2}{dx_1^2} - \beta_1k^2, \quad \text{with } \beta \leq 0. \tag{4.5}$$

We restrict ourselves to the preconditioning operator with $\beta_1 \leq 0$, since in this case a finite difference discretization of (4.5) leads to a symmetric, positive definite matrix (SPD). Numerous efficient methods exist for solving such a matrix. In particular, $\beta_1 = 0$ and $\beta_1 = -1$ give preconditioners used by Bayliss et al. [12] and Laird et al. [64], respectively. In relation with the continuous eigenvalue problem, the preconditioned (generalized) eigenvalue problem reads

$$\left(-\frac{d^2}{dx_1^2} - k^2\right)\widetilde{u}^{\ell_1} = \lambda_r^{\ell_1}\left(-\frac{d^2}{dx_1^2} - \beta_1 k^2\right)\widetilde{u}^{\ell_1}. \tag{4.6}$$

By assuming a solution of the form $\widetilde{u} = \sin(ax_1)$, the eigenvalues are found to be

$$\lambda_r^{\ell_1} = \frac{k_{\ell_1}^2 - k^2}{k_{\ell_1}^2 - \beta_1 k^2} = \frac{1 - (k/k_{\ell_1})^2}{1 - \beta_1 (k/k_{\ell_1})^2}, \tag{4.7}$$

where $k_{\ell_1} = \pi\ell_1$, $\ell_1 \in \mathbb{N}/\{0\}$. For $\ell_1 \to \infty$, $\lambda_r^{\ell_1} \to 1$, i.e., the eigenvalues are bounded above by one. For $\ell_1 \to 0$, the low eigenmodes, we have $\lambda_r^{\ell_1} \to 1/\beta_1$. The modulus of this eigenvalue remains bounded unless $-1 \leq \beta_1 \leq 0$. The maximum eigenvalue can therefore be written as

$$|(\lambda_r^{\ell_1})_{\max}| = \max\left(\left|\frac{1}{\beta_1}\right|, 1\right). \tag{4.8}$$

To estimate the smallest eigenvalue, we use a simple but rough analysis as follows. It is assumed that the minimum eigenvalue is very close (but not equal) to zero. This assumption implies a condition $k_{\ell_1} \approx k$ as obtained from (4.7). To be more precise, let $k_{\ell_1} = k + \epsilon$, where ϵ is any small number. By substituting this relation into (4.7), and by neglecting the higher order terms and assuming that $\epsilon k \ll k^2$, we find that

$$(\lambda_r^{\ell_1})_{\min} = \frac{2}{1-\beta_1}\left(\frac{\epsilon}{k}\right). \tag{4.9}$$

From (4.9), the minimum eigenvalue can be very close to zero as β_1 goes to infinity. The condition number of the preconditioned Helmholtz operator now reads

$$\kappa = \begin{cases} \frac{1}{2}(1-\beta_1)k/\epsilon & \text{if } \beta_1 \leq -1, \\ \frac{1}{2|\beta_1|}(1-\beta_1)k/\epsilon & \text{if } -1 \leq \beta_1 \leq 0. \end{cases} \tag{4.10}$$

If $\beta_1 \leq -1$, κ is a monotonically increasing function with respect to $|\beta_1|$. The best choice is $\beta_1 = -1$, which gives the minimal κ in this context. If $-1 \leq \beta_1 \leq 0$, κ is a monotonically increasing function with respect to β_1. κ is minimal in this range if $\beta_1 = -1$. In the limit we find that

$$\lim_{\beta_1 \downarrow -1} \kappa = \lim_{\beta_2 \uparrow -1} \kappa = k/\epsilon, \tag{4.11}$$

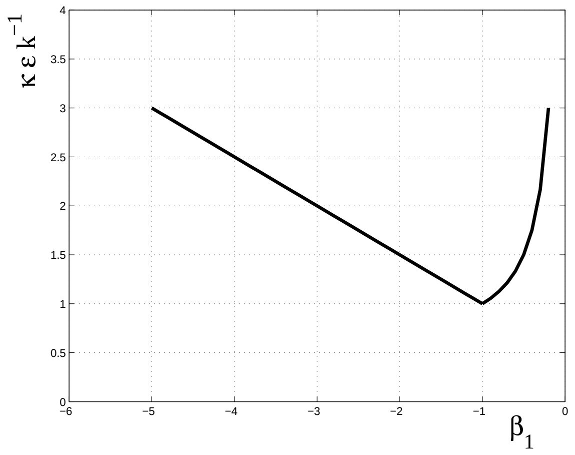

Figure 4.1: Condition number κ of $\mathcal{M}^{-1}\mathcal{A}$ vs. coefficient β_1 for the real shifted Laplace preconditioner

which is the minimum value of κ for $\beta_1 \leq 0 \in \mathbb{R}$ (see Figure 4.1).

The relation (4.10) tells us that taking the leading part of the Helmholtz equation (i.e. the Laplacian term, $\beta_1 = 0$) is generally advisable for high wavenumbers, as advocated in [71]. However, this is no longer true for small wavenumbers, for which the condition number of the preconditioned linear system may become very large. From the convergence bound of CG (3.18) the result (4.10) also gives an indication that setting $\beta_1 = -1$ in (4.5) leads to a fast converging preconditioned iterative method for the Helmholtz equation.

4.1.2 Generalization to complex shift

The basic analysis on 1D shifted Laplace preconditioners for $\beta_1 \leq 0$ in Section 4.1.1 gives $\beta_1 = -1$ as the optimum case. A nice property of the real shifted Laplace operator, at least in 1D, is that the eigenvalues have an upper bound. However, this property does not guarantee that the eigenvalues are favorably distributed. There is the possibility that one or more eigenvalues lie very close to zero. For example, setting $\beta_1 = -1$ gives the minimal condition number κ but, at the same time, results in $\lambda_{\min}$ which is not better than for $\beta_1 = 0$. We can improve the preconditioner by preserving the boundedness and at the same time shifting the minimum eigenvalue as far as possible from zero.

Consider the minimum eigenvalue $\lambda_{\min}$ obtained from the 1D problem (4.2). We may shift this eigenvalue away from zero by adding some real value to λ. In general, this addition will shift all eigenvalues, which is undesirable. An alternative is to multiply the eigenvalues by a factor. From (4.7) the relation between eigenvalues for $\beta_1 = 0$ and $\beta_1 = -1$ can be derived, i.e.,

$$\lambda^{\ell_1}_{\beta_1=-1} = \frac{1}{1 + (k/k_{\ell_1})^2}\lambda^{\ell_1}_{\beta_1=0}. \tag{4.12}$$

Equation (4.12) indicates that $\lambda_{\beta_1=0}$ is scaled by a factor $1/(1 + (k/k_{\ell_1})^2)$ to obtain $\lambda_{\beta=-1}$. Similarly, using (4.9), we obtain the following relation:

$$\left(\lambda^{\ell_1}_{\beta_1=-1}\right)_{\min} = \frac{1}{2}\left(\lambda^{\ell_1}_{\beta_1=0}\right)_{\min}. \tag{4.13}$$

Since the eigenvalues of a general matrix may be complex, relation (4.12) can be considered as a particular case of scaling of the eigenvalues along the real axis in the complex plane. The attempt to improve the clustering here is by introducing an additional shift along the imaginary axis which moves the small eigenvalues away from zero. For that purpose, we consider the complex shifted Laplace operator (4.1), which is written for 1D as

$$\mathcal{M}_{(\beta_1,\beta_2)} := -\frac{d^2}{dx^2} - (\beta_1 - \hat{\jmath}\beta_2)k^2, \quad \beta_1 \leq 0, \beta_2 \in \mathbb{R}. \tag{4.14}$$

In this case by setting $\beta_1 \leq 0$ (as in the previous section) we ensure that the real part of the discrete representation of $\mathcal{M}_{(\beta_1,\beta_2)}$ is positive.

Eigenvalues of the premultiplied equation, denoted by $\lambda^{\ell_1}_c$, are

$$\lambda^{\ell_1}_c = \frac{k^2_{\ell_1} - k^2}{k^2_{\ell_1} - (\beta_1 - \hat{\jmath}\beta_2)k^2} \Rightarrow |\lambda^{\ell_1}_c|^2 = \frac{(k^2_{\ell_1} - k^2)^2}{(k^2_{\ell_1} - \beta_1 k^2)^2 + \beta_2^2 k^4}. \tag{4.15}$$

Evaluating $(\lambda^{\ell_1}_c)_{\max}$ and $(\lambda^{\ell_1}_c)_{\min}$ as in (4.8) and (4.9) one finds

$$|(\lambda^{\ell_1}_c)_{\max}|^2 = \max\left(\frac{1}{\beta_1^2 + \beta_2^2}, 1\right), \quad |(\lambda^{\ell_1}_c)_{\min}|^2 = \frac{4}{(1-\beta_1)^2 + \beta_2^2}\left(\frac{\epsilon}{k}\right)^2. \tag{4.16}$$

These results give the following condition numbers

$$\kappa^2 = \begin{cases} \frac{1}{4}\left(1 + \frac{1-2\beta_1}{\beta_1^2+\beta_2^2}\right)(k/\epsilon)^2, & \beta_1^2 + \beta_2^2 \leq 1, \quad \beta_1 \leq 0, \\ \frac{1}{4}\left((1+\beta_1)^2 + \beta_2^2\right)(k/\epsilon)^2, & \beta_1^2 + \beta_2^2 \geq 1. \end{cases} \tag{4.17}$$

Since $\beta_1^2 + \beta_2^2$ is non-negative, for any given $\beta_1 \leq 0$ the circle $\beta_1^2 + \beta_2^2 = 1$ in the first expression in (4.17) provides the smallest κ^2. Likewise, for any given $\beta_1 \leq 0$, κ^2 is minimal for the second expression in (4.17) whenever $\beta_1^2 + \beta_2^2 = 1$. (One can verify that there is no other circle giving κ^2 lower than that on the circle with radius one. This can be seen, e.g., by introducing condition $\beta_1^2 + \beta_2^2 = 1 + \epsilon_1$, $\epsilon_1 \geq 0$). With condition $\beta_1^2 + \beta_2^2 = 1$, κ is minimal if one takes $\beta_1 = 0$, implying $\beta_2 = \pm 1$. This combination gives the optimal condition number for the shifted Laplace preconditioner, with the setting $\beta_1 \leq 0, \beta_2 \in \mathbb{R}$, for the 1D model problem.

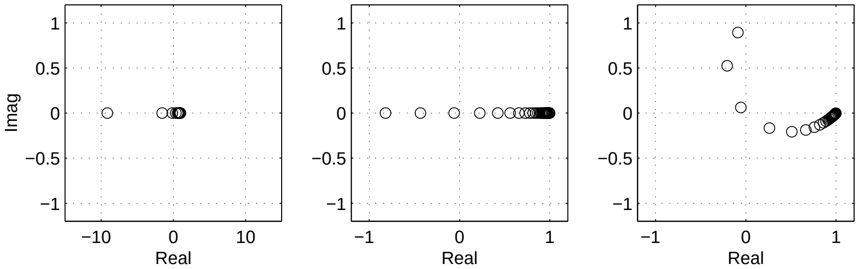

Figure 4.2: Eigenvalues of the preconditioned 1D Helmholtz equation, $k = 10$. Left: $\mathcal{M}_{(0,0)}$, Mid: $\mathcal{M}_{(-1,0)}$, Right: $\mathcal{M}_{(0,1)}$

Figure 4.2 shows the spectra of the 1D Helmholtz problem (4.2) preconditioned by operators $\mathcal{M}_{(\beta_1=0,\beta_2=0)}$, $\mathcal{M}_{(\beta_1=-1,\beta_2=0)}$, and $\mathcal{M}_{(\beta_1=0,\beta_2=+1)}$. For simplicity, we denote these preconditioning operators by $\mathcal{M}_{(0,0)}$, $\mathcal{M}_{(-1,0)}$, and $\mathcal{M}_{(0,1)}$, respectively. Figure 4.2 shows that the preconditioner $\mathcal{M}_{(0,1)}$ clusters the eigenvalues stronger than $\mathcal{M}_{(-1,0)}$ and pushes the eigenvalues in the negative real plane towards the imaginary axis. This clustering may improve the performance of the preconditioned iterative methods. However, with this preconditioner there is still a possibility that some eigenvalues may lie very close to zero, causing an unsatisfactory numerical performance. To estimate the position of the minimum eigenvalues, we consider the real part of (4.15). Similar to (4.9), one finds that

$$\mathrm{Re}((\lambda^{\ell_1}_{(0,1)})_{\min}) = \epsilon/k. \tag{4.18}$$

This estimate is the same as the estimate for $\mathcal{M}_{(-1,0)}$ and smaller than that for $\mathcal{M}_{(0,0)}$. However, the modulus $|(\lambda^{\ell_1}_{(0,1)})_{\min}| = \sqrt{2}(\epsilon/k) > |(\lambda^{\ell_1}_{(-1,0)})_{\min}| = \epsilon/k$ because of the imaginary shift (see Figure 4.3). Because of the same upper bound as $\mathcal{M}_{(-1,0)}$, $\mathcal{M}_{(0,1)}$ may perform better than $\mathcal{M}_{(0,0)}$ and $\mathcal{M}_{(-1,0)}$.

In Figure 4.3, a comparison of the moduli of eigenvalues for $k = 10$ is shown, indicating boundedness of eigenvalues after preconditioning by$\mathcal{M}_{(0,0)}$, $\mathcal{M}_{(-1,0)}$ and $\mathcal{M}_{(0,1)}$ near $|\lambda| = 0$. The right-hand figure zooms in to show the minimum $|\lambda|$. Evidently, $\mathcal{M}_{(0,1)}$ has small eigenvalues with the modulus slightly larger than $\mathcal{M}_{(-1,0)}$, but smaller than $\mathcal{M}_{(0,0)}$.

From the above analysis, by again using the convergence bound of CG (3.18) applied to the normal equations we conclude that for $\beta_1 \leq 0, \beta_2 \in \mathbb{R}$, $\mathcal{M}_{(0,1)}$ turns out to be the optimal preconditioning operator for the Helmholtz equation. This is actually confirmed by the numerical experiments presented in the end of Section 4.2.

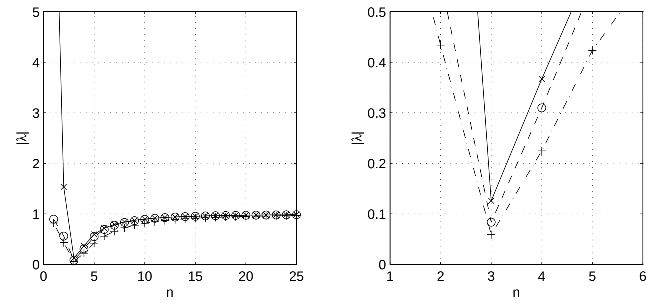

Figure 4.3: The modulus of eigenvalues of the continuous 1D Helmholtz equation, $k = 10$ and $h^{-1} = 100$ for various preconditioners: $\mathcal{M}_{(0,0)}$ ($\times$), $\mathcal{M}_{(-1,0)}$ ($+$), $\mathcal{M}_{(0,1)}$ ($\circ$)

4.2 Spectral properties of the shifted Laplace preconditioner

We extend the analysis for constant wavenumbers to the discrete formulation of (4.2). Suppose that the Helmholtz equation is discretized using central differences, resulting in a linear system $A_h u_h = g_h$. Matrix A_h can be split into two parts: the negative Laplacian component $L_h := -\Delta_h$ and the additional diagonal term $k^2 I_h$ so that $A_h = L_h - k^2 I_h$ and therefore

$$\left(L_h - k^2 I_h\right) u_h = g_h. \tag{4.19}$$

In the present analysis, we assume only Dirichlet or Neumann conditions at the boundaries in order to keep the matrix A_h real-valued. Since A_h is symmetric, all eigenvalues are real-valued. This assumption is of course irrelevant for the exterior problems we are interested in. However, it simplifies the analysis and provides some basic understanding of the preconditioner under consideration. The inclusion of a Sommerfeld radiation condition will lead to different results. Our numerical experiments on exterior problems, however, show similar behavior to the analysis based on the interior problem.

We precondition (4.19) by $M_{h,(\beta_1,\beta_2)} = L_h - (\beta_1 - \hat{j}\beta_2)k^2 I_h$, obtained from a discretization of $\mathcal{M}_{(\beta_1,\beta_2)}$ with the same boundary conditions as for A_h. This gives

$$\begin{aligned}\left(L_h - (\beta_1 - \hat{j}\beta_2)k^2 I_h\right)^{-1} \left(L_h - k^2 I_h\right) u_h = \\ \left(L_h - (\beta_1 - \hat{j}\beta_2)k^2 I_h\right)^{-1} g_h.\end{aligned} \tag{4.20}$$

The generalized eigenvalue problem related to (4.20) is accordingly

$$\left(L_h - k^2 I_h\right) \widehat{u}_h = \widehat{\lambda} \left(L_h - (\beta_1 - \hat{j}\beta_2)k^2 I_h\right) \widehat{u}_h, \tag{4.21}$$

with $\widehat{u}_h$ the eigenvector corresponding to the discrete eigenvalue $\widehat{\lambda}$.

Both systems (4.19) and (4.20) are indefinite if k^2 is larger than the smallest eigenvalue of L_h. In this case, the convergence properties of (4.20) are difficult to estimate. Therefore, for analysis purposes the subsequent analysis will be based on a normal equations formulation of the preconditioned matrix system (as in [64]).

Denote the ordered eigenvalues of L_h as $0 < \mu_1 \leq \cdots \leq \mu_N$, and the normal equations $A_h^* A_h$, $(M_{h,(0,0)}^{-1} A_h)^* (M_{h,(0,0)}^{-1} A_h)$, $(M_{h,(-1,0)}^{-1} A_h)^* (M_{h,(-1,0)}^{-1} A_h)$ and $(M_{h,(0,1)}^{-1} A_h)^* (M_{h,(0,1)}^{-1} A_h)$ as Q_A, $Q_{(0,0)}$, $Q_{(-1,0)}$ and $Q_{(0,1)}$, respectively. We find the eigenvalues in the four following cases as:

$$\lambda\left(Q_A\right) = \left(\mu_m - k^2\right)^2, \tag{4.22}$$

$$\lambda\left(Q_{(0,0)}\right) = \left(\frac{\mu_m - k^2}{\mu_m}\right)^2 = \left(1 - \frac{k^2}{\mu_m}\right)^2, \tag{4.23}$$

$$\lambda\left(Q_{(-1,0)}\right) = \left(\frac{\mu_m - k^2}{\mu_m + k^2}\right)^2 = \left(1 - \frac{2k^2}{\mu_m + k^2}\right)^2, \tag{4.24}$$

$$\lambda\left(Q_{(0,1)}\right) = \left(\frac{\mu_m - k^2}{\mu_m + \hat{j}k^2}\right) \overline{\left(\frac{\mu_m - k^2}{\mu_m + \hat{j}k^2}\right)} = 1 - \frac{2\mu_m k^2}{\mu_m^2 + k^4}. \tag{4.25}$$

We first consider the case where k is small such that $0 < k^2 < \mu_1$, where μ_1 the smallest eigenvalue of $-L_h$. Using (4.22)–(4.25), we find the minimal and maximal eigenvalues as follows:

$$\lambda\left(Q_A\right)_{\min} = \left(\mu_1 - k^2\right)^2, \qquad \lambda\left(Q_A\right)_{\max} = \left(\mu_N - k^2\right)^2,$$

$$\lambda\left(Q_{(0,0)}\right)_{\min} = \left(1 - \frac{k^2}{\mu_1}\right)^2, \qquad \lambda\left(Q_{(0,0)}\right)_{\max} = \left(1 - \frac{k^2}{\mu_N}\right)^2,$$

$$\lambda\left(Q_{(-1,0)}\right)_{\min} = \left(1 - \frac{2k^2}{\mu_1 + k^2}\right)^2, \quad \lambda\left(Q_{(-1,0)}\right)_{\max} = \left(1 - \frac{2k^2}{\mu_N + k^2}\right)^2,$$

$$\lambda\left(Q_{(0,1)}\right)_{\min} = 1 - \frac{2\mu_1 k^2}{\mu_1^2 + k^4}, \qquad \lambda\left(Q_{(0,1)}\right)_{\max} = 1 - \frac{2\mu_N k^2}{\mu_N^2 + k^4}.$$

Since $k^2/\mu_1 < 1$, one easily sees that

$$\lambda\left(Q_{(0,0)}\right)_{\min} > \lambda\left(Q_{(-1,0)}\right)_{\min}.$$

As $m \to \infty$, one finds also that

$$\lim_{\mu_N \to \infty} \lambda\left(Q_{(0,0)}\right)_{\max} = \lim_{\mu_N \to \infty} \lambda\left(Q_{(-1,0)}\right)_{\max} = 1.$$

With respect to the l_2-condition number, it becomes evident that for $k < \sqrt{\mu_1}$, preconditioning with $M_{h,(0,0)}$ gives a lower condition number than preconditioning with $M_{h,(-1,0)}$. Hence, for small k, $M_{h,(0,0)}$ is more effective than $M_{h,(-1,0)}$.

For $M_{h,(0,1)}$, one can compute that

$$\begin{aligned}\lambda\left(Q_{(0,1)}\right)_{\min}/\lambda\left(Q_{(0,0)}\right)_{\min} &= \frac{(\mu_1+k^2)^2}{\mu_1^2+k^4} > 1,\\ \lim_{\mu_N\to\infty}\lambda\left(Q_{(0,1)}\right)_{\max} &= 1.\end{aligned}$$

So, for $k < \sqrt{\mu_1}$, preconditioning with $M_{h,(0,0)}$ gives a better condition number than with $M_{h,(0,1)}$. For k small, $M_{h,(0,0)}$ is more effective than $M_{h,(-1,0)}$ and $M_{h,(0,1)}$.

We consider now k large, such that $\mu_1 < k^2 < \mu_N$. For Q_A (4.22), one finds that

$$\begin{aligned}&\lambda\left(Q_A\right)_{\min} = \left(\mu_{m_1}-k^2\right)^2, \text{ where } |\mu_{m_1}-k^2| \le |\mu_m - k^2|, \ \forall m,\\ &\lambda\left(Q_A\right)_{\max} = \left(\mu_N-k^2\right)^2.\end{aligned} \tag{4.26}$$

The eigenvalues are unbounded either for large μ_N or large k.

For the preconditioned system $Q_{(0,0)}$ one finds

$$\begin{aligned}&\lambda\left(Q_{(0,0)}\right)_{\min} = \left(\frac{\mu_{m_2}-k^2}{\mu_{m_2}}\right)^2, \text{ where } \left|\frac{\mu_{m_2}-k^2}{\mu_{m_2}}\right| \le \left|\frac{\mu_m-k^2}{\mu_m}\right|, \ \forall m,\\ &\lambda\left(Q_{(0,0)}\right)_{\max} = \max\left(\left(\frac{\mu_N-k^2}{\mu_N}\right)^2, \left(\frac{\mu_1-k^2}{\mu_1}\right)^2\right).\end{aligned} \tag{4.27}$$

In this case, there will be boundedness for large μ_N, i.e., for $\mu_N \to \infty$, $\lambda_N = 1$ as long as k is finite (because $\lim_{k\to\infty}((\mu_m-k^2)/(\mu_m))^2 = \infty$). Furthermore, $\lim_{\mu_1\to 0}((\mu_1-k^2)/(\mu_1))^2 = \infty$. Therefore, $\lambda_{\max}$ can become extremely large, which makes $M_{h,(0,0)}$ less favorable for preconditioning.

For the preconditioned system $Q_{(-1,0)}$, one finds that

$$\begin{aligned}&\lambda\left(Q_{(-1,0)}\right)_{\min} = \left(\frac{\mu_{m_3}-k^2}{\mu_{m_3}+k^2}\right)^2, \text{ where } \left|\frac{\mu_{m_3}-k^2}{\mu_{m_3}+k^2}\right| \le \left|\frac{\mu_m-k^2}{\mu_m+\mu_{m_3}}\right|, \ \forall m,\\ &\lambda\left(Q_{(-1,0)}\right)_{\max} = \max\left(\left(\frac{\mu_N-k^2}{\mu_N+k^2}\right)^2, \left(\frac{\mu_1-k^2}{\mu_1+k^2}\right)^2\right).\end{aligned} \tag{4.28}$$

From (4.28), it is found that

$$\lim_{\mu_N\to\infty}\left(\frac{\mu_N-k^2}{\mu_N+k^2}\right)^2 = 1; \quad \lim_{\mu_1\to 0}\left(\frac{\mu_1-k^2}{\mu_1+k^2}\right)^2 = 1; \quad \lim_{k\to\infty}\left(\frac{\mu_m-k^2}{\mu_m+k^2}\right)^2 = 1.$$

The preconditioned system $M_{h,(-1,0)}^{-1}A_h$ is always bounded above by one, i.e. the eigenvalues are always clustered. Moreover, $M_{h,(-1,0)}$ provides a better condition number than $M_{h,(0,0)}$. For large k, $M_{h,(-1,0)}$ is more effective than $M_{h,(0,0)}$.

Finally, we consider the complex shifted preconditioned system with $M_{h,(0,1)}$. One finds that

$$\begin{aligned}\lambda\left(Q_{(0,1)}\right)_{\min} &= \frac{(\mu_{m_4}-k^2)^2}{\mu_{m_4}^2+k^4} \text{ where } \left|\frac{(\mu_{m_4}-k^2)^2}{\mu_{m_4}^2+k^4}\right| \leq \left|\frac{(\mu_m-k^2)^2}{\mu_m^2+k^4}\right|, \ \forall m,\\ \lambda\left(Q_{(0,1)}\right)_{\max} &= \max\left(1-\frac{2\mu_1 k^2}{\mu_1^2+k^4}, 1-\frac{2\mu_N k^2}{\mu_N^2+k^4}\right).\end{aligned} \tag{4.29}$$

The following results follow from (4.29):

$$\lim_{\mu_N\to\infty}\lambda\left(Q_{(0,1)}\right)_{\max}=1; \quad \lim_{\mu_1\to 0}\lambda\left(Q_{(0,1)}\right)_{\max}=1; \quad \lim_{k\to\infty}\lambda\left(Q_{(0,1)}\right)_{\max}=1.$$

Hence, the eigenvalues of $Q_{(0,1)}$ are bounded above by one. Typically, preconditioning with $M_{h,(0,1)}$ gives a better condition number than with $M_{h,(0,0)}$.

To compare $M_{h,(0,1)}$ with $M_{h,(-1,0)}$ we need to estimate the lower bound. In doing this, we assume that $\lambda_{\min}\approx 0$ implying $\mu_m = k^2+\epsilon$, $\epsilon>0$. After substituting this relation to (4.29), one finds that

$$\lambda\left(Q_{(0,1)}\right)_{\min} = \frac{1}{2}\frac{\epsilon^2}{k^4}. \tag{4.30}$$

For $M_{h,(-1,0)}$ we find that

$$\lambda\left(Q_{(-1,0)}\right)_{\min} = \frac{1}{4}\frac{\epsilon^2}{k^4}. \tag{4.31}$$

Therefore,

$$\lambda\left(Q_{(0,1)}\right)_{\min} = 2\lambda\left(Q_{(-1,0)}\right)_{\min}. \tag{4.32}$$

With respect to the l_2−condition number, one finds that

$$\kappa\left(Q_{(0,1)}\right) = 2\left(\frac{k^4}{\epsilon^2}\right) \quad < \quad \kappa\left(Q_{(-1,0)}\right) = 4\left(\frac{k^4}{\epsilon^2}\right).$$

Considering the above result, we conclude that $M_{h,(0,1)}$ is more effective as the preconditioner than $M_{h,(-1,0)}$ for the problems under consideration.

Remark 4.2.1 *For an interior problem where the resulting linear system is real-valued, the use of the complex shifted Laplace preconditioner $M_{h,(0,1)}$ requires more operations because of complex arithmetics. In this situation, it is possible that the gain in the convergence speed-up is overruled by the extra cost of complex arithmetic operations.*

Example 4.1. 2D homogeneous "close-off" problem. We consider a problem in a 2D rectangular homogeneous medium, $\alpha = 0$, governed by

$$\begin{aligned}&\mathcal{A}u := \left(-\Delta - k^2\right)u = (5\pi^2-k^2)\sin(\pi x_1)\sin(2\pi x_2), \quad \Omega=(0,1)^2,\\ &\quad u=0, \quad \text{at the boundaries.}\end{aligned} \tag{4.33}$$

The exact solution of (4.33) is $u = \sin(\pi x_1)\sin(2\pi x_2)$. The numerical solution is displayed in Figure 4.4. We use wavenumbers $k = 5, 10, 20, 30, 40, 50$ and resolve the problem on a mesh of 10 grid points per wavelength ($kh = 0.625$). In Figure 4.5, the spectra of the preconditioned system for $k = 5$, a "slightly" indefinite problem, are shown. All spectra are bounded above by one. For the linear solver, full GMRES is employed and the iteration is terminated if the tolerance (3.25) is reached. For the time being, the inverse of the preconditioner $\mathcal{M}_{\beta_1,\beta_2}$ is computed *exactly*. The initial solution vector is $u_h^0 = 1$.

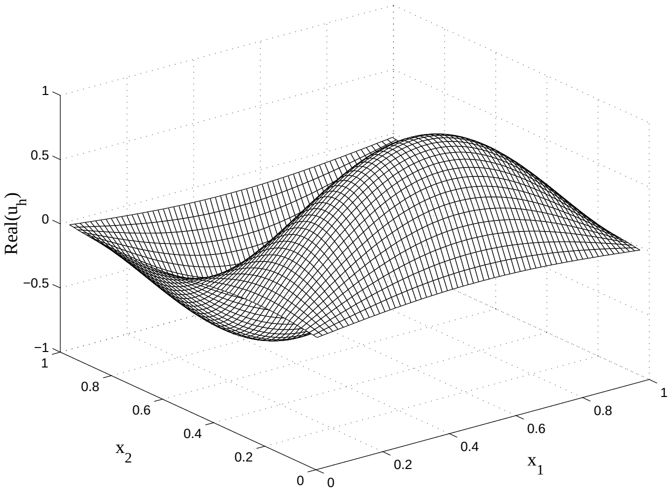

Figure 4.4: Numerical solution of the close-off problem for $\alpha = 0$ (no damping)

Table 4.1 shows the computational performance in terms of the number of iterations and the computational time (using MATLAB) to reach the specified tolerance. For low frequencies, all preconditioners show a very satisfactorily comparable performance. $M_{h,(0,0)}$ becomes less effective for increasing values of k, where the number of iterations increases somewhat faster than for $M_{h,(-1,0)}$ or $M_{h,(0,1)}$. For large k, preconditioning with $M_{h,(0,1)}$ gives the fastest convergence. This behavior is in accordance with the theory. However, preconditioning with $M_{h,(0,1)}$ is expensive. As this problem only requires real arithmetic operations, using $M_{h,(0,1)}$ destroys the cheap operations (about four times more expensive). Furthermore, the computational time shown in Table 4.1 is practically unacceptable due to the *exact inversion* of M. In real-life applications, a cheaper approximate iteration for the preconditioner will be implemented.

Example 4.2. 2D constant k with radiation conditions. In this example we test the preconditioners $M_{h,(0,0)}$,$M_{h,(-1,0)}$ and $M_{h,(0,1)}$ on Model Problem

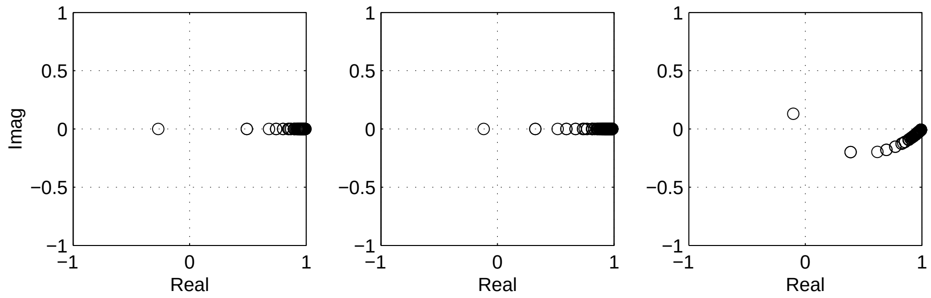

Figure 4.5: Some largest and smallest eigenvalues (in terms of modulus) of the preconditioned systems of "close-off" problems with $k = 5$ and gridsize $h^{-1} = 20$. The preconditioners are $M_{h,(0,0)}$ (left), $M_{h,(-1,0)}$ (middle) and $M_{h,(0,1)}$ (right)

Table 4.1: Computational performance of GMRES to reduce the relative residual by order 7. for 2D closed-off problem. The preconditioner is the shifted Laplace operator. 10 grid points per wavelength are used ($kh = 0.625$). The preconditioners are inverted by using a direct solver

	$M_{h,(0,0)}$		$M_{h,(-1,0)}$		$M_{h,(0,1)}$	
k	Iter	Time(s)	Iter	Time(s)	Iter	Time(s)
5	8	0.01	9	0.01	9	0.03
10	13	0.03	16	0.04	15	0.11
20	28	0.35	30	0.38	26	1.13
30	56	1.77	58	1.84	47	6.54
40	106	7.36	102	7.19	82	26.44
50	149	19.31	142	18.15	114	67.95

1, with $\alpha = 0$ (no damping). So, radiation boundary conditions are included in this test.

The performance of GMRES is compared for these three preconditioners. In the construction of the preconditioning matrix from $\mathcal{M}_{\beta_1,\beta_2}$ (4.1), the same boundary conditions as in Model Problem 1 are used. The grid resolution is $kh = 0.625$. The termination criterion for GMRES is as in (3.25).

Table 4.2 shows the number of preconditioned GMRES iterations to solve Model Problem 1. For all frequencies, $M_{h,(0,1)}$ outperforms $M_{h,(0,0)}$ and $M_{h,(-1,0)}$. $M_{h,(0,0)}$ still performs reasonably well compared to $M_{h,(0,1)}$. This may be due to the influence of Sommerfeld radiation conditions imposed in constructing the preconditioning matrix and because k is not yet large enough.

Figure 4.6 shows the residual at each iteration, computed in the l_2 norm, for $k = 20$. The residual curve indicates slow convergence for the first few iterations

Table 4.2: Computational performance of preconditioned GMRES to solve Model Problem 1. The preconditioner is the shifted Laplace preconditioners: $M_{h,(0,0)}$, $M_{h,(-1,0)}$ and $M_{h,(0,1)}$. 10 grid points per wavelength are used ($kh = 0.625$)

	$M_{h,(0,0)}$		$M_{h,(-1,0)}$		$M_{h,(0,1)}$	
k	Iter	Time (s)	Iter	Time (s)	Iter	Time (s)
10	12	0.15	14	0.10	13	0.10
20	27	1.19	30	1.31	26	1.15
30	55	7.52	56	7.71	44	6.06
40	95	30.23	93	29.58	71	22.70
50	143	85.02	140	83.42	101	59.60

and a convergence improvement later on, indicating a *superlinear* convergence. The slow convergence part is mainly due to the small eigenvalues. Once their components have been removed from the error (or residual) the convergence rate improves. Compared to ILU(0) (in Table 3.2) the use of the shifted Laplace preconditioners results in a faster convergence. For small wavenumbers ILU(0.01) is still more effective to accelerate the convergence (refer to Table 3.3 for comparison) than the shifted Laplace preconditioners.

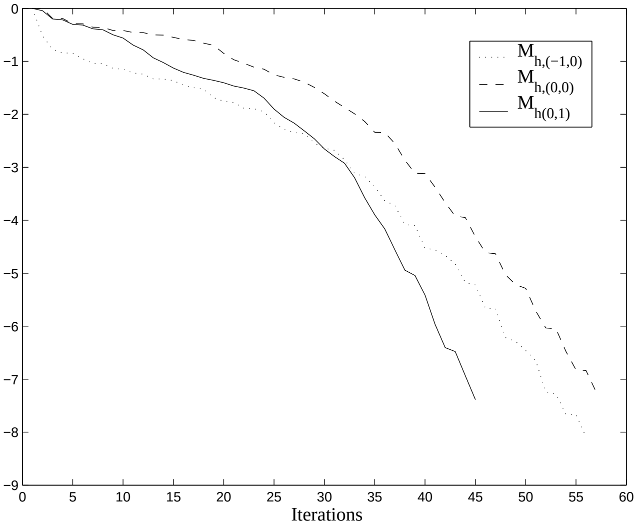

Figure 4.6: Relative residual of preconditioned GMRES iterations, $k = 30$, $\alpha = 0$. $r^j = M_h^{-1}(g_h - A_h u_h^j)$

4.3 Bound of convergence rate of GMRES

In this section we apply GMRES convergence analysis for the discrete 2D Helmholtz equation, preconditioned by the shifted Laplace preconditioner. In particular, we consider $M_{h,(0,1)}$ as the preconditioner. We first provide the spectral properties of a Helmholtz problem with Dirichlet boundary conditions (which is called "closed-off problem"). In Section 4.3.2, we use the results to obtain a bound for the GMRES convergence rate with radiation conditions (2.26). Again, for simplicity we set wavenumber k to be constant over Ω, and $\alpha = 0$ (undamped). We have presented these results in [40].

4.3.1 Closed-off problem

The two-dimensional, "closed-off" Helmholtz problem to be analyzed is defined as follows:

$$\begin{aligned} \mathcal{A}u := -\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2} - k^2 u &= 0, \text{ in } \Omega = (0,1)^2, && (4.34)\\ u(0,x_2) = u(1,x_2) &= 0, \; u(x_1,0) = u(x_1,1) = 0. && (4.35) \end{aligned}$$

The eigenvalue problem related to (4.34)–(4.35) reads

$$\begin{aligned} \left(-\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} - k^2\right)\widetilde{u}^{\ell_1,\ell_2} &= \lambda^{\ell_1,\ell_2}\widetilde{u}^{\ell_1,\ell_2}, \;\; \ell_1,\ell_2 \in \mathbb{N} && (4.36)\\ \widetilde{u}^{\ell_1,\ell_2}(0,x_2) = \widetilde{u}^{\ell_1,\ell_2}(1,x_2) &= 0, && (4.37)\\ \widetilde{u}^{\ell_1,\ell_2}(x_1,0) = \widetilde{u}^{\ell_1,\ell_2}(x_1,1) &= 0, && (4.38) \end{aligned}$$

where $\widetilde{u}^{\ell_1,\ell_2}$ and $\lambda^{\ell_1\ell_2}$ are the eigenfunctions and eigenvalues, respectively. The eigenfunctions

$$\widetilde{u}^{\ell_1,\ell_2} = \sin(\ell_1 \pi x)\sin(\ell_2 \pi y), \quad \text{for} \;\; \ell_1,\ell_2 \in \mathbb{N}. \tag{4.39}$$

satisfy (4.36)-(4.38). Substituting these eigenfunctions in (4.36) yields

$$\lambda^{\ell_1,\ell_2} = k_{\ell_1}^2 + k_{\ell_2}^2 - k^2, \, k_{\ell_1} = \ell_1\pi, \, k_{\ell_2} = \ell_2\pi. \tag{4.40}$$

In this case k_{ℓ_1}, k_{ℓ_2} are the natural frequencies. Resonance occurs if the wavenumber (or reduced frequency), k, is equal to $k_{\ell_1\ell_2} := \sqrt{k_{\ell_1}^2 + k_{\ell_2}^2}$. It resembles the condition $\lambda^{\ell_1,\ell_2} = 0$.

Now, the complex shifted Laplace preconditioner $\mathcal{M}_{(0,1)}$ in 2D is used to speed up the convergence. We can write the preconditioned eigenvalue problem as

$$-\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + k^2\right)\widetilde{u}^{\ell_1,\ell_2} = \lambda_p^{\ell_1,\ell_2}\left(-\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2} + \hat{\jmath}k^2\right)\widetilde{u}^{\ell_1,\ell_2}. \tag{4.41}$$

With the boundary conditions (4.37) and (4.38), the eigenfunctions are given by (4.39), and for the eigenvalues we obtain

$$\lambda_p^{\ell_1,\ell_2} = \frac{k_{\ell_1}^2 + k_{\ell_2}^2 - k^2}{k_{\ell_1}^2 + k_{\ell_2}^2 + \hat{j}k^2}, \ k_{\ell_1} = \ell_1\pi, \ k_{\ell_2} = \ell_2\pi, \ \ell_1, \ell_2 \in \mathbb{N}. \tag{4.42}$$

This means that the preconditioned system is still indefinite, as there is a possibility that $k_{\ell_1}^2 + k_{\ell_2}^2 - k^2$ changes sign, but now it has a clustered spectrum. Adding up $\mathrm{Re}(\lambda_p)$ and $\mathrm{Im}(\lambda_p)$ we find that

$$\mathrm{Re}(\lambda_p^{\ell_1,\ell_2}) + \mathrm{Im}(\lambda_p^{\ell_1,\ell_2}) = (k_{\ell_1,\ell_2}^2 - k^2)^2/(k_{\ell_1,\ell_2}^4 + k^4). \tag{4.43}$$

Assume that resonance does not occur. We can conclude that $\mathrm{Re}(\lambda_p)+\mathrm{Im}(\lambda_p) > 0$. This analysis gives the following result.

Lemma 4.3.1 *Let the 2D Helmholtz equation (4.34) with boundary conditions (4.35) be preconditioned by $\mathcal{M}_{(0,1)}$. If resonance does not occur, then for all $k^2 \neq k_{\ell_1,\ell_2}^2$ the spectrum then completely lies above the line $Re(z) + Im(z) = 0$.*

This lemma suggests a rotation of the spectrum so that the eigenvalues become “definite” (all eigenvalues are translated to either the right or the left half of the complex plane). From Lemma 4.3.1 it is known that the line referred to in Lemma 4.3.1 is at an angle $\theta = \frac{\pi}{4}$ with the imaginary axis (Figure 4.7). We therefore rotate the spectrum by multiplication of the eigenvalue problem by a factor $\exp(\hat{j}\theta)$ (with $\theta = \frac{\pi}{4}$), so that the new eigenvalues $\lambda_{p,\theta}$ satisfy

$$\lambda_{p,\theta} = \lambda_p \exp(\hat{j}\theta). \tag{4.44}$$

By applying (4.44) to (4.42) we obtain

$$\begin{aligned}
\mathrm{Re}(\lambda_{p,\theta}^{\ell_1,\ell_2}) &= \left(\frac{k_{\ell_1,\ell_2}^2 - k^2}{k_{\ell_1,\ell_2}^4 + k^4}\right)\left(k_{\ell_1,\ell_2}^2 \cos\theta + k^2 \sin\theta\right), && (4.45)\\
\mathrm{Im}(\lambda_{p,\theta}^{\ell_1,\ell_2}) &= \left(\frac{k_{\ell_1,\ell_2}^2 - k^2}{k_{\ell_1,\ell_2}^4 + k^4}\right)\left(k_{\ell_1,\ell_2}^2 \sin\theta - k^2 \cos\theta\right), && (4.46)
\end{aligned}$$

where $k_{\ell_1,\ell_2}^4 = (k_{\ell_1}^2 + k_{\ell_2}^2)^2$. Substitution of $\theta = -\pi/4$ gives

$$\begin{aligned}
\mathrm{Re}(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}) &= \frac{1}{2}\sqrt{2}\frac{(k_{\ell_1,\ell_2}^2 - k^2)^2}{k_{\ell_1,\ell_2}^4 + k^4} > 0, \forall k_{\ell_1,\ell_2}, k && (4.47)\\
\mathrm{Im}(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}) &= -\frac{1}{2}\sqrt{2}\frac{k_{\ell_1,\ell_2}^4 - k^4}{k_{\ell_1,\ell_2}^4 + k^4}. && (4.48)
\end{aligned}$$

Note that $\mathrm{Im}(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}) > 0$ if $k_{\ell_1\ell_2}^4 < k^4$.

Elimination of $k_{\ell_1\ell_2}$ from both equations yields

$$\left(\mathrm{Re}\left(\lambda_{p,-\frac{\pi}{4}}^{\ell_1,\ell_2}\right) - \frac{1}{2}\sqrt{2}\right)^2 + \mathrm{Im}\left(\lambda_{p,-\frac{\pi}{4}}^{\ell_1\ell_2}\right)^2 = \frac{1}{2}, \tag{4.49}$$

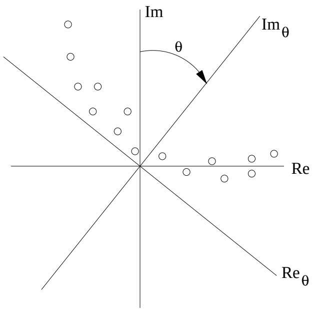

Figure 4.7: Rotation of the z-plane.

or

$$\left|\lambda^{\ell_1,\ell_2}_{p,-\frac{\pi}{4}} - \frac{1}{2}\sqrt{2}\right| = \frac{1}{2}\sqrt{2}, \tag{4.50}$$

which leads to the following lemma.

Lemma 4.3.2 *Let the 2D Helmholtz problem (4.34) with boundary conditions (4.35) be preconditioned by $\mathcal{M}_{(0,1)}$ and assume that resonance does not occur. Furthermore, let the spectrum be rotated according to (4.44). For $\theta = -\pi/4$, the spectrum has the following properties:*

(i) All eigenvalues lie on a circle with center $z_{c,-\frac{\pi}{4}} = \frac{1}{2}\sqrt{2}$ and radius $r = \frac{1}{2}\sqrt{2}$. There are no zero eigenvalues.

(ii) This circle is independent of wavenumber k.

Figure 4.8 illustrates Lemma 4.3.2, for $k = 10$. The grid size affects the position of the small eigenvalues, which asymptotically move closer to the origin. As the circle contains the origin, the classical convergence bound for GMRES cannot be used to estimate the convergence. Therefore, a closer look at the eigenvalues in the vicinity of the origin is required, if one wants to estimate the rate of convergence. On the other hand, the result implies that if an eigenvalue close to the origin is well approximated by a Ritz value during the GMRES process, the convergence of GMRES becomes superlinear [99], which is indeed observed in our numerical experiments in Section 4.2 (see Figure 4.6).

From Lemma 4.3.2, we get the following corollary.

Corollary 4.3.3 *Let the 2D Helmholtz problem (4.34) with boundary conditions (4.35) be preconditioned by operator $\mathcal{M}_{(0,1)}$. Then, the spectrum lies on the circle $|z - z_c| = \frac{1}{2}\sqrt{2}$, with $z_c = \frac{1}{2}(1 + \hat{j})$ the center of circle.*

Proof The proof can be obtained by back transformation of the result in Lemma 4.3.2 by using the rotation matrix (4.44) with $\theta = \pi/4$.

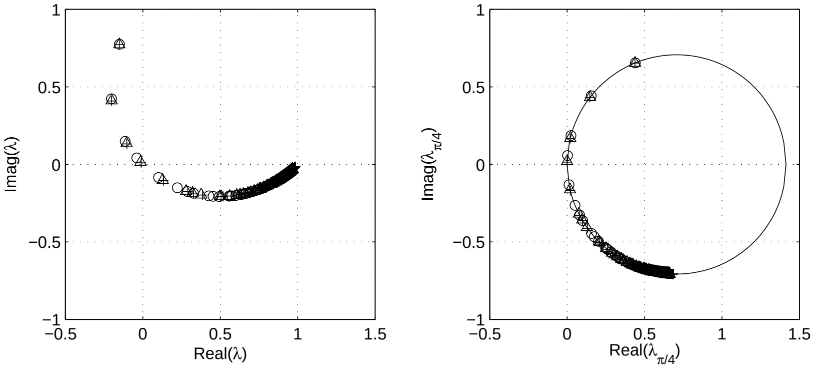

Figure 4.8: Spectrum of the 2D Helmholtz problem preconditioned with the complex shifted Laplace preconditioner before (left) and after (right) rotation, Dirichlet boundary conditions, $k = 10$. Number of gridpoints: $10^2(\circ)$, $20^2(\triangle)$, $30^2(+)$.

Remark 4.3.4 *The introduction of rotation (4.44) is equivalent to solving the system* $PM^{-1}_{h,(0,1)}A_hu_h = PM^{-1}_{h,(0,1)}g_h$ *with* $P = \frac{1}{2}\sqrt{2}diag(1+\hat{j})$*. The rotation is not necessary if Krylov subspace algorithms like GMRES or Bi-CGSTAB are used. These methods are able to handle this type of spectrum automatically. Under rotation (4.44) the condition number* κ *remains the same, so the GMRES or Bi-CGSTAB convergence behavior does not change.*

4.3.2 Analysis for radiation boundary conditions

We consider again the Helmholtz equation (4.34) preconditioned by $\mathcal{M}_{(0,1)}$, but now with the first order radiation boundary conditions (2.26), and discretized by central differences.

Definition A matrix M is complex symmetric positive definite (CSPD) if $\text{Re}(M)$ and $\text{Im}(M)$ are both symmetric, positive definite (SPD).

For a CSPD matrix we can prove the following property.

Lemma 4.3.5 *Let* M *be any CSPD matrix and let* $\lambda_M \in \mathbb{C}$ *be an eigenvalue. Then* $Re(\lambda_M), Im(\lambda_M) > 0$*.*

Proof Consider the eigenvalue problem $Mv = \lambda_M v$ with v the corresponding eigenvector. Thus, $v^*Mv = \lambda_M v^*v$. Using $M = \text{Re}(M) + \hat{j}\text{Im}(M)$ and $v =$

$\mathrm{Re}(v) + \hat{j}\mathrm{Im}(v)$ we have

$$\begin{aligned}\lambda_M v^* v &= \mathrm{Re}(v)^T \mathrm{Re}(M)\mathrm{Re}(v) + \mathrm{Im}(v)^T \mathrm{Re}(M)\mathrm{Im}(v) \\ &+ \hat{j}\left\{\mathrm{Re}(v)^T \mathrm{Re}(M)\mathrm{Im}(v) - \mathrm{Im}(v)^T \mathrm{Re}(M)\mathrm{Re}(v)\right\} \\ &+ \hat{j}\left\{\mathrm{Re}(v)^T \mathrm{Im}(M)\mathrm{Re}(v) + \mathrm{Im}(v)^T \mathrm{Im}(M)\mathrm{Im}(v)\right\} \\ &+ \mathrm{Im}(v)^T \mathrm{Im}(M)\mathrm{Re}(v) - \mathrm{Re}(v)^T \mathrm{Im}(M)\mathrm{Im}(v).\end{aligned}$$

By using the definition of CSPD, the bilinear form and the definition of an SPD matrix, we find that

$$\begin{aligned}\mathrm{Re}(\lambda_M) &= (\mathrm{Re}(v)^T \mathrm{Re}(M)\mathrm{Re}(v) + \mathrm{Im}(v)^T \mathrm{Re}(M)\mathrm{Im}(v))/v^* v > 0, \\ \mathrm{Im}(\lambda_M) &= (\mathrm{Re}(v)^T \mathrm{Im}(M)\mathrm{Re}(v) + \mathrm{Im}(v)^T \mathrm{Im}(M)\mathrm{Im}(v))/v^* v > 0,\end{aligned}$$

which completes the proof.

One can show that the discretization of the negative Laplacian and the boundary conditions results in a CSPD matrix $L_h \in \mathbb{C}^{N\times N}$ for all $kh > 0$. We introduce the splitting

$$A_h = L_h - K_h^2 \Longrightarrow K_h^{-1} L_h K_h^{-1} = K_h^{-1} A_h K_h^{-1} + I_h, \tag{4.51}$$

where $K_h = kI_h, k > 0$. For L_h a CSPD matrix we also have that $K_h^{-1} L_h K_h^{-1} = (kI_h)^{-1} L_h (kI_h)^{-1} = k^{-2} L_h$, is a CSPD matrix. Denote $K_h^{-1} L_h K_h^{-1}$ by C_h. From Lemma 4.3.5, if λ_C is an eigenvalue of C_h, then $\mathrm{Re}(\lambda_C), \mathrm{Im}(\lambda_C) > 0$.

We now consider a similar splitting: $M_{h,(0,1)} := L_h + \hat{j}K_h^2$, with boundary condition (2.26) included in the discretization. For the preconditioned system we have

$$M_{h,(0,1)}^{-1} A_h v = \lambda_{M_{h,(0,1)}^{-1} A_h} v.$$

It is easy to show that

$$\lambda_{M_{h,(0,1)}^{-1} A_h} = \frac{\lambda_C - 1}{\lambda_C + \hat{j}}. \tag{4.52}$$

With this result, we obtain the following theorem.

Theorem 4.3.6 *Let $\lambda_{M_{h,(0,1)}^{-1} A_h}$ be an eigenvalue of $M_{h,(0,1)}^{-1} A_h$ with boundary condition (2.26). Let $|z - z_c| = \frac{1}{2}\sqrt{2}$ with $z_c = \frac{1}{2}(1+\hat{j})$ be the circle corresponding to all eigenvalues of the "closed-off" problem (as described in Corollary 4). Then, $\lambda_{M_{h,(0,1)}^{-1} A_h}$ is enclosed by this circle.*

Proof By using (4.52) and Corollary 4.3.3 we have that

$$\begin{aligned}\lambda_{M_{(0,1),h}^{-1} A_h} - z_c &= \frac{\lambda_C - 1}{\lambda_C + \hat{j}} - \frac{1}{2}(1 + \hat{j}) \\ &= \frac{1}{2}\frac{\lambda_C - 1 - \hat{j}(\lambda_C + 1)}{\lambda_C + \hat{j}} \\ &= \frac{1}{2}\frac{(\lambda_C - 1 - \hat{j}(\lambda_C + 1))(\overline{\lambda}_C - \hat{j})}{(\lambda_C + \hat{j})(\overline{\lambda}_C - \hat{j})}.\end{aligned}$$

With $\left|\lambda_{M^{-1}_{h,(0,1)}A_h} - z_c\right|^2 = (\lambda_{M^{-1}_{h,(0,1)}A_h} - z_c)\overline{(\lambda_{M^{-1}_{h,(0,1)}A_h} - z_c)}$, we find that

$$\begin{aligned}\left|\lambda_{M^{-1}_{h,(0,1)}A_h} - z_c\right| &= \frac{1}{2}\sqrt{2}\sqrt{\frac{\lambda_C - \hat{j}}{\overline{\lambda}_C - \hat{j}} \cdot \frac{\overline{\lambda}_C + \hat{j}}{\lambda_C + \hat{j}}} \\ &= \frac{1}{2}\sqrt{2}\sqrt{\frac{\lambda_C\overline{\lambda}_C - 2\mathrm{Im}(\lambda_C) + 1}{\lambda_C\overline{\lambda}_C + 2\mathrm{Im}(\lambda_C) + 1}} < \frac{1}{2}\sqrt{2}\end{aligned}$$

for every λ_C because of Lemma 4.3.5. Therefore, the eigenvalue $\lambda_{M^{-1}_{h,(0,1)}A_h}$ lies inside the circle. This completes the proof.

Figure 4.9 shows eigenvalues before and after a rotation for $k = 10$ and various grid sizes. For all cases, the eigenvalues are enclosed by the circle. The eigenvalues tend to be more clustered for h increasing. In Figure 4.10 the eigenvalues are shown for different values of k. With increasing k the smallest eigenvalues move closer towards the origin.

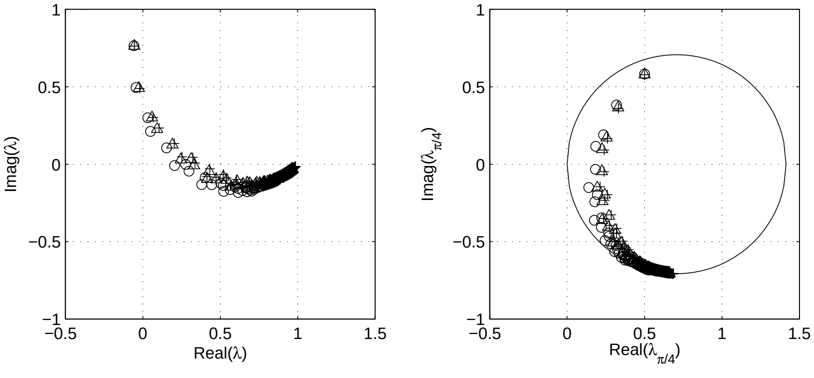

Figure 4.9: Spectrum of the 2D Helmholtz problem ($k = 10$) with radiation boundary conditions, preconditioned by the complex shifted Laplace preconditioner before (left) and after (right) rotation. Number of grid points: $10^2(\circ)$, $20^2(\triangle)$, $30^2(+)$.

In terms of convergence analysis, the results with the radiation boundary conditions are stronger than that with the Dirichlet conditions. By radius $R_c = \max\left|\lambda^{-\pi/4}_{M^{-1}_{h,(0,1)}A_h} - \frac{1}{2}\sqrt{2}\right|$ we denote the eigenvalue of $M^{-1}_{h,(0,1)}A_h$ farthest from the center of the circle after rotation. The circle with center $C_c = \sqrt{2}/2$ and radius R_c encloses the remaining eigenvalues and does not contain the origin. From the convergence bound of GMRES (3.24) the norm of the residual at the j-th iteration $\|r^j\|_2$ is then bounded by $\left(\frac{R_c}{C_c}\right)^j \|r^0\|_2\kappa(X)$ (with the eigenvectors of $M^{-1}_{h,(0,1)}A_h$ being the columns of X).

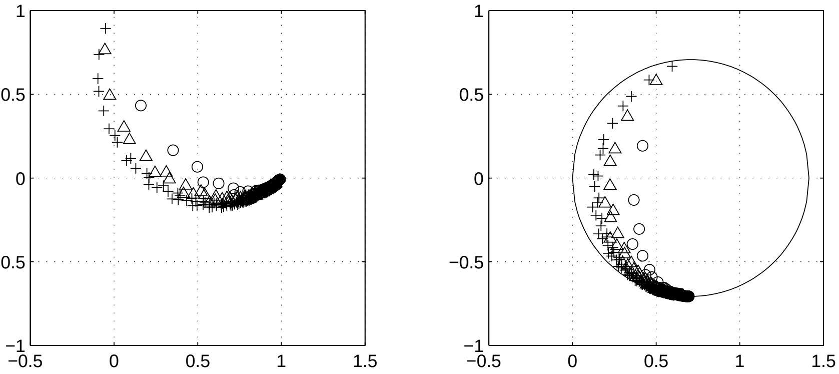

Figure 4.10: Spectrum of the 2D Helmholtz problem (20^2 grid points) with radiation boundary conditions, preconditioned by the complex shifted Laplace preconditioner before (left) and after (right) rotation, $k = 5(\circ)$, $k = 10(\triangle)$, $k = 15(+)$.

Remark 4.3.7 *It is particularly difficult to determine the condition number of X as it does not have the same eigenvectors as A_h for cases with radiation conditions. Nevertheless, for convergence analysis it is already satisfactory to show that the spectrum $M_{h,(0,0)}^{-1}A_h$ is enclosed by a circle that does not contain the origin.*

4.4 h-dependence of convergence

From the previous section it appears that the difference in convergence behavior for the shifted Laplace preconditioner is mainly determined by the smallest eigenvalue. In order to analyze the dependence of the smallest eigenvalue on the step size h we again consider the 1D Helmholtz equation (4.2) with boundary condition $u(0) = 1$ and $u(1) = 0$.

The eigenvalues for this problem with $k = 0$ are $\mu_{\ell_1}^c = (\ell_1\pi)^2$, $\ell_1 = 1, 2, \ldots$. Using the central difference scheme for the Laplace operator, with $N + 1$ grid points and $h = 1/N$, the eigenvalues of the discrete Laplacian operator are given as

$$\mu_{\ell_1} = \frac{4}{h^2}\left(\sin\frac{\pi h \ell_1}{2}\right)^2, \quad \ell_1 = 1, \ldots, N. \tag{4.53}$$

For some of the eigenvalues, say $\widetilde{\ell}_1$, such that $\pi h\widetilde{\ell}_1 \ll 1$ it follows that $|\mu_{\ell_1} - \mu_{\ell_1}^c| = O(h^2)$ for $\ell_1 < \widetilde{\ell}_1$. So, the smallest eigenvalues of the matrix L_h are good approximations of the eigenvalues of the continuous problem. Suppose now that $k^2 \neq \mu_{\ell_1}^c \neq 0$ for all ℓ_1. Then we have that

$$\lim_{h\to 0}\min_{\ell_1}|\mu_{\ell_1} - k^2| = |\mu_m^c - k^2| \neq 0, \tag{4.54}$$

where $|\mu_m^c - k^2| = \min_{\ell_1} |\mu_{\ell_1}^c - k^2|$. Combining this limit with the analysis given in Section 4.2, for $M_{h,(0,1)}$ we can show that [102]

$$\lim_{h \to 0} \lambda_{\min}(Q_{(0,1)}) = \frac{|\mu_m^c - k^2|^2}{2k^4}. \tag{4.55}$$

Since the maximal eigenvalues of $Q_{(0,1)}$ are bounded by 1, we conclude that the convergence behavior of this preconditioners is independent of h (see also [71]). Only initially there can be dependence of the smallest eigenvalue on h. In a similar way, for $M_{h,(-1,0)}$ we find that

$$\lim_{h \to 0} \lambda_{\min}(Q_{(1,0)}) = \frac{|\mu_m^c - k^2|^2}{4k^4}, \tag{4.56}$$

which also indicates an h-independent convergence.

Example 4.3. h-independence convergence. To show the convergence behavior with respect to the gridsize, h, we consider the problem described in Example 4.1 and solve it on different gridsizes using GMRES. The convergence results are shown in Table 4.3 and confirm the h-independent convergence behavior. Similar results are obtained for Bi-CGSTAB.

Table 4.3: Computational performance of GMRES (in terms of number of iterations) to solve the "close-off" problem (see Example 4.1) with different grid resolutions

	$M_{h,(0,0)}$			$M_{h,(-1,0)}$			$M_{h,(0,1)}$		
	h^{-1}			h^{-1}			h^{-1}		
k	50	100	150	50	100	150	50	100	150
10	13	14	14	17	17	17	15	15	16
20	26	26	28	29	30	32	25	26	26
30	52	57	57	54	59	60	44	47	49
40	99	96	98	96	96	99	79	78	80

4.5 Extension to heterogeneous media

In the analysis and numerical tests given so far, we only treated problems in homogeneous media. In real applications we typically encounter problems with heterogeneous properties of the media. Extension of the method to problems with heterogeneity is straightforward. The analysis, however, is much harder.

Example 4.4. Three-layer problems. We show numerical results for a 2D problem in $\Omega = (0,1)^2$ with three layers wherein a simple heterogeneity in k is present, and $\alpha = 0$. The wavenumber k varies in these three layers according to

$$k = \begin{cases} k_{\text{ref}} & 0 \leq x_2 \leq 1/3, \\ 1.5k_{\text{ref}} & 1/3 \leq x_2 \leq 2/3, \\ 2.0k_{\text{ref}} & 2/3 \leq x_2 \leq 1.0. \end{cases} \tag{4.57}$$

The first-order radiation conditions are used on $\Gamma = \partial\Omega$. A unit source is located at $(x_1, x_2) = (0.5, 1)$. The number of grid points used is approximately 30 grid points per reference wavelength ($k_{\text{ref}}h = 0.208$) in the x_1- and x_2-directions. Numerical results are presented in Table 4.4. Here, we compute the solutions using full GMRES, and compare the computational performances with CGNR, COCG and Bi-CGSTAB. For GMRES, we terminate the iterations if the convergence is not reached after 180 iterations, while for the other methods the maximum number of iterations is 600. The convergence is determined based on the criterion (3.25). The preconditioners are $M_{h,(0,0)}$, $M_{h,(-1,0)}$ and $M_{h,(0,1)}$ and inverted *exactly* by using a direct method.

Table 4.4: Computational performance (in terms of number of iterations) of GMRES, CGNR, and Bi-CGSTAB to solve the three-layer problem. The preconditioner is the shifted Laplace operator; 30 grid points per k_{ref} are used

	CGNR			GMRES		
k_{ref}	$M_{h,(0,0)}$	$M_{h,(-1,0)}$	$M_{h,(0,1)}$	$M_{h,(0,0)}$	$M_{h,(-1,0)}$	$M_{h,(0,1)}$
2	10	12	11	8	10	9
5	23	30	25	16	20	18
10	81	81	63	36	47	39
15	259	156	118	67	89	67
20	>600	230	170	112	145	101
30	–	–	192	>180	>180	161

	COCG			Bi-CGSTAB		
k_{ref}	$M_{h,(0,0)}$	$M_{h,(-1,0)}$	$M_{h,(0,1)}$	$M_{h,(0,0)}$	$M_{h,(-1,0)}$	$M_{h,(0,1)}$
2	8	10	9	5	6	6
5	17	20	18	11	13	11
10	45	51	40	60	72	27
15	112	109	74	221	142	41
20	244	182	115	533	185	59
30	>600	353	170	>600	267	77

In this heterogeneous problem, $M_{h,(0,1)}$ outperforms $M_{h,(0,0)}$ and $M_{h,(-1,0)}$ indicated by the smaller number of iterations required to reach convergence. With $M_{h,(0,0)}$, the performance of all iterative methods breaks down for $k_{ref} > 15$. Compared to $M_{h,(0,0)}$, $M_{h,(-1,0)}$ shows a similar performance for low wavenumbers but outperforms $M_{h,(0,0)}$ for high wavenumbers.

From Table 4.4, we also see that preconditioned Bi-CGSTAB does not perform well for $M_{h,(0,0)}$ and $M_{h,(-1,0)}$, as already indicated in [64], but performs satisfactorily for $M_{h,(0,1)}$, with a comparable performance as compared to COCG (in terms of matrix/vector multiplications). Compared to GMRES, Bi-CGSTAB and COCG preconditioned by $M_{h,(0,1)}$ show a better convergence performance for increasing wavenumbers (despite the fact that Bi-CGSTAB requires two preconditioning steps per iteration). If $M_{h,(0,1)}$ is used as the preconditioner, Bi-CGSTAB and COCG can replace full GMRES.

From Table 4.4, one also concludes that CGNR may not be a good iterative method to solve the Helmholtz problem with the shifted Laplace preconditioners. This is due to the squaring of the original eigenvalues in the case of the normal equations, causing too many small eigenvalues. With such a spectrum, CG-type methods often exhibit very slow convergence. However, since our analysis for the preconditioners is based on the normal equations, the results of CGNR are included and are in accordance with the analysis with respect to other preconditioners.

Chapter 5

Multigrid for the preconditioner solve

In the previous chapter we have shown that for high wavenumbers the shifted Laplace preconditioners were able to effectively accelerate the convergence of Krylov subspace methods. Furthermore, we have also shown that the shifted Laplace preconditioner with an *imaginary* shift, denoted by $M_{h,(0,1)}$, leads to the preferred choice. Therefore, for the time being, we will restrict ourselves to a purely *complex symmetric, positive definite* (CSPD) preconditioner. In Chapter 6, we will loosen this requirement. We have also shown that the convergence of the preconditioned Krylov subspace method (in that case GMRES) is independent of gridsize h. In the numerical examples that are given in Chapter 4 we have used, however, a direct solution method to invert the preconditioner in the Krylov subspace method algorithms. This is too expensive, and even more expensive than solving the original problem.

There exist methods to efficiently solve a CSPD system. One idea is to solve the preconditioner by means of other iterative methods. One of the techniques is by using GMRESR [99], which combines GCR as the *outer* iteration with GMRES as the *inner* iteration. This algorithm is, however, expensive in actual tests if compared with Bi-CGSTAB. An attractive approach is to approximately solve the preconditioner by using multigrid [20, 53, 92, 105].

Multigrid is known for its efficiency and robustness when it is applied to a discrete elliptic operator. A multigrid method exhibits fast convergence independent of gridsize, h, and with a complexity of $O(N \log N)$. To achieve such a convergence, however, requires a careful design involving many components in a multigrid method.

This chapter deals with multigrid for solving the preconditioner in the preconditioned Krylov subspace method. We first discuss the principles of multigrid methods and the actual implementation of these principles. We then explain in Section 5.2 why the application of multigrid directly to the Helmholtz equation leads to convergence difficulties. In Section 5.4 we review Fourier analysis which is, among other existing analysis tools, used to quantitatively evaluate the performance of a multigrid algorithm. We restrict ourselves to *geometric* multigrid. *Algebraic* multigrid is beyond the scope of this thesis. The method explained here can, however, be extended to algebraic multigrid (alongside a finite element discretization). Because the geophysical applications are performed on rectangular domains, geometric multigrid is fully satisfactory.

In Chapter 6 we will use the concepts discussed in this chapter.

5.1 Principles of multigrid

The efficiency of multigrid methods to solve a linear system arising from an elliptic partial differential equation comes from two facts. First of all, many classical iterative methods have a strong smoothing effect on the error of any approximation. The error is not necessarily small, but smooth. This is called *smoothing*, which is the first principle of multigrid. The second principle is the so-called *coarse grid correction* (CGC). Any smooth quantity on a fine grid can be well approximated on a coarse grid by a suitable procedure. In the subsequent sections we elaborate these two principles of multigrid more.

5.1.1 From two-grid method to multigrid

To be more technical, we consider a *two-grid* method, with a grid sequence Ω_h, Ω_H, where Ω_h is the fine grid and Ω_H a corresponding coarse grid obtained by doubling the gridsize, i.e. $H = 2h$. We say that Ω_H is obtained by *standard coarsening* of Ω_h. To see how the two multigrid principles work on two grid levels, we consider the Poisson equation: $-\Delta u = g$, with Dirichlet boundary conditions, discretized by the five-point finite difference stencil. This leads to a linear system $L_h u_h = g_h$, with $L_h = -\Delta_h$, the discrete negative Laplacian. The domain is the unit square. If u_h^j is an approximation of u_h after the j-th iteration, the error between these two quantities can be written as

$$v_h^j := u_h - u_h^j = \sum_{\ell_1,\ell_2=1}^{n-1} \alpha_{\ell_1,\ell_2} \sin(\pi \ell_1 x_1) \sin(\pi \ell_2 x_2). \tag{5.1}$$

For $(x_1, x_2) = \Omega_h$, the discrete operator Δ_h has eigenfunctions

$$\phi_h^{\ell_1,\ell_2}(x_1, x_2) = \sin(\pi \ell_1 x_1) \sin(\pi \ell_2 x_2), \;\; \ell_1, \ell_2 = 1, 2, \ldots, \sqrt{N} - 1, \tag{5.2}$$

with N the total unknowns. We consider four eigenfunctions

$$\phi_h^{\ell_1,\ell_2}, \;\; \phi_h^{\sqrt{N}-\ell_1,n-\ell_2}, \;\; \phi_h^{\sqrt{N}-\ell_1,\ell_2}, \;\; \phi_h^{\sqrt{N}-\ell_1,\ell_2}. \tag{5.3}$$

On the coarse grid Ω_H, we observe that

$$\phi_H^{\ell_1,\ell_2} = -\phi_H^{\sqrt{N}-\ell_1,\ell_2} = -\phi_H^{\ell-1,\sqrt{N}-\ell_2} = \phi_H^{\sqrt{N}-\ell_1,\sqrt{N}-\ell_2}. \tag{5.4}$$

This means that the four eigenfunctions (5.3) cannot be distinguished on Ω_H. Since the high frequencies coincide with the low frequencies on Ω_H, only low frequencies are visible on Ω_H. An iterative method with good smoothing properties is one that annihilates the high frequency components quickly.

Now the residual is determined by

$$r_h^j := g_h - L_h u_h^j, \tag{5.5}$$

which is equivalent to the defect equation:

$$L_h v_h^j = r_h^j. \tag{5.6}$$

To solve the defect equation, the two-grid method benefits from the smooth error and uses the approximation of this smooth error on the coarse grid to solve this defect equation *approximately.* Thus, instead of solving (5.6), in the two-grid method one solves

$$L_H \widehat{v}_H^j = r_H^j. \tag{5.7}$$

The size of the matrix L_H is substantially smaller than the size of the matrix L_h. Therefore, the solution of the defect equation is cheap to obtain. Since r_H^j and $\widehat{v}_H^j$ are functions defined on the coarse grid Ω_H, two transfer operators are required to relate the fine-to-coarse grid functions. The first transfer operator is used to *restrict* r_h^j to Ω_H, such that

$$r_H^j := I_h^H r_h^j, \quad I_h^H : \mathcal{G}(\Omega_h) \to \mathcal{G}(\Omega_H). \tag{5.8}$$

The second operator is used to interpolate (*prolongate*) the correction $\widehat{v}_H^j$ back to Ω_h:

$$\widehat{v}_h^j := I_H^h \widehat{v}_H^j, \quad I_H^h : \mathcal{G}(\Omega_H) \to \mathcal{G}(\Omega_h). \tag{5.9}$$

Using this correction, the new approximation can then be computed:

$$u_h^{j+1} = u_h^j + \widehat{v}_h^j. \tag{5.10}$$

This is called the coarse grid correction. The coarse grid correction by itself is, however, not a converging method, as high frequency errors are not annihilated.

A two-grid cycle is a cycle consisting of presmoothing, coarse grid correction and postsmoothing. In pre- and postsmoothing, a classical iterative method with a smoothing property is used. In the coarse grid correction, the defect equation is solved on Ω_H. An algorithm of a two-grid cycle is given as follows.

Algorithm 5.1. Two-grid cycle

```
Given the solution at the $j$-th iteration $u_h^j$.
1. Presmoothing: compute $\bar{u}_h^j$ by applying $\nu_1$ steps of a smoother to $u_h^j$
2. Coarse grid correction (CGC)
```

Compute the defect: $\bar{r}_h^j = g_h - L_h \bar{u}_h^j$
Restrict the defect: $\bar{r}_H^j = I_h^H \bar{r}_h^j$
Solve the defect equation on Ω_H: $L_H \widehat{v}_H^j = \bar{r}_h^j$
Interpolate the correction: $\widehat{v}_h^j = I_H^h \widehat{v}_H^j$
Correct the approximation: $u_h^{j,CGC} = \bar{u}_h^j + \widehat{v}_h^j$

3. Postsmoothing: compute u_h^{j+1} by applying ν_2 steps of a smoother to $u_h^{j,CGC}$.

In multigrid methods we consider more than two grid levels given in a sequence of grids $\Omega_h, \Omega_{2h}, \Omega_{4h}, \ldots, \Omega_{h_0}$, with Ω_{h_0} is the coarsest grid, and recursively apply the same principles as in the two-grid case. Here, the solution of the defect equation on Ω_{2h} is approximately computed by the two-grid method, and so on. On Ω_{h_0} exact solution of the problem can take place.

5.2 Multigrid convergence for the Helmholtz equation

Textbook multigrid methods are typically set up so that a smoothing method reduces high frequency components of an error between the numerical approximation and the exact discrete solution, and a coarse grid correction handles the low frequency error components. Whereas such methods are easily defined for elliptic Poisson-like equations, this is not the case for the Helmholtz equation (without any damping ($\alpha = 0$) in case of (1.26)). Consider for generality the Helmholtz equation for an attenuative medium (1.26)), discretized by the five-point finite difference stencil.

The eigenvalues of the discrete Helmholtz operator (for problems with homogeneous Dirichlet boundary conditions) are determined by

$$\begin{aligned} \lambda_h^{\ell_1,\ell_2} &= \mu_h^{\ell_1,\ell_2} - k^2(1-\alpha\hat{j}) \\ &\equiv \frac{2}{h^2}(2 - \cos\pi\ell_1 h - \cos\pi\ell_2 h) - k^2(1-\alpha\hat{j}), \;\; \ell_1,\ell_2 = 1,\ldots,\sqrt{N}-1 \end{aligned} \tag{5.11}$$

and are not equal to zero as long as $k^2(1-\alpha\hat{j})$ is not equal to any of the eigenvalues of the corresponding discrete Laplace operator $\mu_h^{\ell_1,\ell_2}$. Otherwise, the matrix is singular and its null-space is spanned by the eigenfunctions

$$\widetilde{u}_h^{\ell_1\ell_2} = \sin(\pi\ell_1 x_1)\sin(\pi\ell_2 x_2), \tag{5.12}$$

with ℓ_1, ℓ_2 for which $\lambda_h^{\ell_1,\ell_2} = 0$

Depending on the particular value of k^2, this equation gives rise to both smoothing and coarse grid correction difficulties. The matrix has only eigenvalues in the right half plane as long as k^2 is less than the smallest eigenvalue of the Laplace operator, $\mu_h^{1,1}$. For $k^2 > \mu_h^{1,1}$, the matrix does not have only positive eigenvalues. Point-wise Jacobi iteration with underrelaxation does not *converge* in that case, but since its smoothing properties are satisfactory, the multigrid convergence will deteriorate only gradually for k^2 increasing. By the time k^2 approaches the 6th eigenvalue $\mu_h^{\ell_1,\ell_2}$ ($k^2 \approx 150$), standard multigrid diverges. The Jacobi relaxation now diverges for smooth eigenfrequencies $\widetilde{u}_h^{\ell_1,\ell_2}$ with $\mu_h^{\ell_1,\ell_2} < k^2$. Consequently, multigrid will still converge as long as the coarsest level used is fine enough to represent these smooth eigenfrequencies sufficiently. So, the coarsest level chosen limits the convergence. When k^2 gets larger more variables need to be represented on the coarsest level for standard multigrid convergence. Eventually, this does not result in an $O(N \log N)$ iterative method.

In addition to this feature, the Helmholtz equation also brings a multigrid coarse grid correction difficulty. Eigenvalues close to the origin may undergo a sign change after discretization on a coarser grid. If a sign change occurs the coarse grid solution does not give a convergence acceleration to the finer grid problem, but a severe convergence degradation (or even divergence) instead. In [33] this phenomenon is analyzed and a remedy for the coarse grid correction related to these problematic eigenvalues is proposed. The efficient treatment in [33] is that multigrid is combined with Krylov subspace iteration methods. GMRES is proposed as a smoother and as a cure for the problematic coarse grid correction. The convergence results achieved are impressive.

Standard multigrid will also fail for k^2-values very close to eigenvalues. In that case subspace correction techniques should be employed [23].

An advanced multigrid based solution method for the Helmholtz equation is the wave-ray multigrid method [22]. The method has been adapted for a first-order system least-squares version of the Helmholtz equation in [65]. Wave-ray multigrid has been developed for Helmholtz problems with constant or smoothly varying wavenumbers. A thorough overview for the numerical solution of the Helmholtz equation is presented in [94].

5.3 Multigrid components

In this section, we detail the multigrid components that can be specified for approximately inverting a discrete version of the preconditioning operator $\mathcal{M}$ (4.1). We consider the 5-point discretization and denote the discrete preconditioner as M_h. Furthermore, we consider the *right* preconditioned system. Thus, we solve the equivalent system

$$A_h M_h^{-1} w_h = g_h, \quad M_h u_h = w_h, \tag{5.13}$$

with $M_h u_h = w_h$ solved by one multigrid iteration. Standard multigrid coarsening, i.e., doubling the mesh size h in every direction is chosen. Since we are now applying multigrid to M_h, the discrete operator L_h (and L_H) in Algorithm 4.1 are replaced by M_h (and M_H).

5.3.1 Smoothers

Classical iteration methods like Jacobi with underrelaxation and Gauss-Seidel iterations can be used as smoothers. We denote the two smoothers by ω_r-JAC and ω_r-GS, respectively. In principle one can choose the underrelaxation parameter $\omega_r \in \mathbb{C}$, but the Fourier analysis (discussed in Section 5.4) indicates that there is no real benefit for the problems considered later in this chapter and in Chapter 6. So, we choose $\omega_r \in \mathbb{R}$.

For the Poisson equation, setting $\omega_r = 1$ in Jacobi smoother does not lead to an iterative method with smoothing properties. For Gauss-Seidel, $\omega_r = 1$ already results in good smoothing in combination with lexicographical ordering. This smoothing property improves if red-black ordering is used. We denote the Gauss-Seidel relaxation with red-black ordering by ω_r-GS-RB. Fourier smoothing analysis of these two relaxation methods is explained in [92].

5.3.2 Prolongation and restriction operators

For the intergrid transfer of the grid functions two (linear) transfer operators (5.8) and (5.9) are required. There are several ways to construct the transfer operators.

A frequently used prolongation operator is based on *bilinear interpolation* (denoted by BI) from $\mathcal{G}_H$ to $\mathcal{G}_h$. This interpolation is defined as (see Figure 5.1)

$$I_H^h \widehat{v}_H(x,y) = \begin{cases} \widehat{v}_H(x,y) & \text{for } A \\ \frac{1}{2}[\widehat{v}_H(x_1, x_2+h) + \widehat{v}_H(x_1, x_2-h)] & \text{for } q \\ \frac{1}{2}[\widehat{v}_H(x_1+h, x_2) + \widehat{v}_H(x_1-h, x_2)] & \text{for } p \\ \frac{1}{4}[\widehat{v}_H(x_1+h, x_2+h) + \widehat{v}_H(x_1+h, x_2-h) & \\ +\widehat{v}_H(x_1-h, x_2+h) + \widehat{v}_H(x_1-h, x_2-h) & \text{for } r \end{cases} \tag{5.14}$$

We also consider a prolongation operator which is based on operator-dependent interpolation (denoted by MD) based on de Zeeuw's transfer operators [109], for example, for heterogeneous Helmholtz problems. Originally, this prolongation was set up for general (possibly unsymmetric) real-valued matrices with a splitting of matrix M into a symmetric and an antisymmetric part, see [109]

$$M_{h,s} = \frac{1}{2}(M_h + M_h^T), \quad M_{h,t} = M_h - M_{h,s}.$$

However, since the discretization here leads to a complex symmetric matrix, the prolongation is adapted and briefly explained for such matrices with nine diagonals [35].

The notation in a stencil for the explanation of the prolongation is as in Figure 5.2 (left side). The right picture in Figure 5.2 shows one coarse and four fine grid cells with indices for the explanation of the interpolation weights. Capital letters denote coarse grid, lower case letters fine grid points. Operator element m_p^w, for example, denotes the west element of operator M_h at point p on the fine grid.

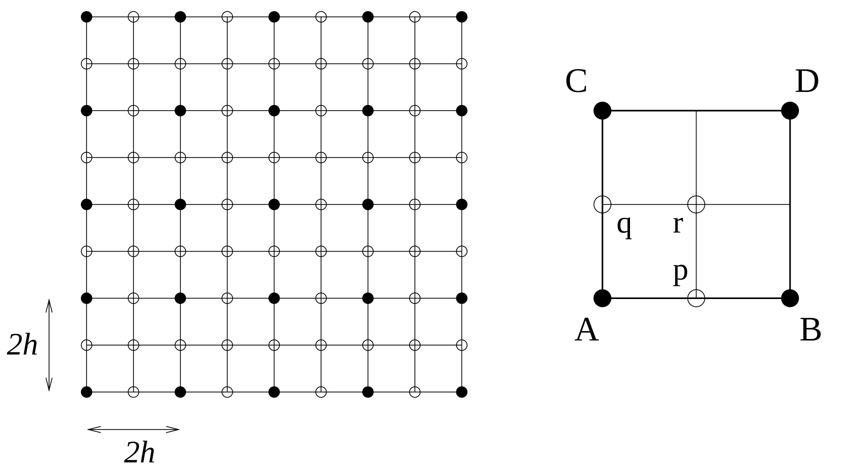

Figure 5.1: Two grids in standard coarsening. The coarse grid is denoted by (•) while the fine grid by (∘)

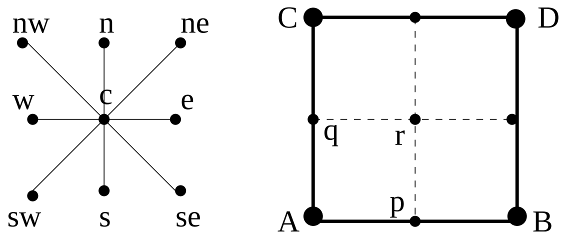

Figure 5.2: Left: Nine point stencil with numbering, Right: Coarse grid cell and four fine cells (Coarse grid indices by capital and fine grid indices by lower case letters)

The corrections from the coarse to the fine grid are obtained by interpolation among nearest coarse grid neighbors. The operator-dependent interpolation weights, w, to determine the fine grid correction quantities e_h are derived with the following formulas:

- for fine grid points p in Fig. 5.2: $e_{h,p} = w_A e_{H,A} + w_B e_{H,B}$.
  $w_A = \min(1, \max(0, w_w))$; $w_B = \min(1, \max(0, w_e))$,

where

$$d_w = \max(|m_p^{sw} + m_p^w + m_p^{nw}|, |m_p^{sw}|, |m_p^{nw}|) \tag{5.15}$$

$$d_e = \max(|m_p^{se} + m_p^s + m_p^{ne}|, |m_p^{se}|, |m_p^{ne}|) \tag{5.16}$$

$$w_w = \frac{d_w}{d_w + d_e}, \; w_e = \frac{d_e}{d_w + d_e} \tag{5.17}$$

- for fine grid points q in Fig. 5.2: $e_{h,q} = w_A e_{H,A} + w_C e_{H,C}$.
  $w_A = \min(1, \max(0, w_s))$; $w_C = \min(1, \max(0, w_n))$,

with

$$\begin{aligned} d_n &= \max(|m_q^{nw} + m_q^{n} + m_q^{ne}|, |m_q^{nw}|, |m_q^{ne}|) && (5.18)\\ d_s &= \max(|m_q^{sw} + m_q^{s} + m_q^{se}|, |m_q^{sw}|, |m_q^{se}|) && (5.19)\\ w_s &= \frac{d_s}{d_s + d_n},\ w_n = \frac{d_n}{d_s + d_n} && (5.20) \end{aligned}$$

On the remaining points the prolongation is defined as follows

- On fine grid points that are also coarse points: $e_h(A) = e_H(A)$
- On points r: $e_h(r)$ is determined so that $M_h I_H^h e_H = 0$ at r.

The interpolation weights are the same as in [109], but especially tailored to the symmetric complex Helmholtz equation. $|.|$ denotes the modulus, in this case. As for symmetric problems with jumping coefficients, the prolongation operator by de Zeeuw [109] is very similar to the original operator-dependent prolongation in [3] and [62]. In [3], for d_w, for example, the lumped sum of three elements, $m_p^{sw} + m_p^{w} + m_p^{nw}$ is chosen. For satisfactory convergence it is, however, important to use the *modulus* of the operator elements, as in (5.15), (5.16), (5.18),(5.19) in the definition of the interpolation weights.

For the restriction, the simplest choice is injecting the value on a fine grid points to the corresponding coarse grid points. This is called *injection.* A commonly used restriction operator is called the *full weighting* (FW) operator, which can be considered as a nine-point weighted average of a (fine) grid function on a coarse grid. For full weighting, the restriction operator reads

$$\begin{aligned} d_H(x_1, x_2) &= I_h^H d_h(x_1, x_2)\\ &= \frac{1}{16}[4d_h(x_1, x_2) + 2d_h(x_1 + h, x_2) + 2d_h(x_1 - h, x_2)\\ &+ 2d_h(x_1, x_2 + h) + 2d_h(x_1, x_2 - h) + d_h(x_1 + h, x_2 + h)\\ &+ d_h(x_1 - h, x_2 + h) + d_h(x_1 + h, x_2 - h) + d_h(x_1 - h, x_2 - h)], \end{aligned} \qquad (5.21)$$

or, in stencil notation,

$$I_h^H = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1 \end{bmatrix}_h^H. \qquad (5.22)$$

In general, we do not choose for the adjoint of the prolongation operator, which is commonly used but not absolutely necessary (as already stated in [3] and an example where the restriction is not the adjoint of the prolongation operator has been given in [30]), unless a combination of bilinear interpolation and full weighting is evaluated. In this case the adjoint of bilinear interpolation operator equals the full weighting operator. Choosing the combination of a full weighting restriction and the operator-dependent interpolation brings to a robust convergence for a variety of Helmholtz problems with constant and nonconstant coefficients. For constant coefficients and mildly varying wavenumbers, bilinear interpolation (in combination with full weighting) also gives very satisfactory convergence results, but for strongly varying coefficients, as in Model Problem 3: the Marmousi problem, a robust and efficient convergence on different grid sizes and for many frequencies is observed for the combination of the transfer operators chosen.

5.3.3 Coarse grid operator, multigrid cycle

A natural choice for the coarse grid operator is to use a direct discretization of the partial differential operator on the coarse grid Ω_H. A different approach is found in the so-called *Galerkin coarse grid operator*, which is defined by

$$M_{2h} := I_h^{2h} M_h I_{2h}^h, \quad M_{4h} := I_{2h}^{4h} M_{2h} I_{4h}^{2h}, \quad \text{etc.} \tag{5.23}$$

In the Fourier analysis results in Chapter 6, this discretization is compared to a direct coarse grid discretization of $\mathcal{M}$. The Galerkin coarse grid discretization seems a natural choice for heterogeneous problems. Also with boundary conditions containing first and second derivatives it is more convenient to choose the Galerkin coarse grid discretization, as it defines the appropriate coarse grid boundary stencils automatically. The transfer operators in building the coarse grid operators are the same as those used for transferring coarse and fine grid quantities to fine and coarse grids, respectively.

In a two-grid method, we have only one possibility of how the sequence of a fine and coarse grid is visited. As the number of grid levels increases, there are many scenarios to process the grids. One typically distinguishes the V-, F- and W-cycles, as depicted in Figure 5.3. On every visit a number of pre- and postsmoothing steps is applied. By using the notation of Algorithm 4.1. V(ν_1,ν_2), for example, means a V-cycle multigrid with ν_1 steps of presmoothing and ν_2 steps of postsmoothing.

5.4 Fourier analysis

In this section we provide analysis for the multigrid convergence of the discrete preconditioning operator M_h. There are several approaches to analyze the multigrid convergence. For example, Hackbusch [53] gives analysis of multigrid based on the *approximation and smoothing* properties of a multigrid method. This approach gives, though important, *qualitative* results. On the other hand, we are interested in *quantitative* estimates of the multigrid convergence. The two-grid Rigorous Fourier Analysis (RFA) [92] is the primary tool in our multigrid analysis.

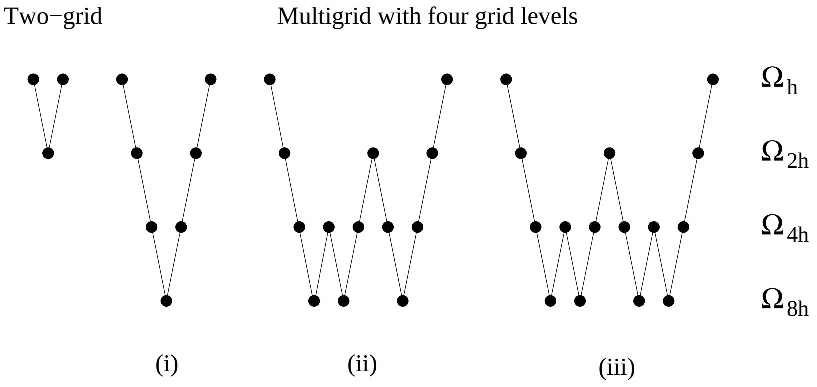

Figure 5.3: Two- and multigrid cycle: (i) V-cycle, (ii) F-cycle and (iii) W-cycle

5.4.1 Two- and three-grid convergence analysis

Fourier smoothing and two-grid analysis, two classical multigrid analysis tools, have been used for quantitative estimates of the smoothing properties and of the other multigrid components in a two-grid method [19, 20, 21, 89, 92]. Consider $M_h u_h = w_h$ in (5.13), where u_h represents the exact discrete solution. The error $v_h^j = u_h^j - u_h$ after the j-th iteration is transformed by a *two-grid* cycle as follows:

$$v_h^{j+1} = T_h^{2h} v_h^j, \quad T_h^{2h} = S_h^{\nu_2} K_h^{2h} S_h^{\nu_1}, \quad K_h^{2h} = I_h - I_{2h}^h (M_{2h})^{-1} I_h^{2h} M_h. \quad (5.24)$$

M_h, M_{2h} correspond to discretizations of $\mathcal{M}$ on the h-, $2h$-grid, S_h is the smoothing operator on the fine grid, I_h the identity operator. ν_1 and ν_2 represent respectively the number of pre- and postsmoothing steps, I_h^{2h} and I_{2h}^h denote the restriction and prolongation operator, respectively. In the analysis we assume an equidistant grid with $\sqrt{N}$ points in each direction.

The $O(h^2)$ discrete complex Helmholtz operator from $\mathcal{M}$ belongs to the class of symmetric stencils. For these stencils it is possible to apply Fourier analysis on the basis of discrete sine-eigenfunctions $\phi_h^{\ell_1,\ell_2}$, $\ell_1, \ell_2 = 1, \ldots, \sqrt{N} - 1$ (5.3), instead of the local Fourier analysis with exponential functions. For problems with symmetric stencils and homogeneous Dirichlet boundary conditions, this analysis can predict h-dependent convergence factors. (The definition of the operator-dependent prolongation and the Galerkin coarse grid stencils in Section 5.3.2 also lead to symmetric operators, that can be analyzed within this framework.) From the discussion of multigrid for the original Helmholtz equation it seems necessary to gain insight into the h-dependency of the multigrid methods developed also for the complex Helmholtz operator.

For the point-wise Jacobi smoother, the $\phi_h^{\ell_1 \ell_2}$ (5.3) are also eigenfunctions of the smoothing operator. This is not true for Gauss-Seidel smoother and the two-grid iteration operator T_h^{2h}. However, the 4-dimensional linearly independent spaces, the *harmonics*:

$$E_h^{\ell_1,\ell_2} = \left[\phi_h^{\ell_1,\ell_2},\ \phi_h^{\sqrt{N}-\ell_1,\sqrt{N}-\ell_2},\ -\phi_h^{\sqrt{N}-\ell_1,\ell_2},\ -\phi_h^{\ell_1,\sqrt{N}-\ell_2}\right] \tag{5.25}$$

for $\ell_1, \ell_2 = 1, \ldots, \frac{\sqrt{N}}{2}$ are invariant under these operators. One can show [89, 92] that

$$\begin{aligned} M_h &: \text{span}\,[\phi_h^{\ell_1,\ell_2}] \to \text{span}\,[\phi_h^{\ell_1,\ell_2}], \quad (M_{2h})^{-1} : \text{span}\,[\phi_{2h}^{\ell_1,\ell_2}] \to \text{span}\,[\phi_{2h}^{\ell_1,\ell_2}], \\ S_h &: \text{span}\,[\phi_h^{\ell_1,\ell_2}] \to \text{span}\,[\phi_h^{\ell_1,\ell_2}], \\ I_h^{2h} &: E_h^{\ell_1,\ell_2} \to \text{span}\,[\phi_{2h}^{\ell_1,\ell_2}], \qquad\qquad I_{2h}^h : \text{span}\,[\phi_{2h}^{\ell_1,\ell_2}] \to E_h^{\ell_1,\ell_2}, \end{aligned}$$

and $T_h^{2h} : E_h^{\ell_1,\ell_2} \to E_h^{\ell_1,\ell_2}$ $(\ell_1, \ell_2 = 1, ..., \frac{\sqrt{N}}{2})$. Therefore, the representation of T_h^{2h} with respect to $E_h^{\ell_1,\ell_2}$ leads to a block-diagonal matrix, $\widetilde{T}_h^{2h}$,

$$T_h^{2h} \triangleq \left[\widehat{T}_h^{2h}(\ell_1, \ell_2)\right]_{\ell_1,\ell_2=1,\ldots,\frac{\sqrt{N}}{2}} =: \widetilde{T}_h^{2h} \ . \tag{5.26}$$

Here the blocks $\widehat{T}_h^{2h}(\ell_1, \ell_2)$ are 4×4 matrices if $\ell_1, \ell_2 < \frac{\sqrt{N}}{2}$, 2×2 (1×1) matrices if either $\ell_1 = \frac{\sqrt{N}}{2}$ or $\ell_2 = \frac{\sqrt{N}}{2}$ $(\ell_1 = \frac{\sqrt{N}}{2}$ and $\ell_2 = \frac{\sqrt{N}}{2})$. The two-grid convergence factor is defined as

$$\rho_{2g} := \max_{1 \le \ell_1, \ell_2 \le \frac{\sqrt{N}}{2}} \rho\Big(\widehat{T}_h^{2h}(\ell_1, \ell_2)\Big) \ . \tag{5.27}$$

Thus, the spectral radii of at most 4×4 matrices $\widehat{T}_h^{2h}(\ell_1, \ell_2)$ have to be determined, and their maximum with respect to ℓ_1 and ℓ_2 has to be found.

The definition of the smoothing factor μ is closely related. The smoothing factor measures the reduction of high frequency error components by an iterative method. It is based on a coarse grid correction operator that annihilates the low frequency error components completely and keeps the high frequency components unchanged. K_h^{2h} is replaced by a projection operator Q_h^{2h} mapping onto the space of high frequencies, i.e., a block diagonal matrix with $\widehat{Q}_h^{2h}$ at most 4×4-diagonal blocks defined by $diag(0, 1, 1, 1)$. So, μ is computed as ρ_{2g} (5.27) with $\widehat{K}_h^{2h}$ in $\widehat{T}_h^{2h}$ replaced by $\widehat{Q}_h^{2h}$.

Recently, three-grid Fourier analysis is proposed in [106]. An issue that can be analyzed in some more detail with a third grid is the coarse grid correction. If a large difference occurs between the two-grid and the three-grid convergence factors, ρ_{2g} and ρ_{3g}, this is an indication for a problematic coarse grid correction. For the complex Helmholtz preconditioner it seems important to analyze the coarse grid correction carefully. The error transformation by a three-grid cycle is given by

$$\begin{gathered} v_h^{j+1} = T_h^{4h}\, v_h^j \quad \text{with} \\ T_h^{4h} = S_h^{\nu_2} K_h^{4h} S_h^{\nu_1} \quad \text{and} \quad K_h^{4h} = I_h - I_{2h}^h (I_{2h} - (T_{2h}^{4h})^{\gamma})(M_{2h})^{-1} I_h^{2h} M_h. \end{gathered} \tag{5.28}$$

Here T_{2h}^{4h}, defined by (5.24), reads $T_{2h}^{4h} = S_{2h}^{\nu_2}(I_{2h} - I_{4h}^{2h}(M_{4h})^{-1}I_{4h}^{2h})S_{2h}^{\nu_1}$. The matrix M_{4h} corresponds to $4h$-grid discretization of $\mathcal{M}$, S_{2h} is the smoothing operator and I_{2h} the identity on the $2h$-grid, I_{2h}^{4h} and I_{4h}^{2h} are transfer operators between the different grids. The $2h$-equation is solved approximately in a three-grid cycle (5.28) by performing γ two-grid iterations T_{2h}^{4h} with zero initial approximation, see also [89, 106].

The three-grid analysis is a recursive application of the two-grid analysis. Not only in the transition from h- to $2h$-grid but also in the transition from the $2h$- to the $4h$-grid, four frequencies are coupled. Thus the three-grid error transformation operator couples 16 Fourier frequencies. As a consequence, T_h^{4h} is unitarily equivalent to a block diagonal matrix $\widetilde{T}_h^{4h}$ with at most 16×16 blocks, $\widehat{T}_h^{4h}(\ell_1, \ell_2)$. The block matrices are composed of the Fourier symbols from the two-grid analysis, which is due to the recursive application of the two-grid analysis. One may compute the three-grid factor ρ_{3g}, as the supremum of the spectral radii from the 16×16 block matrices, $\widehat{T}_h^{4h}(\ell_1, \ell_2)$. For more details about the three-grid analysis, we refer to [106]. Three-grid Fourier analysis software, based on the exponential functions, is freely available, see `www.mgnet.org/mgnet-codes-wienands.html`.

5.4.2 Fourier analysis applied to the preconditioner with imaginary shift

In this section we focus on the shifted Laplace operator with an imaginary shift:

$$\mathcal{M}_{(0,1)} := -\Delta + \hat{\jmath}k^2, \tag{5.29}$$

and apply RFA to its discrete version $M_{h,(0,1)}$.

We first consider multigrid with a V-cycle and ω_r-GS-RB relaxation with $\omega_r = 1$, which is a well-known combination for the Poisson equation. We set $k = 100$ in (5.29), with $kh = 0.625$. The coarse grid matrices are constructed by the Galerkin coarse grid discretization. Table 5.1 shows RFA results for this preconditioning operator. In the table we show the two- and three-grid convergence factor, ρ_{2g} and ρ_{3g} respectively, and the numerical multigrid convergence, ρ_h. In addition, we compute the two-grid operator norm $\|T_h^{2h}\|_S = \sqrt{\rho(T_h^{2h}(T_h^{2h})^*)}$ and the three-grid operator norm $\|T_{2h}^{4h}\|_S = \sqrt{\rho(T_{2h}^{4h}(T_{2h}^{4h})^*)}$, to quantify an upper bound for the error reduction in only one multigrid iteration. This latter quantity is interesting since the method is used as a preconditioner and only one multigrid iteration is applied for preconditioner solves in the Krylov subspace methods. Here we have used the MD-FW transfer operators. The matrix-dependent prolongation operator may be particularly important if a heterogeneity is present in the medium. For constant k the multigrid convergence with this combination resembles that of the bilinear interpolation prolongation operator. The V(1,0)-cycle is compared to V(1,1), where in parentheses the number of pre- and post-smoothing steps are indicated. From RFA, the asymptotic two-grid convergence factor for the V(1,1)-cycle is about 0.06, which is in a good agreement with the numerical convergence. Furthermore, the norm of the two-grid operator is well below 0.2. Multigrid for (5.29) behaves very similarly to the definite real version of the Helmholtz operator (and of the Laplace operator).

Table 5.1: Comparison of asymptotic convergence from Fourier analysis with numerical multigrid convergence, $M_{h,(0,1)}$. μ is the smoothing factor; ρ_{2g}, ρ_{3g} are the two- and three-grid convergence factor from Fourier analysis; ρ_h is the numerical multigrid convergence factor. The smoother is ω_r-RB-GS with $\omega_r = 1$

	μ	ρ_{2g}	$\|T_h^{2h}\|_S$	ρ_{3g}	$\|T_{2h}^{4h}\|_S$	ρ_h, V-cycle
V(1,0)	0.25	0.25	0.56	0.24	0.56	0.235
V(1,1)	0.06	0.06	0.14	0.06	0.14	0.055

We have also analyzed multigrid with $\gamma = 2$ and the ω_r-JAC smoother, applied to (5.29). Notice that for $\gamma = 2$ the F-cycle is identical to the W-cycle. Underrelaxation parameter ω_r is set to $\omega_r = 0.8$, as this is the optimal choice for the Laplace operator [92]. Again, the agreement between the smoothing, two- and three-grid Fourier analysis results with one and two smoothing iterations and the numerical convergence is excellent, as presented in Table 5.2. The results obtained are very similar to the convergence factors for the Laplace operator with ω_r-JAC.

Table 5.2: Comparison of asymptotic convergence from Fourier analysis with numerical multigrid convergence, $M_{h,(0,1)}$. μ is the smoothing factor; ρ_{2g}, ρ_{3g} are the two- and three-grid convergence factor from Fourier analysis; ρ_h is the numerical multigrid convergence factor. The smoother is ω_r-JAC with $\omega_r = 0.8$

	μ	ρ_{2g}	$\|T_h^{2h}\|_S$	ρ_{3g}, $\gamma = 2$	$\|T_{2h}^{4h}\|_S$	ρ_h, F-cycle
F(1,0)	0.60	0.60	0.60	0.60	0.60	0.58
F(1,1)	0.36	0.36	0.36	0.36	0.36	0.34

Remark 5.4.1 *For the case $M_{h,(0,1)}$, one can adopt the well-known multigrid components: direct PDE coarse grid discretization and red-black Gauss-Seidel relaxation. This gives $\rho_{3g} = 0.16$ for $\gamma = 1$ and $\rho_{3g} = 0.08$ for $\gamma = 2$ with two smoothing iterations, very similar to the Laplace situation. The cost of red-black Gauss-Seidel per iteration in CPU time is almost twice as expensive as that of one Jacobi iteration.*

Table 5.3 presents the number of multigrid iterations to solve the *preconditioner* equation: $M_h u_h = w_h$ for varying k. The CPU time is only shown for $k = 100$. The transfer operators are MD and FW. The results are presented for a V-cycle with different numbers of pre- and post-smoothing. By increasing the number of pre- and post-smoothing iterations, the number of multigrid iterations to converge can be reduced, but, of course, the CPU time for one cycle typically increases.

Table 5.3: Number of multigrid V-cycles to solve the preconditioner $\mathcal{M}_{h,(0,1)}$, with MD and FW as the transfer operators. The CPU time is presented for $k = 100$. The termination criterion is $\|r^j/r^0\| \leq 10^{-6}$

	k					time
(ν_1, ν_2)	20	40	60	80	100	(sec)
(1,0)	9	9	9	9	9	1.01
(1,1)	7	8	6	8	8	1.07
(2,1)	4	6	8	5	6	1.16
(1,2)	4	4	7	4	5	0.97

Since we use multigrid as a method to approximately invert the preconditioner M in the Bi-CGSTAB algorithm, we only consider V(1,0) and V(1,1) in our numerical experiments in Section 5.

Remark 5.4.2 *Starting with the 5-point stencil on the finest grid, the Galerkin coarse grid discretization based on matrix-dependent (or bilinear) interpolation and full-weighting restriction results in 9-point coarse grid stencils. The work on the coarse grids can be substantially reduced if the coarse grid stencil would be based on a 5-point stencil. A 5-point coarse grid stencil can be recovered if full-weighting is replaced by half (or full) injection [92]. Using (half) injection, which leads to efficient multigrid Poisson solvers, however, gives a slower method. For half injection combined with bilinear interpolation, the method diverges. From the three-grid analysis half injection results in worse convergence. Extra work on coarser grid, e.g. by using W(1,1) instead of V(1,1) may help improving the convergence on the coarse grid. From RFA, we obtain smaller ρ_{3g} with W(1,1) but this reduction, however, is not so substantial.*

One can also obtain a 5-point stencil on the coarse grids by applying a direct discretization to the operator (5.29). However, it is not trivial how to include the radiation boundary conditions in the discretization on the coarse grids. Inaccurate representation of the boundary condition may lead to an inefficient multigrid method.

5.5 Numerical results

Numerical tests are performed on three model problems which mimic geophysical problems; see Section 2.6. We start with a relatively simple problem with constant k and increase the difficulty of the problem to an irregular heterogeneous medium, the so-called Marmousi problem.

Bi-CGSTAB with right preconditioning is implemented. For the preconditioner solves, two scenarios are implemented: incomplete LU factorization and multigrid. For multigrid, only one multigrid iteration is used to solve the preconditioner. In addition, we compare the convergence with the results using M_I (in Section 3.8.2) and M_{SV} (in Section 3.8.3).

We use the following notation to indicate the different preconditioners implemented.

(1) ILU(A_h,n_{lev}) : n_{lev} level ILU applied to the original matrix A_h,

(2) ILU(M_I,n_{lev}) : n_{lev} level ILU applied to (3.30) and (3.31) (as proposed in [69]).

(3) ILU($M_{h,(0,1)}$,n_{lev}) : n_{lev} level ILU applied to $M_{h,(0,1)}$,

(4) MG(V,ν_1,ν_2) : multigrid applied to $M_{h,(0,1)}$ with V-cycle, ν_1 pre-smoothing and ν_2 post-smoothing steps,

(5) M_{SV} : Separation-of-variables preconditioner as defined in (3.33).

In order to compare with the convergence data from [81] we use

$$\|g_h - A_h u_h^j\| / \|g_h\| \leq 10^{-6} \tag{5.30}$$

for the termination criterion.

5.5.1 Model Problem 1: Constant wavenumber

Here we consider Model Problem 1. The numerical performance of the different preconditioners for Bi-CGSTAB is shown in Tables 5.4 (number of iterations) and 5.5 (CPU time) for increasing k.

All preconditioners accelerate the convergence compared to the unpreconditioned case. Using ILU as the preconditioner based on A_h, M_I, and $M_{h,(0,1)}$ results in a comparable performance here. Further convergence acceleration w.r.t. the number of iterations is achieved by multigrid, especially for k increasing. For example, in case of $k = 100$, the number of iterations is reduced by factor of 4 compared to ILU(1). MG(V(1,0)) improves the CPU time by factor of 2 as compared to ILU(0), but not much gain in CPU time is observed in comparison with ILU(1) (see Table 5.5).

5.5.2 Model Problem 2: The wedge model

A problem of intermediate difficulty, the wedge model, is used to evaluate the behavior of the preconditioners for a simple heterogeneous medium with $\alpha = 0$(see Model Problem 2 in Section 2.6 and Figure 2.3). The problem is adopted from [81] so that we can include the numerical results obtained from the Separation-of-Variables preconditioner. The domain is defined as a rectangle of dimension 600×1000 m^2. The Sommerfeld boundary conditions are set, and a point

Table 5.4: Number of iterations for a constant k Helmholtz problem, shown for various k.

k grid	10 32^2	20 64^2	30 96^2	40 128^2	50 192^2	100 384^2
No-Prec	150	846	1577	1857	3054	6283
ILU(A,0)	75	221	354	474	634	1340
ILU(A,1)	35	113	198	238	295	563
ILU(M_I,0)	79	221	394	475	763	1352
ILU(M_I,1)	42	132	212	238	351	577
ILU($M_{(0,1),h}$,0)	60	188	334	421	684	1293
ILU($M_{(0,1),h}$,1)	36	100	148	206	301	536
MG(V(1,0))	18	36	53	63	71	133
MG(V(1,1))	16	33	49	60	70	133

source is located at the center of the upper surface. Frequency, $f = \overline{k}c/(2\pi)\overline{l}$ (see Remark 1.2.4), is varying from 10 to 60 Hz (with c is the speed of sound). The corresponding values of the local dimensionless wavenumbers $\overline{k}$ vary between 20 (smallest for 10 Hz) and 240 (biggest for 60 Hz). For the problem at 10 Hz approximately 18 points per wavelength are used.

A similar convergence behavior as for Model Problem 1 is observed; ILU can improve the convergence compared to the unpreconditioned case. Multigrid can further accelerate the convergence as compared to ILU(0) and ILU(1) but is in CPU time comparable to ILU(1) and tends to be as fast as ILU($M_{h,(0,1)}$,1) for high f. Complex arithmetic operations for the coarse grid nine-point stencil

Table 5.5: CPU time (in sec.) for a constant k Helmholtz problem, shown for various k.

k grid	10 32^2	20 64^2	30 96^2	40 128^2	50 192^2	100 384^2
No-Prec	0.03	0.9	5.5	15.3	59	483
ILU(A,0)	0.04	0.5	2.3	6.2	19	159
ILU(A,1)	0.02	0.3	1.5	3.5	10	77
ILU(M_I,0)	0.04	0.5	2.4	6.0	22	152
ILU(M_I,1)	0.03	0.4	1.4	3.3	11	73
ILU($M_{(0,1),h}$,0)	0.03	0.5	2.1	5.5	20	153
ILU($M_{(0,1),h}$,1)	0.02	0.3	1.1	3.0	10	72
MG(V(1,0))	0.02	0.2	0.8	2.9	7	65
MG(V(1,1))	0.02	0.2	0.9	3.7	9	89

elements may be the reason for the comparable CPU time of multigrid and ILU($M_{h,(0,1)}$,1).

The use of a V-cycle with only a pre-smoothing step for the preconditioner is the best option w.r.t. multigrid; The combination Bi-CGSTAB and ILU(1) for $M_{h,(0,1)}$ solves the problem fastest among the choices presented. For high frequencies, the complex shifted-Laplace preconditioner $M_{h,(0,1)}$ does not indicate any breakdown of the iterations, whereas the SV preconditioner did (CPU time is not available for SV in [81]). The performance of the complex shifted-Laplace preconditioner behaves well; the number of iterations increases almost linearly against f.

Table 5.6: Number of iterations for the wedge problem, shown for various frequencies f.

f (Hz)	10	20	30	40	50
grid	76 × 126	151 × 251	232 × 386	301 × 501	376 × 626
No-Prec	2379	4057	6907	8248	>10000
ILU(A,0)	571	1339	1917	2443	3287
ILU(A,1)	239	563	832	1120	1418
ILU(M_I,0)	529	1066	1718	2173	2959
ILU(M_I,1)	235	531	787	903	1056
ILU($M_{(0,1),h}$,0)	485	997	1759	2082	2824
ILU($M_{(0,1),h}$,1)	212	426	664	859	1005
MG(V(1,0))	48	92	132	182	213
MG(V(1,1))	44	91	128	182	223
M_{SV}	14	45	167	830	>2000

5.5.3 Model Problem 3: The Marmousi problem

The last example is Model Problem 3 (see Section 2.6): a part of the Marmousi problem which mimics subsurface geology (see also Figure 2.4) with $\alpha = 0$. The domain is taken to be rectangular with a dimension of 6000×1600 m^2. The Sommerfeld radiation condition is imposed at the boundary, and a point source is placed at the center of the upper surface. The minimum number of points per wavelength equals 17. The frequency is varied between 1 and 30 Hz. The number of iterations and CPU time are presented in Tables 5.8 and 5.9, respectively, for frequencies ranging from 1 to 30 Hz, with the termination criterion (5.30).

Whereas the iteration with ILU(0) hardly converges, not even for frequency $f = 1$ Hz, the complex shifted-Laplace preconditioner accelerates the iteration effectively. For low frequencies, the SV preconditioner outperforms the complex shifted Laplace preconditioner, but the latter is faster as the frequency increases.

Even though multigrid can help reduce the number of iterations, it brings only about 20% reduction in the CPU-time compared to ILU(1), see Table 5.9.

Table 5.7: CPU time (in sec.) for the wedge problem, shown for various frequencies f.

f (Hz) grid	10 76×126	20 151×251	30 232×386	40 301×501	50 376×626
No-Prec	9.5	84	333	658	–
ILU(A,0)	3.9	40	139	293	651
ILU(A,1)	1.8	20	70	155	195
ILU(M_I,0)	3.5	31	118	250	539
ILU(M_I,1)	1.7	17	61	117	137
ILU($M_{(0,1),h}$,0)	3.3	30	125	250	535
ILU($M_{(0,1),h}$,1)	1.7	14	55	119	138
MG(V(1,0))	0.8	9	33	77	141
MG(V(1,1))	1.0	12	44	103	198

Table 5.8: Number of iterations for a part of the Marmousi problem, shown for various frequencies f.

f (Hz) grid	1 751×201	10 751×201	20 1501×401	30 2001×534
No-Prec	17446	6623	14687	–
ILU(A,0)	3058	1817	3854	–
ILU(A,1)	715	740	1706	2391
ILU(M_I,0)	3282	1654	3645	–
ILU(M_I,1)	853	755	1718	2444
ILU($M_{(0,1),h}$,0)	2950	1519	3465	–
ILU($M_{(0,1),h}$,1)	715	743	1369	2010
MG(V(1,0))	17	176	307	493
MG(V(1,1))	13	165	307	442
M_{SV}	3	114	648	>2000

In the next chapter we will see that by not restricting the preconditioner $\mathcal{M}$ to a CSPD operator, a faster convergence for the iterative method can be achieved.

Table 5.9: CPU time (in sec.) for a part of the Marmousi problem, shown for various frequencies f.

f (Hz)	1	10	20	30
grid	751×201	751×201	1501×401	2001×534
No-Prec	1375	538	4572	–
ILU(A,0)	365	219	1904	–
ILU(A,1)	98	107	988	2275
ILU(M_I,0)	373	190	1671	–
ILU(M_I,1)	109	98	860	2247
ILU(M_{CSL},0)	371	185	1618	–
ILU(M_{CSL},1)	98	107	763	1894
MG(V(1,0))	7	55	507	1116
MG(V(1,1))	7	67	502	1300

Chapter 6

Multigrid-based preconditioner for the Helmholtz equation

In Chapter 4 we introduced a preconditioner for high wavenumber Helmholtz problems in heterogeneous media. The preconditioner was based on the Laplace operator with a zeroth-order imaginary term. In Chapter 5, in the context of preconditioned Krylov subspace methods, we have advocated the use of multigrid to approximately invert the preconditioner. Numerical results suggested that one multigrid iteration for the preconditioner solve is sufficient to result in a fast converging Krylov subspace method. The choice of preconditioner was, however, restricted by the fact that we only allowed the real part of the preconditioning matrix to be (positive) definite. In this chapter we will be more flexible in the choice of the parameters in the preconditioner. With more flexibility, we will see that the preconditioned system leads to a favorably clustered spectrum for a Krylov subspace convergence acceleration.

In this chapter we benefit from Fourier analysis in several ways. First of all, for idealized (homogeneous boundary conditions, constant coefficients) versions of the preconditioned system it is possible to visualize its spectrum for different values of the wavenumber, as Fourier analysis provides all eigenvalues, Section 6.1. Secondly, for analyzing multigrid algorithms quantitatively Fourier smoothing, two- and three-grid analysis [20, 21, 89, 92, 106] are the analysis tools of choice.

In the end of this chapter, convergence results are shown for the set of 2D model problems described in Section 2.6. These results shows that with more flexibility in the choice of parameters β_1 and β_2 in the preconditioning operator, a significant convergence improvement can be obtained.

6.1 More on the spectral properties of the shifted Laplace preconditioner

Our starting point is the shifted Laplace operator (4.14) written for 2D case as

$$\mathcal{M}_{(\beta_1,\beta_2)} := -\Delta - (\beta_1 - \hat{j}\beta_2)k^2, \tag{6.1}$$

with (β_1, β_2) now any arbitrary real value. Boundary conditions are set identical to those for the original Helmholtz problem. One member of the family is $\mathcal{M}_{(0,1)}$, which was analyzed thoroughly in the previous two chapters. In addition to this choice, we include in this chapter some more values for (β_1, β_2).

We perform Fourier analysis to investigate the effect of the choice of (β_1, β_2) in the preconditioner on the clustering of the eigenvalues of the preconditioned system. For this we consider operator (5.13) with homogeneous Dirichlet boundary conditions, wavenumber k constant, a discrete version of Helmholtz operator (1.26), A_h, and of preconditioner (6.1), M_h. For both A_h and M_h we choose the 5-point stencil. The components (5.12) are eigenfunctions of these discrete operators with constant coefficients. With these eigenfunctions $A_h M_h^{-1}$ is diagonalizable and the eigenvalues are easily determined. In the first tests we do not include damping in A_h, so $\alpha = 0$ in (1.26). We consider several possibilities: $\beta = (\beta_1, \beta_2) = \{(0,0), (-1,0), (0,1), (1,1), (1,0.5), (1,0.3)\}$. The first three pairs are known: $\beta = (0,0)$ is from Bayliss [12], $\beta = (-1,0)$ from [64], and $\beta = (0,1)$ as in [37, 38, 40] (see Chapters 4 and 5). The pairs with $\beta_1 = 1$ are new here. So we reduce the imaginary term in such a way that the operator $\mathcal{M}_{(\beta_1,\beta_2)}$ approximates the Helmholtz operator $\mathcal{A}$. Again, we use the notation $M_{h,(\beta_1,\beta_2)}$ to indicate the discrete version of $\mathcal{M}_{(\beta_1,\beta_2)}$ (6.1).

Figure 6.1 presents the spectra of $A_h M_h^{-1}$ for $M_{h,(0,0)}$, $M_{h,(-1,0)}$, $M_{h,(0,1)}$, $M_{h,(1,1)}$, $M_{h,(1,0.5)}$ and $M_{h,(1,0.3)}$. The results are for $k = 40$ ($k^2 = 1600$), $h = 1/64$. Similar eigenvalue distributions are observed for finer grids.

From the spectra presented with the shifted Laplace preconditioner the lower pictures of Figure 6.1 are favorable as their real parts vary between 0 and 1. Preconditioner $M_{h,(0,0)}$ in Figure 6.1a exhibits large isolated eigenvalues; for $M_{h,(-1,0)}$ the eigenvalues in Figure 6.1b are distributed between -1 and 1 on the real axis. The preconditioners with complex Helmholtz terms give rise to a curved spectrum. Whereas the real part of the spectrum in Figure 6.1c still includes a part of the negative real axis, this is not the case for the (β_1, β_2)-preconditioners with $\beta_1 = 1$. The difference between Figures 6.1d, 6.1e and 6.1f is that with a smaller value of β_2 fewer outliers close to the origin are observed. This is favorable for the convergence of the preconditioned Krylov methods. The approximate inversion of the preconditioner itself by multigrid, however, will be shown to be harder for smaller values of β_2.

In Figure 6.2 the spectra for $k = 100$ ($k^2 = 10^4$) on a grid with $h = 1/160$ are presented for $\beta_1 = 1$ and β_2 varying between 1 and 0.3. The spectra are very similar to those in Figure 6.1. More eigenvalues lie, however, in the vicinity of the origin, due to the higher wavenumber and the correspondingly finer grid.

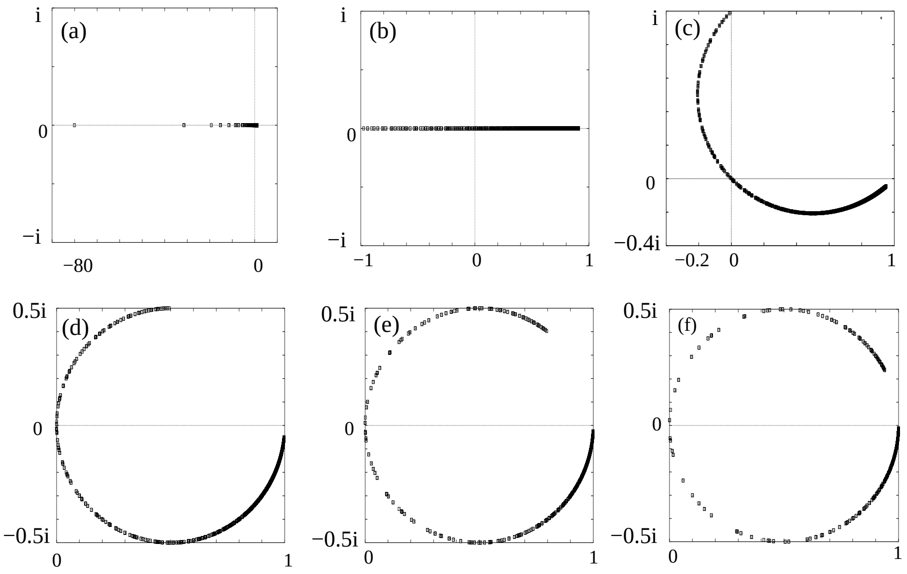

Figure 6.1: Spectral pictures of $A_h M_h^{-1}$ with $\alpha = 0$ and different values of (β_1, β_2) in (6.1): (a) $M_{h,(0,0)}$, (b) $M_{h,(-1,0)}$, (c) $M_{h,(0,1)}$, (d) $M_{h,(1,1)}$, (e) $M_{h,(1,0.5)}$, (f) $M_{h,(1,0.3)}$.

Figure 6.3 then presents the distribution of eigenvalues for the case that 5%s damping ($\alpha = 0.05$) is set in $\mathcal{A}$. Parameters in the preconditioner are $(\beta, \beta_2) = (1, 0.5)$. Again the 5-point stencil is used for discretization. Figure 6.3a presents the spectrum for $k = 40, h = 1/64$, and Figure 6.3b for $k = 100, h = 1/160$. An interesting observation is that now the eigenvalues move away from the origin into the right half plane. This is beneficial for iterative solution methods. From the spectra in Figure 6.3 it is expected that the Bi-CGSTAB convergence in the case of damping will be considerably faster than for the undamped case. As the circles have moved away from the origin it is possible to apply the classical theory of the GMRES convergence [84, 83], for example. One can place an ellipse around the spectrum. As it becomes a circle in the case of damping, it is expected that the resulting bounds for the GMRES convergence will be sharp.

6.2 Multigrid for the preconditioner

6.2.1 Fourier analysis and multigrid results

We first consider here solely the preconditioner $\mathcal{M}_{(\beta_1,\beta_2)}$ (6.1) and compare the numerical multigrid convergence with asymptotic convergence factors $\mu, \rho_{2g}, \rho_{3g}$ from Fourier analysis. (The behavior of the complete solution method will be considered in the next section.) Wavenumber k is taken constant here and a square domain with an equidistant grid is used. The second order Sommerfeld

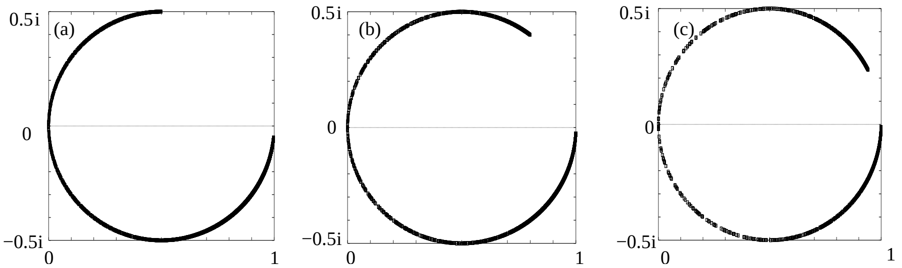

Figure 6.2: Spectral pictures of $A_h M_h^{-1}$ for $k = 100$, $h = 1/160$ and $\alpha = 0$; (a) $M_{h,(1,1)}$, (b) $M_{h,(1,0.5)}$, (c) $M_{h,(1,0.3)}$.

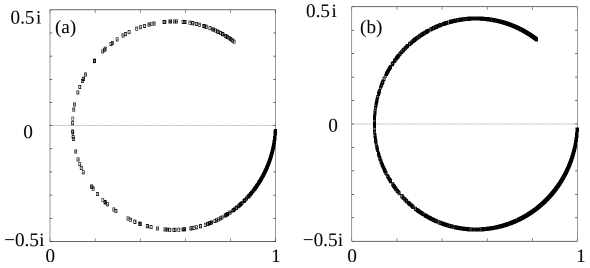

Figure 6.3: Spectral pictures of $A_h M^{-1}$ with 5 % damping in A_h and $M_{h,(1,0.5)}$; (a) $k = 40, h = 1/64$, (b) $k = 100, h = 1/160$.

radiation conditions (2.23) are set in the numerical experiments to mimic reality.

An interesting aspect is that almost identical convergence factors are obtained, both from the analysis and from the actual experiments, for constant values of kh (see Table 2.1). The results are validated from $k = 40$ up to $k = 600$ (the highest wavenumber tested, $k^2 = 3.6 \times 10^5$). The following abbreviations are used: 'ω_r-JAC' is the Jacobi smoother with underrelaxation. The Galerkin coarse grid discretization is denoted by 'galerkin' and a direct coarse grid discretization of the PDE 'direct'. The option 'direct' has not been implemented in the numerical code, but it is used in the analysis framework.

Multigrid coarsening is continued until fewer than 10×10 points are present on the coarsest grid. The number of levels is h- and therefore also k-dependent, as kh is kept constant on the finest grid, and varies between 5 and 9 grids.

The F-cycle is always used in the numerical tests with $\beta_1 = 1$; even though we used the V-cycle in Chapter 5 we found out that the V-cycle's performance was generally too poor for $\beta_1 = 1$ compared to the F-cycle, and the W-cycle is considered too expensive on the very fine grids processed at high wavenumbers. In the three-grid analysis, $\gamma = 2$, the W-cycle is used.

Remark 6.2.1 *The Fourier analysis applied directly to the Helmholtz equation (1.26) with $\alpha = 0$ and the specified meshsizes gives a satisfactory smoothing factor, but the two- and three-grid analysis convergence factors and also the* *actual multigrid results show a strong divergence, as expected.*

The Case ($\boldsymbol{\beta}_1$,$\boldsymbol{\beta}_2$)= $(\mathbf{1},\mathbf{1})$. The first test is for $M_{h,(1,1)}$. In this test we employ ω_r-JAC smoothing with $\omega_r = 0.7$ in an F(1,1)-cycle ($\nu_1 = \nu_2 = 1$). We compare the Galerkin discretization with the direct coarse grid PDE discretization. Analysis results with two smoothing iterations are shown in Table 6.1 and they are compared to the numerical F(1,1) multigrid convergence.

Table 6.1: Comparison of convergence $(\beta_1, \beta_2) = (1, 1)$, Fourier analysis convergence ($\gamma = 1$), ω_r-JAC, $\omega_r = 0.7$ and F(1,1)-cycle. Coarse grid discretizations are compared. (The direct discretization has not been implemented)

coarse discr.	μ	ρ_{2g}	ρ_{3g}, $\gamma = 2$	ρ_h, F(1,1)
galerkin	0.47	0.47	0.47	0.45
direct	0.47	0.47	0.47	-

Convergence factors well below 0.5 are obtained with the F(1,1)-cycle, and ω_r-JAC relaxation with $\omega_r = 0.7$. The Fourier analysis results with the Galerkin coarse grid discretization are very similar to those obtained with a direct coarse grid PDE discretization. We observe no difference between the results for the two coarse grid discretizations.

The Case ($\boldsymbol{\beta}_1$,$\boldsymbol{\beta}_2$) = $(\mathbf{1},\mathbf{0.5})$. This is our preconditioner of choice. For this parameter set it is possible to obtain a converging multigrid iteration by means of an F(1,1)-cycle, ω_r-JAC relaxation with $\omega_r = 0.5$, and a Galerkin coarse grid discretization. The underrelaxation parameter needs to be adapted for a robust convergence for a variety of heterogeneous Helmholtz problems. For values $\beta_2 < 0.5$ it is very difficult to find a satisfactory converging multigrid F(1,1)-cycle with the components at hand. They are therefore not considered.

Table 6.2 compares the Galerkin with the direct PDE coarse grid discretization. Here, the operator-dependent interpolation and full weighting restriction are chosen and two smoothing iterations are applied.

Table 6.2: Fourier analysis convergence factors compared to multigrid convergence $(\beta_1, \beta_2) = (1, 0.5)$. The smoother is ω_r-JAC with $\omega_r = 0.5$. (The direct discretization has not been implemented)

coarse discr.	μ	ρ_{2g}	ρ_{3g}, $\gamma = 2$	ρ_h, F(1,1)
galerkin	0.60	0.60	0.60	0.61
direct	0.60	0.60	0.60	-

The smoothing, two- and three-grid factors are very similar, which is an indication for the proper choice of coarse grid correction components for the problems under investigation. The numerical convergence with the F(1,1)-cycle is again very similar to the Fourier results.

In the following three remarks we explain the satisfactorily convergence with standard multigrid for the complex Helmholtz equation and $\beta_1 = 1$ with some heuristic arguments.

Remark 6.2.2 Smoothing. *The Fourier symbol of ω_r-JAC for the complex Helmholtz equation reads*

$$S_h = 1 - \frac{\omega_r}{4 - (\beta_1 - \beta_2 \hat{j})(hk)^2} \left(4 - (\beta_1 - \beta_2 \hat{j})(hk)^2 - 2\cos \ell_1 \pi h - 2 \cos \ell_2 \pi h\right), \quad \ell_1, \ell_2 = 1, \ldots, \sqrt{N} - 1.$$

We consider the case $k = 40, h = 1/64$ and take ω_r as in the previous experiments. Table 6.3 presents smoothing factors on four consecutive grids for $(\beta_1, \beta_2) = (1, 0)$ (original Helmholtz equation) and for $(\beta_1, \beta_2) = (0, 1), (1, 1)$ and $(1, 0.5)$. For simplicity, a direct PDE discretization on the coarse grids has been used.

Table 6.3: Smoothing factors μ_h for ω_r-JAC on different coarse grids and various (β_1, β_2)-values.

(β_1, β_2)	ω_r in ω_r-JAC	h :			
		1/64	1/32	1/16	1/8
$(1, 0)$	0.7	0.47	0.75	2.31	0.18
$(0, 1)$	0.8	0.36	0.32	0.13	0.05
$(1, 1)$	0.7	0.47	0.56	0.35	0.13
$(1, 0.5)$	0.5	0.60	0.77	0.81	0.32

From Table 6.3, one finds that for $h = 1/16$ ω_r-JAC diverges for the original Helmholtz operator (also found with other relaxation parameters). This is in accordance with the remarks in [23, 33], that smoothing problems do not occur on the very fine or the very coarse grids, but on the intermediate grids. Furthermore, it can be observed that the $(\beta_1, \beta_2) = (0, 1)$-preconditioner resembles a Laplace-type situation, with excellent smoothing factors on all grids. The preconditioners with $\beta_1 = 1$ give smoothing factors less than one on every grid. The $(1, 1)$-preconditioner exhibits better smoothing factors than the set $(\beta_1, \beta_2) = (1, 0.5)$, which represents a limit case for which smoothing factors are still below one.

Remark 6.2.3 Simplified coarse grid analysis. *Some insight into the coarse grid correction can be gained from the so-called 'simplified coarse grid analysis' or first-differential-approximation analysis [21, 23, 92]. As in [33] we apply this analysis for a 1D Helmholtz operator. Assuming that transfer operators do not have any effect on the lowest frequencies, the quantity $1 - \lambda_h^{\ell_1} / \lambda_{2h}^{2\ell_1}$ (ℓ_1*

small) gives some insight into the relation between the discrete fine and coarse grid operators. This quantity should be close to zero and is an indication of the suitability of a coarse grid operator in multigrid. For the original 1D Helmholtz equation and $\alpha = 0$ (no damping) this quantity reads [33]:

$$1 - \lambda_h^{\ell_1}/\lambda_{2h}^{2\ell_1} = \frac{\sin^4(\ell_1 h\pi/2)}{\sin^2(\ell_1 h\pi/2)\cos^2(\ell_1 h\pi/2) - (kh/2)^2}, \ell_1 = 1, \ldots, N.$$

It may give rise to a problematic coarse grid correction in the range where

$$\sin^2(\ell_1 h\pi/2)\cos^2(\ell_1 h\pi/2) \approx (kh/2)^2$$

and ℓ_1 is associated with a smooth mode. For a 1D version of the complex Helmholtz operator, this quantity reads

$$\begin{aligned} 1 - \lambda_h^{\ell_1}/\lambda_{2h}^{2\ell_1} &= \frac{\sin^4(\ell_1 h\pi/2)}{\sin^2(\ell_1 h\pi/2)\cos^2(\ell_1 h\pi/2) - (kh/2)^2(\beta_1 - \beta_2\hat{\jmath})} \\ &= \frac{\sin^4(\ell_1 h\pi/2)\left(\sin^2(\ell_1 h\pi/2)\cos^2(\ell_1 h\pi/2) - (kh/2)^2(\beta_1 + \beta_2\hat{\jmath})\right)}{\left(\sin^2(\ell_1 h\pi/2)\cos^2(\ell_1 h\pi/2) - (kh/2)^2\beta_1\right)^2 + (kh/2)^2\beta_2^2}, \\ &\ell_1 = 1, \ldots, N. \end{aligned}$$

This expression for the complex Helmholtz operator is close to zero for the (β_1, β_2)-sets under consideration: the denominator does not reach zero, and the numerator contains the term $\sin^4 \ell h\pi/2$ which is very small for smooth eigenmodes.

Remark 6.2.4 h-ellipticity. *When a Galerkin coarse grid discretization is used, it is difficult to gain insight into the coarse grid correction, as the coarse grid stencil elements are constructed with nontrivial formulas. Therefore, we discuss here for the case $(\beta_1, \beta_2) = (1, 0.5)$ two coarse grid discretizations. With $h = 1/64, k = 40$, $\alpha = 0$ in the five-point stencil, we obtain by direct PDE discretization similar coarse grid stencils as the fine grid stencil with grid sizes $2h$ or $4h$, respectively. In that case, only the central stencil element contains an imaginary contribution. When the Galerkin coarse grid operator is employed, the imaginary part is distributed over all entries. With operator-dependent interpolation and full weighting restriction we find,*

$$\begin{aligned} A_{2h} &\mathrel{\hat{=}} \begin{bmatrix} -282.9 + 15.3\hat{\jmath} & -665.8 + 80.6\hat{\jmath} & -282.9 + 15.3\hat{\jmath} \\ -665.8 + 80.6\hat{\jmath} & 2164.5 + 461.2\hat{\jmath} & -665.8 + 80.6\hat{\jmath} \\ -282.9 + 15.3\hat{\jmath} & -665.8 + 80.6\hat{\jmath} & -282.9 + 15.3\hat{\jmath} \end{bmatrix}, \\ A_{4h} &\mathrel{\hat{=}} \begin{bmatrix} -129.5 + 43.0\hat{\jmath} & -290.1 + 135.0\hat{\jmath} & -129.5 + 43.0\hat{\jmath} \\ -290.1 + 135.0\hat{\jmath} & -101.4 + 483.2\hat{\jmath} & -290.1 + 135.0\hat{\jmath} \\ -129.5 + 43.0\hat{\jmath} & -290.1 + 135.0\hat{\jmath} & -129.5 + 43.0\hat{\jmath} \end{bmatrix}. \end{aligned} \quad (6.2)$$

The h-ellipticity measures, indicating the suitability of the stencils for pointwise smoothing [21, 92], are 0.28 and 0.18. For the direct PDE discretization *the h-ellipticity measures are 0.13 and 0.45 for the $2h$- and $4h$-discretization, respectively. The fact that these qualitative measures are not close to zero means that point-wise smoothers can be constructed for these stencils. From these complicated coarse grid stencils it is, however, difficult to judge between the different smoothers, relaxation parameters etc. but the three-grid Fourier analysis helps to some extent. We obtain very satisfactory multigrid convergence with simple multigrid components, although the coarse grid discretization (6.2) seems awkward. At least it does not spoil the h-independent multigrid convergence. One merely needs to choose the underrelaxation parameter in the smoother with some care.*

6.2.2 Multigrid for the preconditioner

One multigrid iteration is taken for approximating the inverse of the operator in (5.13). After some experimentation it was found that it is sufficient to employ a multigrid iteration with a convergence factor $\rho_h \approx 0.6$ for the preconditioner. This can also be observed qualitatively from spectral pictures obtained by Fourier analysis as follows. Let us start with a regular splitting of M_h:

$$C_h u_h^{j+1} = (C_h - M_h)u_h^j + w_h, \quad \text{or} \quad u_h^{j+1} = (I_h - C_h^{-1}M_h)u_h^j + C_h^{-1}w_h. \quad (6.3)$$

This splitting is considered to represent a multigrid iteration, with iteration matrix $(I_h - C_h^{-1}M_h)$ and C_h^{-1} an *approximation* of M_h^{-1}. T_h^{2h} in (5.24) represents the two-grid version of a multigrid iteration matrix. Therefore, we equate $T_h^{2h} = I_h - C_h^{-1}M_h$. Matrix $\widetilde{T}_h^{2h}$ in (5.26) is a block matrix related to T_h^{2h}: $\widetilde{T}_h^{2h} = U_h T_h^{2h} U_h^{-1}$, where U_h is a unitary matrix with four consecutive rows defined by the orthogonal eigenvectors related to (5.12). U_h transforms the two-grid iteration matrix into the block diagonal matrix $\widetilde{T}_h^{2h}$. Clearly,

$$\begin{aligned} \widetilde{T}_h^{2h} &= I_h - U_h C_h^{-1} M_h U_h^{-1}, \text{ and} \\ U_h C_h^{-1} M_h U_h^{-1} &= U_h C_h^{-1} U_h^{-1} U_h M_h U_h^{-1} =: \widetilde{C}_h^{-1} \widetilde{M}_h \end{aligned}$$

is in block diagonal form. We have $\widetilde{C}_h^{-1}\widetilde{M}_h\widetilde{M}_h^{-1} = (I_h - \widetilde{T}_h^{2h})\widetilde{M}_h^{-1}$. So, the expression for the block diagonal form $\widetilde{A}_h\widetilde{C}_h^{-1}$ ($\widetilde{C}_h^{-1}$ the approximation of $\widetilde{M}_h^{-1}$) from (5.13) reads

$$\widetilde{A}_h\widetilde{C}_h^{-1} = \widetilde{A}_h(I_h - \widetilde{T}_h^{2h})\widetilde{M}_h^{-1} \quad (6.4)$$

As all symbols of the operators in the right-hand side of (6.4) can be formed easily with Fourier two-grid analysis, the corresponding eigenvalues can be visualized for various multigrid cycles. These spectra can be compared to those in Figure 6.1, where operator M_h from (6.1) is inverted exactly. Figure 6.4, for example, presents the spectrum of the $(\beta_1, \beta_2) = (1, 1)$-preconditioned system where a two-grid iteration is used for preconditioning, for wavenumber $k = 40$ ($h = 1/64$). The left-side picture shows the spectrum with one ω_r-JAC ($\omega_r = 0.7$) smoothing iteration for which $\rho_{2g} \approx 0.7$, whereas the right-side picture shows the two-grid spectral picture with two ω_r-JAC smoothing iterations,

$\nu_1 + \nu_2 = 2$, and operator-dependent interpolation, full weighting restriction, Galerkin coarse grid discretization ($\rho_{2g} = 0.45$). The right-side picture shows a spectrum that coincides well with the spectrum related to the exact inversion in Figure 6.1d, whereas in the left-side picture eigenvalues are also outside the circle obtained with the exact inversion.

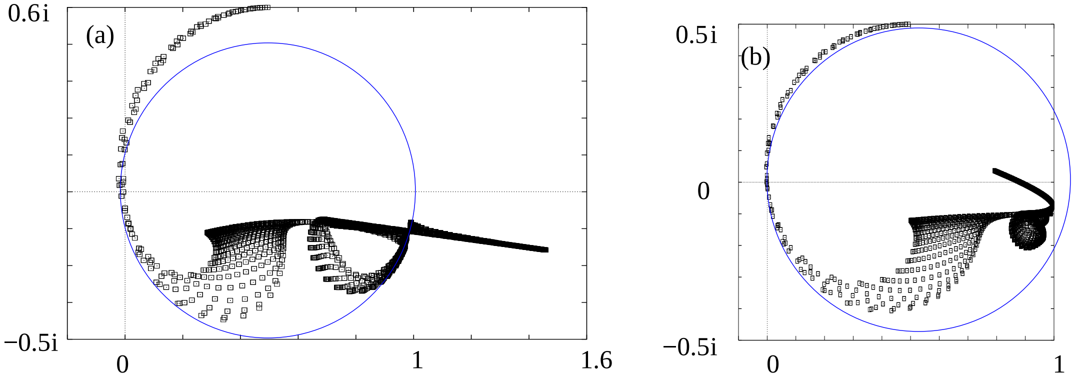

Figure 6.4: Spectral pictures of preconditioned system with one two-grid iteration used for preconditioning $(\beta_1, \beta_2) = (1, 1)$, $k = 40, h = 1/64$, (a) one ω_r-JAC relaxation, (b) two ω_r-JAC relaxations, $\omega_r = 0.7$ (the eigenvalues with the exact inversion lie at the circles).

Figure 6.5 presents the spectra with a two-grid iteration for the $(\beta_1, \beta_2) = (1, 0.5)$-preconditioner and Galerkin coarsening, ω_r-JAC relaxation ($\omega_r = 0.5$). The left-side picture is for $\nu = 1$; the right-side picture for $\nu = 2$. Also for this approximate inversion of the preconditioner the spectrum obtained in the right-side picture compares well with the exact inversion in Figure 6.1e, indicating that one multigrid iteration with two ω_r-JAC smoothing steps may be sufficient for approximating M_h^{-1}.

6.3 Numerical results

In this section the overall solution method, preconditioned Bi-CGSTAB for the indefinite heterogeneous Helmholtz problems with the complex Helmholtz (β_1, β_2)-preconditioner is evaluated. One multigrid F(1,1)-cycle is used for approximately inverting preconditioner equation with the complex Helmholtz operator. The three model problems of increasing difficulty, as described in Section 2.6, are again discussed.

6.3.1 Model Problem 1: Constant wavenumber

For constant wavenumbers k (Model Problem 1) the Bi-CGSTAB convergence history for the Helmholtz equation with the three preconditioners is presented.

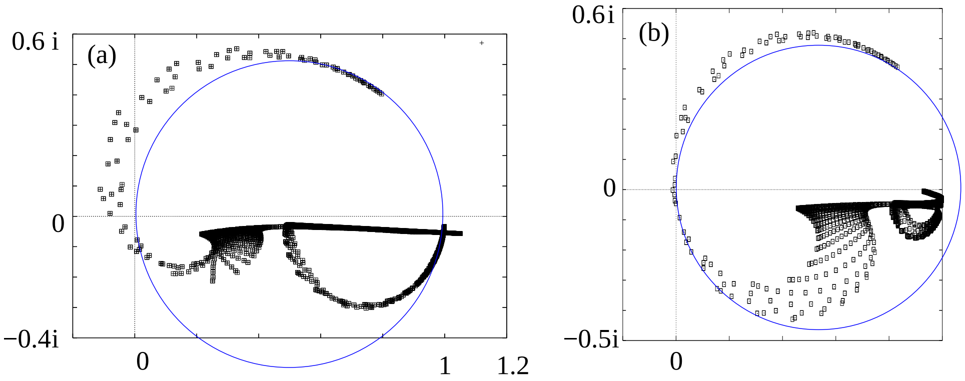

Figure 6.5: Spectral pictures of preconditioned system with one two-grid iteration used for preconditioning $(\beta_1, \beta_2) = (1, 0.5)$, $k = 40, h = 1/64$, (a) one ω_r-JAC relaxation, (b) two ω_r-JAC relaxations, $\omega_r = 0.5$ (the eigenvalues with exact inversion lie at the circles).

At the boundaries we apply the second order Sommerfeld radiation conditions (2.23). In these experiments the finest grid size for each wavenumber is as in Table 2.1. The numerical solution corresponding to $k = 50$ has been presented in Figure 2.2.

A zero initial guess has been used during the computations. The Bi-CGSTAB iteration is terminated as soon as the criterion (3.25) is satisfied.

For all three preconditioners, $(\beta_1, \beta_2) = (0, 1), (1, 1)$ and $(1, 0.5)$, the method chosen to approximately invert the preconditioner consists of the multigrid F(1,1)-cycle with ω_r-JAC, operator-dependent interpolation plus full weighting as the transfer operators and a Galerkin coarse grid discretization. The only difference is the value of the underrelaxation parameter in ω_r-JAC, which is $\omega_r = 0.8$ for $(\beta_1, \beta_2) = (0, 1)$, $\omega_r = 0.7$ for $(\beta_1, \beta_2) = (1, 1)$ and $\omega_r = 0.5$ for $(\beta_1, \beta_2) = (1, 0.5)$.

The results for different values of k and $(\beta_1, \beta_2) = (0, 1)$ are presented in the upper part of Tables 6.4 (for the number of iterations) and 6.5 (for CPU time). In the middle part of Tables 6.4 and 6.5, the Bi-CGSTAB convergence with the $(\beta_1, \beta_2) = (1, 1)$-preconditioner is presented. In the lower lines of Tables 6.4 and 6.5 the $(\beta_1, \beta_2) = (1, 0.5)$-preconditioner is employed. Next to the results for the Helmholtz equation without any damping ($\alpha = 0$) we also show the convergence with 2.5% ($\alpha = 0.025$), and 5% ($\alpha = 0.05$) damping. The number of Bi-CGSTAB iterations are presented as well as the CPU time. From the results in Tables 6.4 and 6.5 we conclude that the preferred methods among the choices are the preconditioners with $\beta_1 = 1$. This was already expected from the spectra in Figure 6.1. Fastest convergence is obtained for $(\beta_1, \beta_2) = (1, 0.5)$.

Table 6.4 shows that the Bi-CGSTAB convergence with some damping in the Helmholtz problem is considerably faster than for $\alpha = 0$. This was also expected from the spectra in Figure 6.3. Furthermore, the number of iterations

Table 6.4: Number of prec. Bi-CGSTAB iterations to reduce the initial residual by 7 orders. Damping parameter α is varied in the Helmholtz problem.

		$k:$				
(β_1, β_2)	α from (1.26)	40	50	80	100	150
$(0, 1)$	$\alpha = 0$	57	73	112	126	188
	2.5% damping	48	61	84	93	121
	5% damping	45	55	69	75	97
(1,1)	$\alpha = 0$	36	39	54	74	90
	2.5% damping	33	37	44	51	61
	5% damping	28	30	36	41	49
(1,0.5)	$\alpha = 0$	26	31	44	52	73
	2.5% damping	24	26	33	39	47
	5% damping	21	23	28	32	37

in the case of damping grows only slowly for increasing wavenumbers, especially for the $(\beta_1, \beta_2) = (1, 0.5)$-preconditioner.

The difference between the two preconditioners with $\beta_1 = 1$ is more pronounced if we compute higher wavenumbers. The Bi-CGSTAB convergence and CPU times for the higher wavenumbers, without and with damping in the Helmholtz problem are presented in Table 6.6.

Also for the higher wavenumbers damping in the Helmholtz problem by means of $\alpha \neq 0$ improves the convergence significantly. Very satisfactory convergence is found for high wavenumbers on fine grids.

6.3.2 Model Problem 2: The wedge model

A problem of intermediate difficulty is Model Problem 2, used to evaluate the preconditioner's behavior for a simple heterogeneous medium (see Figure 2.3 for the velocity variation and the solution for frequency 30 and 50 Hz). Here, the second order Sommerfeld radiation boundary conditions (2.23) are applied.

In the preconditioner with the complex Helmholtz equation, wavenumber $k(x, y)$ is chosen as in the original problem. Also the boundary conditions in the preconditioner are as for the original problem. The number of Bi-CGSTAB iterations with one multigrid iteration for the preconditioner with $(\beta_1, \beta_2) = (0, 1), (1, 1)$ and $(1, 0.5)$ are displayed in Table 6.7, for frequencies ranging from 10 to 60 Hz on corresponding grid sizes. Results with and without damping in the Helmholtz problem are presented. The only difference in the multigrid methods for the preconditioner is the value of the relaxation parameter: for $(\beta_1, \beta_2) = (0, 1)$ $\omega_r = 0.8$, for $(\beta_1, \beta_2) = (1, 1)$ $\omega_r = 0.7$, for $(\beta_1, \beta_2) = (1, 0.5)$ $\omega_r = 0.5$. A zero initial guess has been used as starting approximation.

The convergence results for $(\beta_1, \beta_2) = (1, 0.5)$ are best, also without any damping in the original problem. The convergence with the (1,0.5)-preconditioner

Table 6.5: CPU time (in seconds) to reduce the initial residual by 7 orders. Damping parameter α is varied in the Helmholtz problem.

		k :				
(β_1, β_2)	α in (1.26)	40	50	80	100	150
$(0, 1)$	$\alpha = 0$	0.44	0.92	4.3	7.7	28.5
	2.5% damping	0.38	0.77	3.3	5.6	18.5
	5% damping	0.35	0.70	2.7	4.7	14.9
(1,1)	$\alpha = 0$	0.30	0.51	2.2	4.5	13.9
	2.5% damping	0.27	0.48	1.8	3.2	9.6
	5% damping	0.24	0.39	1.5	2.6	7.5
(1,0.5)	$\alpha = 0$	0.21	0.40	1.8	3.3	10.8
	2.5% damping	0.20	0.35	1.4	2.5	7.3
	5% damping	0.18	0.32	1.2	2.1	5.8

Table 6.6: High wavenumbers, number of Bi-CGSTAB iterations and CPU time in seconds (in parentheses) to reduce the initial residual by 7 orders with and without damping in the Helmholtz problem.

		k :		
(β_1, β_2)	α in (1.26)	200	500	600
(1,1)	$\alpha = 0$	114 (30.8)	291 (515)	352 (890)
	2.5 % damping	74 (20.2)	125 (227)	145 (372)
	5 % damping	56 (15.5)	95 (174)	80 (205)
(1,0.5)	$\alpha = 0$	92 (25.4)	250 (425)	298 (726)
	2.5 % damping	57 (15.2)	91 (164)	102 (252)
	5 % damping	44 (11.9)	64 (115)	66 (165)

is about 1.5 times faster than with the (1,1)-preconditioner and about 3 times faster than with the (0,1)-preconditioner. The Bi-CGSTAB convergence for the wedge problem for $\alpha = 0$ and different frequencies is visualized for $(\beta_1, \beta_2) = (1, 0.5)$ in Figure 6.6.

6.3.3 Model Problem 3: The Marmousi problem

Finally, we present numerical results for Model Problem 3 with the second order Sommerfeld conditions imposed at the boundaries (see Figure 2.4a; see also numerical results in Section 5.5).

Preconditioning consists of one multigrid iteration for the complex Helmholtz equation with the multigrid components prescribed. The underrelaxation parameter in ω-JAC is depending on (β_1, β_2). In the preconditioner again the wavenumbers $k(x_1, x_2)$ are as in the original problem. Also the boundary con-

Table 6.7: Bi-CGSTAB convergence for the wedge problem with and without damping and the three multigrid based (β_1, β_2)-preconditioners compared. Number of Bi-CGSTAB iterations and CPU time in seconds (in parentheses).

			(β_1, β_2)		
f (Hz)	Grid	damping	(0,1)	(1,1)	(1,0.5)
10	75 × 125	0.0%	52 (1.2)	30 (0.67)	19 (0.42)
		2.5%	48 (1.1)	27 (0.62)	17 (0.39)
		5.0%	42 (0.91)	25 (0.57)	16 (0.38)
20	149 × 249	0.0%	91 (8.8)	45 (4.5)	27 (2.8)
		2.5%	75 (7.2)	39 (4.0)	23 (2.4)
		5.0%	65 (6.3)	35 (3.5)	20 (2.1)
30	232 × 386	0.0%	128 (30.6)	64 (15.8)	37 (9.4)
		2.5%	94 (22.8)	49 (12.3)	29 (7.5)
		5.0%	86 (21.0)	42 (10.7)	25 (6.6)
40	301 × 501	0.0%	161 (66.1)	80 (33.5)	49 (20.8)
		2.5%	116 (48.0)	60 (25.4)	35 (15.2)
		5.0%	91 (37.9)	46 (19.8)	28 (12.4)
50	376 × 626	0.0%	205 (134.5)	98 (65.5)	58 (38.7)
		2.5%	135 (89.0)	67 (45.5)	37 (24.8)
		5.0%	99 (66.5)	54 (37.1)	32 (22.0)
60	481 × 801	0.0%	232 (247.3)	118 (127.6)	66 (71.9)
		2.5%	147 (159.1)	74 (81.1)	42 (47.1)
		5.0%	110 (119.6)	58 (64.5)	32 (36.7)

ditions are as in the original problem. Table 6.8 presents the number of Bi-CGSTAB iterations to solve the indefinite Helmholtz Marmousi problem with in parentheses the CPU times required. Results are presented for $\alpha = 0, 0.025$ and 0.05. A zero initial guess has been used. The $(\beta_1, \beta_2) = (1, 0.5)$-preconditioner shows a satisfactory and robust convergence, also for this problem with irregularly varying wavenumbers. For $\alpha = 0.05$ the number of iterations increases only very slowly for increasing frequencies. With the $(\beta_1, \beta_2) = (1, 0.5)$-preconditioner the CPU time is reduced by a factor 3, compared to the performance of the $(\beta_1, \beta_2) = (0, 1)$-preconditioner for the challenging problems. The difference with the $(\beta_1, \beta_2) = (1, 1)$ is less pronounced, but still significant.

The real parts of the solutions at 20 Hz for $\alpha = 0$ and $\alpha = 0.025$ are presented in Figure 2.4b and 2.4c. The effect of damping of the solution is significant, as can be deduced from these global pictures. However, in the actual applications, some damping is present. The right side picture may therefore be a more realistic solution for the real application.

In comparison with convergence results based on the ILU(1) preconditioner, presented in Chapter 5, an acceleration by a factor of more than four in terms of CPU time is obtained by using $(\beta_1, \beta_2) = (1, 0.5)$. Therefore, the method

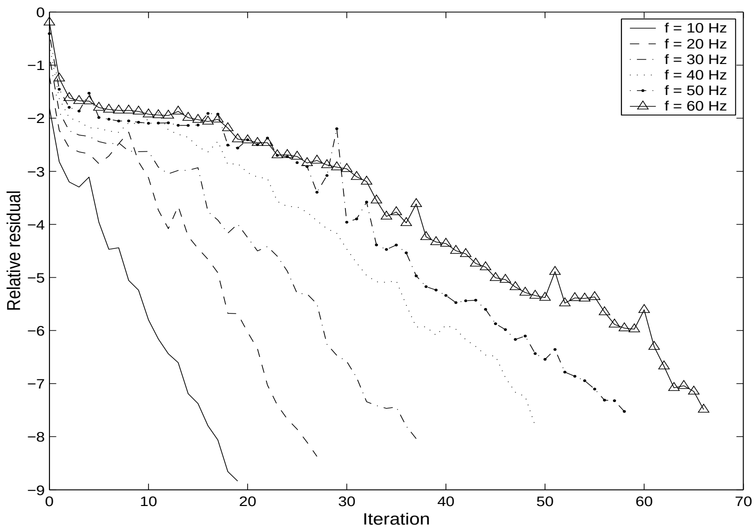

Figure 6.6: Bi-CGSTAB convergence plot for $(\beta_1, \beta_2) = (1, 0.5)$ for the wedge problem at different frequencies, $\alpha = 0$.

Table 6.8: Bi-CGSTAB convergence for the Marmousi problem with and without damping and the three multigrid based (β_1, β_2)-preconditioners. Number of Bi-CGSTAB iterations and CPU time in seconds (in parentheses).

			(β_1, β_2)		
f (Hz)	Grid	damping	(0,1)	(1,1)	(1,0.5)
1	751 × 201	0.0%	91 (36)	55 (22)	39 (16)
		2.5%	87 (34)	46 (18)	36 (15)
		5.0%	78 (31)	47 (19)	32 (14)
10	751 × 201	0.0%	187 (72)	89 (35)	54 (22)
		2.5%	123 (48)	59 (23)	34 (15)
		5.0%	95 (37)	49 (20)	29 (12)
20	1501 × 401	0.0%	388 (642)	162 (273)	98 (167)
		2.5%	202 (411)	86 (147)	54 (94)
		5.0%	142 (295)	60 (104)	35 (64)
30	2001 × 534	0.0%	484 (1472)	212 (649)	144 (444)
		2.5%	208 (640)	96 (300)	60 (192)
		5.0%	121 (380)	63 (200)	37 (122)

shows its effectiveness.

Remark 6.3.1 *In Chapter 5, the first order radiation conditions (2.26) are used, and the termination criterion is one order of magnitude larger than that* *we use in this chapter. Our observation reveals that different boundary conditions do not significantly affect the convergence. As long as the same boundary conditions are used in both the Helmholtz and the preconditioning operator, the convergence for a given frequency remains practically the same.*

An adaptation of the solution method presented to a variant of the Helmholtz problem in which the damping parameter α is varying *locally* will be an easy generalization.

6.3.4 Convergence result for the 9-point stencil

Numerical tests have also been performed by using the 9-point stencil (2.13) with $\gamma = 4/5$. Absorption layers (denoted by Ω_2) are added at the physical boundaries (Ω_1) with a thickness of 50 gridpoints (see Figure 6.7). In Ω_2 the damped Helmholtz equation is used with

$$\alpha = 0.25\frac{(x-x_d)^2}{(x_m-x_d)^2}, \tag{6.5}$$

where x_d is the position of the physical boundary Γ_1 and x_m is the position of the boundary of the absorption layer, Γ_2. On Γ_2 the first order radiation condition (2.26) is imposed. The addition of absorption layers may further reduce undesired reflections from the physical boundaries.

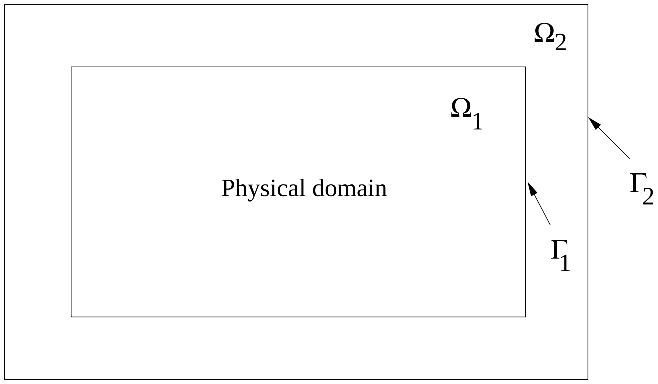

Figure 6.7: Domain with absorption layers (Ω_2)

We use $(\beta_1, \beta_2) = (1, 0.5)$ for the preconditioner, which is discretized by the *5-point* stencil (this is in contrast with Am which is given by a 9-point stencil). Numerical results are shown here for Model Problem 3; see Table 6.9. The real part of the solutions at 20 Hz for $\alpha = 0$ and $\alpha = 0.025$ has been presented in Figure 2.5b and 2.5c.

Compared to the results in Table 6.8, the use of the 9-point stencil does not influence the convergence negatively, as the convergence is similar to that

Table 6.9: Bi-CGSTAB convergence for the Marmousi problem with and without damping. The Helmholtz equation is discretized by using a 9-point stencil ($\gamma = 4/5$). The preconditioner is based on $(\beta_1, \beta_2) = (1, 0.5)$, discretized with the 5-point stencil. 50 gridpoints are added for absorption layers.

	damping, α		
f(Hz)	0.0%	2.5%	5.0%
1	9 (9)	9 (9)	8 (8)
10	47 (37)	33 (27)	24 (20)
20	87 (193)	46 (106)	33 (78)
30	138 (513)	56 (215)	39 (154)

with the 5-point stencil. The CPU time increases somewhat due to the larger problem size. This is due the inclusion of absorption layers. Furthermore, this result shows that the preconditioning operator needs not be discretized by the same discretization for the Helmholtz equation. Similar results are also observed for cases without absorption layers (not presented).

Chapter 7

Three-dimensional Helmholtz equation

In the preceding chapters we have show that multigrid applied to the shifted Laplace preconditioner can effectively enhance the convergence of Krylov subspace methods (which, in this case, is Bi-CGSTAB) for the 2D heterogeneous Helmholtz equation at high wavenumbers. It has been shown, theoretically and numerically, that the convergence is h-independent and *mildly* depends on the wavenumber k for the undamped case ($\alpha = 0$). For Helmholtz problems with damping ($\alpha \sim 0.05$) the convergence is observed to be also independent of k.

In this chapter we discuss the extension of the method to 3D [39]. The general algorithm is similar: multigrid-based preconditioners within Bi-CGSTAB. For Bi-CGSTAB the algorithm is immediately extendable to 3D. For multigrid, however, a multigrid algorithm with semicoarsening is employed. By this the coarsening is not done in all three directions. Rather, one does coarsening in one or two directions, while the other directions are uncoarsened, in which line (or plane) smoothing is applied. After introducing the 3D Helmholtz equation and the 3D preconditioner, a multigrid method with semicoarsening for approximately inverting the preconditioner is discussed in Section 7.2. Applications of the method to 3D model problems are given in Section 7.3.

7.1 3D Helmholtz equation

We consider the three-dimensional Helmholtz equation in a 3D rectangular domain $\Omega \subset \mathbb{R}^3$:

$$\mathcal{A}u := -\left(\partial_{x_1x_1} + \partial_{x_2x_2} + \partial_{x_3x_3} + (1 - \alpha\hat{j})k^2\right) u = g, \tag{7.1}$$

with boundary conditions on $\Gamma \equiv \partial\Omega$ as determined by (2.23)–(2.26). Equation (7.1) is discretized using the standard 7-point finite difference stencil.

In 3D the preconditioning operator reads

$$\mathcal{M} := -\left(\partial_{x_1x_1} + \partial_{x_2x_2} + \partial_{x_3x_3} + (1 - 0.5\hat{j})k^2\right). \tag{7.2}$$

In this chapter we simplify the notation for the shifted Laplace preconditioner as we only consider one particular case with $(\beta_1, \beta_2) = (1, 0.5)$. So, here $\mathcal{M}$ always refers to $\mathcal{M}_{(1,0.5)}$. The preconditioner M is then constructed from discretization of (7.2) using the 7-point finite difference stencil, with the same boundary conditions as for (7.1). The right preconditioned linear system to solve is then given as

$$A_h M_h^{-1} w_h = g_h, \quad w_h = M_h u_h, \tag{7.3}$$

where $M_h u_h = w_h$ is to be solved approximately using one multigrid iteration.

7.2 Multigrid with semicoarsening for the preconditioner

For a three-dimensional problem, the choice of multigrid components is richer than in two dimensions. For the 3D Poisson equation, multigrid method with standard coarsening and point-wise smoothing already results in a good convergence. For anisotropic Poisson problems, however, multigrid with standard coarsening and *point* smoother is not sufficient; the convergence rate is not the typically fast multigrid convergence. For such problems, the use of *line* (or *plane*) smoothers is important. A point smoother can only produce fast multigrid convergence if a semicoarsening strategy is incorporated (see [91]). For more complicated situations, where strong variations exist in an arbitrary way in the domain, the use of *algebraic* multigrid is a natural choice (not treated in this thesis).

The shifted Laplace operator does not seem to require a non-standard multigrid method; multigrid method with point smoother seems sufficient. This is an observation in 2D, even for Helmholtz problems with strong heterogeneities (see Chapter 6). We, however, prefer to use a multigrid method with semicoarsening because if coarsening is only done in two directions, all 2D multigrid components (which were analyzed in Chapters 5 and 6) can be immediately used in a 3D setting. For example, the interpolation operator can be built based on bilinear interpolation or operator-dependent interpolation by de Zeeuw. The restriction operator can also be determined in the same way as in 2D. In the third direction, where the grid is kept uncoarsened, line smoothing is employed. Furthermore, a 2D parallelization strategy can immediately be adopted in 3D.

A discussion on multigrid with semicoarsening can be found, e.g. in [73, 78], for multiple semicoarsened grids (MSG) method, or in [77] for the so-called MG-S method, which is also based on multiple semicoarsening. For MG-S, the behavior of the method can be analyzed based on standard multigrid methods. Processing the semicoarsened grids in the case of MG-S can be viewed as a smoothing iteration in the standard multigrid sequence. Extension of *flexible* MG-S to 3D problems with anisotropy is presented in [104].

The 3D multigrid with (x_1, x_2) semicoarsening is illustrated in Figure 7.1 for three grid levels. In this case standard coarsening, i.e. doubling the mesh size, only takes place in x_1- and x_2-directions, while the x_3-direction is kept uncoarsened. For the shifted Laplace operator (and the Helmholtz operator) anisotropies do not appear. Therefore, the choice of coarsening is rather arbitrary. We show in the numerical results in Section 7.3 that the choice of coarsening direction does not have a significant effect to the convergence of the method.

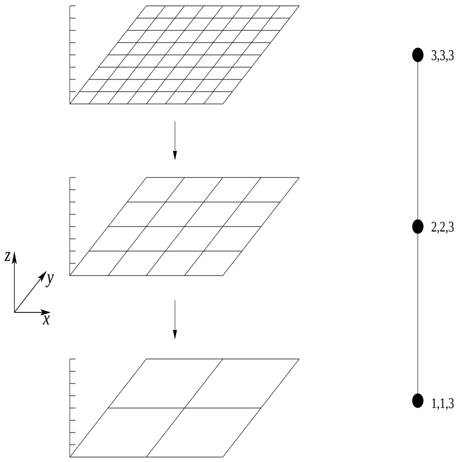

Figure 7.1: Semicoarsening of three grid levels: standard coarsening in two directions (x_1 and x_2), the third (x_3) direction is kept uncoarsened

Consider the preconditioning operator (7.2) which is discretized by the 7-point stencil on a grid $\mathcal{G}_h = \{(i_1h, i_2h, i_3h)|i_1, i_2, i_3 = 1, \ldots, \sqrt[3]{N}-1\}$, with $h = 1/\sqrt[3]{N}$. For a damped x_3-line Jacobi relaxation with relaxation factor $0 < \omega_r \leq 1$ we have the following iteration:

$$(M_{h,x_3} + D_h)\overline{u}_h^{j+1} + (M_{h,x_1} + M_{h,x_2})u_h^j = w_h, \tag{7.4}$$

$$u_h^{j+1} = \omega_r \overline{u}_h^{j+1} + (1-\omega_r)u_h^j, \tag{7.5}$$

with

$$(M_{h,x_1}u)_{i_1,i_2,i_3} := -\frac{1}{h^2}\left(u_{i_1+1,i_2,i_3} + u_{i_1-1,i_2,i_3}\right), \tag{7.6}$$

$$(M_{h,x_2}u)_{i_1,i_2,i_3} := -\frac{1}{h^2}\left(u_{i_1,i_2+1,i_3} + u_{i_1,i_2-1,i_3}\right), \tag{7.7}$$

$$(M_{h,x_3}u)_{i_1,i_2,i_3} := -\frac{1}{h^2}\left(u_{i_1,i_2,i_3+1} + u_{i_1,i_2,i_3-1}\right), \tag{7.8}$$

$$(D_h u)_{i_1,i_2,i_3} := \frac{1}{h^2}\left(6 - (1-0.5\hat{\jmath})k^2h^2\right)u_{i_1,i_2,i_3}. \tag{7.9}$$

Equations (7.4)– (7.5) can be rewritten as

$$\frac{1}{\omega_r}(M_{h,x_3}+D_h)u_h^{j+1} = w_h - (M_{h,x_1}+M_{h,x_2})u_h^j + \frac{1-\omega_r}{\omega_r}(M_{h,x_3}+D_h)u_h^j,$$

If the error at the j-th iteration is defined by $e^j = u_h^j - u_h$, where u_h is the exact solution of $M_h u_h = w_h$, we find that

$$\frac{1}{\omega_r}(M_{h,x_3}+D_h)e^{j+1} = \left(\frac{1-\omega_r}{\omega_r}(M_{h,x_3}+D_h) - (M_{h,x_1}+M_{h,x_2})\right)e^j. \quad (7.10)$$

By using Fourier decomposition of the error function in the form of

$$e^j = E_\theta^j \exp\left\{\hat{\jmath}\,(\boldsymbol{\theta}\cdot\boldsymbol{x})\right\}, \quad \boldsymbol{\theta} = (\theta_{x_1},\theta_{x_2},\theta_{x_3}), \quad \boldsymbol{x} = \frac{1}{h}(x_1,x_2,x_3), \qquad (7.11)$$

the new error after an iteration can be related to the previous error by

$$\frac{1}{\omega_r}E_{\boldsymbol{\theta}}^{j+1}\left(\frac{6}{h^2}-\frac{2}{h^2}\cos\theta_{x_3}-(1-0.5\hat{\jmath})k^2\right) =$$
$$E_{\boldsymbol{\theta}}^{j}\left(\frac{1-\omega_r}{\omega_r}\left(\frac{6}{h^2}-\frac{2}{h^2}\cos\theta_{x_3}-(1-0.5\hat{\jmath})k^2\right)+\frac{2}{h^2}\left(\cos\theta_{x_1}+\cos\theta_{x_2}\right)\right).$$

The convergence factor of the $\boldsymbol{\theta}$ component, $\mu(\boldsymbol{\theta})$, is determined by

$$\begin{aligned}
\mu(\boldsymbol{\theta}) \quad &\equiv \quad \left|\frac{E_{\boldsymbol{\theta}}^{j+1}}{E_{\boldsymbol{\theta}}^{j}}\right| \\
&= \quad \left|\frac{(1-\omega_r)\left(6-2\cos\theta_{x_3}-(1-0.5\hat{\jmath})k^2h^2\right)+2\omega_r\left(\cos\theta_{x_1}+\cos\theta_{x_2}\right)}{6-2\cos\theta_{x_3}-(1-0.5\hat{\jmath})k^2h^2}\right| \\
&= \quad \left|\frac{(1-\omega_r)(2-2\cos\theta_{x_3})+(1-\omega_r)(4-(1-0.5\hat{\jmath})k^2h^2)+2\cos\theta_{x_1}+2\cos\theta_{x_2}}{2-2\cos\theta_{x_3}+4-(1-0.5\hat{\jmath})k^2h^2}\right| \\
&\leq \quad \max\left(1-\omega_r, \left|\frac{(1-\omega_r)(4-(1-0.5\hat{\jmath})k^2h^2)+2\cos\theta_{x_1}+2\cos\theta_{x_2}}{4-(1-0.5\hat{\jmath})k^2h^2}\right|\right).
\end{aligned}$$

The second term in the above expression is the convergence factor of the 2D point Jacobi with underrelaxation ω_r, denoted by $\mu(\theta_{x_1},\theta_{x_2})$. Since $1-\omega_r \leq 1$ for $0 < \omega_r \leq 1$ this second term plays an important role in determining a converging 3D multigrid solver. If $\mu(\theta_{x_1},\theta_{x_2}) \leq 1$, a robust 3D iterative solver can be designed by comprising of the damped x_3-line Jacobi and the 2D damped point Jacobi with (x_1,x_2) coarsening.

In Chapter 6 we gave an analysis which showed that a point Jacobi smoother with relaxation $\omega_r = 0.5$ results in a robust F(1,1)-multigrid method for the 2D version of the preconditioner (7.2). We have used the following multigrid components in the 2D multigrid method for the shifted Laplace preconditioner:

- Restriction, $I_h^H : \mathcal{G}_h \to \mathcal{G}_H$, is full weighting,
- Prolongation, $I_H^h : \mathcal{G}_H \to \mathcal{G}_h$, is bilinear interpolation or de Zeeuw's operator-dependent interpolation [109], and
- Galerkin coarse grid discretization $A_H = I_h^H A_h I_H^h$.

In the 3D multigrid method with semicoarsening, the same multigrid components as in 2D are used. However, a slight adaptation is needed before one realizes the same 2D interpolation and restriction operators in 3D semicoarsening.

Prolongation operator. To determine the 3D interpolation weights, we consider the 27-point stencil matrix (see Figure 7.2), written as follows:

$$
\begin{aligned}
&(Mu)_{i_1,i_2,i_3} \\
&= \sum_{iz=-1,0,1} (m(iz)^1_{i_1,i_2,i_3} u_{i_1-1,i_2-1,i_3+iz} \quad +m(iz)^2_{i_1,i_2,i_3} u_{i_1,i_2-1,i_3+iz} \\
&\qquad +m(iz)^3_{i_1,i_2,i_3} u_{i_1+1,i_2-1,i_3+iz} \quad +m(iz)^4_{i_1,i_2,i_3} u_{i_1-1,i_2,i_3+iz} \\
&\qquad +m(iz)^5_{i_1,i_2,i_3} u_{i_1,i_2,i_3+iz} \quad +m(iz)^6_{i_1,i_2,i_3} u_{i_1+1,i_2,i_3+iz} \\
&\qquad +m(iz)^7_{i_1,i_2,i_3} u_{i_1-1,i_2+1,i_3+iz} \quad +m(iz)^8_{i_1,i_2,i_3} u_{i_1,i_2+1,i_3+iz} \\
&\qquad +m(iz)^9_{i_1,i_2,i_3} u_{i_1+1,i_2+1,i_3+iz}).
\end{aligned}
\tag{7.12}
$$

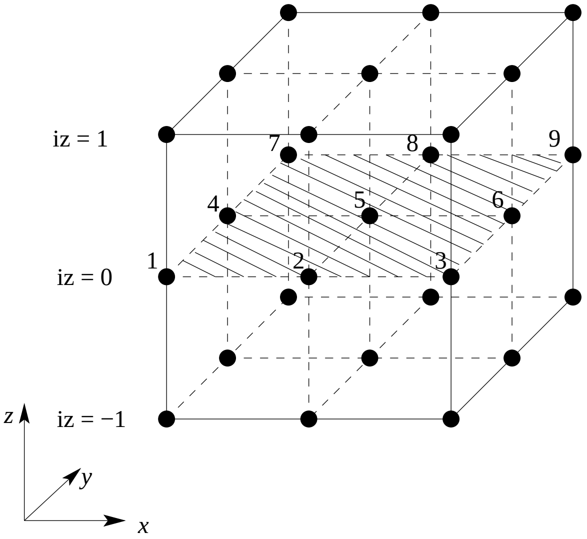

Figure 7.2: The 27-point stencil

In case the coarsening is only done in (x_1, x_2) direction, a lumped 9-point stencil matrix $\widetilde{M}_h$ in an (x_1, x_2)-plane is defined as:

$$\begin{aligned}(\widetilde{M}\phi)_{i_1,i_2,i_3} &= \widetilde{m}^1_{i_1,i_2,i_3} u_{i_1-1,i_2-1,i_3} + \widetilde{m}^2_{i_1,i_2,i_3} u_{i_1,i_2-1,i_3} + \widetilde{m}^3_{i_1,i_2,i_3} u_{i_1+1,i_2-1,i_3} \\ &+ \widetilde{m}^4_{i_1,i_2,i_3} u_{i_1-1,i_2,i_3} + \widetilde{m}^5_{i_1,i_2,i_3} u_{i_1,i_2,i_3} + \widetilde{m}^6_{i_1,i_2,i_3} u_{i_1+1,i_2,i_3} \\ &+ \widetilde{m}^7_{i_1,i_2,i_3} u_{i_1-1,i_2+1,i_3} + \widetilde{m}^8_{i_1,i_2,i_3} u_{i_1,i_2+1,i_3} + \widetilde{m}^9_{i_1,i_2,i_3} u_{i_1+1,i_2+1,i_3},\end{aligned}$$

with

$$\widetilde{m}^p_{i_1,i_2,i_3} = m(-1)^p_{i_1,i_2,i_3} + m(0)^p_{i_1,i_2,i_3} + m(1)^p_{i_1,i_2,i_3}, \quad p = 1, 2, \ldots, 9.$$

Based on the lumped 9-point stencil the coarse-to-fine grid interpolation can be determined in the same way as described in Chapter 5 for bilinear interpolation (BI) or in [109] for operator-dependent interpolation (MD).

Restriction operator. For the restriction operator we use the full weighting (5.22) instead of the adjoint of the interpolation operator. In the (x_1, x_2) semicoarsening setting the implementation of the 2D full weighting operator is straightforward.

7.3 Numerical results

In this section we present numerical results from several 3D model problems, as depicted in Figure 7.3 and 7.4. Since the numerical experiments are performed on a single processor machine, the problem size that can be resolved is limited. A parallel implementation in 3D becomes inevitable. The examples here, however, are representative to show that the shifted Laplace preconditioner is robust and effective to accelerate the Krylov subspace methods applied to the Helmholtz equation in general dimensions with regular heterogeneities. There is no reason to believe that the solver will not perform well for irregular heterogeneities.

7.3.1 3D constant wavenumber

We first consider a problem at constant wavenumber and $\alpha = 0$ (undamped) in $\Omega = (0, 1)^3$. At the boundaries $\Gamma = \partial\Omega$ the first-order radiation conditions are prescribed, that are discretized by a one-sided finite difference scheme. A unit source is situated at $\boldsymbol{x} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Starting with a zero initial guess, Bi-CGSTAB runs until the residual is reduced by 7 orders of magnitude, as in (3.25).

The numerical performance is shown in Table 7.1 for various wavenumbers k, obtained on grids of resolution $kh = 0.625$ ($\sim$ 10 grid points per wavelength). Compared to Model Problem 1 in 2D, the 3D convergence results show a similar performance as their 2D counterparts, or even faster convergence is observed. In the constant wavenumber case, the use of bilinear or operator-dependent interpolation in multigrid does not lead to a different computational performance.

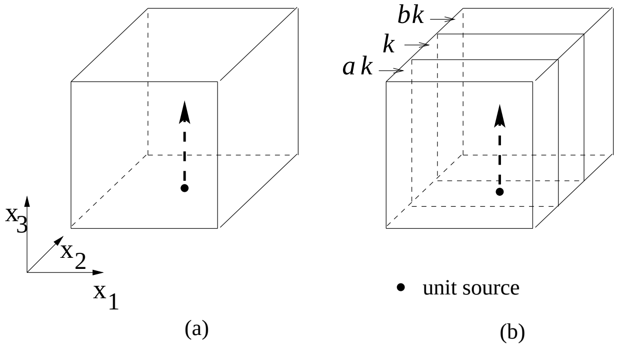

Figure 7.3: Three-dimensional problems: (a) constant k (b) three layers, k is varied in the x_2-direction.

The effect of different grid resolutions on the numerical performance is shown in Table 7.2 for $k = 10, 20$ and 30. Except for $k = 10$, for which the number of iterations slightly increases on the finer grids, the results indicate an almost h-independent convergence of the method with multigrid.

However, with the operator-dependent interpolation an increasing number of iterations is observed for fine grids. For example, for $k = 10$ on a 96^3 mesh ($\sim$ 60 grid points per wavelength), Bi-CGSTAB does not converge in 30 iterations. In this case, we find multigrid does not lead to a converging method for the preconditioner. The h-independent convergence is, however, recovered for high wavenumbers. As we currently are not able to compute a larger problem, due to hardware restrictions, further investigation of this fact are needed in the future (especially in a parallel implementation).

Table 7.1: Performance of preconditioned Bi-CGSTAB in terms of the number of iterations and CPU time (in sec.) to reach convergence for the Helmholtz equation with constant wavenumber k, $\alpha = 0$; $kh = 0.625$

	BI		MD	
k	Iter	Time	Iter	Time
10	9	0.65	9	0.71
20	13	6.89	13	6.64
30	17	25.52	18	27.63
40	21	75.99	21	71.35
50	24	136.31	24	136.33
60	26	251.62	27	276.88

Table 7.2: Number of iterations of preconditioned Bi-CGSTAB to reach convergence for the constant wavenumber Helmholtz equation, $\alpha = 0$, solved on a different grid resolution. The "–" means that the computation is not performed because $kh > 0.625$

	BI					MD				
	Grid:					Grid:				
k	16^3	32^3	48^3	64^3	96^3	16^3	32^3	48^3	64^3	96^3
10	9	9	10	11	18	9	10	15	16	>30
20	–	13	13	12	14	–	13	13	13	19
30	–	–	17	16	17	–	–	17	17	17

7.3.2 3D three-layer problems

The second problem to be considered is a three-layer problem where the wavenumber varies in Ω according to

$$k = \begin{cases} ak_{\text{ref}} & 0 \leq x_2 < 1/3, \\ k_{\text{ref}} & 1/3 \leq x_2 < 2/3, \\ bk_{\text{ref}} & 2/3 \leq x_2 \leq 1. \end{cases} \tag{7.13}$$

The problem is sketched in Figure 7.3b. A source is located at $\boldsymbol{x} = (\frac{1}{2}, \frac{1}{2}, 0)$. With this model problem we investigate the influence of different semicoarsening directions on the convergence of the preconditioned Bi-CGSTAB in the presence of a simple heterogeneity. We first look at the case where coarsening is done in the direction of strong variation in k, i.e. (x_1, x_2) semicoarsening. We use a zero initial guess for Bi-CGSTAB and terminate the iterations if criterion (3.25) is satisfied.

Table 7.3 shows the convergence results for two pairs (a, b), that determine the physical contrast in the media. For (a, b) used in Table 7.3 we see that there is a significant effect of an increasing contrast on the numerical performance. The number of iterations increases linearly with respect to k_{ref}. With respect to the interpolation operator in multigrid, the use of the operator-dependent interpolation is found to be somewhat more effective than bilinear interpolation. For small wavenumbers the bilinear interpolation outperforms the operator-dependent interpolation.

In order to see the effect of different coarsening directions on the convergence, we also perform (x_1, x_3) semicoarsening. This means that the direction with variation in k (i.e. x_2) is not coarsened. The convergence results are shown in Table 7.4. Compared to Table 7.3 the results from the two semicoarsening strategies are almost identical, which implies that the semicoarsening directions can be chosen independent from the direction of the contrast. Similar convergence results as the (x_1, x_2)-semicoarsening case are also obtained for different choices of interpolation operator.

Table 7.3: Bi-CGSTAB iteration to reach convergence for three layers problems with (x_1, x_2) semicoarsening, $k_{\text{ref}}h = 0.625$

k_{ref}	$(a,b) = (1.2, 1.5)$				$(a,b) = (1.2, 2.0)$			
	Iter		Time(s)		Iter		Time(s)	
	BI	MD	BI	MD	BI	MD	BI	MD
10	9	12	0.71	0.81	14	16	0.98	0.99
20	18	19	8.47	9.02	24	29	12.07	12.89
30	30	29	42.50	41.91	36	43	57.07	53.33
40	36	33	114.44	107.68	49	56	173.50	172.07
50	49	40	261.18	218.40	65	68	382.73	381.49
60	51	48	470.62	449.48	78	75	736.10	713.43

Table 7.4: Bi-CGSTAB iteration to reach convergence for three layers problems with (x_1, x_3) semicoarsening, $k_{\text{ref}}h = 0.625$

$(a,b) =$	$(1.2, 1.5)$		$(1.5, 2.0)$	
k_{ref}	BI	MD	BI	MD
10	11	12	13	14
20	18	18	23	25
30	30	28	36	35
40	36	34	50	49
50	49	40	65	62
60	51	49	79	68

The numerical tests related to Tables 7.3 and 7.4 are performed on grids whose resolution is based on the criterion $k_{\text{ref}}h \leq 0.625$, which is related to the minimal number of grid points per wavelength for the 7-point stencil. On the top and bottom layers, however, the grid resolution falls below this criterion. The safest is to set a grid resolution based on the largest wavenumber in the medium. For example, in case $(a, b) = (1.2, 1.5)$ for $k = 20$, $k_{\max} = 30$. Hence, instead of using $h^{-1} = 32$ we require that $h^{-1} = 48$. Table 7.5 displays results on this finer grids for $k_{\text{ref}} = 10$ and 20. In this case we only use the operator-dependent interpolation. Here we observe a convergence dependence on h only for $k_{\text{ref}} = 10$. This is because the preconditioner is not accurately inverted by multigrid. This dependence, however, becomes much less significant as the wavenumber increases. The convergence there is asymptotically independent of h.

Table 7.5: Number of Bi-CGSTAB iterations to reach convergence for a three layers problem with (x_1, x_2) semicoarsening. The "–" means that the computation is not performed because $k_{\text{ref}}h > 0.625$

	$(a,b) = (1.2, 1.5)$					
k_{ref}	16^3	32^3	48^3	64^3	80^3	96^3
10	12	12	11	14	19	35
20	–	19	16	23	14	16
30	–	–	29	23	20	20

7.3.3 The 3D wedge problem

The final 3D example is the so-called 3D wedge problem as depicted in Figure 7.4. A unit source is located at $\boldsymbol{x} = (\frac{1}{2}, \frac{1}{2}, 0)$. On the boundaries the first-order radiation conditions (2.26) are imposed. For multigrid iteration, the bilinear (BI) and operator-dependent interpolation (MD) are used and compared. The restriction is the full weighting (FW) operator. The convergence results are shown in Table 7.6 for various values of k_{ref}.

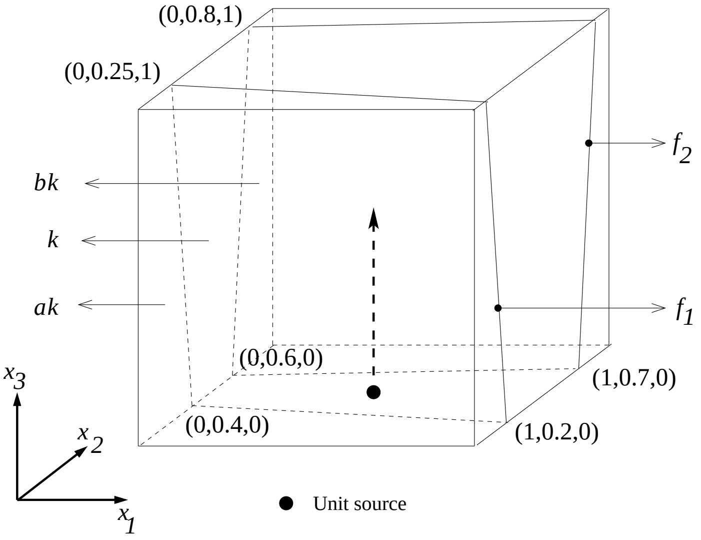

Figure 7.4: Wedge problem: $f_a(x_1, x_2, x_3) = 0.5x_1 + 2.5x_2 + 0.375x_3 - 1 = 0$, $f_b(x_1, x_2, x_3) = -\frac{1}{6}x_1 + \frac{5}{3}x_2 - \frac{1}{3}x_3 - 1 = 0$

For this problem, similar convergence results as for the previous problems are observed. For high wavenumbers the operator-dependent interpolation is superior to the bilinear interpolation.

Table 7.6: Bi-CGSTAB iteration to reach convergence for a three-dimensional wedge problem with (x_1, x_2) semicoarsening, $k_{\text{ref}}h = 0.625$

	$(a, b) = (1.2, 1.5)$				$(a, b) = (1.2, 2.0)$			
	Iter		Time(s)		Iter		Time(s)	
k_{ref}	BI	MD	BI	MD	BI	MD	BI	MD
10	11	12	0.80	0.86	14	14	0.96	0.97
20	17	18	8.83	9.06	27	28	13.09	13.76
30	25	23	40.03	37.89	37	36	56.68	55.94
40	31	29	111.27	106.27	53	50	181.05	175.53
50	40	38	258.67	249.64	66	62	413.04	392.29
60	47	43	510.92	474.95	79	76	831.94	808.33

In summary, for these 3D model problems with regular heterogeneities we find a similar, very satisfactorily numerical convergence as we saw for corresponding 2D model problems.

Chapter 8

Conclusion

8.1 Overview

Due to resolution requirements, finite difference/element discretizations of the Helmholtz equation at high wavenumbers always lead to a large, sparse linear system. This linear system is symmetric, but *highly* indefinite, non-Hermitian and extremely ill-conditioned. These properties bring complications when one needs to iteratively solve the linear system. Direct methods are only viable in 2D, while their use in 3D is simply out of reach. When one considers iterative methods for solving indefinite, non-Hermitian and ill-conditioned linear systems, one is often faced with the slow convergence of the methods, or even divergence. Direct applications of standard, existing methods, like preconditioned Krylov subspace or multigrid methods, do not result in satisfactory convergence. For preconditioning, the well-known incomplete LU factorization is not effective.

In this thesis we have proposed and discussed an iterative method to solve the discrete Helmholtz equation in 2D and 3D at very high wavenumbers. The method is robust and efficient, and can solve problems with strong heterogeneities arising, e.g., from geophysical applications. The method can be viewed as a preconditioned Krylov subspace iteration, with multigrid serving as the preconditioner. Multigrid, however, is not applied to the discrete Helmholtz equation at hand. Rather, multigrid is applied to a properly chosen preconditioning operator, which is called in this thesis the *shifted Laplace* operator. This operator can be considered as the Laplace operator with a “complex perturbation” of the zeroth order term. Thus, to solve the Helmholtz equation, multigrid applied to the shifted Laplace operator has to be used in the Krylov subspace iteration context.

We have theoretically analyzed the preconditioner in continuous and discrete case, and provided the “optimal” preconditioning operator for convergence acceleration. Some convergence analysis has also been given. The theoretical results have been confirmed by numerical results.

We have used a finite difference discretization and the numerical results are obtained on regular, finite difference meshes. Therefore, geometric multigrid is a natural choice in this case. As multigrid is the key element in our method, satisfactory multigrid convergence for the preconditioner is important. It is not required that multigrid applied to the preconditioner converges at the same rate as the multigrid convergence for the Poisson equation, for instance. As the error due to multigrid iterations is corrected by the Krylov subspace method, what is important here is that the multigrid iterations must lead to a convergence acceleration of the Krylov subspace methods. Furthermore, the 5-point and 9-point stencils have been used to discretize the Helmholtz equation. For both cases, the preconditioner is, however, obtained from discretization based on the 5-point stencil. The convergence for both cases is very similar.

To analyze the performance of multigrid we have used intensively Rigorous Fourier Analysis (RFA). With RFA, we are not only able to quantify the multigrid performance for two and three grids, but also to study the spectral properties of the preconditioned linear system. These two benefits of using RFA make it possible to make a good choice of multigrid components for a Krylov subspace acceleration. For example, from RFA we found that W(1,1)-multigrid with point Jacobi smoother with underrelaxation results in a good multigrid-based preconditioner for Bi-CGSTAB. As one can not distinguish the significance between bilinear and operator-dependent interpolation in constant wavenumber Helmholtz problems, the use of operator-dependent interpolation in cases with strong heterogeneities becomes necessarily important. We also found that Galerkin coarse grid discretization is better done based on a combination of an interpolation operator and an *explicit* restriction operator, and *not* of an interpolation operator and its adjoint. In this case, for general Helmholtz problems, the operator-dependent interpolation of de Zeeuw and the full weighting operator lead to the better combination among other possibilities. The results of RFA were numerically confirmed, with the W-cycle was replaced by the F-cycle to reduce the amount of work.

Extension to 3D is straightforward. In this thesis, we have used a 3D multigrid method with semicoarsening. This is in general a robust method for problems with anisotropies. One advantage of using multigrid with semicoarsening is that if one applies coarsening only in two directions, a robust 2D multigrid method with point smoother can be used in the two directions, while in the third direction a line smoother is applied. Therefore, the behavior of the method can be foreseen immediately from the 2D multigrid methods.

Numerical experiments in 2D and 3D have been performed and were presented in this thesis based on problems of varying difficulties. The numerical results showed the robustness and efficiency of the method. The convergence is independent of gridsize for constant wavenumber problems, and *asymptotically* independent of gridsize for problems with heterogeneities. The convergence depends *linearly* on the wavenumber, with only a small linear constant. In cases with damping (i.e. in an attenuative medium), the convergence is wavenumber-independent.

The 2D Marmousi problem used in this thesis is considered as a difficult problem to solve. Beside our method, the only method known so far used to iteratively solve this problem is Bi-CGSTAB with the separation-of-variables preconditioner [81]. This method, however, does not converge after 2000 iterations at $f = 30$ Hz. To our knowledge, our method is the first iterative method which is able to reach convergence for the Marmousi problem with satisfactory performance (for $f = 30$ Hz, 136 iterations).

8.2 Remark

Deflation

From the spectral analysis of the preconditioned system we observed that many eigenvalues lie very close to zero in the complex plane. This bunch of small eigenvalues is responsible for slow convergence in the first few iterations of Krylov subspace iterations, before the fast convergence is obtained. This is called superlinear convergence, and observed in our numerical tests.

In [76, 24, 43] so-called *deflation* is used to project out the smallest eigenvalues from the original spectrum. As their components no longer contribute to the residual (or error), the convergence rate improves. In deflation a deflation subspace must be constructed, which is usually based on eigenvectors corresponding to the removed eigenvalues. For a small number of small eigenvalues, say ~ 5, corresponding to a small matrix A, the projection matrix can be easily determined. A problem arises when a bunch of too many small eigenvalues has to be projected out from the spectrum. Computing the corresponding eigenvectors is then too expensive.

As advocated in [43, 103] the projection matrix can be approximately constructed based on spectral properties of the problem. Thus, one needs not define the projection matrix *exactly*. For our case, if a similar approach can also be realized, a faster convergence of the method can be expected.

Finite element implementation

Many applications of the Helmholtz equation involve complicated geometries. For such geometries a finite element discretization is a natural choice. In this thesis we did not treat finite element methods. The method, however, is easily extended to finite elements. For finite element matrices, *algebraic* multigrid is then the natural choice.

Parallel implementation

In geophysical applications, one usually encounters huge problems that lead to huge matrices. In 3D, for instance, the number of unknowns to be resolved is of order of billions, which is too huge to be fitted in one computer. For a realistic 3D problem, and to some extent for 2D, parallel implementation of the method becomes inevitable.

The method proposed in this thesis is fully parallelizable. Every step in the Bi-CGSTAB algorithm is parallelizable: the matrix-vector multiplications, the inner products and preconditioner solves. For the latter we have chosen multigrid components such that the multigrid algorithm can be fully implemented in a parallel machine. The Jacobi smoother, for example, not only turns out to be an efficient smoother for the shifted Laplace operator, but it is also fully parallelizable.

Currently, a parallel implementation is carried out and preliminary results from 2D tests indicate a good scalability of the method.

Seismic application: migration

As stated in Chapter 1, the target application of the method is in 2D/3D frequency-domain migration in seismics. This is done by resolving the Helmholtz equation for a range of frequencies and recording the solution of each frequency on some locations of the receiver on the earth's surface. Once all of these solutions are obtained, time domain solution can be recovered by Fast-Fourier transforming the frequency-domain solutions.

We currently carry out a migration computation for a synthetic model based on the full Marmousi problem, and the results will be part of our publication in [36].

Bibliography

[1] S. Abarbanel and D. Gottlieb, *A mathematical analysis of the PML method*, J. Comput. Phys., 134 (1997), pp. 357–363.

[2] ——, *On the construction and analysis of absorbing layers in CEM*, Appl. Numer. Math., 27 (1998), pp. 331–340.

[3] R. E. Alcouffe, A. Brandt, J. E. Dendy Jr, and J. W. Painter, *The multi-grid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Comput., 2 (1981), pp. 430–454.

[4] A. Arnold and M. Ehrhardt, *Discrete transparent boundary conditions for wide angle parabolic equations in underwater acoustics*, J. Comput. Phys., 145 (1998), pp. 611–638.

[5] O. Axelsson, *A survey of preconditioned iterative methods for linear systems of algebraic equations*, BIT, 25 (1985), pp. 166–187.

[6] O. Axelsson and G. Lindskog, *On the rate of convergence of the preconditioned conjugate gradient method*, Numer. Math., 48 (1986), pp. 499–523.

[7] I. Babuska, F. Ihlenburg, T. Strouboulis, and S. K. Gangaraj, *Posteriori error estimation for finite element solutions of Helmholtz's equation. Part I: The quality of local indicators and estimators*, Int. J. Numer. Methods. Engrg., 40 (1997), pp. 3443–3462.

[8] ——, *Posteriori error estimation for finite element solutions of Helmholtz's equation. Part II: Estimation of the pollution error*, Int. J. Numer. Methods. Engrg., 40 (1997), pp. 3883–3900.

[9] I. Babuska and S. Sauter, *Is the pollution effect of the FEM avoidable for the Helmholtz Equation considering high wave numbers?*, SIAM J. Numer. Anal., 27 (1997), pp. 323–352.

[10] A. Bamberger, B. Enquist, L. Halpern, and P. Joly, *Parabolic wave equation approximation in heterogeneous media*, SIAM J. Appl. Math., 48 (1988), pp. 99–128.

[11] A. Bamberger, P. Joly, and J. E. Roberts, *Second-order absorbing boundary conditions for the wave equation: A solution for the corner problem*, SIAM J. Numer. Anal., 27 (1990), pp. 323–352.

[12] A. Bayliss, C. I. Goldstein, and E. Turkel, *An iterative method for Helmholtz equation*, J. Comput. Phys., 49 (1983), pp. 443–457.

[13] ——, *On accuracy conditions for the numerical computation of waves*, J. Comput. Phys., 59 (1985), pp. 396–404.

[14] J. P. Berenger, *A perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200.

[15] ——, *Three-dimensional perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys., 127 (1996), pp. 363–379.

[16] A. J. Berkhout, *Seismic Migration: Imaging of Acoustic Energy by Wave Field Extrapolation*, Elsevier, Amsterdam, 1982.

[17] B. Biondi and G. Palacharla, *3-d prestack migration of common-azimuth data*, Geophysics, 61 (1996), pp. 1822–1832.

[18] A. Bourgeois, M. Bourget, P. Lailly, M. Poulet, P. Ricarte, and R. Versteeg, *Marmousi, model and data*, in Marmousi Experience, 1991, pp. 5–16.

[19] K. Brackenridge, *Multigrid and cyclic reduction applied to the Helmholtz equation*, in Proc. 6th Copper Mountain Conf. on Multigrid Methods, N. D. Melson, T. A. Manteuffel, and S. F. McCormick, eds., 1993, pp. 31–41.

[20] A. Brandt, *Multi–level adaptive solutions to boundary–value problems*, Math. Comp., 31 (1977), pp. 333–390.

[21] ——, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Tech. Rep. GMD-Studie 85, GMD Sankt Augustine, Germany, 2002.

[22] A. Brandt and I. Livshits, *Wave-ray multigrid methods for standing wave equations*, Electr. Trans. Num. Anal., 6 (1997), pp. 162–181.

[23] A. Brandt and S. Ta'asan, *Multigrid method for nearly singular and slightly indefinite problems*, in Proc. EMG'85 Cologne, 1986, pp. 99–121.

[24] A. Chapman and Y. Saad, *Deflated and augmented krylov subspace techniques*, Numer. Linear Algebra Appl., 4(1) (1997), pp. 43–66.

[25] J. F. Claerbout, *Imaging the Earth's Interior*, Blackwell Scientific Publications, 1985.

[26] R. Clayton and B. Engquist, *Absorbing boundary conditions for acoustic and elastic wave equations*, Bull. Seis. Soc. America, 67(6) (1977), pp. 1529–1540.

[27] ———, *Absorbing boundary conditions for wave-equation migration*, Geophysics, 45(5) (1980), pp. 895–904.

[28] F. Collino and P. Joly, *Splitting of operators, alternate directions, and paraxial approximations for the three-dimensional wave equation*, SIAM J. Sci. Comput., 16 (1995), pp. 1019–1048.

[29] D. Colton and R. Kress, *Inverse matrix and electromagnetic scattering theory*, Springer Verlag, Berlin-Heidelberg, 1998.

[30] J. Dendy, Jr., *Blackbox multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–283.

[31] A. Deraemaeker, I. Babuska, and P. Bouillard, *Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two, and three Dimensions*, Int. J. Numer. Methods Engrg., 46 (1999), pp. 471–499.

[32] J. Douglas, J. Hensley, and J. E. Roberts, *An alternating-direction iteration method for Helmholtz problems*, Appl. Math., 38 (1993), pp. 289–300.

[33] H. R. 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., 31 (2001), pp. 1291–1315.

[34] B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comput., 31 (1977), pp. 629–651.

[35] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik, *A novel multigrid-based preconditioner for the heterogeneous helmholtz equation*, SIAM J. Sci. Comput., (to appear).

[36] Y. A. Erlangga, R. E. Plessix, W. A. Mulder, C. vuik, and C. W. Oosterlee, *A new iterative solver for the time-harmonic wave equation applied to seismic problems*, Geophysics, submitted (2005).

[37] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, *On a class of preconditioners for solving the the discrete helmholtz equation*, in Mathematical and Numerical Aspects of Wave Propagation, G. Cohen, E. Heikkola, P. Joly, and P. Neittaanm akki, eds., Univ. Jyväskylä, Finland, 2003, pp. 788–793.

[38] ———, *On a class of preconditioners for solving the Helmholtz equation*, Appl. Numer. Math., 50 (2004), pp. 409–425.

[39] ———, *Krylov subspace iterative methods accelerated with multigrid for the 3d heterogeneous helmholtz equation*, in Mathematical and Numerical Aspects of Wave Propagation, Brown University, Rhode Island, 2005, pp. 418–419.

[40] ——, *Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation*, Appl. Numer. Math., (in press).

[41] G. J. Fix and S. P. Marin, *Variational methods for underwater acoustic problems*, J. Comput. Phys., 28 (1978), pp. 253–270.

[42] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Proc. the 1974 Dundee Biennial Conf. on Numerical Analysis, G. A. Watson, ed., 1975, pp. 73–89.

[43] J. Frank and C. Vuik, *On the construction of the deflated-based preconditioner*, SIAM J. Sci. Comput., 23(2) (2001), pp. 442–462.

[44] R. W. Freund, *Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices*, SIAM J. Sci. Stat. Comput., 13(1) (1992), pp. 425–448.

[45] ——, *Preconditioning of symmetric but highly indefinite linear systems*, in 15th IMACS World Congress on Scientific Computation Modelling and Applied Mathematics, Vol. 2 Numerical Mathematics, A. Sydow, ed., 1997, pp. 551–556.

[46] R. W. Freund and N. M. Nachtigal, *QMR: A quasi minimum residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[47] M. J. Gander and F. Nataf, *AILU for Helmholtz problems: A new preconditioner based on the analytic parabolic factorization*, J. Comput. Acoustics, 9 (2001), pp. 1499–1509.

[48] A. George and J. W. Liu, *Computer solution of large sparse positive definite systems*, Prentice-Hall Inc., NJ, 1981.

[49] D. N. Ghosh-Roy and L. S. Couchman, *Inverse problems and inverse scattering of plane waves*, Academic Press, London, 2002.

[50] D. Givoli and I. Patlashenko, *Optimal local non-reflecting boundary conditions*, Appl. Numer. Math., 27 (1998), pp. 367–384.

[51] J. Gozani, A. Nachshon, and E. Turkel, *Conjugate gradient coupled with multigrid for an indefinite problem*, in Advances in Comput. Methods for PDEs V, 1984, pp. 425–427.

[52] M. H. Gutknecht, *Variants of BICGSTAB for matrices with complex spectrum*, SIAM J. Sci. Comput., 14(5) (1993), pp. 1022–1033.

[53] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 2003.

[54] I. Harari and E. Turkel, *Accurate finite difference methods for time-harmonic wave propagation*, J. Comput. Phys., 119 (1995), pp. 252–270.

[55] E. Heikkola, T. Rossi, and J. Toivanen, *A parallel fictitious domain decomposition method for the three-dimensional Helmholtz equation*, tech. rep., Dept. Math. Info. Tech., Univ. Jÿvaskÿla, No. B 9/2000, 2000.

[56] R. L. Higdon, *Numerical absorbing boundary conditions for the wave equation*, Math. Comp., 49(179) (1987), pp. 65–90.

[57] F. Ihlenburg and I. Babuska, *Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation*, Int. J. Numer. Methods Engrg., 38 (1995), pp. 3745–3774.

[58] ———, *Finite element solution of the Helmholtz equation with high wave number. Part I: The h-version of the FEM*, Comput. Math. Appl., 30(9) (1995), pp. 9–37.

[59] ———, *Finite element solution of the Helmholtz equation with high wave number. Part II: The hp-version of the FEM*, SIAM J. Numer. Anal., 34 (1997), pp. 315–358.

[60] S. W. Jin and G. Peng, *Prestack depth migration using a hybrid pseudo-screen propagator*, in 68th SEG Annual Mtg., 1998, pp. 1819–1822.

[61] C.-H. Jo, C. Shin, and J. H. Suh, *An optimal 9-point, finite difference, frequency space, 2-D scalar wave extrapolator*, Geophysics, 61(2) (1996), pp. 529–537.

[62] R. Kettler, *Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods*, in Multigrid methods. Lecture Notes in Mathematics 960, W. Hackbusch and U. Trottenberg, eds., 1982, pp. 502–534.

[63] S. Kim and S. Kim, *Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media*, SIAM J. Sci. Comput., 24 (2002), pp. 684–701.

[64] A. L. Laird and M. B. Giles, *Preconditioned iterative solution of the 2D Helmholtz equation*, Tech. Rep. NA 02-12, Comp. Lab., Oxford Univ., 2002.

[65] B. Lee, T. A. Manteuffel, S. F. McCormick, and J. Ruge, *First-order system least-squares for the Helmholtz equation*, SIAM J. Sci. Comput., 21 (2000), pp. 1927–1949.

[66] S. K. Lele, *Compact finite difference schemes with spectral-like resolution*, J. Comput. Phys., 103(1) (1992), pp. 16–42.

[67] E. L. Lindman, *Free space boundary conditions for time dependent wave equation*, J. Comput. Phys., 18 (1975), pp. 66–78.

[68] R. E. Lynch and J. R. Rice, *A high-order difference method for differential equations*, Math. Comp., 34(150) (1980), pp. 333–372.

[69] M. M. M. Made, *Incomplete factorization-based preconditionings for solving the Helmholtz equation*, Int. J. Numer. Meth. Engng., 50 (2001), pp. 1077–1101.

[70] A. Majda and S. Osher, *Reflection of singularities at the boundary*, Comm. Pure Appl. Math., 28 (1975), pp. 277–298.

[71] T. A. Manteuffel and S. V. Parter, *Preconditioning and boundary conditions*, SIAM J. Numer. Anal., 27(3) (1990), pp. 656–694.

[72] K. J. Marfurt, *Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations*, Geophysics, 49 (1984), pp. 533–549.

[73] W. A. Mulder, *A new multigrid approach to convection problems*, J. Comput. Phys., 83 (1989), pp. 303–323.

[74] W. A. Mulder and R. E. Plessix, *How to choose a subset of frequencies in frequency-domain finite-difference migration*, Geophys. J. Int., 158 (2004), pp. 801–812.

[75] ———, *One-way and two-way wave-equation migration*, Geophys. J. Int., 69 (2004), pp. 1491–1504.

[76] R. A. Nicolaides, *Deflation of conjugate gradients with applications to boundary value problems*, SIAM J. Numer. Anal., 24(2) (1987), pp. 355–365.

[77] C. W. Oosterlee, *The convergence of parallel multiblock multigrid methods*, Appl. Numer. Math., 19 (1995), pp. 115–128.

[78] C. W. Oosterlee and P. Wesseling, *On the robustness of multiple a semi-coarsened grid method*, Z. Angew. Math. Mech., 75 (1995), pp. 251–257.

[79] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12(4) (1975), pp. 617–629.

[80] R. E. Plessix and W. A. Mulder, *Frequency-domain finite-difference amplitude-preserving migration*, Geophys. J. Int., 157 (2004), pp. 975–987.

[81] ———, *Separation-of-variables as a preconditioner for an iterative helmholtz solver*, Appl. Numer. Math., 44 (2004), pp. 385–400.

[82] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126.

[83] ——, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.

[84] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7(12) (1986), pp. 856–869.

[85] I. Singer and E. Turkel, *High-order finite difference methods for the Helmholtz equation*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 343–358.

[86] G. L. G. Sleijpen and D. R. Fokkema, *BICGSTAB(L) for linear equations involving unsymmetric matrices with complex spectrum*, Electr. Trans. Numer. Anal., 1 (1993), pp. 11–32.

[87] P. Sonneveld, *CGS: A fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 36–52.

[88] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, Pacific Groove, CA, 1989.

[89] K. Stüben and U. Trottenberg, *Multigrid methods: fundamental algorithms, model problem analysis and applications*, in Lecture Notes in Math. 960, W. Hackbusch and U. Trottenberg, eds., 1982, pp. 1–176.

[90] C. K. W. Tam and J. C. Webb, *Dispersion-relation-preserving finite difference schemes for computational acoustics*, J. Comput. Phys., 107(2) (1993), pp. 262–281.

[91] C. A. Thole and U. Trottenberg, *Basic smoothing procedures for the multigrid treatment of elliptic 3D operators*, Appl. Math. Comput., 19 (1986), pp. 333–345.

[92] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, New York, 2001.

[93] S. Tsynkov and E. Turkel, *A cartesian perfectly matched layer for the Helmholtz equation*, in Absrobing Boundaries and Layers, Domain Decomposition Methods Applications to Large Scale Computation, L. Tourette and L. Harpern, eds., Springer, Berlin, 2001, pp. 279–309.

[94] E. Turkel, *Numerical difficulties solving time harmonic equations*, in Multiscale Computational Methods in Chemistry and Physics, A. Brandt, ed., IOS Press, Ohmsha, 2001, pp. 319–337.

[95] E. Turkel and A. Yefet, *Absorbing PML boundary layers for wave-like equations*, Appl. Numer. Math., 27 (1998), pp. 533–557.

[96] H. P. Urbach and R. T. M. Merkx, *Finite element simulation of electromagnetic plane wave diffraction at gratings for arbitrary angles of incidence*, in Mathematical and Numerical Aspects of Wave Propagation Phenomena, G. Cohen, L. Halpern, and P. Joly, eds., 1991, pp. –.

[97] H. A. van der Vorst, *Bi-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13(2) (1992), pp. 631–644.

[98] H. A. van der Vorst and J. B. M. Melissen, *A Petrov-Galerkin type method for solving $Ax = b$, where A is symmetric complex systems*, IEEE Trans. Magnetics, 26(2) (1990), pp. 706–708.

[99] H. A. van der Vorst and C. Vuik, *GMRESR: A family for nested GMRES methods*, Numer. Linear Algebra Appl., 1(4) (1994), pp. 369–386.

[100] P. Vanek, J. Mandel, and M. Brezina, *Two-level algebraic multigrid for the Helmholtz problem*, Contemp. Math., 218 (1998), pp. 349–356.

[101] R. S. Varga, *Matrix iterative analysis*, Prentice-Hall, Inc., New Jersey, 1962.

[102] C. Vuik, Y. A. Erlangga, and C. W. Oosterlee, *Shifted Laplace preconditioner for the Helmholtz equations*, Tech. Rep. Report 03-18, Dept. Appl. Math. Anal., Delft Univ. Tech., The Netherlands, 2003.

[103] C. Vuik, A. Segal, L. el Yaakoubi, and E. Dufour, *A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients*, Appl. Numer. Math., 41 (2002), pp. 219–233.

[104] T. Washio and C. W. Oosterlee, *Flexible multiple semicoarsening for three-dimensional singularly perturbed problems*, SIAM J. Sci. Comput., 19(5) (1998), pp. 1646–1666.

[105] P. Wesseling, *An Introduction to Multigrid Methods*, John Willey and Sons, London, 1992.

[106] R. Wienands and C. W. Oosterlee, *On three-grid fourier analysis of multigrid*, SIAM J. Sci. Comput., 23 (2001), pp. 651–671.

[107] Y. Yoon, K. J. Marfurt, and W. Starr, *Challenges in reverse-time migration*, in SEG Expanded Abstracts 23, 2004.

[108] H. Yserentant, *Preconditioning indefinite discretization matrices*, Numer. Math., 54 (1988), pp. 719–734.

[109] P. M. d. Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.

[110] L. Zhou and H. F. Walker, *Residual smoothing techniques for iterative methods*, SIAM J. Sci. Comput., 15(2) (1994), pp. 297–312.