Iterative methods for time-harmonic waves

Towards accuracy and scalability

Dwarka, V.N.S.R.

DOI
10.4233/uuid:4d26359a-89ad-4a5a-9fac-cd8e9e59c743

Publication date
2022

Document Version
Final published version

Citation (APA)
Dwarka, V. N. S. R. (2022). *Iterative methods for time-harmonic waves: Towards accuracy and scalability*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:4d26359a-89ad-4a5a-9fac-cd8e9e59c743

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

ITERATIVE METHODS FOR TIME-HARMONIC WAVES

TOWARDS ACCURACY AND SCALABILITY

VANDANA DWARKA

ITERATIVE METHODS FOR TIME-HARMONIC WAVES

TOWARDS ACCURACY AND SCALABILITY

ITERATIVE METHODS FOR TIME-HARMONIC WAVES

TOWARDS ACCURACY AND SCALABILITY

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 4 juli 2022 om 15:00 uur

door

Vandana DWARKA

Wiskundig ingenieur, Technische Universiteit Delft, Nederland,
geboren te Amsterdam, Nederland.

Dit proefschrift is goedgekeurd door de promotoren.

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. C. Vuik,	Technische Universiteit Delft, promotor
Prof. dr. ir. M. B. van Gijzen,	Technische Universiteit Delft, promotor
Onafhankelijke leden:	
Prof. dr. V. Dolean	Universiteit van Strathclyde, Schotland
Prof. dr. Y. A. Erlangga	Zayed Universiteit, Dubai
Prof. dr. ir. C. W. Oosterlee	Universiteit Utrecht
Prof. dr. H. M. Schuttelaars	Technische Universiteit Delft
Prof. dr. ir. E. C. Slob	Technische Universiteit Delft

TU Delft | Delft Institute of Applied Mathematics

Keywords: Helmholtz, Deflation, Multigrid, Pollution Error, Isogeometric Analysis

Printed by: Ridderprint | www.ridderprint.nl

ISBN 978-94-6458-348-9

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

CONTENTS

PREFACE

With our current complex society and fast paced technological developments, the branch of numerical analysis and scientific computing has played a paramount role in providing the tools to translate (abstract) models into computer aided simulation. For example, for seismic exploration, we need to understand how waves travel through the earth's subsurface. For nodules to show up on MRI and ultrasound, we need to know how electromagnetic waves or sound travel and scatter through human tissue. For wireless communication, we require insights into wave propagation at different frequencies. Scientists working on these diverse topics have on thing in common: they all rely on accurate simulation tools.

At the heart of the previously mentioned applications lies the Helmholtz equation. While the equation in itself, which is essentially the shifted Laplace equation, appears simple and elegant, retrieving accurate and scalable numerical solutions leads to a wide array of issues. Due to the shift, which represents the wavenumber, the operator and consequently resulting discretized linear system matrix become indefinite. To ensure accurate solutions and the minimization of numerical dispersion, we are required to use very fine grids. Consequently, we end up with very large linear systems. Despite increased computer power, direct solution methods are no viable alternative and we resort to iterative solvers. As the wavenumber increases, the number of iterations to reach convergence increases as well, leading to inscalabilty of the solver.

For more than 15 years, the industry has relied on using the Complex Shifted Laplacian (CSL) as an effective preconditioner to accelerate the convergence. While this works efficiently for medium sized wavenumbers, the number of iterations are still too high for practical applications and the problem sizes become too large when we move to modern high-frequency problems and applications, such as numerical weather prediction models and plasma fusion simulations. The main culprit behind the deteriorating performance of the solver are the near-zero eigenvalues of the preconditioned matrix.

So how can we design simulation tools which remain scalable both in terms of the computational complexity and the wavenumber? Answering this question lies at the core of this dissertation.

SCOPE

The bottleneck in designing iterative solvers lies in balancing the trade-off between accuracy and scalability. To work on these issues, we formulated three research pillars which guide the progression of this dissertation. For accurate solutions, we require fine grids leading to uneconomical simulations which could take up days (accuracy). Moreover, solutions always suffer from the so called 'pollution error' due to numerical dispersion. For fast solutions, we need iterative solvers which converge in steps independent of the wavenumber

(wavenumber independent convergence). We also require that the amount of computational work depends linearly on the problem size (linear complexity). At the same time, the presence of heterogeneous media and variable frequencies further complicates the design of numerical solvers. In these instances, no analytical solution is available.

The scope of this dissertation lies in sequential implementations and we will therefore not discuss parallel scalability. Moreover, we distinguish between two research parts: one dealing with the accuracy, which discusses the first research pillar (Part II) and the second part dealing with the numerical solvers (Part III). We mostly focus on using second order finite differences schemes, and thus other discretization techniques unless states otherwise, are not within the scope of this dissertation.
Given that the study of the accuracy and pollution error requires model problems where the analytical solution is known, we will use different model problems for each chapter depending on the context. Thus, model problems using Sommerfeld boundary conditions will be discussed in the numerical solver part (Part III).
Furthermore, the two-level and multilevel solvers we develop in this dissertation will also be subjected to different model problems as, for example, the two-level method requires more memory. To make things more clear, each chapter will start with a clear definition of the model problems which will be discussed within the context of that chapter. Moreover, notation-wise, due to the broad set of methods discussed in this dissertation, we will reintroduce notations at the beginning of each chapter. One reason for this is that describing the accuracy issues, for example, allows for a more compact notation of the eigenvalues compared to when we are dealing with eigenvalues of the systems in a multilevel hierarchy.

Vandana Dwarka
Amsterdam, March 2022

SUMMARY

The bottleneck in designing iterative solvers for the Helmholtz equation lies in balancing the trade-off between accuracy and scalability. Both the accuracy of the numerical solution and the number of iterations to reach convergence deteriorate in higher dimensions and increase with the wavenumber. To address these issues in this dissertation, we formulated three research pillars: accuracy, wavenumber independent convergence and linear complexity. Below, we summarize the core findings of this dissertation:

WAVENUMBER INDEPENDENT CONVERGENCE

We develop the first preconditioning technique which leads to close to wavenumber independent convergence for very large wavenumbers in 1D, 2D and 3D. Building on a two-level deflation projection method, we incorporated Quadratic Rational Bézier curves to construct the deflation space and vectors (Chapter 7). As a result, the near-zero eigenvalues of the coarse grid operator remain aligned with the fine-grid operator, keeping the spectrum of the preconditioned system clustered, leading to superior convergence properties compared to previous methods.

LINEAR COMPLEXITY

For over 30 years, applied mathematicians have tried to make convergent (standard) multigrid solvers for the Helmholtz equation. Multigrid solvers use sequences of smaller problem sizes and are computationally cheap and easy to implement. Unfortunately, multigrid methods diverge for Helmholtz and solving this issue remained an open problem. Using standard smoothing techniques, combined with similar higher-order coarse spaces, we constructed a fully convergent V- and W-cycle algorithm (Chapter 9). The key features of the algorithm are the use of higher-order transfer operators (instead of deflation vectors in the previous application) and a complex shift in the smoothing operator. While the method converges and the preliminary results have been proven, much research can still be conducted in this area, as this could support a paradigm shift in solving the complexity issue for very large wavenumbers in 2D and 3D.
In light of this, we extended the two-level deflation solver to a multi-level deflation solver to address both the issue of wavenumber and problem size dependence (Chapter 8). In this part, we show better convergence properties and provide numerical experiments on challenging 2D and 3D test problems to corroborate the theoretical results.

ACCURACY

Finally, we developed an unprecedented way to study the accuracy of the numerical solutions by studying the eigenvalues of systems where the analytical solution is known (Chapter 5). Expressing the pollution error in terms of these eigenmodes, enabled theoretical accuracy studies and dispersion corrections in higher dimensions, irrespective of the wave propagation angles. Something which was previously impossible. We also studied the application of Isogeometric Analysis (IgA) to improve the accuracy and reduce the pollution

error (Chapter 6). Our results showed that the use of IgA was able to significantly suppress the pollution error compared to Finite Elements Discretizations of the same order.

SAMENVATTING

Het belangrijkste knelpunt in het ontwerpen van iteratieve solvers voor de Helmholtz vergelijking ligt in het vinden van een balans tussen de nauwkeurigheid van de numerieke oplossing en de schaalbaarheid van de solver. Zowel de nauwkeurigheid van de numerieke oplossing, als het totaal aantal iteraties, verslechteren in rap tempo wanneer men multidimensionele problemen wilt oplossen en naar mate het golfgetal groter wordt. Om deze problemen gericht te onderzoeken, worden in deze dissertatie drie onderzoekspijlers geaddreseerd: nauwkeurigheid, golfgetal onafhankelijke convergentie en lineaire complexiteit. Hieronder vatten we de bevindingen per pijler kort samen:

GOLFGETAL ONAFHANKELIJKE CONVERGENTIE

We hebben de eerste preconditionering techniek ontwikkeld waarbij we zo goed als golfgetal onafhankelijke convergentie krijgen voor grote golfgetallen in 1D, 2D en 3D toepassingen. Hiervoor gebruiken we een two-level deflatie projectie methode waarbij we kwadratische rationale Bézier krommen gebruiken om de deflatie ruimte en vectoren op te spannen. Als gevolg ligt het spectrum van de coarse-grid operator in het verlengde van de fine-grid operator, met als resultaat dat spectrum van het complete gepreconditioneerde systeem geclusterd blijft en een significante verbetering oplevert in convergentie gedrag ten opzichte van andere methodes.

LINEAIRE COMPLEXITEIT

De afgelopen 30 jaar hebben veel toegepaste wiskundigen hun tijd en toewijding besteed aan het werkend krijgen van een convergerend multigrid algoritme voor de Helmholtz vergelijking. Multigrid solvers staan bekend om het gebruik van een serie aan steeds kleiner wordende problemen en zijn hierdoor goedkoper in rekenkracht en makkelijk te implementeren. Helaas divergeren multigrid methoden voor de Helmholtz vergelijking en is dit nogsteeds een open probleem in de toegepaste wiskunde. De combinatie van hogere orde coarse spaces en efficiente standaard smoothing technieken uit de klassieke multigrid literatuur, blijken succesvol in het construeren van een volledig convergente V- en W-cycle algoritme (hoofdstuk 9). De belangrijkste componenten van het algoritme zijn het gebruik van hogere orde transfer operators, in plaats van de deflatie vectoren zoals in het vorige algoritme, en een complexe shift in de smoothing operator. Ondanks dat de methode convergeert en de eerste resultaten worden ondersteund door convergentie bewijzen, is er nog veel ruimte voor verder onderzoek. Met name omdat een efficiente toepassing van multigrid methode voor Helmholtz vergelijking een paradigm shift kan ondersteunen in het oplossing van het complexiteitsvraagstuk voor toepassingen waarbij grote golfgetallen worden gebruikt in 2D en 3D.

Mede vanwege het complexiteitsvraagstuk en de golfafhankelijke convergentie, breiden we tevens de two-level deflation solver uit naar een multi-level deflatie solver (hoofdstuk 8). In dit deel bewijzen we betere convergentie eigenschappen en rapporteren we de resultaten

van complexe numerieke experimenten voor 2D en 3D testproblemen die onze theoretische resultaten onderschrijven en ondersteunen.

NAUWKEURIGHEID

Tot slot, hebben we unprecedente manier gevonden om de nauwkeurigheid van de numeriek oplossing uit te drukken in de eigenwaarden van de operatoren waarvan de analytische oplossing bekend is (hoofdstuk 5). Door de pollution error te bekijken vanuit het perspectief van de eigenmodes is het mogelijk om theorie te ontwikkelen die de nauwkeurigheid en numerieke dispersie correcties in kaart kan brengen van multidimensionale problemen, onafhankelijk van de golf propagatie hoek. Dit was voorheen onmogelijk omdat de dispersiefout enkel opgesteld kon worden indien de hoek van propagatie bekend was. Voorts bestuderen we ook de toepassing van isogeometrische analyse (IgA) om de nauwkeurigheid te verbeteren en de 'pollution error' te minimaliseren (hoofdstuk 6). Onze resultaten leggen bloot dat het gebruik van IgA de pollution fout in sterke mate weet te onderdrukken in vergelijking met standaard eindige elementen methode van dezelfde orde.

I

Background Fundamentals

1

Introduction

Each individual fact taken by itself,
can indeed arouse our curiosity,
or our astonishment or be useful to us
in its practical applications.

Hermann von Helmholtz

1

At the heart of many fields like optics, acoustics, electrostatics and quantum mechanics, lies the wave equation. It's time-harmonic equivalent, the Helmholtz equation, has been studied to answer a variety of questions governing modern day applications. The Helmholtz equation, named after its creator, Hermann von Helmholtz, German physician and physicist, is a second order partial differential equation which models wave phenomena in the frequency domain. These phenomena are widely studied in various engineering practices. For example, for seismic exploration, we need to know how waves travel through the earth's subsurface. For nodules to show up on MRI and ultrasound, we need to know how electromagnetic waves travel and scatter through human tissue. For wireless telecommunications, we require insights into wave propagation at different frequencies. For cells to be inactivated or loaded with DNA, we need an understanding of the permeability of the electromagnetic field on the cell membrane.
But with our society facing new challenges, its relevance is becoming increasingly important. Even more innovative and futuristic projects rely the study of charged particles with electromagnetic fields, which are crucial for the advancement and understanding of nuclear fusion devices in our current energy transition. In particular, the study of more complex wave phenomena in the time domain can be supported by studying the Helmholtz equation.

1.1. HELMHOLTZ EQUATION

The Helmholtz equation can be derived from the wave equation, given that it models harmonic wave propagation in the frequency domain through a homogeneous medium. We start by considering the propagation of time harmonic waves, which is governed by (1.1)

$$(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2})\varphi(\mathbf{x}, t) = \mathbf{0} \tag{1.1}$$

In equation(1.1) the vector $\mathbf{x}$ denotes the spatial variable in some subspace Ω of $\mathbb{R}^n$, which represents the physical domain. The real constant c and the real variable t represent the wave speed and time parameter respectively.
A solution to equation (1.1) can be obtained by separating the variables into a spatial and time component

$$\varphi(\mathbf{x}, t) = u(\mathbf{x})\, T(t) \tag{1.2}$$

Letting equation (1.2) represent a potential solution to equation(1.1), we substitute the previous equation into the former to obtain

$$(\nabla^2 - \frac{1}{c^2}\frac{\partial^2}{\partial t^2})(u(\mathbf{x})\, T(t)) = \mathbf{0} \tag{1.3}$$

$$\frac{\partial^2 u}{u} = \frac{1}{Tc^2}\frac{\partial^2 T}{\partial t^2} = -k^2 \tag{1.4}$$

Note that in order for the solution to satisfy equation (1.1), we have to equate both sides of equation to a constant $-k^2$. Rearranging the left hand side of equation (1.1), which now is completely separated from the time component, we obtain the homogeneous Helmholtz equation

$$(-\nabla^2 - k^2)\, u(\mathbf{x}) = \mathbf{0} \tag{1.5}$$

Intuitively $u(\mathbf{x})$ can best be interpreted as the wave function, whereas k stands for the wavenumber, which relates the wavelength λ and the angular frequency. General expressions for the before mentioned are

$$k = \frac{2\pi}{\lambda} \tag{1.6}$$

Practical applications of the Helmholtz equation often involve the non-homogeneous Helmholtz equation. In this case the right hand side of (1.5) consist of a source function $f(\mathbf{x})$

$$f(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x_s}), \tag{1.7}$$

where $\mathbf{x_s}$ denotes the location of source in the domain. Additionally, in some applications, such as modeling phenomena through an inhomogeneous medium, a non-constant wavenumber $k(\mathbf{x})$ is enforced to capture different velocity profiles. Also, especially in geophysical applications, a damping constant is added to the wavenumber.

1.2. BOUNDARY CONDITIONS

Solving the Helmholtz equation on a bounded physical domain Ω requires the reinforcement of boundary conditions. In the absence of such conditions the problem becomes ill-posed; the equation in its current form models the indefinite propagation of waves. Therefore, we define either vanishing or reflecting boundary conditions at the boundary of Ω, which we denote by $\partial\Omega$.

- Vanishing boundary conditions: vanishing boundary conditions can be modelled by imposing homogeneous Dirichlet conditions

$$u(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \partial\Omega \tag{1.8}$$

- Reflecting boundary conditions: reflecting boundary conditions can be modelled by imposing homogeneous Neumann conditions, where $\mathbf{n}$ denotes the outward normal unit vector with respect to the boundary $\partial\Omega$

$$\left(\frac{\partial}{\partial \mathbf{n}}\right) u(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \partial\Omega \tag{1.9}$$

- Mixed boundary conditions: mixed boundary conditions can be modelled by imposing both homogeneous Dirichlet and Neumann conditions instantaneously. Within the context of the Helmholtz equation, if the equation is solved on an infinite domain, these mixed boundary conditions are often referred to as Sommerfeld Radiation conditions, where i represents the imaginary unit

$$\lim_{r\to\infty} \sqrt{r}\left(\frac{\partial}{\partial r} + ik\right) u(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \partial\Omega \tag{1.10}$$

  This condition ensures uniqueness of the solution and represents that there are no incoming waves from infinity. To approximate the Sommerfeld conditions on bounded domains, we use an approximation,

$$\left(\frac{\partial}{\partial \mathbf{n}} + ik\right) u(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in \partial\Omega, \tag{1.11}$$

where **n** denotes the outward normal unit vector with respect to the boundary $\partial\Omega$. This type of boundary condition is also known as the non-reflecting or absorbing boundary condition and it models the wave disappearing at infinity.

1.3. APPLICATIONS

In order to illustrate the wide range of applications of the Helmholtz equation, we give an example of a scattering problem and an example of a wave propagation problem. Scattering problems are considered to be both mathematical and diagnostic in nature. Inverse problems are solved to extract hidden features of natural phenomena. Here, one starts with some data and proceeds backwards to the source of that data. An important class of inverse problems is the study of inverse scattering of plane waves from material objects. Here, the data exists physically in the form of the scattered fields. The inverse problem is to determine the scatterer, which is governed by the Helmholtz equation. Similarly, the Helmholtz equation is also used in the modelling of the propagation of electromagnetic waves.

1.3.1. WIFI ANTENNA

In Fig. 1.1 we observe the propagation and scattering of electromagnetic waves for a particular floor plan in a building. The WiFi router is placed in the lower right corner of the floor and is modelled by an inhomogeneous source function. The source function will be zero everywhere, except where the antenna is located, leading to a point-source function. If walls are assumed to be concrete with a high refractive index, some absorption will be visible and reflections will be stopped.

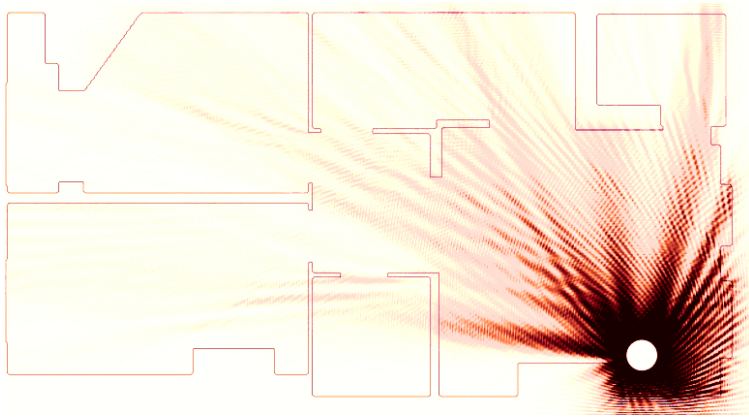


Figure 1.1: Wave propagation throughout the floor. Image from Jason Cole.

1.3.2. NUCLEAR FUSION

The study of electromagnetic waves in plasma are of paramount importance to propel the development of alternative energy sources, such as nuclear fusion. In this field of study, both cold and hot plasma problems are investigated. Cold plasma problems, in particular, have become ubiquitous in the study of radio frequency power in fusion plasma's. For magnetically confined fusion plasma's, it is convenient to examine the wave propagation of

the reduced Maxwell equations, which in fact leads to the Helmholtz equation. In Fig. 1.2 a combination of radio waves is able to stabilize fusion.

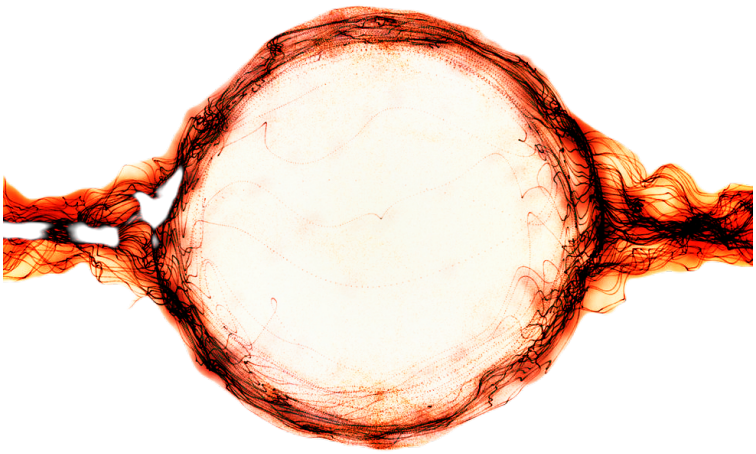


Figure 1.2: Illustration of controlled plasma by the emittance of radio waves.

1.3.3. EARTHQUAKE SIGNALS

The increasing number of earthquakes has led earth scientists to develop new methods in surface wave tomography in order to track phase fronts and map the travel times for earthquakes [1]. This method is based on the eikonal equation and is, therefore, referred to as 'eikonal tomography'.

Eikonal tomography does not account for frequency effects such as wave interference or backward scattering. This shortcoming potentially may lead to both systematic bias and random error in the phase velocity measurements, which would be particularly important at longer periods studied with earthquakes. It is shown here that eikonal tomography can be improved by additionally solving the Helmholtz equation as the latter allows the inclusion of the effects of both wave interference and backward scattering. As a result, a geographically localized correction can be applied leading a reduction in the uncertainties in the phase velocity maps of the earthquakes studied.

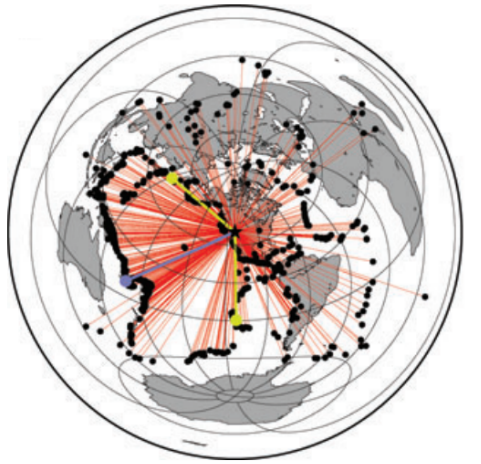


Figure 1.3: Data obtained from earthquakes. Circles mark the location of the earthquakes. Image from [1].

1.4. SCIENTIFIC COMPUTING CONSIDERATIONS

The common denominator in assessing examples such as the ones from Section 1.3 across so many different fields is numerical simulation. Simulation tools are developed through the utilization of a broad set of both theoretical and applied mathematical expertise. It enables savings in financial and computational resources before setting up costly experiments. It can even serve as a proxy in certain cases where such experiments can not be realized. While the general Helmholtz equation may appear very simple and elegant, retrieving the numerical solutions leads to a wide range of issues. Some problems are, to this day, considered an open problem in applied mathematics after decades of research.

In particular, the underlying complexity of the numerical solver grows with the frequency, which we have denoted as the wavenumber k. Despite increased computer power, direct solution methods are no viable alternative and we resort to iterative solvers. The bottleneck in designing iterative solvers lies in balancing the trade-off between accuracy and scalability. For accurate solutions, we require fine grids leading to uneconomical simulations which could take up days to weeks. For fast solutions, we need iterative solvers which converge in steps independent of the frequency. We also require that the amount of computational work depends linearly on the problem size. At the same time, the presence of heterogeneous media and variable frequencies further complicates the design of numerical solvers. In these instances, no analytical solution is available.

1.4.1. RESEARCH PILLARS

To work on these issues, we determined three research pillars.

- We aim on constructing a scalable iterative solver where the number of iterations are independent of the frequency or wavenumber (wavenumber independent convergence).
- We also require that the computational work depends linearly on the problem size

(linear complexity).

- At higher frequencies the quality of the analytical solution improves, while the quality of the numerical solution rapidly deteriorates. Consequently, we aim to develop new ways to study the accuracy (accuracy).

1.5. DISSERTATION OUTLINE

In order to solve the Helmholtz equation numerically, some fundamental concepts from numerical analysis themes are introduced in Part I. After introducing these themes such as numerical discretization, the array of suitable iterative solvers and the current span of multi-level solvers for the Helmholtz equation (1.5), we move our focus onto the study of the accuracy and scalability of numerical solutions.
In Part II and Part III we elaborate on the research pillars. Part II discusses the numerical accuracy and Part III focuses on the first two pillars. In each respective chapter, a literature overview will be presented as regards the topic concerned. Finally in Part IV we provide our conclusions, discussions and outlook for this research topic. In particular, this thesis consists of 4 parts and 11 chapters.

2

Numerical Discretization

The true and best way of learning any Art,
is not to see a great many examples done by another person,
but to possess ones self first of the principles of it,
and then to make them familiar, by exercising ones self in the practice.

Brook Taylor

Solving the Helmholtz equation numerically requires the translation from the continuous partial differential equation into its discrete counterpart. This process is what we refer to as numerical discretization, which relies heavily on approximation theory developed by the British Mathematician Brook Taylor. Several methods are at our disposal to discretize the Helmholtz equation, such as the finite difference and finite element method respectively. In this work we focus on the finite difference method, but a vast collection of works has been dedicated to studying the finite element method for the Helmholtz equation. The latter is particularly of interest in the case of complicated (geometric) domains.

2.1. FINITE DIFFERENCES

We elaborate on the concept of numerical discretization using the following one-dimensional example. Starting with the one-dimensional case, we can naturally extend the discretization to the two-dimensional case. We discretize the following continuous problem on a simple finite domain using a second-order accurate central difference scheme

$$\begin{aligned} -\frac{d^2u(x)}{dx^2} - k^2\,u(x) &= f(x), \quad x \in \Omega = (0,L), \\ u(x) &= 0, \quad x = 0, \\ u(x) &= 0, \quad x = L,. \end{aligned} \tag{2.1}$$

2.1.1. DISCRETIZATION OF THE GEOMETRY

For the discretization, we take $\Omega_{1,h} = [0,1]$ and we let n denote the number of elements on an uniform grid consisting of $n+1$ nodes, including the boundary $\partial\Omega_{1,h}$. Given the unit interval, we get the following numerical domain, with step size $h = \frac{1}{n}$

$$\Omega_{1,h} = \{(x_i)\,|\,x_j = jh, h = \frac{1}{n}, 1 \leqslant j \leqslant n+1,\, n \in \mathbb{N}\backslash\{0\}\}$$

In the two-dimensional case, our finite domain becomes the unit square domain $\Omega_{2,h} = [0,1]\times[0,1]$. A geometrical representation of these grids is illustrated below in Fig. 2.1.

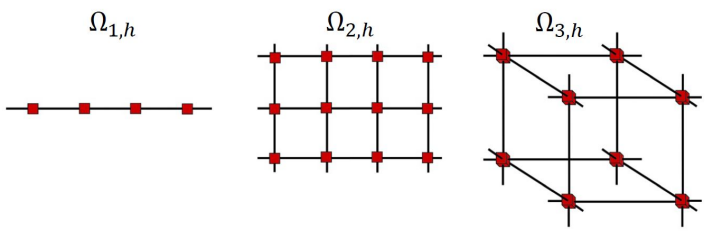


Figure 2.1: An equidistant numerical domain with internal nodes for 1D, 2D and 3D.

2.1.2. DISCRETIZATION OF THE PHYSICS

On both $\Omega_{1,h}$ and $\Omega_{2,h}$ respectively, we introduce spatial grid vectors in order to approximate the source function $f(x)$ and the wave function $u(x)$. Due to the vanishing Dirichlet boundary conditions, the numerical wave function will be completely defined at the internal grid nodes

$$\begin{aligned} &f(x) \approx f(x_j) = f_{h,j}, \\ &u(x) \approx u(x_j) = u_{h,j}, \\ &x \in \Omega_{1,h}. \end{aligned}$$

2.1.3. LINEAR SYSTEM FORMULATION

We arrive at a linear system formulation after approximating the continuous second order derivatives by central finite difference approximations. This scheme provides second order accuracy $\mathcal{O}(h^2)$ for smooth solutions on uniform grids. For the 1D case we have

$$\frac{-u_{h,j-1} + 2u_j - u_{h,j+1}}{h^2} - k^2 u_{h,j} = f_{h,j},\ 1 \leqslant j \leqslant n-1,$$

Implementing a x-line lexicographic ordering of the internal nodes, allows us to assemble the unknown grid values $u_{h,j}$ and $f_{h,j}$ into column vectors of dimension $(n-1)$ for the 1D case. Consequently, for the 1D problem, we construct a linear system of equations as follows

$$A_h u_h = \begin{bmatrix} -1 & 2-k^2h^2 & -1 \end{bmatrix} u_h = f_h$$

In the 1D case, the wave vector u_h and source function f_h are vectors with length $n-1$. We have transformed the continuous partial differential Helmholtz equation into a linear system of equations. Solving the Helmholtz boundary value problem now boils down to solving the system

$$\begin{aligned} &A_h u_h = f_h, \\ &A_h \in \mathbb{R}^{(n-1)\times(n-1)}, \\ &u_h, f_h \in \mathbb{R}^{(n-1)}. \end{aligned}$$

2.1.4. DISCRETIZATION OF THE BOUNDARY CONDITIONS

In Chapter 1 we described the common boundary conditions used in order to solve the Helmholtz equation. So far we have used Dirichlet boundary conditions for discretization purposes, but Sommerfeld boundary conditions are an integral part of well-posed Helmholtz boundary value problems. Numerically this condition is fulfilled approximately as an infinite domain is always reduced to a finite domain. Recall from Section 1.2 that the Sommerfeld boundary condition is modelled by

$$\left(\frac{\partial}{\partial \mathbf{n}}\right) u(\mathbf{x}) = -iku(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega \tag{2.2}$$

We start by discretisizing Eq. (2.2) for the 1D case. We again use a centered second order difference scheme to approximate the first order derivative. This translates to

$$\frac{\partial u}{\partial n} - iku \approx \frac{u_2 - u_0}{2h} - iku_1 = 0, \tag{2.3}$$

where u_0 is called a ghost-point to the left of u_1, given that the indices in our discretization scheme go from $j = 1$ to $j = n+1$ whenever the boundary nodes are included. The ghost point can be eliminated by observing that $u_0 = u_2 - 2hiku_1$.

2

2.1.5. LINEAR SYSTEM PROPERTIES

The matrix A obtained after discretization of the Helmholtz equation has several characteristic properties. Depending on whether Dirichlet or Sommerfeld boundary conditions are used, the matrix can be either *real- or complex*-valued. While A is *(complex) symmetric*, it is *non-normal* and *non-Hermitian* in case Sommerfeld conditions are used. If homogeneous Dirichlet boundary conditions are used, the matrix remains normal and thus self-adjoint. One can immediately notice that the coefficient matrix A is in fact the discretized Laplacian including a term involving $-k^2$

$$A = -\Delta - k^2 I, \tag{2.4}$$

where I represents the identity matrix and Δ the discretized Laplacian. For large enough k^2, the matrix becomes highly *indefinite*. As a result, the real part of the (complex) eigenvalues of A can be negative as well and the condition number of the matrix A becomes inevitably large. This is why Helmholtz problems are often referred to as being *ill-conditioned*.

SPECTRUM

When homogeneous Dirichlet boundary conditions are enforced, we can easily construct closed form expressions of the eigenvalues of the matrix A. In this case, looking closely at Eq. (2.4) reveals that the eigenvalues of the Helmholtz operator are similar to the eigenvalues of the Laplace operator including a shift $-k^2$. Thus, we obtain the following expressions for the 1D continuous (λ^j) and discrete $(\hat{\lambda}^j)$ eigenvalues

$$\lambda^j = j^2\pi^2 - k^2,\ j = 1,2,3,\ldots \tag{2.5}$$

$$\hat{\lambda}^j = \frac{1}{h^2}\left(2 - 2\cos(j\pi h) - k^2h^2\right),\ j = 1,2,\ldots,n-1. \tag{2.6}$$

The eigenvectors of the matrix A are

$$\hat{v}_h^j = \begin{pmatrix} \sin\pi jh \\ \sin\pi j2h \\ \vdots \\ \sin\pi j(n-1)h \end{pmatrix},$$
$$1 \leqslant j \leqslant n-1. \tag{2.7}$$

In case of homogeneous Dirichlet conditions, we explicitly need to ensure that $j^2\pi^2 \neq k^2$ and $i^2\pi^2 + j^2\pi^2 \neq k^2$, which would imply resonance and unbounded oscillations in the absence of dissipation. In the latter case, the problem would become ill-posed due to zero eigenvalues being present. The use of these homogeneous Dirichlet conditions in this work will serve as a theoretical test problem in order to determine preliminary convergence and accuracy properties.

2.1.6. HIGHER-DIMENSIONS

The discretization for a 2D problem follows naturally from the 1D case. Using a second-order finite difference scheme, we obtain

$$\frac{-u_{h,i(j-1)} - u_{h,(i-1)j} + 4u_{h,ij} - u_{h,i(j+1)} - u_{h,(i+1)j}}{h^2} - k^2 u_{h,ij} = f_{h,ij},\ 1 \leqslant i,j \leqslant n-1$$

The lexicographic ordering for the 2D case results in a mapping from a 2-index coordinate to a single index coordinate as illustrated in Fig. 2.2.

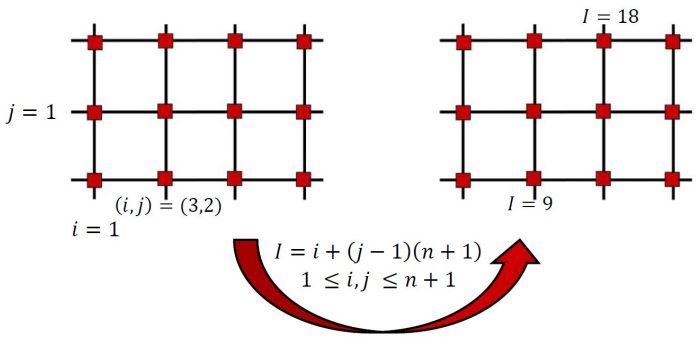


Figure 2.2: Illustration of lexicographic ordering.

The approximations of the derivatives on $\Omega_{2,h}$ can also be written in the following stencil notation

$$[A_h u_h]_{1\leqslant i,j\leqslant n} = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4-k^2h^2 & -1 \\ & -1 & \end{bmatrix} [u_h]_{1\leqslant i,j\leqslant n} = [f_h]_{1\leqslant i,j\leqslant n}.$$

Using the lexicographic ordering, we formulate a linear system of equations.Note that now, the vectors have length $(n-1)^2$. The extension to 3D can be derived analogously. Solving the Helmholtz boundary value problem now boils down to solving the system

$$\begin{aligned} &A_h u_h = f_h, \\ &A_h \in \mathbb{R}^{(n-1)^2 \times (n-1)^2}, \\ &u_h, f_h \in \mathbb{R}^{(n-1)^2}. \end{aligned}$$

DISCRETIZATION OF THE BOUNDARY CONDITIONS

The 2D boundary condition can be discretized by

$$\frac{\partial u}{\partial n} - iku \approx \frac{u_{h,2j} - u_{h,0j}}{2h} - iku_{h,1j} = 0. \tag{2.8}$$

In this case, we again define a stencil for the boundary nodes $u_{h,1j}$, for $2 \leqslant j \leqslant n-1$ as

$$\frac{1}{h^2}\begin{bmatrix} 0 & -1 & 0 \\ 0 & 4-k^2h^2+2hik & -2 \\ 0 & -1 & 0 \end{bmatrix}, \tag{2.9}$$

whereas for the corner point $u_{h,11}$ we obtain the stencil

$$\frac{1}{h^2}\begin{bmatrix} 0 & -2 & 0 \\ 0 & 4-k^2h^2+4hik & -2 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.10}$$

Repeating the above process for the other boundary nodes results in a complex symmetric linear system. The complex symmetry is obtained by dividing the non-corner stencil (Eq. (2.9)) and the corner stencil (Eq. (2.10)) points by 2 and 4 respectively.

SPECTRUM

If we use homogeneous Dirichlet conditions, we again determine the analytical eigenvalues. In this case, the eigenvalues for the 2D system are given by

$$\lambda^{i,j} = i^2\pi^2 + j^2\pi^2 - k^2,\ i,j = 1,2,3,\ldots \tag{2.11}$$

$$\lambda^{\hat{i},j} = \frac{1}{h^2}\left(4 - 2\cos(i\pi h) - 2\cos(j\pi h) - k^2h^2\right),\ i,j = 1,2,\ldots,n-1. \tag{2.12}$$

Note that also here we need to warrant for the case where $k^2 = i^2\pi^2 + j^2\pi^2$, as this would imply resonance. Thus, whenever we use this model problem, we explicitly check whether the matrix remains non-singular.

2.2. FINITE ELEMENTS

In this section we briefly explain the finite element discretization (FEM) of the Helmholtz equation using piece-wise linear elements. The main idea behind the method is to use simple basis functions (piece-wise linear) to construct local elements which approximate the solution on these subdomains. The combination of the subdomains then leads to a global system, which can also be represented by a linear system of equations. We discretize the same model problem.

2.2.1. DISCRETIZATION OF THE GEOMETRY

We again start with a discretization of the geometry where we take the domain $\Omega_{1,h} = [0,1]$ and divide it into $2^n = N$ elements with length $h = \frac{1}{N}$, which we denote by e_j. In this case, e_j represents the interval $[x_j, x_{j+1}]$ and is an element within the physical domain. The end point of the interval x_{N+1} is located at the last element e_N.
Thus, in 1D the interval is divided into elements and in 2D the discrete domain is divided into triangles, see Fig. 2.4 for an example.

BASIS FUNCTIONS

Moving on with the 1D example, we define piece-wise linear basis functions on each element.

$$\varphi_j(x) = \begin{cases} \frac{x-x_{j-1}}{h} & x_{j-1} \leqslant x \leqslant x_j \\ \frac{x_{j+1}-x}{h} & x_j \leqslant x \leqslant x_{j+1} \\ 0 & \text{elsewhere} \end{cases} \qquad \varphi_j(x_i) = \delta_{ij}, \tag{2.13}$$

where δ_{ij} is the Kronecker delta function. Fig. 2.5 shows the global shape functions over the space containing all elements.

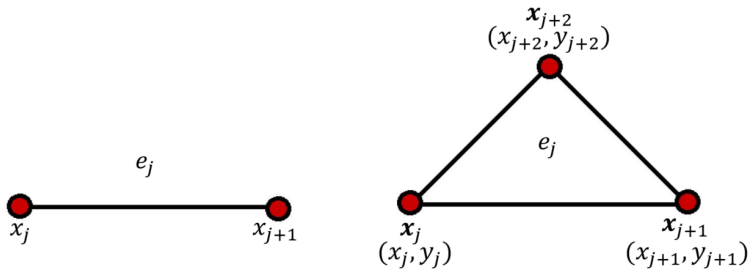


Figure 2.3: Illustration of an element e_j on $\Omega_{1,h}$ (L) and $\Omega_{2,h}$ (R). Note that in the 2D case we need 1 more node to construct an element and the nodes are 2D-coordinates rather than points.

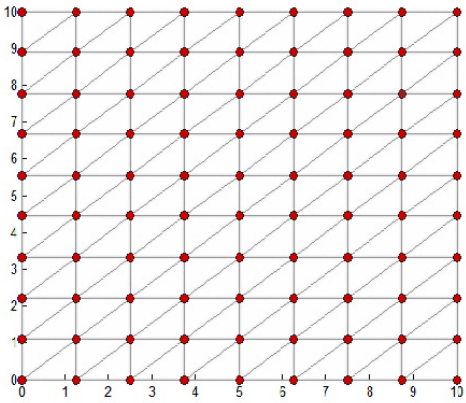


Figure 2.4: Example of triangularization of the unit square. The domain is divided into equally spaced triangle elements to create the depicted mesh.

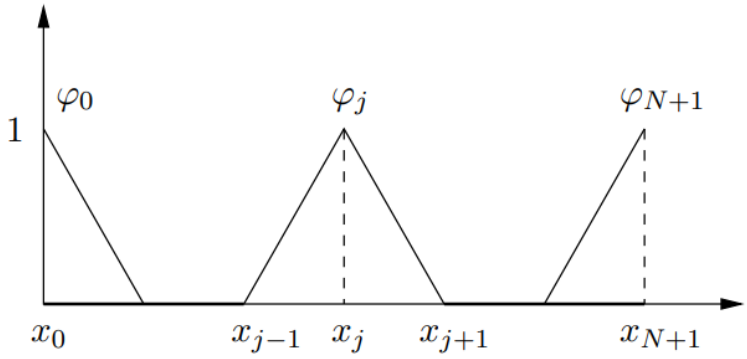


Figure 2.5: Shape functions φ_j for elements on the unit interval. These are also known as *hat functions*.

2.2.2. DISCRETIZATION OF THE PHYSICS

Before we obtain a discretized version of the solution, we need to put the original PDE into the *weak form*. In order to do this, we assume there exist a test function w, which obeys the same boundary conditions as our original solution function u. We also assume that there exist a set of basis functions such that u and w can be represented by this same basis. We multiply Section 2.1 by w and integrate both sides. This gives

$$-\int_0^1 w\frac{d^2u}{dx^2}dx - k^2\int_0^1 wudx = \int_0^1 wfdx \tag{2.14}$$

Integration by parts gives the weak formulation of our boundary value problem

$$-\left[w\frac{du}{dx}\right]_0^1 + \int_0^1 \frac{dw}{dx}\frac{du}{dx}dx - k\int_0^1 wudx = \int_0^1 wfdx. \tag{2.15}$$

Given that w satisfies the same boundary conditions as v, the first term vanishes. Next, we need to find a suitable basis for u and w such that both functions can be written as a superposition of the basis functions. This will allow us to represent the discrete solution as the sum of the elements of the basis by solving the integrals corresponding to each element. If we denote the basis functions by φ, then we can write

$$w \approx w_h = \sum_{i=1}^{N} w_{h,i}\varphi_i, \quad \varphi_i(0) = 0, \quad \varphi_i(1) = 0 \tag{2.16}$$

$$u \approx u_h = \sum_{j=1}^{N} u_{h,j}\varphi_j, \quad \varphi_j(0) = 0, \quad \varphi_j(1) = 0. \tag{2.17}$$

Substituting Eq. (2.16) and Eq. (2.17) into Eq. (2.15) gives us the discretized version of the weak form

$$\sum_{j=1}^{N} u_{h,j}\left[\int_0^1 \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx - k^2\int_0^1 \varphi_i\varphi_j dx\right] = \int_0^1 \varphi_i f dx \quad i = 1,2,\ldots N. \tag{2.18}$$

2.2.3. LINEAR SYSTEM FORMULATION

Note that the formulation in Eq. (2.18) is equivalent to solving a linear system of equations $A_h u_h = f_h$ where

$$u_h = u_{h,j}{}_{1\leqslant j\leqslant N}, \quad f_h = \left[\int_0^1 \varphi_i f dx\right]_{1\leqslant i\leqslant N}, \tag{2.19}$$

$$A_h = \left[\int_0^1 \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx - k^2\int_0^1 \varphi_i\varphi_j dx\right]_{1\leqslant i,j\leqslant N}.$$

For the remainder of this section, we proceed by dropping the subscript h for the linear system matrix A. We can split the matrix A in terms of a stiffness matrix K and mass matrix M by writing

$$K = \left[\int_0^1 \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx\right]_{1\leqslant i,j\leqslant N}, \quad M = \left[\int_0^1 \varphi_i\varphi_j dx\right]_{1\leqslant i,j\leqslant N}. \tag{2.20}$$

ELEMENT MATRIX

Given that the basis functions are defined in an element-wise manner, the integrals can also be evaluated element-wise. Moreover, the shape functions φ_j have small support, which implies that most of the coefficients in A will be zero. Note that for an arbitrary element $e_j = [x_j, x_{j+1}]$, there will only be two non-zero piece-wise linear shape functions. Using this, we thus obtain

$$K = \left[\int_0^1 \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx\right]_{1\leqslant i,j\leqslant N} = \sum_{l=1}^{N}\int_{e_l}\frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx. \tag{2.21}$$

Thus, instead of the global linear system in Eq. (2.20), we obtain the element-matrix $A^j = K^j + M^j$, with respect to element ξ_j

$$K^j = \begin{bmatrix} k^j_{11} & k^j_{12} \\ k^j_{21} & k^j_{22}\end{bmatrix}, \quad k_{ij} = \left(\int_{x_j}^{x_{j+1}}\frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx\right)_{i,j\in|e_j|}, \tag{2.22}$$

$$M^j = \begin{bmatrix} m^j_{11} & m^j_{12} \\ m^j_{21} & m^j_{22}\end{bmatrix}, \; m_{ij} = \left(-k^2\int_{x_j}^{x_{j+1}}\varphi_i\varphi_j dx\right)_{i,j\in|e_j|}. \tag{2.23}$$

Here, $|e_j|$ denotes the cardinality of the nodes in element ξ_j. Using the piece-wise linear basis functions, we get the following stencil for the element-matrix

$$K^j = \frac{1}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1\end{bmatrix}, \quad M^j = \frac{1}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2\end{bmatrix} \tag{2.24}$$

GLOBAL MATRIX ASSEMBLY

The global matrix can be assembled by looping over all the mesh elements and adding their respective contributions. An advantage of this method is that all the information to solve the Helmholtz equation is now stored locally. For the right-hand function given in Eq. (2.26), a numerical quadrature rule has to be used to obtain the element vector and global right-hand vector respectively. The term in Eq. (2.26) contains an integral with $f(x)$ and can also be numerically integrated and stored locally to give the *element vector*. Assembly in the same way then leads to the global right-hand vector.

2.2.4. DISCRETIZATION OF THE BOUNDARY CONDITIONS

So far we have assumed homogeneous Dirichlet conditions. However, as mentioned previously, for the Helmholtz equation, homogeneous Sommerfeld conditions on one end can also be applied. In this case, the variational formulation of the internal approximation becomes

$$\sum_{j=1}^{N+1} u_j\left[\int_0^1 \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx - k^2\int_0^1\varphi_i\varphi_j dx + ik\varphi_i(1)\varphi_j(1)\right] = \int_0^1\varphi_i f dx \quad i = 1,2,\ldots N+1. \tag{2.25}$$

In this case, we include the basis function for the node x_{N+1}, φ_{N+1} as well which can be visualized as half a triangle at the end of the interval (see Fig. 2.5). As a result, we need to

solve a linear system $Au = f, A \in \mathbb{C}^{N+1}$, where we have

$$u = u_{j\,1\leqslant j\leqslant N+1}, \quad f = \left[\int_0^1 \varphi_i f\,dx\right]_{1\leqslant i\leqslant N+1}, \tag{2.26}$$
$$A = \left[\int_0^1 \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx - k^2\int_0^1 \varphi_i\varphi_j dx + ik\varphi_i(1)\varphi_j(1)\right]_{1\leqslant i,j\leqslant N+1}.$$

Again, in terms of the element matrix, we obtain $A^j = K^j + M^j + R^j$ for element e_j, where R^j now contains the contributions of the boundary terms. We thus have

$$K^j = \begin{bmatrix} k^j_{11} & k^j_{12} \\ k^j_{21} & k^j_{22}\end{bmatrix}, \quad k_{ij} = \left(\int_{x_j}^{x_{j+1}} \frac{d\varphi_i}{dx}\frac{d\varphi_j}{dx}dx\right)_{i,j\in|e_j|}, \tag{2.27}$$

$$M^j = \begin{bmatrix} m^j_{11} & m^j_{12} \\ m^j_{21} & m^j_{22}\end{bmatrix}, \; m_{ij} = \left(\int_{x_j}^{x_{j+1}} \varphi_i\varphi_j dx\right)_{i,j\in|e_j|}, \tag{2.28}$$

$$R^j = \begin{bmatrix} r^j_{11} & r^j_{12} \\ r^j_{21} & r^j_{22}\end{bmatrix}, \quad r_{ij} = (\varphi_i(1)\varphi_j(1))_{i,j\in|e_j|}. \tag{2.29}$$

2.2.5. HIGHER-DIMENSIONS

In 2D or 3D, if we let the domain be represented by Ω and the boundary by $\partial\Omega = \Gamma_1 \cup \Gamma_2$, then the weak formulation of our model problem becomes

$$\int_\Omega \nabla u\cdot\nabla w\,\mathrm{d}\Omega - \int_\Omega k^2 uw\,\mathrm{d}\Omega - ik\int_{\Gamma_2} uw\,\mathrm{d}\Gamma = \int_\Omega fw\,\mathrm{d}\Omega. \tag{2.30}$$

Note that here the Sommerfeld condition is enforced on Γ_2, where the homogeneous Dirichlet condition is enforced on Γ_1. Next, a geometry function $\mathbf{F}$ is then defined to parameterize the physical domain Ω by describing an invertible mapping to connect the parameter domain $\Omega_0 = (0,1)^2$ with the physical domain Ω.

$$\mathbf{F} = \Omega_0 \to \Omega, \quad \mathbf{F}(\xi,\eta) = (x,y). \tag{2.31}$$

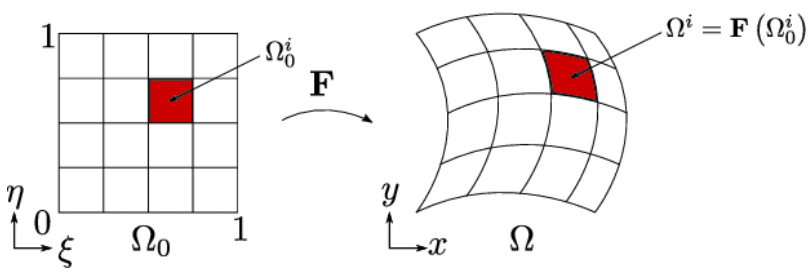


Figure 2.6: Example of the geometry function $\mathbf{F}$ and its parametrization of the unit square.

Next, a Lagrange polynomial basis is constructed on the parameter domain Ω_0, where the basis functions are now taken over a reference element $e_{0\,j}$ instead of the element e_j in the physical domain. Using these basis functions, we obtain the discretization of the interior

$$K = \left[\sum_e \int_e \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}\Omega(e)\right]_{1\leqslant i,j\leqslant N}, \quad M = \int_e \varphi_i \varphi_j \, \mathrm{d}\Omega(e), \tag{2.32}$$

and

$$R = \left[\sum_{\Gamma(e)} \int_{\Gamma(e)} \varphi_i \varphi_j \, \mathrm{d}\Gamma_2(e)\right]_{1\leqslant i,j\leqslant N} \tag{2.33}$$

of the boundary. We thus obtain the following global linear system

$$Au = \left(K + -k^2 M + -ikR\right) u = f. \tag{2.34}$$

A few remarks are in place. The basis functions are applied to reference parametrized elements e_0 and thus the integration is done locally per element. Given that $\mathbf{F}$ is invertible, the integration can then be mapped to the physical domain. In this way, the global linear system can still be solved with respect to the physical domain, which is also why the integrals in Eq. (2.32) and Eq. (2.33) are stated with respect to the elements in the physical domain.

2.3. NUMERICAL DISPERSION

One difficulty we encounter when trying to solve the Helmholtz equation numerically is that of the *pollution error*. Whether we use the finite difference or finite element method, the accuracy of the resulting solution strongly depends on the granularity of the grid. For larger k there appears to be a difference between the analytical wavenumber k and its non-continuous counterpart. This effect aggravates as the wavenumber k increases as a result of the solutions becoming more oscillatory in nature. Due to this, the numerical solution contains phase differences relative to the analytical solution. In order to understand this effect, a measure of the pollution error is given in terms of the the step-size or mesh-width h relative to the wavenumber k. The smaller h, the less pollution error we have. However, as h gets smaller, the linear system of equations becomes larger and retrieving the solution becomes more computationally difficult. In this work we focus on the pollution error after using a finite difference discretization, but the original pollution studies were conducted using finite elements discretization. This will be the main topic of Part II.

3

KRYLOV SOLVERS

Mathematics, like a millstone,
grinds everything placed under it and,
just as you won't get wheat flour
by grinding Deadly Nightshade,
you won't get the truth from the false premises,
even if you cover the page with formulae.

Alexei N. Krylov

After discretization, we obtain a linear system $Au = f$. Given that our coefficient matrix A is indefinite, we are limited in our choice of iterative methods. Moreover as the wavenumber k increases, the problem size increases as well in order to guarantee accurate solutions. Direct numerical solution methods, such as the lower-upper (LU) factorization method, become impractical for solving medium sized 2D and 3D problems. Despite these drawbacks, direct numerical solution methods can serve as subdomain solvers in domain decomposition methods and multigrid methods, which we will come to in Chapter 4.
In this work, we focus on Krylov subspace methods as the underlying method of choice, given that basic iterative methods suffer from reluctant convergence behavior or even divergence in the case of indefinite Helmholtz problems. We also discuss the inclusion of a preconditioner and its effect on convergence.

3.1. KRYLOV SUBSPACE METHODS

Consider a general linear system

$$\begin{aligned} &Au = f, \\ &A \in \mathbb{C}^{n\times n},\ u, f \in \mathbb{C}^n. \end{aligned} \tag{3.1}$$

Definition 1. *(Petrov-Galerkin Method) Given a linear system $Au = f$, let A be a matrix in $\mathbb{C}^{n\times n}$, u, f vectors in $\mathbb{C}^n$. Then a solution of equation 3.2 can be approximated by*

$$y = u_0 + s, s \in S \subset \mathbb{C}^n, \tag{3.2}$$

where u_0 is a predefined initial approximation and S is denoted as the search space. Let $r \in \mathbb{C}^n$ be defined as the residual vector such that we can define a constraint space C satisfying

$$r := f - Ay \perp C \subset \mathbb{C}^n. \tag{3.3}$$

Then a Petrov-Galerkin method is well defined if $\langle C, AS\rangle$ is nonsingular for any C and Y, where $C, Y \subseteq \mathbb{C}^n$.

Here $\langle \bullet \rangle$ denotes the standard inner product defined on the complex space. If the latter condition is satisfied, we get an approximate solution using the following theorem

Theorem 1: Petrov-Galerkin Method

Let A be a matrix in $\mathbb{C}^{n\times n}$, u, f vectors in $\mathbb{C}^n$ such that the Petrov-Galerkin method with search space S and constraint space C is well-defined. Then the approximate solution y and the corresponding residual r that satisfy Definition 1 are given by

$$y = u_0 + S\langle C, AS\rangle^{-1}\langle C, r_0\rangle, \tag{3.4}$$

$$r = f - Ay = P_{C^\perp, AS} r_0, \tag{3.5}$$

where $r_0 = f - Au_0$ is the initial residual. Furthermore, the linear system from Definition 1 is solved if and only if $r_0 \in AS$

Proof. For a proof of this theorem, see [2] corollary 2.26, p. 23. ■

Using Theorem 1, we now have a practical way to find an approximate solution $y \approx x$ which solves the linear system $Au \approx Ay = f$. However, we would like to find an approximate solution which is not only optimal, but also unique. For this purpose, we use the following theorem

Theorem 2: Well-definedness and Optimality

Consider a linear system $Au = f$ with A a matrix in $\mathbb{C}^{n\times n}$, u, f vectors in $\mathbb{C}^n$. Furthermore, let $u_0 \in \mathbb{C}^n$ be the inital guess vector and let S be an $n-$dimensional subspace. Then Petrov-Galerkin method with search space S and constraint space C is well defined and defines an unique approximate solution $y + u_0 \in S$ if one of the following conditions holds:

1. $C = S, S \cap \mathcal{N}(A) = \{0\}$, A is self-adjoint and positive semi-definite. Then
$$\|u - y\|_A = \inf_{z \in u_0 + S} \|u - z\|_A,$$
where $\|\bullet\|_A$ is the norm defined by $\|z\|_A = \sqrt{\langle z, Az\rangle}$.
2. $C = AS, S \cap \mathcal{N}(A) = \{0\}$. Then
$$\|f - Ay\| = \inf_{z \in u_0 + S} \|f - Az\|.$$

Proof. For a proof of this theorem, see [2] lemma 2.28, p. 23. ■

Note that either the residual or the difference between the true and approximate solution is minimized, and thus we obtain an optimality certificate for constructing an approximate solution to the original linear system 3.2.
We now proceed by giving the definition of a general Krylov subspace, using an arbitrary vector $v \in \mathbb{C}^n$:

Definition 2. *(Krylov Subspace) Given a linear system $Au = f$, with u, f, v vectors in $\mathbb{C}^n$ Then the m-th Krylov subspace is defined by*

$$\begin{aligned} K_m(A, v) &= \text{span}\{v, Av, ...A^{m-1}v\}, \\ K_0(A, v) &= \{0\}, m \geqslant 1. \end{aligned} \tag{3.6}$$

If the vectors from Definition 2, i.e. $v, Av, \ldots, A^{m-1}v$ are linearly independent, they form a basis for the Krylov subspace $\mathcal{K}_m(A, v)$. Furthermore, it has been shown in [2] that there exists a minimal index d at which the Krylov subspace becomes invariant, i.e., $A\mathcal{K}_d(A, v) \subseteq \mathcal{K}_d(A, v)$. As a result, applying A to v will not result in an additional vector which can span the Krylov subspace any further. Using this index d, it has also been shown that a Krylov subspace, for a nonsingular matrix A, has the following properties:

1. *Dimension*: $\dim \mathcal{K}_m(A, v) = m$ for $m \leqslant d \leqslant n$.
2. *Nested sequence of subspaces*: $K_{m-1}(A, v) \subseteq \mathcal{K}_m(A, v)$ for $m \geqslant 1$.

3. *The following statements are equivalent*:
   - $A\mathcal{K}_d(A,v) = \mathcal{K}_d(A,v)$
   - $\mathcal{K}_d(A,v) \cap \mathcal{N}(A) = \{0\}$
   - $v \in A\mathcal{K}_d(A,v)$

A Krylov subspace method is essentially an iterative implementation of the Petrov-Galerkin method over the Krylov subpace from Definition 2 using $v = r_0$. It was developed by the Russian Mathematician Alexei Krylov in 1931.
If we take the Krylov subspace K_m as a basis for the search space as defined in Definition 1 and apply Theorem 2 up to the point where the subspace becomes invariant, we arrive at the heart of all Krylov subspace methods.

Corollary 2.1: Krylov Subspace Method

Consider a consistent linear system $Au = f$ with $A \in \mathbb{C}^{n\times n}$ and $f \in \mathbb{C}^n$. Let $u_0 \in \mathbb{C}^n$ be an initial guess corresponding to the initial residual $r_0 = f - Au_0$. Let $d < \infty$ be the minimal index at which $A\mathcal{K}_d(A,r_0) \subseteq \mathcal{K}_d(A,r_0)$ and let $\mathcal{K}_d(A,r_0) \cap \mathcal{N}(A) = \{0\}$. The sequence of iterates $\{u_m\}_{m\in 1,..,d}$ that satisfy

$$u_m = u_0 + s_m,\ s_m \in S = \mathcal{K}_m(A,r_0),$$

$$r_m := f - Au_m \perp C_m,$$

is well defined and u_d is a solution of the linear system $Au_d = f$ if one of the following conditions holds:

1. $C_m = \mathcal{K}_m(A,v)$, A is self-adjoint and positive semidefinite. Then the iterates u_m satisfy the optimality property
$$\|u - u_m\|_A = \inf_{z\in u_0+\mathcal{K}_m(A,r_0)} \|u - z\|_A. \tag{3.7}$$
2. $C_m = A\mathcal{K}_m(A,r_0)$. Then the iterates u_m satisfy the optimality property
$$\|f - Au_m\| = \inf_{z\in u_0+\mathcal{K}_m(A,r_0)} \|f - Az\|. \tag{3.8}$$

Proof. In both cases the well-definedness and optimality of the approximate solutions follow from Theorem 2. $Au_d = f$ follows from Theorem 1 and using the second property of the Krylov subspaces. For more details, see [2] corollary 2.41, p. 31. ∎

Theoretically, for $m \leqslant d \leqslant n$, the vectors $r_0, Ar_0, ..A^{m-1}r_0$ are linearly independent. They also form a basis for the Krylov subspace $\mathcal{K}_m(A,r_0)$. However, numerically this basis becomes indistinguishable from linear independence as the computation of the vector $A^i r_0$ using the power method usually points in the direction of the dominant eigenvector as i increases. As a result, if n is large, most of the vectors in $\mathcal{K}_m(A,r_0)$ will point to the same direction, rendering an ill-conditioned basis. Consequently, a Krylov subspace method is always

constructed by implementing an basis orthonormalization process, such as the Arnoldi or Lanczos method (modified Gram-Schmidt), see [3] and [4].

As a result of Corollary 2.1, different iterative Krylov subspace methods can be obtained by varying the constraint space C to be equal to either $\mathcal{K}_m$ or $A\mathcal{K}_m$. Indefiniteness of the coefficient matrix A restricts the applicability of several Krylov subspace methods for the Helmholtz equation, which are based on equation 3.7 from Corollary 2.1. For example, the well known CG-method[1] requires the input of a symmetric and positive-definite coefficient matrix A. In case of a complex matrix, we require A to be Hermitian.

3.2. GMRES-METHOD

The GMRES-method is based on the MINRES-method. The MINRES method was particularly developed as an extension of the Lanczos method to solve a linear system with a self-adjoint but indefinite coefficient matrix A. The GMRES method was proposed for general matrices, interchanging the Lanczos method for the Arnoldi method. Both methods are characterized by minimizing the residual norm over the Krylov subspace. In essence, this translates into the minimization problem from Corollary 2.1, equation 3.8, which we now reformulate specifically as

Theorem 3: Minimized residual

Consider a consistent linear system $Au = f$ with $A \in \mathbb{C}^{n\times n}$ and $f \in \mathbb{C}^n$. Let $u_0 \in \mathbb{C}^n$ and $r_0 = f - Au_0$ be such that $d < \infty$ and $\mathcal{K}_d(A, r_0) \cap \mathcal{N}(A) = \{0\}$ are fulfilled. Then, for $S_m = \mathcal{K}_m(A, r_0)$ and $C_m = A\mathcal{K}_m(A, r_0)$, the iterates $u_m = u_0 + s_m$, $s_m \in \mathcal{K}_m(A, r_0)$ minimize the residual norm, i.e.

$$\|f - Au_m\| = \inf_{z \in u_0 + \mathcal{K}_m(A,r_0)} \|f - Az\|, \tag{3.9}$$

and u_d is a solution for the linear system .

Proof. Applying Theorem 2 and Corollary 2.1 leads to the GMRES-method. For more details, please refer to [2], section 2.9.1. ∎

Note that Theorem 3 only holds for u_0 and r_0 satisfying $d < \infty$ and $\mathcal{K}_d(A, r_0) \cap \mathcal{N}(A) = \{0\}$. However, as long as A is non-singular, these conditions are automatically satisfied and the GMRES-method is well-defined for any initial choice u_0 [2]. Consequently, in upcoming sections we will present the results assuming that the coefficient matrix A is non-singular.

3.2.1. ARNOLDI'S-METHOD

We have previously mentioned that the application of Krylov subspace methods goes hand in hand with an orthonormalization procedure in order to obtain a well-conditioned ba-

[1] The *Conjugate Gradient* method falls into the first category of Corollary 2.1, i.e. equation 3.7 and minimizes the error in terms of the A-norm. Where the GMRES-method uses an Arnoldi procedure for orthonormalizing the Krylov basis vectors, the CG-method uses the Lanczos method. The CG-method is widely used for large sparse SPD systems due to its superlinear convergence behavior. For more information, please refer to [5] section 7.1.3 and [2] section 2.8.

sis for the Krylov subspace. As the GMRES-method is applicable to general and thus non-symmetric matrices, the Arnoldi procedure is used to construct a set of orthonormal basis vectors, which in algorithmic form is given below

Algorithm 1: Arnoldi's Orthonormalization Algorithm

Initialization:
Choose v_1 with $\|v_1\| = 1$
for $j = 1,2,..n$ **do**
  $w_j := Av_j$
  for $i = 1,2,..j$ **do**
    $h_{i,j} := (w_j, v_i)$
    $w_j := w_j - h_{i,j} v_i$
  end
  $h_{j+1,j} := \|w\|$
  $v_{j+1} := \frac{w}{h_{j+1,j}}$
end

Each step in the algorithm multiplies v_j by A and orthonormalizes the vector w_j with respect to all previous Arnolid vectors v_i from $i = 1$ to j. Using the Arnoldi method, we arrive at two widely used propositions, see [6], p. 129.

Theorem 4: Orthonormal basis

Assume that Arnoldi's algorithm does not stop before the $m-$th step, then the vectors $v_1, v_2, \ldots, v_m$ form an orthonormal basis of the Krylov subspace $\mathcal{K}_m(A, v_1)$.

Theorem 5: Hessenberg matrix

Let V_m be the $m \times m$ matrix with column vectors $v_1, v_2, \ldots, v_m$. Let $\widehat{H_m}$ be the $((m+1) \times m)$ Hessenberg matrix whose nonzero entries $h_{i,j}$ are defined by Arnoldi's method and let $e_m = \{0,0,\ldots,1\}^T$. If we let H_m be the matrix obtained from $\widehat{H_m}$ by deleting its last row, then the following relation holds

$$AV_m = V_m H_m + w_m {e_m}^T, \tag{3.10}$$

$$= V_{m+1} \widehat{H_m}, \tag{3.11}$$

$${V_m}^T A V_m = H_m. \tag{3.12}$$

We can implement Arnoldi's method into Theorem 3, by noting that iterate vectors u_m can be written as $u_m = u_0 + V_m s_m$, where s_m is a vector in $\mathbb{C}^m$ and V_m is an orthonormal basis for the Krylov subspace. If we let $\beta = \|r_0\|$ and $v_1 = r_0/\|r_0\|$, we use Eq. (3.11) to obtain

$$\begin{aligned} \|f - Au_m\| &= \|f - A(u_0 + V_m s_m)\|, \\ &= \|r_0 - AV_m s_m\|, \\ &= \left\|\beta v_1 - V_{m+1} \widehat{H_m} s_m\right\|, \\ &= \left\|V_{m+1}(\beta e_1 - \widehat{H_m} s_m)\right\|. \end{aligned} \tag{3.13}$$

By definition, the columns of V_{m+1} are orthonormal and we can rewrite equation 3.13 as follows

$$\left\| V_{m+1}(\beta e_1 - \widehat{H_m} s_m) \right\| = \left\| \beta e_1 - \widehat{H_m} s_m \right\|. \tag{3.14}$$

The optimality property from equation 3.9, Theorem 3 becomes

$$\begin{aligned} \|f - Au_m\| &= \left\| \beta e_1 - \widehat{H_m} s_m \right\|_m, \\ &= \min_{z \in \mathbb{C}^n} \left\| \beta e_1 - \widehat{H_m} z \right\|. \end{aligned} \tag{3.15}$$

As a result, the approximate solution is the unique z vector which minimizes $F(z) = \min_{z \in \mathbb{C}^n} \left\| \beta e_1 - \widehat{H_m} z \right\|$ over $\mathcal{K}_m(A, r_0)$ which iteratively reduces to finding

$$s_m = \arg\min_{z \in \mathbb{C}^n} \left\| \beta e_1 - \widehat{H_m} z \right\|.$$

3.2.2. GMRES-ALGORITHM

The GMRES-method can be implemented using the following algorithm:

Algorithm 2: GMRES-method $Au = f$

Initialization:
Choose u_0 and compute $r_0 = f - Au_0$, $b_0 = ||r_0||$ and $v_1 = r_0/b_0$
for $j = 1, 2, ..n$ **do**
  $w_j := Av_j$
  for $i = 1, 2, ..j$ **do**
    $h_{i,j} := (w_j, v_i)$
    $w_j := w_j - h_{i,j} v_i$
  end
  $h_{j+1,j} := \|w\|$
  $v_{j+1} := \frac{w}{h_{j+1,j}}$
end

Note that this includes the Arnoldi orthonormalization algorithm. The GMRES-method is stable and only breaks down if $h_{j+1,j} = 0$. However, if $h_{j+1,j} = 0$ then $u_j = u$ and we retrieve the exact solution.

3.2.3. CONVERGENCE

In this section we briefly describe the convergence properties of the GMRES-method, which is based on the following theorem

Theorem 6: GMRES Convergence

Let P_m be the space of all polynomials of degree less than m and let $\sigma = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ represent the spectrum of A. Moreover, we define

$$\varepsilon^m = \min_{p \in P_m, p(0)=1} \max_{\lambda_i \in \sigma} |p(\lambda_i)|.$$

Suppose that A is diagonalizable so that $A = XDX^{-1}$ where D is a diagonal matrix containing $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. Then the residual norm of the $m-$th iterate satisfies

$$\frac{\|r_m\|_2}{\|r_0\|_2} = \min_{p \in P_m, p(0)=1} \frac{\|XP(D)X^{-1}r_0\|_2}{\|r_0\|_2} \leqslant K(X)\varepsilon_m \|r_0\|_2, \tag{3.16}$$

where $K(X) = \|X\|_2 \|X^{-1}\|_2$.

Proof. For a proof see [5], Theorem 7.3.1. and [7], section 3.1. ■

It has been stated that it would be impossible to predict the convergence behavior of the GMRES-method *solely* in terms of the eigenvalues of A [8]. In fact, the author argues that in case convergence is monitored through the spectrum, additional assumptions on *departure* from normality are a necessity. [7] have presented an extensive overview of the convergence properties of Krylov subspace methods. The problem with non-normality seems to be related to ill-conditioned eigenvectors resulting in very large $K(X)$ due to $\|X^{-1}r_0\| > \|r_0\|$. As a result, the bound in equation 3.16 may not be sharp and information regarding the convergence may be disconnected from spectral properties. However, [9], [7] and [10] all argue that for a large class of matrices, such as general normal and Hermitian matrices, the convergence results in terms of the spectral distribution properties hold. [7] even emphasize that theoretically, non-normality of a matrix does *not* lead to slower convergence, as for each non-normal matrix A there exists a normal matrix B with the same convergence behavior.

For normal matrices A in general, the eigenvectors form an orthonormal set making X in equation 3.16 well-conditioned. Due to the orthonormality, the eigenvalues have a predominant influence on the rate of convergence. Consequently, clustering and favorably distributed eigenvalues stimulate convergence, while eigenvalues close to the origin impede convergence.

The GMRES-method is considered inefficient in case a large number of iterations are needed. Due to its long recurrences, it requires increasing memory storage and computational force for the orthonormalization process. Several remedies have been opted to circumvent this drawback, such as for example restarted GMRES [11].

In this disseration we have mainly focus on GMRES, but other Krylov subspace methods can also be used, such as Biconjugate Gradient stabilized (Bi-CGSTAB) and Induced Dimension Reduction (IDR) [12–14]. Especially for subsequent parallelization strategies, the use of IDR methods could provide more efficiency as the method uses a shorter chain of recurrences [15].

3.2.4. PRECONDITIONERS

In order to accelerate the convergence, preconditioning techniques are available for iterative methods. The study of preconditioning techniques for Krylov subspace methods, such as GMRES, comprises a large part of of numerical analysis and scientific computing. Essentially, a preconditioner matrix M is designed in order to accelerate convergence by requiring that the (spectral) properties of the linear system $M^{-1}Au = M^{-1}f$ are more favourable. For GMRES, a preconditioned variant can be obtained by applying GMRES to the following linear system

$$M^{-1}Au = M^{-1}f \Leftrightarrow AMy = f, u = My,$$
$$A \in \mathbb{C}^{n\times n},\ u, x, f \in \mathbb{C}^n,$$

where M is an invertible matrix in $\mathbb{C}^{n\times n}$. In general, subject to the points discussed about the normality conditions in Section 3.2.3, a matrix M is eligible as a preconditioner if the eigenvalues of $M^{-1}A$ are clustered around $(1,0)$ in the complex plane and $M^{-1}y$ can be obtained at low cost. The algorithm for preconditioned GMRES is given in Algorithm 3. The preconditioning step is indicated in red.

Algorithm 3: Preconditioned GMRES-method $M^{-1}Au = M^{-1}f$

Initialization:
Choose u_0 and compute $r_0 = f - Au_0$, $b_0 = ||r_0||$ and $v_1 = r_0/b_0$
for $j = 1,2,..n$ **do**
  $z_j := Av_j$
  $w_j := M^{-1}z_j$
  for $i = 1,2,..j$ **do**
    $h_{i,j} := (w_j, v_i)$
    $w_j := w_j - h_{i,j}v_i$
  end
  $h_{j+1,j} := \|w\|$
  $v_{j+1} := \frac{w}{h_{j+1,j}}$
end

3.3. PRECONDITIONING FOR THE HELMHOLTZ PROBLEM

Preconditioning for the Helmholtz problem has been studied widely throughout the years. Suitable Krylov subspace methods generally do not perform well without incorporating a preconditioner. Several preconditioners have been tailored for the Helmholtz problem.
An important class is mentioned in [16] and [17], where an incomplete LU factorization of the coefficient matrix A serves as a preconditioner. However, ILU preconditioners are notoriously known to cause fill-in, destroying the original sparsity of the coefficient matrix and can especially become problematic for large wavenumbers.
An alternative has been opted by [18], [19] and [20], where an analytical ILU factorization has been proposed. A drawback of the AILU preconditioner is its applicability to constant wavenumber problems as it diverges for non-constant wavenumber problems.
Finally, a class of preconditioners has been constructed which focuses on the operator in question. In [21] the preconditioner matrix M is equal to the discretizized Laplacian operator, which is equivalent to letting $k = 0$. [22] have further developed this class by including

a positive real shift.
For large wavenumbers it seems that the most effective and robust results can be achieved by combining a real and complex shift in the Laplacian operator based preconditioner. [23] and [24] have first examined the behavior of the Complex shifted Laplacian (CSL) preconditioner for the Helmholtz equation, which is still considered the industry standard. However, despite achieving a substantial speed-up, small eigenvalues of the preconditioned system rush to zero for the Helmholtz problem as the wavenumber increases instead of remaining clustered near the point $(1,0)$ in the complex plane.
A different approach, which is mainly used to solve the problem in parallel, can be found by using preconditioning techniques based on domain decomposition methods. This branch of preconditioners has recently received a lot of attention and is largely based on the work in [25]. These methods split the computational domain into subdomains and solve a local sub problem using a direct method. However an iterative method could also be used at the level of the subdomains [26].
Balancing between wavenumber independent convergence and practical constraints created the opportunity to consider a deflation strategy, which lies at the basis of the methods developed in this thesis (see Chapter 7 and Chapter 8). Its use for time-harmonic wave problems was first proposed in[27]. Deflation, in essence, aims to move the unwanted eigenvalues to zero or one and has been studied widely, see ([28], [29],[30]).

3.3.1. CSL PRECONDITIONER

Let A be the resulting coefficient matrix after discretization. Recall that we can write A in terms of the discrete Laplacian operator $-\Delta$ and the $n \times n$ identity matrix I: as:

$$A = -\Delta - k^2 I, A, \in \mathbb{C}^{n\times n} \tag{3.17}$$

The CSL preconditioner is accordingly defined as

$$M = -\Delta - (\beta_1 + i\beta_2)k^2 I,\ A, \in \mathbb{C}^{n\times n},\ \beta_1, \beta_2 \in [0,1] \tag{3.18}$$

where i denotes the imaginary unit and β_1 and β_2 represent the real and complex shift respectively. Initially, the coefficient matrix A is an indefinite real symmetric matrix in the absence of Sommerfeld radiation conditions. For the sake of brevity, we introduce a notation for the preconditioned linear system $\widehat{A}x = M^{-1}Ax = \widehat{b} = M^{-1}b$.
The preconditioned system has a convenient way of relating the eigenvalues of the matrix A to the eigenvalues of the transformed system $\widehat{A}$ given that A and M^{-1} commute

$$\begin{aligned}\widehat{A} &= M^{-1}A \\ &= M^{-1}(M + (\beta_1 + i\beta_2 - 1)k^2 I) \\ &= I + (\beta_1 + i\beta_2 - 1)k^2 M^{-1} \\ &= (M + (\beta_1 + i\beta_2 - 1)k^2 I)M^{-1} = AM^{-1}\end{aligned} \tag{3.19}$$

As it has been pointed out in the previous chapter, the use of homogeneous Dirichlet conditions leads to a normal and symmetric matrix. This implies that both A and M share an orthonormal basis of eigenvectors, the eigenvalues of the preconditioned system $\widehat{A}$ are

given by

$$\lambda_{\widehat{A}} = \lambda_{M^{-1}A} = \lambda_{M^{-1}}\lambda_A = \frac{\lambda_A}{\lambda_M}. \tag{3.20}$$

The eigenvalues for the discretized Helmholtz operator are given in Section 2.1.5.1. In the case homogeneous Dirichlet conditions are used, the preconditioned system shares the same orthonormal eigenvectors as the original coefficient matrix A and we obtain an elegant expression for the eigenvalues of the preconditioned system as well. Thus in 1D, for the continuous operator we have

$$\lambda^j(\widehat{A}) = \frac{j^2\pi^2 - k^2}{j^2\pi^2 - (\beta_1 + i\beta_2)k^2}, \beta_1, \beta_2 \in [0,1]. \tag{3.21}$$

Note that in case of a zero eigenvalue, the matrix A is singular. The eigenvalues for the discretized Helmholtz operator can be constructed from the eigenvalues of the discretized Laplacian. If we let $\hat{\lambda}_j(L)$ denote the eigenvalues of the Laplacian, then in 1D, the preconditioned system has the following eigenvalues

$$\begin{aligned} \hat{\lambda}^j(L) &= \frac{1}{h^2}(2 - 2\cos(j\pi h)),\ j = 1, 2, \dots n-1, \\ \Rightarrow \lambda^j(M^{-1}A) &= \frac{\hat{\lambda}^j(L) - k^2}{\hat{\lambda}^j(L) - (\beta_1 + i\beta_2)k^2},\ \beta_1, \beta_2 \in [0,1] \end{aligned}$$

3.3.2. OPTIMAL SHIFT

Various options for the shift parameters β_1 and β_2 have been considered, while respecting the condition that $\beta_1, \beta_2 \in [0,1]$. When the real shift parameter β_1 is set to 1 the condition number of the preconditioned coefficient matrix $\widehat{A}$ is minimized [31]. Letting $\beta_1 = 1$ leads to a tight circular distribution of the eigenvalues, remedying the high indefiniteness of the original coefficient matrix A and eventually positively affecting rate of convergence iterative Krylov subspace methods [2].

Unless the shift is kept $\mathcal{O}(k)$ and the preconditioner is inverted exactly, the small eigenvalues of the preconditioned system still rush to zero as the wavenumber increases [32]. In order to properly manage the computational costs, in practice one multigrid iteration is used to obtain an approximation of the inverse, which will be the main focus of Chapter 4. Using rigorous Fourier analysis, it has been shown that the use of multigrid to obtain a cost effective preconditioner came at the price of having to keep the complex shift rather large, i.e. of $\mathcal{O}(k^2)$. A more recent analysis provided a generalization for this claim without having to restrict to Dirichlet boundary conditions [33]. In light of this, [34] have studied the optimal complex shift parameter β_2, affirming that the complex shift parameter can be interpreted as the radius of the circular eigenvalue distribution when β_1 is fixed at 1. However, a word of caution is in place as decreasing the magnitude of β_2 leads to the matrix M resembling the original coefficient matrix A, making the inversion and implementation of the preconditioner redundant. [34] postulate that the optimal shift (β_1, β_2) is obtained by letting $\beta_1 = 1$ and $\beta_2 = 0.5$, causing the real part of the eigenvalues to be bounded below by 0 and above by 1, while allowing the complex part to vary between $-0.5i$ and $0.5i$.

[2] Choosing β_1 any larger than 1 would lead to a more indefinite preconditioner matrix M than the original matrix A.

Figure 3.1: One-dimensional spectrum of CSL preconditioned system in the complex plane with $\beta_2 = 0.5$. Left we have $k = 50$ and right we have $k = 250$. The grid resolution has been set at $kh = 0.625$.

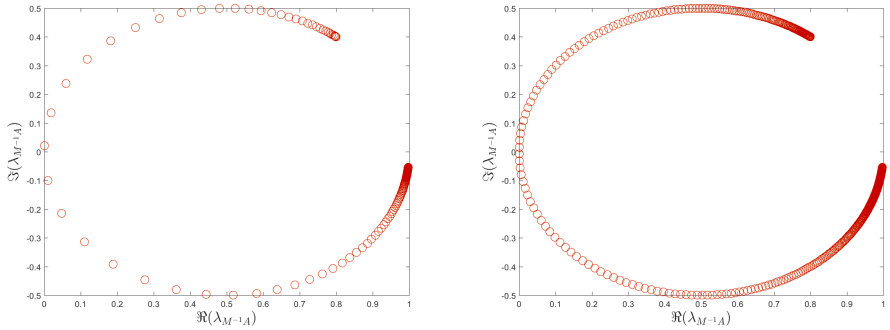
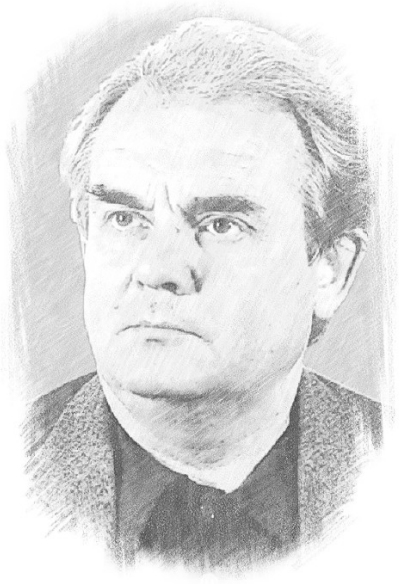


Fig. 3.1 is illustrative of the problem at hand. For increasing k, we observe that small eigenvalues start moving towards the origin. Taking $k = 250$ already shows that the clustering near the origin starts becoming more dominant. When k grows very large, this effect accumulates.

4

MULTIGRID METHODS



This story (creation of multigrid) is related to the history
of the establishment of computational mathematics
where in the era of the first computers,
the history still has to be written.

Radii Petrovich Fedorenko

In this chapter we describe the basic idea behind multigrid methods and how they are used in solving the Helmholtz equation. The first multigrid method was developed during the seventies by Russian mathematician Radii Petrovich Fedorenko. Ever since, the method has become an industry standard in solving a class of partial differential equations.
To illustrate the mathematical properties of the methods, we use the 1D Helmholtz equation accompanied with Dirichlet conditions as an example. The main idea behind the use of different grid refinement levels in multigrid methods was the notion that the low-frequency modes of the iteration error from solving a linear system using Basic Iterative Methods (BIMs) was not being reduced sufficiently. These low-frequency modes are related to the eigenvectors corresponding to the small eigenvalues of the linear system. To understand these low- and high-frequency error components, we consider the linear system obtained from discretisizing a simple one-dimensional Poisson problem

$$Au = f, A \in \mathbb{R}^{n\times n}, u, f \in \mathbb{R}^n.$$

The eigenmodes can be divided into low and high-frequency modes. The low-frequency modes are slowly varying grid vectors that correspond to the small eigenvalues of A. The eigenvectors of the matrix A are

$$v_h^j = \begin{pmatrix} \sin \pi jh \\ \sin \pi j2h \\ \vdots \\ \sin \pi j(n-1)h \end{pmatrix}, \quad 1 \leqslant j \leqslant n-1. \tag{4.1}$$

For now we assume $n-1$ to be even. The eigenvectors are sine-functions applied to the grid vectors $\mathbf{x} = [x_i] = ih$, with $i = 1, 2, \ldots, n-1$. For increasing j, the eigenvectors become more oscillatory. The indices $j = 1$ to $\frac{n}{2} - 1$ therefore adhere to the low-frequency modes, whereas the remaining eigenmodes represent high-frequency modes. By transferring these low-frequency eigenvectors onto a coarse grid, their smooth components become oscillatory and can be reduced.

4.1. TWO-GRID METHOD

The key ingredient of multigrid methods is the use of coarser grids, where smooth components become oscillatory. Note that the eigenvectors of the discretisized 1D Helmholtz operator with Dirichlet conditions coincide with the eigenvectors of the Laplacian given in equation 4.1.

4.1.1. COARSE GRID CORRECTION

We first start by constructing intergrid transfer functions, which will allow us to move from the fine grid with stepsize h to the coarse grid with stepsize $H = 2h$ and vice versa. Using standard linear interpolation, we define the coarse grid vector $u_H = [u_{H_1}, \ldots, u_{H_n}]$ from Ω_H to the fine grid Ω_h such that

$$I_H^h : \Omega_H \to \Omega_h, \quad u_H \to I_H^h u_H \tag{4.2}$$

such that

$$\begin{cases} [u_H]_{i/2} & \text{if } i \text{ is even,} \\ \frac{1}{2}\left([u_H]_{(i-1)/2} + [u_H]_{(i-1)/2}\right) & \text{if } i \text{ is odd,} \end{cases} \quad i = 1,\dots,n-1 \tag{4.3}$$

with matrix representation

$$I_H^h = \frac{1}{2}\begin{bmatrix} 1 & & & & \\ 2 & & & & \\ 1 & 1 & & & \\ & 2 & & & \\ & 1 & & & \\ & & \ddots & 1 & \\ & & & 2 & \\ & & & 1 & \end{bmatrix} \in \mathbb{R}^{n\times(n)/2-1} \tag{4.4}$$

Using the eigenvectors given in equation 4.1, we obtain the following theorem

Theorem 7: Coarse grid eigenvectors

The coarse-grid eigenvectors are mapped by the interpolation operator I_H^h according to

$$I_H^h v_H^j = (c^j)^2 v_h^j - (s^j)^2 v_h^{n-1-j}, \quad j = 1,\dots,\frac{n}{2} - j, \tag{4.5}$$

where we define

$$c^j := \cos\frac{j\pi h}{2}, \quad s^j := \sin\frac{j\pi h}{2}, \quad j = 1,\dots,\frac{n}{2} - 1. \tag{4.6}$$

Proof. The proof is given in [35] ■

As a result, the coarse-grid modes v_H^j are mapped to a linear combination of their fine grid counterparts v_h^j and a complementary mode $v_h^{j'}$, where $j' := n-1-j$. Moreover, we have

$$c^{j'} = s^j \quad s^{j'} = c^j, \quad j = 1,\dots,\frac{n}{2} - 1, \tag{4.7}$$

In order to transfer fine-grid functions to a coarse grid, we define the restriction operator

$$I_h^H : \Omega_h \to \Omega_H, \quad u_h \to I_h^H u_h \tag{4.8}$$

by

$$\left[I_h^H u_h\right]_i = \frac{1}{4}\left([u_h]_{2i-1} + 2[u_h]_{2i} + [u_h]_{2i+1}\right), \quad i = 1,\dots,\frac{n}{2} - 1. \tag{4.9}$$

The associated matrix representation is given by $I_h^H = \frac{1}{2}\left[I_H^h\right]^T$. The following theorem can be proven for I_h^H, see [35], p. 20.

Theorem 8: Fine grid eigenvectors

The fine-grid eigenvectors are mapped by the restriction operator I_h^H according to

$$I_h^H v_h^j = (c^j)^2 v_H^j, \quad j = 1,\dots,\frac{n}{2}-1, \tag{4.10}$$

$$I_h^H v_h^{N+1-j} = -(s^j)^2 v_H^j, \quad j = 1,\dots,\frac{n}{2}-1, \tag{4.11}$$

$$I_h^H v_h^{n+1} = 0. \tag{4.12}$$

Proof. The proof is given in [35]. ■

4

Let u_h be an approximate solution to our model problem. Then the coarse-grid correction of u_h can be obtained by solving the error equation $A_h e_h = f - A_h u_h = r_h$ on the coarse grid. We start by defining a coarse-grid representation A_H of A_h and solve for $A_H^{-1} I_h^H r_h$, where r_h is first restricted to the coarse grid. A_H is more commonly referred to as the *Galerkin Coarsening Matrix*. Note that $A_H^{-1} I_h^H r_h$ approximates the error $e_H = A_h^{-1} r_h$ on Ω_H. As a last step, e_h is interpolated to the fine grid by

$$u_h \leftarrow u_h + I_H^h A_H^{-1} I_h^H (b - A_h u_h), \tag{4.13}$$

with the associated *error propagation operator*

$$C := I - I_H^h A_H^{-1} I_h^H A_h. \tag{4.14}$$

We thus get the following recursive relation for the error

$$e_h^{j+1} = C_h^j e_0 \tag{4.15}$$

It has been noted that C spans two invariant subspaces corresponding to the index set $j = 1,\dots,\frac{n}{2}-1$ and $j' = n-1-j$. Recall that the eigenvalues of the 1D Laplacian operator on Ω_h and Ω_H are given by

$$\lambda_h^l = \frac{4}{h^2}\sin^2\frac{j\pi h}{2} - k^2, \quad j = 1,\dots,n-1 \tag{4.16}$$

and

$$\lambda_H^l = \frac{4}{H^2}\sin^2\frac{j\pi H}{2} - k^2, \quad j = 1,\dots,\frac{n}{2}-1, \tag{4.17}$$

Letting span $\left\{ v_h^j, v_h^{j'} \right\}$ denote an invariant subspace, i.e.

$$C\left[v_h^j, v_h^{j'} \right] = \left[v_h^{j'}, v_h^{j'} \right] C^j, \quad j = 1,\dots,\frac{n}{2}-1, \tag{4.18}$$

$$C v_h^{n/2} = v_h^{n/2}, \tag{4.19}$$

we can write C from equation 4.14 as follows

$$C^j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} (c^j)^2 \\ -(s^j)^2 \end{bmatrix} \frac{1}{\lambda_H^j} \left[(c^j)^2 - (s^j)^2 \right] \begin{bmatrix} \lambda_h^j & 0 \\ 0 & \lambda_h^{j'} \end{bmatrix} = \begin{bmatrix} 1-(c^j)^4 \frac{\lambda_h^j}{\lambda_H^j} & (c^j)^2 \frac{\lambda_h^{j'}}{\lambda_H^j} \\ (c^j)^2 (s^j)^2 \frac{\lambda_h^j}{\lambda_H^j} & 1-(s^j)^4 \frac{\lambda_h^{j'}}{\lambda_H^j} \end{bmatrix}. \tag{4.20}$$

Moreover, the following theorem can be proven,

Theorem 9: Coarse grid correction

The eigenvalues of the 2×2 blocks from equation 4.20 representing the coarse grid correction operator are given by

$$\Lambda(C^j) = \left\{ 1 - \frac{(c^j)^4 \lambda_h^j + (s^j)^4 \lambda_h^{j'}}{\lambda_H^j}, 1 \right\}, j = 1, \ldots, \frac{n}{2} - 1 \tag{4.21}$$

with eigenvectors

$$w^{j,1} = \begin{bmatrix} (c^j)^2 \\ -(s^j)^2 \end{bmatrix} \text{ and } w^{j,2} = \frac{4}{h^2} \begin{bmatrix} (s^j)^2 \left((c^j)^2 - \frac{hk}{2}^2 \right) \\ (c^j)^2 \left((s^j)^2 - \frac{hk}{2}^2 \right) \end{bmatrix}. \tag{4.22}$$

Proof. The proof is given in [35]. ∎

If k is zero, we obtain the discrete 1D Laplace operator and the expressions from equation 4.16, 4.17 and 4.20 simplify to

$$\frac{\lambda_h^j}{\lambda_H^j} = \frac{4(s^j)^2}{(2s^j c^j)^2} = \frac{1}{(c^j)^2} \quad \text{as well as} \quad \frac{\lambda_h^{j'}}{\lambda_H^j} = \frac{4(cs^j)^2}{(2s^j c^j)^2} = \frac{1}{(s^j)^2}, \quad j = 1, \ldots, \frac{n}{2} - 1, \tag{4.23}$$

and therefore

$$C^j = \begin{bmatrix} 1 - (c^j)^2 & (c^j)^2 \\ (s^j)^2 & 1 - (s^j)^2 \end{bmatrix} = \begin{bmatrix} (s^j)^2 & (c^j)^2 \\ (s^j)^2 & (c^j)^2 \end{bmatrix}, \quad j = 1, \ldots, \frac{n}{2} - 1. \tag{4.24}$$

For $k = 0$, the operator C is an orthogonal projection and has only two eigenvalues 0 and 1. Also, the eigenvectors corresponding to the 0 and 1 block respectively are

$$w^{j,1} = \begin{bmatrix} (c^j)^2 \\ -(s^j)^2 \end{bmatrix}, w^{j,2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{4.25}$$

For small j, $w^{j,1}$ reduces to approximately $[0,1]^T$ since $(c^j)^2 \approx 1$ and $(s^j)^2 \approx 0$. As such, the eigenmode $w^{j,1}$ eliminated by the coarse grid correction is closely aligned with the low-frequency eigenmode v_h^j. This alignment becomes less as j increases.
In case the wavenumber $k > 0$ is positive, the unit eigenvalues of C remain, but the zero eigenvalue starts to shift. As a result, low-frequency modes corresponding to small eigenvalues of the Helmholtz operator may be partially unaffected by the coarse grid correction.

ERROR PROPAGATION

We show this by means of an example. Suppose the small eigenvalues of C corresponding to the eigenvectors in the low-frequency range are of order ε, with $0 < \varepsilon << 1$. Thus, for a lower index j up to some index j^*, where $j \leqslant j^* \leqslant \frac{n}{2} - 1$, we assume that the eigenmodes

$w^{j,1}$ corresponding to the zero eigenvalue of the operator C^j and the low-frequency modes v_h^j are closely aligned. From equation 4.15 we know that the error propagates as follows

$$e_h^{j+1} = C^j e_0. \tag{4.26}$$

If we decompose the initial error e_0 in terms of the eigenvectors of A_h we obtain

$$e_{h_0} = \begin{bmatrix} \gamma^j v_h^j \\ \gamma^{j'} v_h^{j'} \end{bmatrix}^T \text{ for } j = 1, \ldots, j^*,$$

where γ^j corresponds to suitable coefficients for the low-frequency range and $\gamma^{j'}$ represents the coefficients with respect to the high-frequency range. Similarly, for e_h^{j+1} we write

$$e_h^{j+1} = \begin{bmatrix} \widehat{\gamma^j} v_h^j \\ \widehat{\gamma^{j'}} v_h^{j'} \end{bmatrix}^T \text{ for } j = 1, \ldots, j^*.$$

Applying the coarse-grid error propagation matrix C^j according to equation 4.26 thus gives

$$\begin{bmatrix} \widehat{\gamma^j} v_h^j \\ \widehat{\gamma^{j'}} v_h^{j'} \end{bmatrix}^T = C^j \begin{bmatrix} \gamma^j v_h^j \\ \gamma^{j'} v_h^{j'} \end{bmatrix}^T \tag{4.27}$$

Multiplying by $w^{j,1} = [1,0]^T$ on both sides gives

$$e_h^{j+1} w^{j,1} = \begin{bmatrix} \widehat{\gamma^j} v_h^j \\ \widehat{\gamma^{j'}} v_h^{j'} \end{bmatrix}^T \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \widehat{\gamma^j} v_h^j.$$

The left hand side of equation 4.27 becomes

$$C_j \begin{bmatrix} \gamma^j v_h^j \\ \gamma^{j'} v_h^{j'} \end{bmatrix}^T \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Using that $w^{j,1} = [1,0]^T$ is an eigenvector of C^j corresponding to $j = 1$ up to $j = j^*$ we can rewrite the expressions into

$$e_h^{j+1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \widehat{\gamma^j} v_h^j = C^j \begin{bmatrix} \gamma^j v_h^j \\ \gamma^{j'} v_h^{j'} \end{bmatrix}^T \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \gamma^j v_h^j \\ \gamma^{j'} v_h^{j'} \end{bmatrix}^T \begin{bmatrix} \varepsilon \\ 0 \end{bmatrix} = \gamma^j \varepsilon v_h^j$$

Thus, if ε is not equal to zero, the initial low-frequency modes for $j = 1$ to $j = j^*$, do not get removed and propagate further as the error develops. In fact, for $k > 0$ some of these low-frequency modes, instead of being projected onto zero, become amplified and do not partake in the error smoothing process.

4.1.2. SMOOTHING

So far, we have focused on the low-frequency error components. The coarse grid correction operator generally takes care of these modes. In order to reduce the high-frequency error components, the coarse grid correction scheme is often complemented with a *smoother*. Basic iterative methods such as weighted Jacobi or Gauss-Seidel are often used as smoothers. While these smoothers work efficiently for the Laplace equation, several problems arise in case of the Helmholtz equation. If we take the general form of the weighted Jacobi scheme, the iteration matrix is given by

$$S_{JAC} = I - \omega D^{-1} A, \tag{4.28}$$

where $\omega \in \mathbb{R}$ and D is the diagonal matrix containing the diagonal of the matrix A. Note that we can represent ωD^{-1} as

$$\frac{1}{\omega} D = \frac{(2 - k^2 h^2)}{\omega h^2} I. \tag{4.29}$$

As A and D are simulaneously diagonalizable, the eigenvalues for $j = 1, 2, \ldots, n-1$ of Eq. (4.28) are given by

$$\lambda^j_{S_{JAC}} = 1 - \omega \frac{h^2 \lambda^j_A}{2 - k^2 h^2} \tag{4.30}$$

Given that A is indefinite, we know that the eigenvalues are both negative and positive. As the Jacobi scheme is a stationary iterative method, it converges if $\rho(S_{JAC}) < 1$, where denotes the spectral radius, which implies that the largest eigenvalue in magnitude should be strictly less than 1. However, for $\omega > 0$ and $k^2 h^2 < \sqrt{2}$, this can only be satisfied if each eigenvalue λ^j_A in Eq. (4.30) is positive. Hence, there is no ω which can satisfy

$$\rho(S_{JAC}) = \max \left\{ \left| 1 - \omega \frac{h^2 \lambda_j(A)}{2 - k^2 h^2} \right|, \forall j = 1, 2, \ldots n \right\} < 1, \tag{4.31}$$

simultaneously for the positive and negative eigenvalues of A.

TWO-GRID ITERATION

The combination of the coarse grid correction and the smoother gives us the iteration operator B of the two grid cycle. If for simplicity we consider pre-smoothing only, then the two-grid cycle can be represented by a fixed point iterative method with

$$B^H_h = S^{\nu_1} K^H_h, \text{ where } K^H_h = I_h - I^h_H A^{-1}_H I^H_h A_h \tag{4.32}$$

as its iteration matrix. Here, I^h_H and I^H_h denote the interpolation and restriction operator respectively, A_H is the coarse grid linear system and S is the smoothing operator.

4.1.3. ALGORITHM

We arrive at the full two-grid method, which is also called a *two-grid cycle* by combining the coarse grid correction scheme with a smoothing scheme. The algorithm is given in Algorithm 4, where ν_1 and ν_2 denote the number of pre- and post-smoothing steps respectively.

Algorithm 4: Two-grid cycle: $u_h^{(k+1)} = TG(u_h^k, A_h, f, \nu_1, \nu_2)$

$\tilde{u}_h^k = S^{\nu_1}\left(u_h^k, A_h, f\right)$	▷ Pre-smoothing
$r_H = Rr_h = R\left(f - A_h\tilde{u}_h^k\right)$	▷ Restrict residual
$e_H = A_H^{-1} r_H$	▷ Direct solve on Ω_H
$\tilde{u}_h^k = \tilde{u}_h^k + \tilde{e}_h$	▷ Prolong residual
$u_h^k = S^{\nu_2}\left(\tilde{u}_h^k, A_h, f\right)$	▷ Post-smoothing

In Algorithm 4, S respresents the smoothing operator. To obtain a better overview of how the two-grid cycle works, a schematic representation is given in Section 4.1.3. Note that the cycle depicted represents one two-grid cycle iteration.

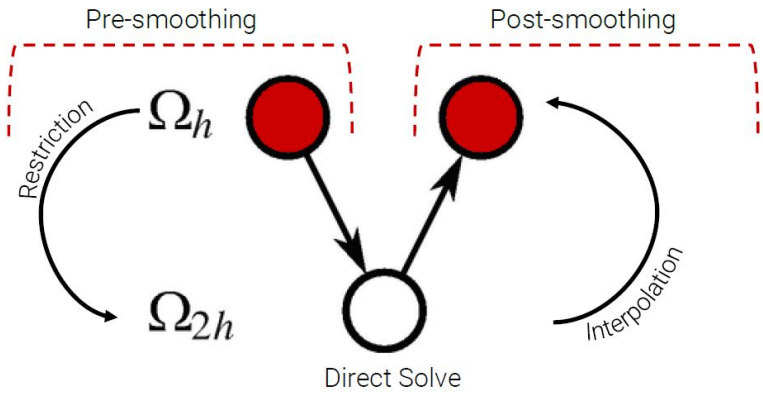


Figure 4.1: Schematic overview of the two-cycle multigrid algorithm.

4.1.4. CONVERGENCE

The two grid cycle converges if $\rho(B_h^H) < 1$. In fact the smaller the spectral radius, the faster the convergence. We extend the analysis of the eigenvalues from Section 4.1.1 to now include the effect of the smoother. Using the same ordering of the orthonormal basis, we

again obtain 2×2 blocks for the eigenvalues of B_h^H.

$$C^j = \begin{bmatrix} 1-\omega h^2\frac{\lambda_h^j}{2-k^2h^2} & 0 \\ 0 & 1-\omega h^2\frac{\lambda_h^{j'}}{2-k^2h^2} \end{bmatrix}^{\nu_1} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} (c^j)^2 \\ -(s^j)^2 \end{bmatrix} \frac{1}{\lambda_H^j} \left[(c^j)^2 - (s^j)^2 \right] \begin{bmatrix} \lambda_h^j & 0 \\ 0 & \lambda_h^{j'} \end{bmatrix} \right), \tag{4.33}$$

$$= \begin{bmatrix} 1-\omega h^2\frac{\lambda_h^j}{2-k^2h^2} & 0 \\ 0 & 1-\omega h^2\frac{\lambda_h^{j'}}{2-k^2h^2} \end{bmatrix}^{\nu_1} \begin{bmatrix} 1-(c^j)^4\frac{\lambda_h^j}{\lambda_H^j} & (c^j)^2(s^j)^2\frac{\lambda_h^{j'}}{\lambda_H^j} \\ (c^j)^2(s^j)^2\frac{\lambda_h^j}{\lambda_H^j} & 1-(s^j)^4\frac{\lambda_h^{j'}}{\lambda_H^j} \end{bmatrix}, \tag{4.34}$$

$$= \left(\frac{1-\omega h^2}{2-k^2h^2}\right)^{\nu_1} \begin{bmatrix} \left(\lambda_h^j\right)^{\nu_1}\left(1-(c^j)^4\frac{\lambda_h^j}{\lambda_H^j}\right) & \left(\lambda_h^j\right)^{\nu_1}(c^j)^2(s^j)^2\frac{\lambda_h^{j'}}{\lambda_H^j} \\ \left(\lambda_h^{j'}\right)^{\nu_1}(c^j)^2(s^j)^2\frac{\lambda_h^j}{\lambda_H^j} & \left(\lambda_h^{j'}\right)^{\nu_1}\left(1-(s^j)^4\frac{\lambda_h^{j'}}{\lambda_H^j}\right) \end{bmatrix}, \text{ with} \tag{4.35}$$

$$c^j := \cos\frac{j\pi h}{2}, \quad s^j := \sin\frac{j\pi h}{2}, \quad j = 1,\ldots,\frac{n}{2}-1. \tag{4.36}$$

In this case the non-unit eigenvalues are given by the trace of each respective block in Eq. (4.20). Table 4.1 contains the spectral radius for various k using only pre-smoothing.

Table 4.1: Spectral radius $\rho(B_h^H)$ of the two-grid operator for the 1D Helmholtz equation using $h = 2^{-5}$ using Eq. (4.36). ν denotes both the number of pre- and post-smoothing steps.

$\nu\backslash k$	$k=0$	$k=1.3\pi$	$k=4.3\pi$	$k=3.6\pi$
1	0.333	0.336	0.409	0.885
2	0.111	0.117	0.239	1.853
3	0.079	0.077	0.262	1.645
4	0.062	0.061	0.248	1.634
5	0.051	0.049	0.256	1.583

From Table 4.1 we observe that already for $k > 3.6\pi$ the two-grid cycle diverges. Consequently, the two-grid cycle is unsuitable as a solver for the Helmholtz equation, but works efficiently for Poisson type problems.

4.2. MULTIGRID

The extension from a two-grid method to a multigrid method is fairly straightforward. The first step is to extend the two grids to a collection of coarser grids. In practice this translates to coarsening until we reach only one grid point. Apart from Ω_h which is the finest grid, we construct

$$\Omega_m^H = \left\{ (x_i) | x_i = i(2^{m-1}H) = i2^m h, h = n^{-1}, 1 \leqslant i \leqslant n-1 \right\},$$

where m denotes an integer representing the total number of levels or grids needed. So for example, if we need m levels or grids including the finest one, we obtain the set

$$\Omega = \left\{ \Omega_l^H | l = 1,2,\ldots m-1 \right\} \bigcup \Omega_0^h,$$

where $\Omega_0^h = \left\{(x_i)|x_i = ih, h = n^{-1}, 1 \leqslant i \leqslant n-1\right\}$. Apart from the two-grid interpolation I_h^H operator, we define

$$I_m^{m+1} : \Omega_{m+1}^H \to \Omega_m^H, \quad u_H^{m+1} \to I_m^{m+1} u_H^{m+1}, \tag{4.37}$$

such that we construct a collection of Galerkin coarsened matrices. We previously had A and A_H for the two-grid cycle. We now construct a family of Galerkin matrices $E_1^H, E_2^H, \ldots E_m^H$, where we define $E_m^H = I_{m+1}^m E_{m-1}^H I_m^{m+1}$, with $E_0^H = A$ such that we obtain

$$E = \left\{E_l^H | l = 1, 2, \ldots m-1\right\}. \tag{4.38}$$

The next step is to apply the two-grid cycle recursively. In general convergence of the multigrid method is regarded to be feasible if the spectral radius of the two-grid method is less than 1 [36].

4.2.1. ALGORITHM

The multigrid algorithm is given in Algorithm 5. Observe that line number 3 in Algorithm 4, is now replaced with a recursive application of the two-grid cycle. Instead of solving the equation $e_H = A_H^{-1} r_H$ directly on Ω_1^H, we apply the two-grid cycle and repeat the steps until we reach the coarsest grid. There (see line 2), a direct solve is performed and the solution is prolonged back onto the fine grid.

Algorithm 5: Multigrid V-cycle: $u_h^{(k+1)} = TG(u_h^k, A_h, f, \nu_1, \nu_2, m)$

if *coarse level* **then**
  Solve $A_h u_h = f$ ▷ Direct solve on Ω_m
else
  $\tilde{u}_h^k = S^{\nu_1}\left(u_h^k, A_h, f\right)$ ▷ Pre-smoothing
  $r_H = R r_h = R\left(f - A_h \tilde{u}_h^k\right)$ ▷ Restrict residual
  $e_l^H = TG\left(0, EA_l^H, r_l^H - E_l^H e_{l-1}^H \nu_1, \nu_2, m\right), l = 1, 2, \ldots, m$ ▷ Recursion
  $\tilde{u}_h^k = \tilde{u}_h^k + P e_m^h$ ▷ Prolong and coarse grid correction
  $u_h^k = S^{\nu_2}\left(\tilde{u}_h^k, A_h, f\right)$ ▷ Post-smoothing
end
end

In Algorithm 5, R and P stand for the two-grid restriction and prolongation operators moving between levels $l-1$ and l respectively.

4.3. MULTIGRID PRECONDITIONING

We have established that multigrid as a stand-alone solver for Helmholtz problems diverges. However, multigrid can still be used as a preconditioner, especially when we include a complex shift like in the CSL. Table 4.2 contains the two-grid spectral radius when we use the CSL matrix M instead of the linear system corresponding to the Helmholtz equation.

Table 4.2: Spectral radius $\rho(B_h^H)$ of the two-grid operator using only pre-smoothing for the 1D Complex Shifted Laplacian with shifts $(\beta_1, \beta_2) = (1, 0.5)$ and $h = 2^{-5}$. ν denotes both the number of pre- and post-smoothing steps. Weighted Jacobi is used as a smoother with $\omega = \frac{2}{3}$.

$\nu \backslash \rho(B_h^H)$	$k = 0$	$k = 1.3\pi$	$k = 4.3\pi$	$k = 6.3\pi$
1	0.333	0.497	0.486	0.485
2	0.111	0.422	0.441	0.478
3	0.079	0.337	0.350	0.385
4	0.062	0.295	0.319	0.376
5	0.051	0.259	0.289	0.364

The results show that the scheme is convergent for all values of k as the spectral radius is always strictly less than 1. Because the scheme is convergent, we can extend the two-grid method to a multigrid method for the CSL.

4.3.1. ALGORITHM

In Section 3.3.1 we showed that if we want to solve the original Helmholtz equation, we can use CSL as a preconditioner. Unlike multigrid as a stand-alone solver, we incorporate the multigrid method as a preconditioning step in order to avoid computing M^{-1} exactly. We therefore need to combine it with an iterative method, such as GMRES. By doing so, we obtain an approximation to the exact inverse of M. In the preconditioned GMRES algorithm, which we again state below, this boils down to solving the equation $Mw_j = z_j$, with $z_j = Av_j$ with a few multigrid iterations.

Algorithm 6: Preconditioned GMRES-method $M^{-1}Au = M^{-1}f$

Initialization:
Choose u_0 and compute $r_0 = f - Au_0$, $b_0 = ||r_0||$ and $v_1 = r_0/b_0$
for $j = 1, 2, ..n$ **do**
  $z_j := Av_j$
  $w_j := M^{-1}z_j$ ▷ Solve $Mw_j = z_j$ with multigrid
  for $i := 1, 2, \ldots, j$ **do**
    $h_{i,j} := (w_j, v_i)$
    $w_j := w_j - h_{i,j}v_i$
  end
  $h_{j+1,j} := \|w\|$
  $v_{j+1} := \frac{w}{h_{j+1,j}}$
end

4.3.2. CONVERGENCE

The inclusion of a preconditioner in the algorithm is considered in order to accelerate convergence of an iterative method, such as GMRES. To illustrate this, Table 4.3 contains the number of iterations to reach convergence for both GMRES with and without CSL preconditioning.

Table 4.3: Number of iterations to reach convergence with (L) and without (R) CSL for the 1D Helmholtz equation. The shift has been set $\beta_1 = 1$ and $\beta_2 = 1$. The iteration is stopped once the relative residual has reached a tolerance of 10^{-6}. With CSL (L) uses one multigrid V-cycle to approximate the inverse, with one pre- and post-smoothing step. Weighted Jacobi is used as a smoother with $\omega = 1.5$.

k	w CSL	w/o CSL
10	9	9
50	24	41
100	38	81
200	66	161
400	118	321

We can make several observations. First of all, note that GMRES without CSL converges in exactly $\frac{n}{2}$ iterations, which will be way too high in 2D and 3D applications.
For GMRES with CSL, the number of iterations grows linearly with the wavenumber k. The number of iterations with the CSL is significantly lower, which is more beneficial for GMRES. GMRES becomes more and more expensive as the number of iterations increases. With each added iteration, the storage and orthogonalization costs accumulate. Unlike other Krylov subspace methods, GMRES has better optimality conditions at the expense of having long recurrences, which is why a lower iteration count is preferred. This effect is even more pronounced once we move to 2D and 3D problems.
In Section 3.3.1 we inspected the spectrum of the CSL preconditioned system and observed that as the wavenumber k increases, the eigenvalues of $M^{-1}A$ with exact inversion start moving towards the origin. The deteriorating GMRES convergence can be ascribed to this aspect, granted that for normal matrices, GMRES convergence is predominantly governed by the clustering of the eigenvalues. Improved convergence can be anticipated if the eigenvalues are close to $(1,0)$ in the complex plain. Evidently, if we use the exact inverse of the CSL combined with a small complex shift, the spectrum stays clustered near this point due to the closer resemblance of M and A [32].

II

Numerical Modelling and Accuracy

5

Pollution Error

Parts of this chapter have been published in Journal of Computational and Applied Mathematics **395**, (2021) [37].

In Chapter 2 we have discussed the process of discretisizing the Helmholtz equation. In the absence of any numerical errors, the waves modelled by the Helmholtz equation will propagate without any dissipation or dispersion. However, as mentioned previously, shifting from the continuous problem to its discrete counterpart, gives rise to the *pollution error*. In essence, the pollution effect is directly related to numerical dispersion errors due to differences between the actual and numerical wavenumber [38–41]. This error grows with the wavenumber as in the high-frequency range the solutions become very oscillatory.
As a result of this discrepancy, there may be large errors between the actual solution and the obtained numerical solution. Therefore, the solution obtained using fast and efficient solvers, may therefore still be inaccurate. The fact that the pollution effect for finite element and finite difference methods can not be avoided in higher-dimensions adds to the problem [39]. No simple solution exists, as it has been shown that for a certain accuracy, the number of grid points needed to retain that accuracy grows along with the wavenumber. However, it grows slower than the order of accuracy of the schemes. In particular, if we let k denote the wavenumber, n the problem size in one-dimension and p the order of a finite difference or finite element scheme, then

5

$$n = Ck^{\left(\frac{p+1}{p}\right)},$$

where C is a constant that only depends on the accuracy achieved [42]. Therefore, if we wish to increase k while keeping the accuracy of the same order, we need to increase n as well, which leads to larger linear systems.

In this chapter we derive the analytical solution to our model problem and derive the bounds which reveal the pollution error. We also provide an overview of studies dealing with the pollution error so far.

5.1. PROBLEM DEFINITION

In this section we start by defining two model problems. Following a similar approach in the literature, we use the constant wavenumber model with Dirichlet conditions, such that the analytical solution and eigenvalues can be derived [43–51]. We therefore start by focusing on a 1D model problem, which we denote by MP 1.

MP 1

$$-\frac{d^2u}{dx^2} - k^2\,u = \delta(x-x'), x \in \Omega = [0,L] \subset \mathbb{R}, \qquad (5.1)$$
$$u(0) = 0, u(L) = 0,\ k \in \mathbb{R}\backslash\{0\}.$$

Working on the unit-domain $(L = 1)$, the second order difference scheme with step-size $h = \frac{1}{n}$ leads to

$$\frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} - k^2 u_j = f_j, j = 1,2,3,\ldots,n,\ x_j = jh.$$

Using a lexicographic ordering, the linear system can be formulated exclusively on the internal grid points due to the homogeneous Dirichlet boundary conditions. We obtain the

following system and eigenvalues

$$Au = \frac{1}{h^2}\text{tridiag}[-1\ 2-k^2h^2\ -1]u = f,$$
$$\hat{\lambda}^j = \frac{1}{h^2}(2-2\cos(j\pi h)) - k^2,\ j = 1,2,\dots n. \tag{5.2}$$

In order to investigate the pollution error in higher dimensions, we define MP 2 to be the 2D version of the original model problem on the standard 2D square unit domain $\Omega = [0,1] \times [0,1]$ with constant wavenumber k.

MP 2

$$-\Delta u(x,y) - k^2 u(x,y) = \delta(x-\frac{1}{2}, y-\frac{1}{2}),\ (x,y) \in \Omega \backslash \partial\Omega \subset \mathbb{R}^2, \tag{5.3}$$
$$u(x,y) = 0,\ (x,y) \in \partial\Omega,$$

Next to find the error bound, we need an expression for the analytical solution. Given that we have used Dirichlet boundary conditions, these can be derived in the section below.

5.2. ANALYTICAL SOLUTION

We can express the exact solution to MP 1 in terms of the Green's function $G(x,x')$ given that this contains the eigenvalues. We need to use the Green's function given that we are working with the non-homogeneous Helmholtz equation. We therefore seek a solution of the form

$$u(x) = \int_0^L G(x,x')f(x)dx', \tag{5.4}$$

where the Green's function satisfies

$$\left(\frac{d^2}{dx^2} - k^2\right) G(x,x') = \delta(x-x').$$

To obtain the Green's function, we need to rewrite the differential operator from MP 1 in the Sturm-Liouville form [52]. Let $\mathscr{L}(x)$ be the general Sturm-Liouville operator

$$\mathscr{L}(x) = \frac{d}{dx}\left[p(x)\frac{d}{dx}\right] + q(x) \tag{5.5}$$

Setting $p(x) = -1$ and $q(x) = -k^2$, we obtain the Sturm-Liouville operator for the Helmholtz boundary value problem, which we denote by $\mathscr{L}(x)$. Using the Sturm-Liouville operator for the Helmholtz problem, we can rewrite the problem as

$$\mathscr{L}(x)u(x) = f(x).$$

The related eigenvalue problem is

$$\mathscr{L}(x)u(x) = \lambda u(x).$$

Using the eigenfunction expansion, we can rewrite MP 1 (5.1) as

$$\left(\frac{d^2}{dx^2}+\lambda^j\right)u_j(x)=0,$$
$$u_j(0)=u_j(L)=0.$$

Normalizing with a factor $\sqrt{\frac{2}{L}}$ gives the following solution

$$u_j(x)=\sqrt{\frac{2}{L}}\sin\left(\frac{j\pi x}{L}\right) \text{ with } \lambda^j=\left(\frac{j\pi}{L}\right)^2,\ j=1,2,3,\ldots.$$

Integrating over the eigenfunctions for the eigenvalue problem gives

$$\frac{2}{L}\int_0^L \sin\left(\frac{j\pi x}{L}\right)\sin\left(\frac{i\pi x}{L}\right)dx=\delta_{ij}. \tag{5.6}$$

The Green's function for equation 5.6 is given by

$$G(x,x')=\frac{2}{L}\sum_{j=1}^{\infty}\sum_{i=1}^{\infty}\frac{\sin\left(\frac{j\pi x}{L}\right)\sin\left(\frac{i\pi x'}{L}\right)}{\lambda^j},k^2\neq j^2\pi^2,j=1,2,3,\ldots. \tag{5.7}$$

Consequently on the unit interval, $G(x,x')$ satisfies

$$\mathscr{L}(x)G(x,x')=\delta(x-x'),x\in\Omega=[0,1]\subset\mathbb{R}, \tag{5.8}$$
$$G(0,x')=G(1,x')=0,\ x\in\partial\Omega.$$

In the event that $k^2=j^2\pi^2$, the eigenfunction expansion would become defective as this would imply resonance and unbounded oscillations in the absence of dissipation. Therefore, we explicitly need to warrant for the latter case and impose the extra condition $k^2\neq j^2\pi^2$ asserting that our Green's function exists.
Equation 5.7 immediately provides us with an expression for the analytical eigenvalues. It is apparent that within the bounded domain $[0,1]$ there are an infinite number of eigenpairs. We employ this expression for the eigenvalues in upcoming sections, where we compare them with the numerical eigenvalues for the linear system of equations. We have expressed the exact solution to MP 1 as an eigenfunction expansion using Green's function. A similar approach will allow us to obtain the exact solution for the 2D MP 2, which is given by

$$u(x,y)=\int_\Omega f(x,y)G(x,y,x',y')dx'dy', \tag{5.9}$$
$$=\int_\Omega \delta(x-x',y-x')G(x,y,x',y')dx' \tag{5.10}$$
$$=G(x,y,x',y'). \tag{5.11}$$

The Green's function $G(x,y,x',y')$ on the unit square becomes

$$G(x,y,x',y')=\frac{4}{L}\sum_{j=1}^{\infty}\sum_{j=1}^{\infty}\frac{\sin\left(\frac{j\pi x}{L}\right)\sin\left(\frac{j\pi x'}{L}\right)\sin\left(\frac{j\pi y}{L}\right)\sin\left(\frac{j\pi y'}{L}\right)}{\frac{i^2\pi^2+j^2\pi^2}{L^2}-k^2}, \tag{5.12}$$
$$k^2\neq i^2\pi^2+j^2\pi^2,i,j=1,2,3,\ldots.$$

and satisfies

$$\begin{aligned}
&\mathscr{L}(x,y)G(x,y,x',y') = \delta(x-x',y-y')\\
&G(x,0,x',y') = G(x,1,x',y') = 0,\ y\in\partial\Omega\\
&G(0,y,x',y') = G(1,y,x',y') = 0,\ x\in\partial\Omega\\
&(x,y)\in\Omega = [0,1]\times[0,1]\subset\mathbb{R}^2,
\end{aligned} \tag{5.13}$$

where $\mathscr{L}(x,y)$ is the 2D Sturm-Liouville operator corresponding to the Helmholtz equation from MP 2.

5.3. ERROR BOUNDS

We now briefly explain the classical error bound for the pollution error. It was mentioned, that in order to keep the pollution error at bay, the grid should be refined such that $k^3h^2 < 1$ [38, 53]. Such a severe restriction on the step-size is necessary, as the accuracy of the numerical solution deteriorates rapidly if the wavenumber increases. In fact, the numerical wave has dispersive properties, which are not present in the analytical wave. Consequently, a phase shift occurs which forms the primary source of error in the pollution term. Thus, in the case FEM and FDM solutions, a phase lag between the computed and the exact wave is directly related to the dispersive character of the discrete medium (i.e. the computed wave does not propagate at the speed of sound), which causes a difference between the exact and numerical wavenumber. This effect accumulates into the pollution term as k increases.

5.3.1. NUMERICAL DISPERSION

To understand how the pollution error depends on the numerical dispersion and consequently on the wavenumber k, note that the dimensionless wavenumber is represented by

$$k = \frac{2\pi f}{\lambda},$$

where $2\pi f$ denotes the angular frequency and λ denotes the phase velocity. Discretisizing the 1D Helmholtz equation leads to

$$-\left(\frac{u_{j+1}-2u_j+u_{j-1}}{h^2}\right) - k^2u_j = 0. \tag{5.14}$$

Moreover, a general continuous solution is given by

$$u(x) = e^{ikx}. \tag{5.15}$$

Evaluation of expression 5.15 in the discrete points gives

$$u_j = e^{i\tilde{k}x_j}. \tag{5.16}$$

Here i denotes the imaginary unit and $\tilde{k}$ represents the perturbed wavenumber due to having a velocity which is different from the speed of sound. Substituting expression 5.16 into 5.14 results in

$$-u_{j+1}+2u_j-u_{j-1} = e^{i\tilde{k}x_j}\left(-e^{i\tilde{k}h}+2-e^{-i\tilde{k}h}\right) = 2\left(\cos\left(\tilde{k}h\right)-1\right)e^{i\tilde{k}x_j}. \tag{5.17}$$

Equation 5.17 is a good approximation of the exact solution if $\tilde{k}$ solves

$$\frac{2\left(\cos\left(\tilde{k}h\right)-1\right)}{h^2}-k^2=0. \tag{5.18}$$

Applying Taylor's expansion on the cosine term and substituting into equation 5.18 gives

$$k-\tilde{k}=\mathcal{O}\left(k^2h^2\right)$$

The a priori error estimation due to $\left|\tilde{k}-k\right|\neq 0$ becomes

$$\text{error}_{\text{pollution}}=\left|e^{ikx_j}-e^{i\tilde{k}x_j}\right|=\left|1-e^{i(\tilde{k}-k)x_j}\right|\leqslant Ck\left|\tilde{k}-k\right|\leqslant Ck^3h^2. \tag{5.19}$$

The factor Ck^3h^2 can be decomposed as follows. $\mathcal{O}(k^2h^2)$ provides the error in the numerical wave speed for a wave travelling one period. The extra factor k is called the *pollution error* and corrects the total pollution error by scaling the error over one wavelength by the total number of wavelengths travelled over the entire numerical domain [41, 53].
In [38] it is noted that the error given in equation 5.19 mainly relates to the dispersion caused by the differing wavenumbers. The total error for the discretized 1D Helmholtz operator is given by

$$\text{error}_{\text{total}}=\frac{\|u-\hat{u}\|}{\|u\|}\leqslant C_1kh+C_2k^3h^2,\ kh<1. \tag{5.20}$$

While applying the rule of thumb $kh\leqslant 0.625$ is sufficient for keeping the first term under control, it does not harbour properly against the propagation of the pollution error which grows with k, even if kh is kept small enough. Thus, it has been advocated to set the grid resolution to $k^3h^2\leqslant\epsilon$ instead of $kh\leqslant 0.625$. [38] and [41] have proved that while it is possible to eliminate the pollution effect in 1D Helmholtz problems by implementing a modified wavenumber, a similar conclusion can not be extended to higher dimensional problems, see Section 5.4 for more details. As a result, much research has been conducted towards minimizing the pollution error. Note that the bound in equation 5.20 also holds in higher dimensions, as long as the second order finite difference method is used. For any general p-th order scheme, we obtain the following error bound:

$$\text{error}_{\text{total}}=\frac{\|u-\hat{u}\|}{\|u\|}\leqslant C_1kh+C_2k(k^ph^p),\ kh<1. \tag{5.21}$$

5.3.2. LITERATURE OVERVIEW

The literature has proposed several ways to mitigate this persisting issue. One branch has focused on formulating new higher-order discretization schemes. Among the first are a rotated 9-point finite difference scheme [54]. This method is extended by including a 'perfectly matched layer' (PML) [55]. In both works, optimal parameters for the difference scheme were computed in order to improve the accuracy of the numerical solution. A similar strategy was used for the three-dimensional Helmholtz operator, where the 9-point stencil was extended to a 27-point stencil [56]. Furthermore, some line of work developed accurate higher order schemes for the one- and 2D Helmholtz equation, under the assumption that separation of variables can be used [43–46].

In line of this strategy lies the use of compact finite difference schemes [42, 47, 57, 58]. One advantage of the compact scheme is that no additional boundary conditions are required due to having a larger stencil. While both compact fourth- and sixth-order schemes were developed in the literature, it has been shown that at best sixth-order accuracy can be achieved using compact stencils for the Poisson, and thus inherently, the Helmholtz equation [59]. Apart from using compact higher-order finite difference schemes, others have incorporated wave-ray theory to obtain more accurate solutions [60] or have constructed a modified wavenumber which is closer to the exact wavenumber in order to reduce the numerical dispersion [61]. When using such strategies, all methods depend on a pre-specified propagation angle to provide an accurate solution, as the exact propagation angle is unknown. As a result, for specified angles an accurate solution can be obtained by either incorporating a modified wavenumber or by switching to a higher-order dispersion corrected discretization. A combination of both has been studied by [61], where the standard 5-point stencil is replaced by a parametrized 9-point difference scheme including a modified wavenumber. Recently, using an asymptotic dispersion correction for 2D constant wavenumber problems, these methods have shown to provide up to sixth order accuracy for plane waves given an angle of propagation [51].

5.4. CLASSICAL DISPERSION CORRECTION

As mentioned earlier, it is possible to eliminate the pollution error for the 1D MP 1. Recall from the previous section that the discretization of MP 1 using second order finite-differences was given by

$$\frac{-u_{j-1}+2u_j-u_{j+1}}{h^2}-k^2u_j=0,\ 1\leqslant j\leqslant n-1, \tag{5.22}$$

with general solution

$$u(x)=e^{ikx}. \tag{5.23}$$

Evaluation of expression 5.23 in the discrete points led to

$$u_j=e^{i\tilde{k}x_j},\ 1\leqslant j\leqslant n-1, \tag{5.24}$$

which can be considered as plane-wave solutions of the discrete homogeneous Helmholtz equation, where $\tilde{k}$ represents the numerical wavenumber. Substituting 5.24 into 5.22 and using Euler's trigonometric identity to decompose the exponential function, leads to

$$\begin{aligned}
&-2\cos(\tilde{k}h)+2-k^2h^2=0,\\
&2\cos(\tilde{k}h)=2-k^2h^2,\\
&\tilde{k}h=\arccos(1-\frac{k^2h^2}{2}),\\
&\tilde{k}=\frac{1}{h}\arccos(1-\frac{k^2h^2}{2})=k-\frac{k^3h^2}{24}+\mathcal{O}(k^5h^4).
\end{aligned}$$

If we want to eliminate the discretization error introduced into the scheme, we need to set $\tilde{k}=k$, i.e.

$$\tilde{k}=\frac{1}{h}\arccos(1-\frac{k^2h^2}{2})=k\Rightarrow\tilde{k}=\sqrt{\frac{2(1-\cos(kh)}{h^2}}. \tag{5.25}$$

Unfortunately, this approach only works for 1D problems. To see this, we look at the 2D second order finite difference scheme

$$\frac{-u_{i-1,j} - u_{i,j-1} + 4u_{i,j} - u_{i+1,j} - u_{i,j+1}}{h^2} - k^2 u_{i,j} = 0,\ 1 \leqslant i,j \leqslant n-1. \tag{5.26}$$

Again, using plane-wave solutions, we write $u(x,y) = e^{i(k_1 x + k_2 y_j)}$, with $(k_1,k_2) = (k\cos\theta, k\sin\theta)$. Evaluating the solution in the discrete grid points (x_i,y_j) gives $u(x_i,y_j) = e^{i(\tilde{k}_1 x + \tilde{k}_2 y_j)}$, where $(\tilde{k}_1,\tilde{k}_2) = (\tilde{k}\cos\theta, \tilde{k}\sin\theta)$ denotes the numerical wavenumber. Substituting these expressions into the difference scheme 5.26, the problem becomes

$$-2\cos(\tilde{k}\cos(\theta)h) - 2\cos(\tilde{k}\sin(\theta)h) + 4 - k^2h^2 = 0. \tag{5.27}$$

Generally the direction of the plane waves θ is unavailable. This is due to the fact that plane waves propagate in an infinite number of directions. Even if there are directionally prevalent components in this decomposition they are not necessarily known apriori [40, 62]. Therefore, in order to solve for $\tilde{k}$ to obtain a 2D dispersion correction, equation 5.27 needs to be minimized over all angles θ, which remains problematic.

5.5. POLLUTION AND SPECTRAL PROPERTIES

The vast majority of works regarding the pollution error focuses on developing numerical discretization schemes to mitigate the pollution effect. Note that in order to study the pollution error, the analytical solution solution must be known, which limits the scope of potential test problems. Moreover, the a priori upper bound from expression 5.20 shows that the pollution error can be bounded from above by a term which grows linearly with k. This bound is known to be sharp, but provides little detail as regards the underlying characteristics with respect to its dependence on the numerical dispersion. As we have seen in Section 5.4, this becomes even more problematic in higher-dimensions.

Thus, in order to investigate the explicit translation of the numerical dispersion effect into the pollution error, we will use the information from the eigenvalues. To our current knowledge, this provides a novel theoretical perspective on the pollution error. How the pollution effect influences spectral properties and vice versa has remained an unconventional approach in researching the pollution error. In order to research these properties, we start by looking at the differences between the exact and numerical solution of MP 1. The explicit use of the eigenvalues requires that we use a model problem with Dirichlet boundary conditions. The latter model problem has also been researched using the conventional method [45, 47–49].

5.5.1. GENERAL PROPERTIES

Recall from Section 5.2 that the 1D MP 1 is given by

$$-\frac{d^2u}{dx^2} - k^2\,u = \delta(x-x'), x \in \Omega = [0,L] \subset \mathbb{R},$$
$$u(0) = 0, u(L) = 0,\ k \in \mathbb{R}\backslash\{0\}.$$

We also showed that the analytical solution $u(x,x')$ can be expressed in terms of the Green's function by

$$u(x,x') = 2\sum_{j=1}^{\infty} \frac{\sin(j\pi x')}{j^2\pi^2 - k^2}\sin(j\pi x),\ k \neq j\pi \text{ for } j = 1,2,3,\ldots. \tag{5.28}$$

If we define $u_j = u(x_j), j = 1,2,\ldots,n$, where u is evaluated at the discrete grid points, we can represent the $n-$th term finite solution as a vector $u(\bar{x})$ by

$$u(\bar{x}) = 2\sum_{j=1}^{n} \frac{\sin(j\pi x')}{\lambda^j} v^j(\bar{x}),\ k \neq \pi \text{ for } j = 1,2,3,\ldots,n, \tag{5.29}$$

where $\bar{x} = [x_1, x_2, \ldots, x_n]^T$ and $v^j(\bar{x}) = \frac{\sin(j\pi\bar{x})}{\|\sin(j\pi\bar{x})\|}$ is now the $j-$th orthonormal eigenvector corresponding to the $j-$th eigenvalue. The eigenvectors are exact discretizations of the continuous eigenfunctions. Note that the denominator of each term in the sum consists of the analytical eigenvalues. The right-hand side function $f(\bar{x})$ of MP 1 is known and can also be represented using the same basis of orthonormal eigenvectors

$$f(\bar{x}) = 2\sum_{j=1}^{n} \sin(j\pi x')v^j(\bar{x}). \tag{5.30}$$

Similarly, we can write the numerical solution vector $\hat{u}$ as follows

$$\begin{aligned}\hat{u} = A^{-1}f(\bar{x}) &= A^{-1}2\sum_{j=1}^{n}\sin(j\pi x')v^j(\bar{x}) \\ &= 2\sum_{j=1}^{n}\frac{\sin(j\pi x')}{\hat{\lambda}^j}v^j(\bar{x}),\end{aligned} \tag{5.31}$$

where $\hat{\lambda^j}$ are the numerical eigenvalues. We proceed by using the notation u, $\hat{u}$ and f respectively.

5.5.2. ONE-DIMENSIONAL SPECTRAL PROPERTIES

We now have a simple expression which can be decomposed into terms containing the eigenvalues. This allows us to identify the polluting terms of the numerical solution. We start by investigating some general properties of the differences between the analytical and numerical eigenvalues.

Lemma 9.1: Difference Eigenvalues

Let λ^j be the analytical eigenvalue and $\hat{\lambda}^j$ be the numerical eigenvalue for $j = 1,\dots,n$, where $n > \pi$. If the expressions for the eigenvalues are given by

$$\lambda^j = j^2\pi^2 - k^2, \hat{\lambda}^j = \frac{2}{h^2}(1-\cos(j\pi h)) - k^2,$$

then the difference between the eigenvalues is bounded from above by

$$\lambda^j - \hat{\lambda}^j < \frac{j^4\pi^4 h^2}{12}, \tag{5.32}$$

and from below by

$$\lambda^j - \hat{\lambda}^j \geqslant \frac{j^4\pi^4 h^2}{12} - \frac{2j^6\pi^6 h^4}{6!}. \tag{5.33}$$

5

Proof. We start by showing expression 5.32. The difference between the eigenvalues is given by

$$\lambda^j - \hat{\lambda}^j = j^2\pi^2 - k^2 - \left(\frac{2}{h^2}(1-\cos(j\pi h)) - k^2\right).$$

Substituting the power series for the cosine term and letting ζ represent our cut-off point, we obtain

$$\begin{aligned}
\lambda^j - \hat{\lambda}^j &= j^2\pi^2 - k^2 - \left(\frac{2}{h^2}\left(1 - \left(\sum_{l=0}^{\infty}\frac{(-1)^l(j\pi h)^{2l}}{(2l)!}\right)\right) - k^2\right), \\
&< j^2\pi^2 - k^2 - \left(\frac{2}{h^2}\left(1 - 1(1 - \frac{j^2\pi^2h^2}{2} + \frac{j^4\pi^4h^4}{24} - \zeta^6)\right) - k^2\right), \\
&< j^2\pi^2 - k^2 - \left(\frac{2}{h^2}\left(1 - 1 + j^2\pi^2\frac{h^2}{2} - j^4\pi^4\frac{h^4}{24}\right) - k^2\right), \\
&= j^2\pi^2 - k^2 - \left(j^2\pi^2 - k^2 - \frac{j^4\pi^4h^2}{12}\right), \\
&= \frac{j^4\pi^4h^2}{12}.
\end{aligned}$$

This gives us an upper bound with respect to the difference between the analytical and numerical eigenvalue. Now to construct the lower bound in expression 5.33, we need to show that

$$\lambda^j - \hat{\lambda}^j \geqslant \frac{j^4\pi^4 h^2}{12} - \frac{2j^6\pi^6 h^4}{6!}. \tag{5.34}$$

We again substitute the power series for the cosine term in the difference equation of the

eigenvalues, which gives

$$\begin{aligned}\lambda^j - \hat{\lambda}^j &= j^2\pi^2 - k^2 - \left(\frac{2}{h^2}\left(1 - \left(\sum_{l=1}^{\infty}\frac{(-1)^l(j\pi h)^{2l}}{(2l)!}\right)\right) - k^2\right),\\ &= j^2\pi^2 - k^2 - \left(j^2\pi^2 - k^2 - \frac{j^4\pi^4h^2}{12} + \frac{2j^6\pi^6h^4}{6!} - \frac{2j^8\pi^8h^6}{8!} + \frac{2j^{10}\pi^{10}h^8}{10!}\ldots\right),\\ &= \frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!} + \frac{2j^8\pi^8h^6}{8!} - \frac{2j^{10}\pi^{10}h^8}{10!}\ldots.\end{aligned}$$

Substituting the difference expression into 5.34 and grouping terms on the left-hand side leads to a true statement if each of the term in parenthesis is non-negative.

$$\left(\frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!}\right) + \left(\frac{2j^8\pi^8h^6}{8!} - \frac{2j^{10}\pi^{10}h^8}{10!}\right) + \ldots \geqslant \left(\frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!}\right). \quad (5.35)$$

Thus, in order to show that this holds for all j we need to show that each term in parenthesis is non-negative. We write expression 5.35 as

$$\left(\frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!}\right) + \sum_{l=2}^{\infty}\left(\frac{2j^{4l}\pi^{4l}h^{4l-2}}{(4l)!} - \frac{2j^{4l+2}\pi^{4l+2}h^{4l}}{(4l+2)!}\right) \geqslant \left(\frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!}\right). \quad (5.36)$$

The sum on the left-hand side of expression 5.36 will be greater than the right-hand side if we can proof that each grouped term is non-negative. Thus, we need to show that for each $j = 1,2,\ldots n$

$$\begin{aligned}&\left(\frac{2j^{4l}\pi^{4l}h^{4l-2}}{(4l)!} - \frac{2j^{4l+2}\pi^{4l+2}h^{4l}}{(4l+2)!}\right) \geqslant 0 \Leftrightarrow,\\ &\frac{2j^{4l}\pi^{4l}h^{4l-2}}{(4l)!}\left(1 - \frac{j^2\pi^2h^2}{(4l+2)(4l+1)}\right) \geqslant 0. \quad (5.37)\end{aligned}$$

For a positive integer j and $0 < h < 1$, this boils down to showing that for each $j = 1,2,\ldots n$ and $l \geqslant 2$

$$1 \geqslant \frac{j^2\pi^2h^2}{(4l+2)(4l+1)} \Leftrightarrow (4l+2)(4l+1) \geqslant j^2\pi^2h^2. \quad (5.38)$$

Given that the right-hand side of inequality 5.38 is strictly increasing with respect to j, we can evaluate the minimum at $j = 1$ and maximum at $j = n$ to evaluate the lower bound.

$$(4l+2)(4l+1) \geqslant \begin{cases}\pi^2h^2, & \text{if } j = 1\\ \pi^2, & \text{if } j = n,\end{cases} \quad (5.39)$$

where we used that $h = n^{-1} < 1$, where $n > \pi$. In both cases and already for the smallest value of l ($l = 2$), the statement holds. Consequently, we must have

$$\lambda^j - \hat{\lambda}^j \geqslant \frac{j^4\pi^4h^2e}{12} - \frac{2j^6\pi^6h^4}{6!}.$$

■

Corollary 9.1: Bound for analytical eigenvalue

Let λ^j be the analytical eigenvalue and $\hat{\lambda}^j$ be the numerical eigenvalue for $j = 1,\dots,n$, where $n > \pi$. Then for each j, the analytical eigenvalue λ^j is bounded in terms of the numerical eigenvalue $\hat{\lambda}^j$ by

$$\hat{\lambda}^j + \left(\frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!}\right) \leqslant \lambda^j < \hat{\lambda}^j + \frac{j^4\pi^4h^2}{12}.$$

Proof. This result follows directly from Lemma 9.1, where we have

$$\lambda^j - \hat{\lambda}^j < \frac{j^4\pi^4h^2}{12} \Rightarrow \lambda^j < \hat{\lambda}^j + \frac{j^4\pi^4h^2}{12},$$
$$\lambda^j - \hat{\lambda}^j \geqslant \left(\frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!}\right) \Rightarrow \lambda^j < \hat{\lambda}^j + \left(\frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^4}{6!}\right).$$

5

■

Note that the upper and lower bound are dependent on the truncation error of the numerical discretization method. We use Lemma 9.1 and Corollary 9.1 to obtain a more detailed understanding of the pollution error and how the numerical dispersion contributes to it. Moreover, we aim to find the eigenmodes which are responsible for this dispersive pattern. By writing the numerical eigenvalue as a function of the discretization error to approximate the analytical eigenvalue, we propose a dispersion correction depending on the discretization scheme (see Section 5.5.1).

Corollary 9.2: Sum Eigenvalues

Let λ^j be the analytical eigenvalue and $\hat{\lambda}^j$ be the numerical eigenvalue for $j = 1,\dots,n$. Then the sum of the reciprocal of the analytical eigenvalues can be bounded in terms of the numerical eigenvalues by

$$\sum_{j=1}^{n} \left|\frac{1}{\lambda^j}\right| < \sum_{j=1}^{n} \frac{1}{\tilde{\lambda}^j},$$

where we let $\tilde{\lambda}^j = \min\left\{\left|\hat{\lambda}^j + \frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^2}{6!}\right|, \left|\hat{\lambda}^j + \frac{j^4\pi^4h^2}{12}\right|\right\}$.

Proof. We use Corollary 9.1 . By taking the minimum, we ensure that the analytical eigenvalue is bounded in terms of magnitude. This is necessary as both the continuous and discrete operator are indefinite, which leads to positive and negative eigenvalues. Taking the reciprocal and summing over all eigenvalues gives the statement. ■

Lemma 9.1 and Corollary 9.2 provides us with a way to express the analytical eigenvalues in terms of the numerical eigenvalues by adding a correction term. This correction term depends on the truncation error of the discretization method. We now construct an upper bound for the error term between the exact and numerical solution in the theorem below.

Theorem 10: Pollution

Let u be the (exact) solution to MP 1 and let $\hat{u}$ be the numerical solution obtained by solving $A\hat{u} = f$, where A is a non-singular matrix obtained by using a p-th order finite difference scheme. If kh is kept constant, then the absolute error in the L^2-norm is bounded from above by

$$\|u - \hat{u}\| < 2\sqrt{\sum_{j=1}^{n}\left(\frac{\frac{j^4\pi^4h^2}{12}}{\tilde{\lambda}^j\hat{\lambda}^j}\right)^2},$$

where $\tilde{\lambda}^j = \min\left\{\left|\hat{\lambda}^j + \frac{j^4\pi^4h^2}{12} - \frac{2j^6\pi^6h^2}{6!}\right|, \left|\hat{\lambda}^j + \frac{j^4\pi^4h^2}{12}\right|\right\}$.

Proof. Using the expansion for the right-hand side function $f(x)$, We write the numerical solution vector $\hat{u}$ as

$$\begin{aligned}\hat{u} = A^{-1}f(\bar{x}) &= A^{-1}(2\sum_{j=1}^{n}\sin(j\pi x'))v^j(\bar{x}) \\ &= 2\sum_{j=1}^{n}\frac{\sin(j\pi x')}{\hat{\lambda}^j}v^j(\bar{x}). \end{aligned} \tag{5.40}$$

Note that this is based on the eigenfunctions evaluated at the discrete grid points and scaled to yield an orthonormal basis (see Section 5.5.1). Consequently, we have

$$\begin{aligned}\|u - \hat{u}\| &= \left\|2\sum_{j=1}^{n}\frac{\sin(j\pi x')}{\lambda^j}v^j(\bar{x}) - 2\sum_{j=1}^{n}\frac{\sin(j\pi x')}{\hat{\lambda}^j}v^j(\bar{x})\right\|, \\ &= \left\|2\sum_{j=1}^{n}\left(\frac{\sin(j\pi x')}{\lambda^j} - \frac{\sin(j\pi x')}{\hat{\lambda}^j}\right)v^j(\bar{x})\right\|, \\ &= \left\|2\sum_{j=1}^{n}\sin(j\pi x')\left(\frac{1}{\lambda^j} - \frac{1}{\hat{\lambda}^j}\right)\right\|,\end{aligned}$$

where we used that the eigenvectors are orthonormal. We can write the error in the 2-norm

as

$$\begin{aligned}\|u-\hat{u}\| &= \sqrt{4\sin(\pi x')^2(\frac{1}{\lambda^1}-\frac{1}{\hat{\lambda}^1})^2+\ldots+4\sin(n\pi x')^2(\frac{1}{\lambda^n}-\frac{1}{\hat{\lambda}^n})^2},\\ &= \sqrt{4\sum_{j=1}^{n}\sin(j\pi x')^2\left(\frac{1}{\lambda^j}-\frac{1}{\hat{\lambda}^j}\right)^2},\\ &< \sqrt{4\sum_{j=1}^{n}\left(\frac{1}{\lambda^j}-\frac{1}{\hat{\lambda}^j}\right)^2},\\ &= \sqrt{4\sum_{j=1}^{n}\left(\frac{\hat{\lambda}^j-\lambda^j}{\hat{\lambda}^j\lambda^j}\right)^2}, \end{aligned} \tag{5.41}$$

5

where we used that the eigenvectors are orthonormal and each sine term containing the location of the source is less than one. We would like to find an upper bound for expression 5.41. We use Lemma 9.1 and Corollary 9.2 , to provide element-wise upper bounds. From Lemma 9.1 it follows that

$$\sum_{j=1}^{n}(\lambda^j-\hat{\lambda}^j)^2 < \sum_{j=1}^{n}\left(\frac{j^4\pi^4h^2}{12}\right)^2. \tag{5.42}$$

For the denominator, Corollary 9.2 provides us with

$$\sum_{j=1}^{n}\left(\frac{1}{\hat{\lambda}^j\lambda^j}\right)^2 \leqslant \sum_{j=1}^{n}\left(\frac{1}{\hat{\lambda}^j}\right)^2\left(\frac{1}{\tilde{\lambda}^j}\right)^2, \tag{5.43}$$

where we have $\tilde{\lambda}^j = \min\left\{\left|\hat{\lambda}^j+\frac{j^4\pi^4h^2}{12}-\frac{2j^6\pi^6h^2}{6!}\right|,\left|\hat{\lambda}^j+\frac{j^4\pi^4h^2}{12}\right|\right\}$. Substituting 5.42 and 5.43 into inequality 5.41 gives

$$\sqrt{4\sum_{j=1}^{n}\left(\frac{\hat{\lambda}^j-\lambda^j}{\hat{\lambda}^j\lambda^j}\right)^2} < 2\sqrt{\sum_{j=1}^{n}\left(\frac{\frac{j^4\pi^4h^2}{12}}{\hat{\lambda}^j\tilde{\lambda}^j}\right)^2}.$$

■

5.5.3. TWO-DIMENSIONAL SPECTRAL PROPERTIES

In this section we extend the results from Section 5.5.2 to the 2D case for MP 2. We start by defining the error estimation for the 2D case.

Lemma 10.1: Difference Eigenvalues

Let $\lambda^{i,j}$ be the analytical eigenvalue and $\hat{\lambda}^{i,j}$ be the numerical eigenvalue for $i,j = 1,\dots,n$, where $n > \pi$. If the expressions for the eigenvalues are given by

$$\lambda^{j} = (i^2 + j^2)\pi^2 - k^2,$$
$$\hat{\lambda}^{j} = \frac{1}{h^2}\left(4 - 2\cos(i\pi h) - 2\cos(j\pi h)\right) - k^2,$$

then the difference between the eigenvalues is bounded from above by

$$\lambda^{i,j} - \hat{\lambda}^{i,j} < \frac{(i^4 + j^4)\pi^4 h^2}{12}, \tag{5.44}$$

and from below by

$$\lambda^{i,j} - \hat{\lambda}^{i,j} \geqslant \frac{(i^4 + j^4)\pi^4 h^2}{12} - \frac{2(i^6 + j^6)\pi^6 h^4}{6!}. \tag{5.45}$$

5

Proof. Similar to the 1D case, substituting the power series for both the i-th and j-th cosine term and letting ζ represent our cut-off point, we obtain

$$\begin{aligned}
\lambda^{i,j} - \hat{\lambda}^{i,j} &= i^2\pi^2 + j^2\pi^2 - k^2 \\
&\quad - \left(\frac{1}{h^2}\left(4 - 2\left(\sum_{l=0}^{\infty}\frac{(-1)^l(i\pi h)^{2l}}{(2l)!}\right) - 2\left(\sum_{l=0}^{\infty}\frac{(-1)^l(j\pi h)^{2l}}{(2l)!}\right)\right) - k^2\right), \\
&< i^2\pi^2 + j^2\pi^2 - k^2 \\
&\quad - \left(\frac{1}{h^2}\left(4 - 2 + i^2\pi^2\frac{h^2}{2} - 2 + j^2\pi^2\frac{h^2}{2} - i^4\pi^4\frac{h^4}{24} - j^4\pi^4\frac{h^4}{24} + \zeta^6\right) - k^2\right), \\
&= i^2\pi^2 + j^2\pi^2 - k^2 \\
&\quad - \left(i^2\pi^2 + j^2\pi^2 - k^2 - \frac{i^4\pi^4 h^2}{12} - \frac{j^4\pi^4 h^2}{12}\right), \\
&= \frac{(i^4 + j^4)\pi^4 h^2}{12}.
\end{aligned}$$

To construct the lower bound, we again substitute the power series for the cosine terms in

the difference equation, which gives

$$\begin{aligned}\lambda^{i,j} - \hat{\lambda}^{i,j} &= i^2\pi^2 + j^2\pi^2 - k^2 \\ &- \left(\frac{1}{h^2}\left(4 - 2\left(\sum_{l=0}^{\infty}\frac{(-1)^l(i\pi h)^{2l}}{(2l)!}\right) - 2\left(\sum_{l=0}^{\infty}\frac{(-1)^l(j\pi h)^{2l}}{(2l)!}\right)\right) - k^2\right), \\ &= i^2\pi^2 + j^2\pi^2 - k^2 \\ &- \left(i^2\pi^2 + j^2\pi^2 - k^2 - \frac{i^4\pi^4h^2}{12} - \frac{j^4\pi^4h^2}{12} + \frac{2i^6\pi^6h^4}{6!} + \frac{2j^6\pi^6h^4}{6!} - \ldots\right), \\ &= \frac{(i^4+j^4)\pi^4h^2}{12} - \frac{2(i^6+j^6)\pi^6h^4}{6!} + \frac{2(i^8+j^8)\pi^8h^6}{8!} - \frac{2(i^{10}+j^{10})\pi^{10}h^8}{10!}\ldots\end{aligned}$$

Substituting the difference expression into 5.45 and grouping terms on the left-hand side only leads to

$$\begin{aligned}\left(\frac{(i^4+j^4)\pi^4h^2}{12} - \frac{2(i^6+j^6)\pi^6h^4}{6!}\right) + \left(\frac{2(i^8+j^8)\pi^8h^6}{8!} - \frac{2(i^{10}+j^{10})\pi^{10}h^8}{10!}\right) + \ldots, \\ \geqslant \left(\frac{(i^4+j^4)\pi^4h^2}{12} - \frac{2(i^6+j^6)\pi^6h^4}{6!}\right).\end{aligned}$$

We can write this as

$$\begin{aligned}\left(\frac{(i^4+j^4)\pi^4h^2}{12} - \frac{2(i^6+j^6)\pi^6h^4}{6!}\right) + \sum_{l=2}^{\infty}\left(\frac{2(i^{4l}+j^{4l})\pi^{4l}h^{4l-2}}{(4l)!} - \frac{2(i^{4l+2}+j^{4l+2})\pi^{4l+2}h^{4l}}{(4l+2)!}\right) \\ \geqslant \left(\frac{(i^4+j^4)\pi^4h^2}{12} - \frac{2(i^6+j^6)\pi^6h^4}{6!}\right). \qquad (5.46)\end{aligned}$$

The sum on the left-hand side of expression 5.46 will be greater than the right-hand side if we can proof that each grouped term is non-negative. Thus, we need to show that for each $i, j = 1, 2, \ldots n$

$$\begin{aligned}\left(\frac{2(i^{4l}+j^{4l})\pi^{4l}h^{4l-2}}{(4l)!} - \frac{2(i^{4l+2}+j^{4l+2})\pi^{4l+2}h^{4l}}{(4l+2)!}\right) \geqslant 0 \Leftrightarrow, \\ \frac{2(i^{4l}+j^{4l})\pi^{4l}h^{4l-2}}{(4l)!}\left(1 - \frac{(j^2+i^2)\pi^2h^2}{(4l+1)(4l+2)}\right) \geqslant 0. \qquad (5.47)\end{aligned}$$

For positive integers i, j and $0 < h < 1$, this boils down to showing that for each $i, j = 1, 2, \ldots n$ and $l \geqslant 2$

$$1 > \frac{(j^2+i^2)\pi^2h^2}{(4l+1)(4l+2)} \Leftrightarrow (4l+2)(4l+1) \geqslant i^2\pi^2h^2 + j^2\pi^2h^2. \qquad (5.48)$$

Given that the right-hand side of inequality 5.48 is strictly increasing with respect to i and j, we can evaluate the minimum at $i, j = 1$ and maximum at $i, j = n$ to evaluate the lower bound.

$$(4l+2)(4l+1) \geqslant \begin{cases} 2\pi^2h^2, & \text{if } i, j = 1 \\ 2\pi^2, & \text{if } i, j = n, \end{cases} \qquad (5.49)$$

where we used that $h = n^{-1} < 1$ such that $nh = 1$ and $n > \pi$. In both cases and already for the smallest value of l ($l = 2$), the statement holds. Consequently, we must have

$$\lambda^{i,j} - \hat{\lambda}^{i,j} \geqslant \frac{(i^4 + j^4)\pi^4 h^2}{12} - \frac{2(i^6 + j^6)\pi^6 h^4}{6!}.$$

■

Similar to the 1D case, we can now bound the analytical eigenvalues in terms of the numerical eigenvalues by using the lower bound.

Corollary 10.1: Sum Eigenvalues

Let $\lambda^{i,j}$ be the analytical eigenvalue and $\hat{\lambda}^{i,j}$ be the numerical eigenvalue for $i, j = 1, \ldots, n$. Then the sum of the reciprocal of the analytical eigenvalues can be bounded in terms of the numerical eigenvalues by

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\left|\frac{1}{\lambda^{i,j}}\right| < \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\tilde{\lambda}^{i,j}},$$

where we let $\tilde{\lambda}^{i,j} = \min\left\{\left|\hat{\lambda}^{i,j} + \frac{(i^4+j^4)\pi^4h^2}{12} - \frac{2(i^6+j^6)\pi^6h^2}{6!}\right|, \left|\hat{\lambda}^{i,j} + \frac{(i^4+j^4)\pi^4h^2}{12}\right|\right\}$

Proof. The proof is exactly the same as in the 1D case. Using the lower bound and taking the reciprocal of each respective term will give the statement after summing over all i and j. ■

We use Lemma 10.1 and Corollary 10.1 to find a similar upper bound for the 2D pollution error. We proceed by extending Theorem 10 to the 2D case.

Corollary 10.2: Pollution

Let u be the (exact) solution to MP 2 and let $\hat{u}$ be the numerical solution obtained by solving $A\hat{u} = f$, where A is a non-singular matrix obtained by using a p-th order finite difference scheme.
If kh is kept constant, then the absolute error in the L^2-norm is bounded from above by

$$\|u - \hat{u}\| < 4\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\frac{(i^4+j^4)\pi^4h^2}{12}}{\hat{\lambda}^{i,j}\tilde{\lambda}^{i,j}}\right)^2},$$

where $\tilde{\lambda}^{i,j} = \min\left\{\left|\hat{\lambda}^{i,j} + \frac{(i^4+j^4)\pi^4h^2}{12} - \frac{2(i^6+j^6)\pi^6h^2}{6!}\right|, \left|\hat{\lambda}^{i,j} + \frac{(i^4+j^4)\pi^4h^2}{12}\right|\right\}$.

Proof. See proof of Theorem 10 for the 1D case and extend it to the case where the index i also goes from 1 to n. ■

We now have an upper bound for the total error in terms of the numerical eigenvalues. If we compare this to the conventional pollution term,

$$\text{error}_{\text{pollution}} = \|u - \hat{u}\| \leqslant Ck(k^2h^2),$$

we observe that the explicit linear dependence on k has been replaced by the explicit dependence on a superposition of the numerical eigenvalues. One advantage of writing the upper bound in this way is that we immediately observe that the pollution error can be minimized in both one- and two-dimensions for this model problem. Even for this simple model problem, the latter was deemed impossible due to the wave travelling in infinite directions for the 2D model problem, see Section 5.5.4.1. It is easy to see that if we minimize the largest term of the sum, then all other terms, which are by definition smaller, will allow the total sum to be minimized as well.

Corollary 10.3: Minimized Pollution 2D

Let u be the (exact) solution to MP 2 given by expression 5.29 and suppose the L^2-norm of the exact solution is always smaller than 1, i.e. $\|u\| < 1$.
Let $(i_{\min}, j_{\min})$ and $(\hat{i}_{\min}, \hat{j}_{\min})$ denote the location of the smallest analytical and numerical eigenvalue respectively and suppose $\left|\lambda^{i_{\min},j_{\min}}\right| \leqslant \left|\hat{\lambda}^{\hat{i}_{\min},\hat{j}_{\min}}\right|$.
If

$$\left(\frac{4\frac{(i_{\min}^4 + j_{\min}^4)\pi^4 h^2}{12}}{\hat{\lambda}^{i_{\min},j_{\min}}\left(\hat{\lambda}^{i_{\min},j_{\min}} + \frac{(i_{\min}^4 + j_{\min}^4)\pi^4 h^2}{12} - \frac{2(i_{\min}^6 + j_{\min}^6)\pi^6 h^4}{6!}\right)} \right)^2 = \mathcal{O}(h^2),$$

then the relative error is bounded by

$$\frac{\|u - \hat{u}\|}{\|u\|} \leqslant 1.$$

Proof. Note that reciprocal of the smallest analytical value in terms of magnitude is the largest term in the set of the reciprocals of both the analytical and numerical eigenvalues. Now, unless $(i_{\min}, j_{\min}) = (\hat{i}_{\min}, \hat{j}_{\min})$, and $\lambda^{i_{\min},j_{\min}} \approx \hat{\lambda}^{i_{\min},j_{\min}}$, the difference between the reciprocals will be largest there and thus it will provide the largest contribution to the sum. As a result, we must have

$$\left(\frac{4\frac{(i_{\min}^4 + j_{\min}^4)\pi^4 h^2}{12}}{\hat{\lambda}^{i_{\min},j_{\min}}\left(\hat{\lambda}^{i_{\min},j_{\min}} + \frac{(i_{\min}^4 + j_{\min}^4)\pi^4 h^2}{12} - \frac{2(i_{\min}^6 + j_{\min}^6)\pi^6 h^4}{6!}\right)} \right)^2 \geqslant \left(\frac{4\frac{(i^4 + j^4)\pi^4 h^2}{12}}{\hat{\lambda}^{i,j}\left(\hat{\lambda}^{i,j} + \frac{(i^4 + j^4)\pi^4 h^2}{12} - \frac{2(i^6 + j^6)\pi^6 h^4}{6!}\right)} \right)^2, \tag{5.50}$$

for all $i, j = 1, 2, \ldots, n$. Each (i, j)-term can be bounded from above by the left-hand side of inequality 5.50. Substituting for each term in the upper bound from Corollary 10.2, we

obtain

$$
\begin{aligned}
\|u-\hat{u}\| &< 4\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{\frac{(i^4+j^4)\pi^4h^2}{12}}{\hat{\lambda}^{i,j}\left(\hat{\lambda}^{i,j}+\frac{(i^4+j^4)\pi^4h^2}{12}-\frac{2(i^6+j^6)\pi^6h^4}{6!}\right)}\right)^2},\\
&= \left(\left(\frac{4\frac{(i^4_{\min}+j^4_{\min})\pi^4h^2}{12}}{\hat{\lambda}^{i_{\min},j_{\min}}\left(\hat{\lambda}^{i_{\min},j_{\min}}+\frac{(i^4_{\min}+j^4_{\min})\pi^4h^2}{12}-\frac{2(i^6_{\min}+j^6_{\min})\pi^6h^4}{6!}\right)}\right)^2\right.\\
&\quad \left.+\sum_{\substack{i=1\\ i\neq i_{\min}}}^{n-1}\sum_{\substack{j=1\\ j\neq j_{\min}}}^{n-1}\left(\frac{4\frac{(i^4+j^4)\pi^4h^2}{12}}{\hat{\lambda}^{i,j}\left(\hat{\lambda}^{i,j}+\frac{(i^4+j^4)\pi^4h^2}{12}-\frac{2(i^6+j^6)\pi^6h^4}{6!}\right)}\right)^2\right),\\
&= \sqrt{\mathcal{O}(h^2)+(n-1)\mathcal{O}(h^2)},\\
&= 1.
\end{aligned}
\tag{5.51}
$$

The proof for the case $\left|\lambda^{i_{\min},j_{\min}}\right| \geqslant \left|\hat{\lambda}^{\hat{i}_{\min},\hat{j}_{\min}}\right|$ is exactly the same. ■

The upper corollary reveals the paramount importance of the accuracy of the near-zero eigenvalues and eigenvectors. These dictate the upper bound for the remaining terms in the sum. If the near-zero eigenmodes are approximated with high accuracy, then the dispersion part of the pollution error can be minimized. This also means that if we need a rough estimate which is in the ball park of the true error, we can simply take the reciprocal of the smallest eigenvalue in magnitude due to its largest contribution to the entire sum. In the next section we use the results from this section to construct a dispersion correction for the one- and 2D model problems.

5.5.4. EIGENVALUE BASED DISPERSION CORRECTION

Using this novel perspective, we construct a dispersion correction using the eigenvalues and eigenvectors. Note that for the 1D MP 1, this can easily be constructed and produces similar results compared to using the modified wavenumber, see Section 5.5.4.1. However, one advantage we now have is that we can use the same method in the higher-dimensional problem MP 2 to explicitly study how the numerical dispersion translates into the pollution error. In the next section, we provide numerical evidence for the accuracy ranging from fine to very coarse grids ($kh \geqslant 1$). The latter will allow to solve and study the current model problem very intricately, while keeping the problem size economically feasible compared to determining the step-size according to $k^3h^2 \leqslant 1$.

ONE-DIMENSIONAL DISPERSION CORRECTION

We start by rewriting our original system as follows. Note that for our matrix A, if $\hat{\lambda}^j$ is an eigenvalue of A corresponding to eigenvector v^j, then

$$
Av^j = \hat{\lambda}^j v^j \Longrightarrow (A+cI)v^j = (\hat{\lambda}^j + c)v^j,
$$

and thus $\hat{\lambda}^j + c$ is an eigenvalue of $(A+cI)$. Consequently, if the analytical solution is known, a very simple remedy to obtain better accuracy according to our proposition, would be to

let

$$c = -\hat{\lambda}^{j_{\min}} + \lambda^{j_{\min}}. \tag{5.52}$$

This alleviates the mismatch between the exact near zero eigenvalue and the numerical eigenvalue at index $j_{\min}$. Recall from Section 5.4 that the pollution error for MP 1 can be eliminated by incorporating a modified wavenumber $\tilde{k}$. The latter represents an explicit correction of the wavenumber with respect to the dispersion error. Consequently, we test for the elimination of pollution by comparing the relative error between the exact and numerical solution after solving the following two systems

$$\tilde{A} = A - \tilde{k}I, \text{ where } \tilde{k} = \sqrt{\frac{2(1-\cos(kh))}{h^2}},$$
$$A_c = A + cI, \text{ where } c = -\hat{\lambda}^{j_{\min}} + \lambda^{j_{\min}}.$$

We furthermore denote

$$\hat{u}_{\tilde{k}} : \tilde{A}\hat{u}_{\tilde{k}} = f \text{ and } \hat{u}_c : A_c\hat{u}_c = f.$$

5

For the 1D case, our results from Section 5.5.4.1 suggest that this is often enough to alleviate the adverse effects of numerical dispersion by adding the constant c. However, in some cases, and especially for the 2D model problem, we need a way to shift more smaller eigenvalues while keeping the corresponding eigenvectors unchanged. The reason for this is that in the 2D case there may be a higher algebraic multiplicity and corresponding locations $(i_{\min}, j_{\min})$ where the smallest eigenvalue is located and consequently there may be more than one value for c. In order to circumvent this difficulty, we make use of some theorems, starting with Brauer's theorem [63].

Theorem 11: Brauer

Let A be a diagonalizable matrix with $Av^j = \lambda^j v^j$ and suppose r is a vector such that $r^\intercal v^j = 1$, then for any scalar $\hat{\lambda}^j$, the eigenvalues of the matrix

$$\hat{A} = A + (\hat{\lambda}^j - \lambda^j) v^j r^\intercal,$$

consist of those of A, except that one eigenvalue λ^j of A is replaced with $\hat{\lambda}^j$. Moreover, the eigenvector v^j is unchanged, that is $\hat{A}v^j = \hat{\lambda}^j v^j$.

Proof. For a proof see [63] ■

Corollary 11.1: Brauer

Let A be a diagonalizable matrix with $Av^j = \lambda^j v^j$ and suppose $r = v^j$ then for any scalar $\hat{\lambda}^j$, the eigenvalues of the matrix

$$\hat{A} = A + (\hat{\lambda}^j - \lambda^j) v^j v^{j\intercal},$$

consist of those of A, except that one eigenvalue λ^j of A is replaced with $\hat{\lambda}^j$. Moreover, all the eigenvectors remain unchanged.

Proof. By the diagonalization property of A, we can write $A = P\Sigma P^{-1}$, where Σ consist of the diagonal matrix containing the eigenvalues of A. Then v^j lies in the $j-$th column of P. Let e^j be the $j-$th column of the identity matrix. Then we take

$$\begin{aligned}\hat{A} &= A + (\hat{\lambda^j} - \lambda^j)P(e^j e^{j\top})P^{-1},\\ &= A + (\hat{\lambda^j} - \lambda^j)(Pe^j)(e^{j\top}P^{-1}),\end{aligned}$$

where $r^\top = e^{j\top}P^{-1}$, is precisely the $j-$th column of the matrix P^{-1}. ■

Using the above theorem and lemma, we can correct each eigenvalue, without shifting the eigenvectors of the previous system. Our dispersion correction for the 2D case will use the above theorem recursively, which is extended into the following lemma.

Lemma 11.1: Brauer

Let A be a diagonalizable matrix such that we can write $A = P^{-1}\Sigma P$, where P is the matrix containing the eigenvectors of A. Then, the same basis can be used for diagonalizing $\hat{A}$, where $\hat{\Sigma}$ is the matrix containing the shifted eigenvalues of A such that $\hat{\Sigma}(j,j) = \hat{\lambda^j}$ and we can write $\hat{A} = P\hat{\Sigma}P^{-1}$.

5

Proof. We start by applying Theorem 11 and Corollary 11.1 recursively. For the first eigenvalue λ^1 we obtain

$$\hat{A} = A + (\hat{\lambda^1} - \lambda^1)(Pe^1)(e^{1\top}P^{-1}),$$

where $\hat{A}$ has exactly the same eigenvectors as the original matrix A, but the first eigenvalue λ_1 is shifted to $\hat{\lambda_1}$. Applying this for all $j = 1, 2, \ldots, n$, we finally obtain

$$\hat{A} = A + \sum_{j=1}^{n} (\hat{\lambda^j} - \lambda_1)(Pe^j)(e^{j\top}P^{-1}). \tag{5.53}$$

We proceed by multiplying equation 5.53 from the left by P^{-1}. If we let I, denote the identity matrix, we obtain

$$\begin{aligned}P^{-1}\hat{A} &= P^{-1}A + \sum_{j=1}^{n} (\hat{\lambda^j} - \lambda_1)(P^{-1}Pe^j)(e^{j\top}P^{-1}),\\ &= P^{-1}A + \sum_{j=1}^{n} (\hat{\lambda^j} - \lambda_1)(Ie^j)(e^{j\top}P^{-1}).\end{aligned} \tag{5.54}$$

Note that for each j the term $(e^j)(e^{j\top}P^{-1})$ is an all zero matrix apart from the $j-$th row vector of P^{-1}. Next we multiply equation 5.54 from the right by P, which leads to

$$\begin{aligned}P^{-1}\hat{A}P &= P^{-1}AP + \sum_{j=1}^{n} (\hat{\lambda^j} - \lambda_1)(e^j)(e^{j\top}P^{-1}P),\\ &= \Sigma + \sum_{j=1}^{n} (\hat{\lambda^j} - \lambda^1)(e^j)(e^{j\top}I),\\ &= \Sigma + (\hat{\Sigma} - \Sigma) = \hat{\Sigma}.\end{aligned}$$

We can use Lemma 11.1 to correct the eigenvalues, while keeping the eigenvectors of the original matrix unchanged. We now proceed by constructing the corrected eigenvalues of the new matrix $\hat{A}$. We know that the eigenvalues are bounded from above by a term which is in fact similar to the remainder from the truncation error of the discretization method used. Thus, the method is reminiscent of switching to a higher cut-off point in constructing higher-order discretization stencils. One advantage of this approach is that can now explicitly study the eigenmodes which cause the pollution error as a direct result of numerical dispersion to grow. When constructing higher-order pollution-free discretization schemes, each gridfunction can not be tied explicitly to a measure of having numerical dispersion inducing properties. Whereas, the contribution of the particular eigenmodes are now clearly visible in the solution and therefore the error. In our case, we therefore correct the eigenvalues by adding a finite part of the remainder in order to better approximate the analytical eigenvalue. When using Dirichlet boundary conditions, the effect of each eigenmode contributing to the overall pollution term can be studied in one-, two- and three-dimensions.

$$\tilde{\lambda^j} = \hat{\lambda}^j + \sum_{n=2}^{10} \frac{(-1)^n (j\pi)^{2n} h^{2(n-1)}}{(2n)!}.$$

For the 1D case in particular, we need the eigendecomposition and the new matrix containing the corrected eigenvalues to obtain the solution. With respect to the 1D model problem, it is much more efficient to solely correct one eigenvalue, in particular the smallest eigenvalue (see Section 5.5.4.1). However, for the 2D dispersion correction, we propose a different method, which is based on using the 1D eigendecomposition. As a result, for our model problem, the pollution error can be studied for large wavenumbers in higher-dimensions at reasonable computational costs.

TWO-DIMENSIONAL DISPERSION CORRECTION

As mentioned previously, we use the 1D eigendecomposition to construct the new 2D coefficient matrix $\hat{A}$. One important feature we need is that the original partial differential equation can be solved using separation of variables. A similar prerequisite is needed and posed in some methods developed in the literature [43–46].

Algorithm 7: Pollution corrected coefficient matrix $\hat{A}_{2D}$ using A_{1D}

Initialization:
Construct eigendecomp. of 1D A_{1D} such that $D = P^{-1} A_{1D} P$
for $j = 1, 2, ..n$ **do**
  $\tilde{\lambda^j} = \hat{\lambda}^j + \sum_{n=2}^{10} \frac{(-1)^n (j\pi)^{2n} h^{2(n-1)}}{(2n)!}$
  Replace $\hat{\lambda}^j$ in D with $\tilde{\lambda^j}$
  $\tilde{D}(j,j) = \tilde{\lambda^j}$
  $E_{c\varepsilon_0} = v_h^{j_{\min} T} v_h^{j_{\min}} - wy2$
end
Use corrected matrix $\tilde{D}$ to construct $\hat{A}_{1D} = P^{-1} \tilde{D} P$
Construct 2D coefficient matrix $\hat{A}_{2D} = (\hat{A}_{1D} \otimes I_{1D}) + (I_{1D} \otimes \hat{A}_{1D})$

5.6. NUMERICAL RESULTS

We start by examining the error estimates for the pollution error for MP 1 and MP 2. In both cases we evaluate how close our error estimates are to the true error. We then continue by examining the performance of our eigenvalue-based dispersion correction for both model problems. We mentioned that the conventional approach to studying pollution focuses on the notion of a discrepancy between the numerical and exact wavenumber k. In these instances, the exact solution is generally expressed in exponential form, and the eigenvalues are not expressed explicitly. An interesting observation is that this discrepancy between the numerical and exact wavenumber manifests itself through inaccurate near zero eigenvalues. Thus, if the numerical eigenvalues were better approximations of their continuous counterparts, then we expect the relative error to decrease. Section 5.6.1 contains the results for MP 1, while Section 5.6.2 covers MP 2. All 1D systems are solved using a direct method in Matlab R2018a. For the 2D model problems with large k ($k > 300$), we use a standard preconditioned GMRES-solver to obtain the numerical solution, due to the increasing density of the coefficient matrix.

5.6.1. ONE-DIMENSIONAL CONSTANT WAVENUMBER MODEL

ERROR ESTIMATION

In Figure 5.1 we plot the relative error (red) for random values of k between 100 and 2000 and the upper bound (light red) based on Theorem 10. Additionally, the dashed line is the reciprocal of the smallest numerical eigenvalue in magnitude. This allows us to assess how well this estimate is in the ballpark of the true relative error.

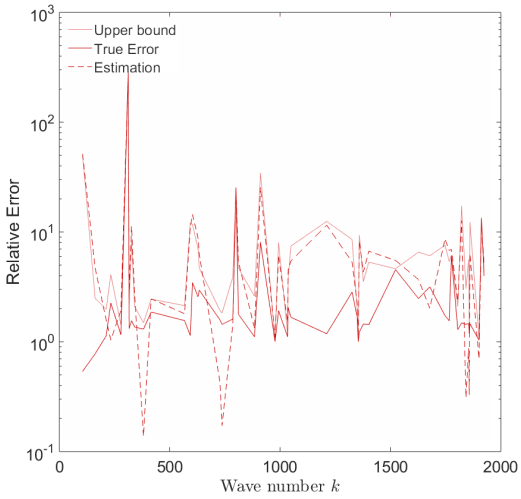


Figure 5.1: 1D Relative error and upper bound for various randomly generated k using $kh = 0.625$.

From Figure 5.1 we observe that the upper bound always holds, as the light red line is always either above or exactly on the dark red line. The lines for the error (red) and upper bound never intersect, and the bound is sharp. Moreover, it shows that the true error behaves more erratically and has a more oscillatory nature which is in direct relation to the smallest eigenvalue in magnitude (dotted line). In particular for example, $k = 1000$ yields a true relative error of 1.493. If we use the bound where the error grows linearly with k, then we have that the pollution term is estimated to be bounded by $k^3h^2 = 390.625$. Using the information from the eigenvalues, our upper bound gives 3.238. Note that the true error (red) follows an oscillatory pattern with peaks appearing for certain k. These are instances where one of the eigenmodes are close to resonant modes and the numerical approximation is poor. If $\lambda^{j_{\min}}$ or $\hat{\lambda}^{j_{\min}}$ is closer to zero than its counterpart, the reciprocal becomes very large. As the intrinsic oscillatory behavior of the actual error become visible, we observe that the proxy based solely on the smallest eigenvalue (dashed black line) provides a close representation of the actual relative error. Thus, a lot of information can be deduced by simply taking into account the smallest eigenvalue in terms of magnitude. Note the proxy is meant to perform as an estimate of the true relative error and not as an upperbound. In some cases, the bound underestimates the actual relative error.

ONE-DIMENSIONAL DISPERSION CORRECTION

For the 1D case, we use the dispersion correction in equation 5.52. It is also possible to correct each eigenvalue in order to obtain very accurate solutions. However, the results we obtain by using the simple correction with respect to the smallest eigenvalue produces comparable results relative to including the modified wavenumber, which is known to eliminate the pollution error to a satisfactory level. Thus, we start by adding the correction term, which is based on adding terms of the truncation error, to the coefficient matrix A,

$$c = -\hat{\lambda}^{j_{\min}} + \sum_{n=2}^{10} \frac{(-1)^n (j_{\min}\pi)^{2n} h^{2(n-1)}}{(2n)!}. \tag{5.55}$$

This alleviates the mismatch between the exact near zero eigenvalue and the numerical eigenvalue at index $j_{\min}$. As mentioned, recall from Section 5.4 that the pollution error for MP 1 can be eliminated by incorporating a modified wavenumber $\tilde{k}$. The latter represents an explicit correction of the wavenumber with respect to the dispersion error. Consequently, we test for the elimination of pollution by comparing the relative error between the exact and numerical solution after solving the following two systems

$$\tilde{A} = A - \tilde{k}I, \text{ where } \tilde{k} = \sqrt{\frac{2(1-\cos(kh))}{h^2}},$$
$$A_c = A + cI, \text{ where } c = -\hat{\lambda}^{j_{\min}} + \sum_{n=2}^{10} \frac{(-1)^n (j_{\min}\pi)^{2n} h^{2(n-1)}}{(2n)!}.$$

We furthermore denote

$$\hat{u}_{\tilde{k}} : \tilde{A}\hat{u}_{\tilde{k}} = f \text{ and }, \hat{u}_c : A_c\hat{u}_c = f,$$

and

$$e_{\tilde{k}} = \frac{\|u - \hat{u}_{\tilde{k}}\|}{\|u\|}, \; e_c = \frac{\|u - \hat{u}_c\|}{\|u\|}.$$

Table 5.1 contains the results for randomly chosen wavenumbers k between 100 and 1000 using 10 grid points per wavelength ($kh = 0.625$) and approximately 6 grid points per wavelength ($kh = 1$). The latter represents the results of applying the dispersion correction on a very coarse grid. The reason we consider a coarse grid is that in absence of dominating pollution, which has been corrected by either $\tilde{k}$ or c, we should be able to obtain accurate results. The results from Table 5.1 show that using the eigenvalue correction c leads to a significant reduction of the relative error. In some instances it provides even better accuracy than using the adjusted wavenumber $\tilde{k}$. Similar conclusions can be drawn from the results when letting $kh = 1$. While e_c exceeds $e_{\tilde{k}}$ occasionally, we see that $e_{\tilde{k}}$ is more much insensitive to changes in the grid resolution. In particular for $\tilde{k}$, the average error for $kh = 0.625$ appears to be fixed around 0.06, and increases to about 0.18 for $kh = 1$, whereas for $kh = 1$ even further reductions of the error can be obtained by using the eigenvalue correction c.

Table 5.1: Relative error e before and after dispersion correction using the eigenvalue-correction c and $\tilde{k}$ for $kh = 0.625$ (left) and $kh = 1$ (right).

$kh = 0.625$				$kh = 1$			
k	e	$e_{\tilde{k}}$	e_c	k	e	$e_{\tilde{k}}$	e_c
104	0.830	0.067	0.006	168	1.153	0.185	0.092
170	24.732	0.068	0.071	175	1.369	0.186	0.251
175	1.331	0.068	0.088	210	1.091	0.186	0.027
195	6.204	0.068	0.069	222	1.241	0.187	0.061
245	2.836	0.068	0.068	230	1.127	0.186	0.037
249	0.965	0.067	0.016	263	1.607	0.187	0.247
306	6.945	0.068	0.027	265	1.499	0.188	0.283
380	4.085	0.068	0.002	315	19.641	0.188	0.240
498	2.564	0.068	0.033	333	21.482	0.188	0.188
505	1.270	0.068	0.016	337	1.071	0.186	0.048
575	0.991	0.068	0.001	415	1.598	0.188	0.195
584	12.136	0.068	0.070	459	21.213	0.188	0.195
641	1.881	0.068	0.068	461	1.182	0.187	0.046
688	1.000	0.068	0.003	488	1.400	0.187	0.061
720	2.597	0.068	0.011	561	13.429	0.188	0.081
773	1.476	0.068	0.069	594	0.999	0.187	0.013
797	1.318	0.068	0.089	621	18.673	0.187	0.271
814	1.007	0.068	0.006	659	1.638	0.187	0.227
835	1.426	0.068	0.078	820	1.000	0.187	0.002
843	6.106	0.068	0.094	867	21.485	0.188	0.188
922	1.310	0.068	0.033	881	1.501	0.188	0.345
965	1.018	0.068	0.010	882	1.112	0.188	0.044
996	0.995	0.068	0.002	919	1.340	0.188	0.092

5.6.2. TWO-DIMENSIONAL CONSTANT WAVENUMBER MODEL

ERROR ANALYSIS

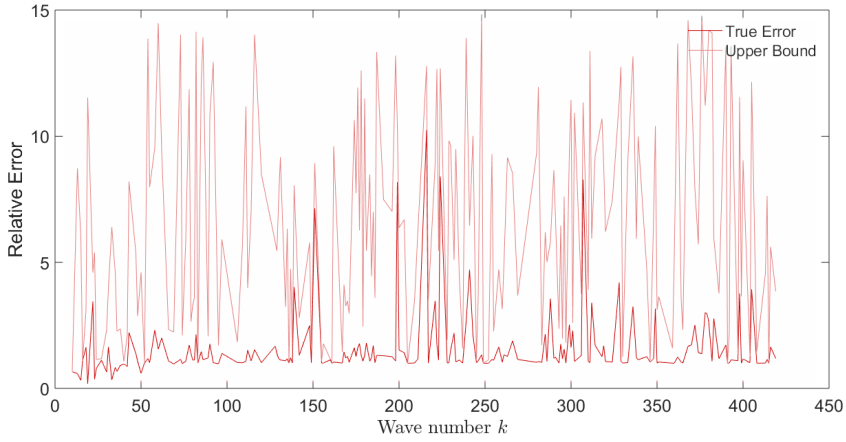
In this section we provide numerical results for MP 2. We start by presenting the error and the upper bound using the eigenvalues in Figure 5.2. To illustrate the pollution effect, we present the solution and error for various examples in Figure 5.3 and 8.20.

Starting with Figure 5.2, we observe that the upper bound always holds. Similar to the one dimensional case, we again observe the oscillatory nature of the actual true error. The spikes in the error provide great insight relative to the linear relation between k and the increasing error. From Figure 5.2 we additionally notice that almost for all k, the relative error is always larger than one. While the upper bound is of the same order as the true error, it is often larger than the true error. Yet, it follows the same oscillatory pattern as the true error from which we can deduce how much each eigenmode contributes to the error. For the first time to our knowledge, we are therefore able to break down and study the dispersive property of the numerical solution in higher-dimensions. The oscillatory error pattern also reveals that the largest contribution in terms of the dispersion can be pointed to the smallest eigenvalues which determine the total sum in Corollary 10.1 and Corollary 10.2.

Secondly, as mentioned previously, in some cases the upper bound is much larger than the actual error. This can be understood by noting that in this model problem the source is located at the center of the numerical domain. Thus, at all even indices j, the sine-term related to the source will be zero and these terms will not be included into the sum. In cases where we see an overshoot, either the smallest numerical or analytical eigenvalue is located

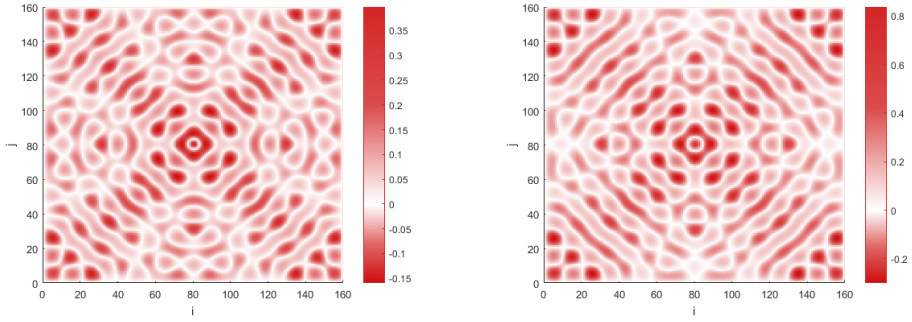
at an even index. While it is not part of the actual error, due to being eliminated by the sine-term containing $\frac{\pi}{2}$, it is in fact still included in our upper bound. Note that in creating the upper bound, we do not differentiate between even and odd indices. The reason for this is that we prefer an upper bound which covers the worst case scenario and is not limited to fixing the location of the point source for this model problem.

Figure 5.2: 2D Relative error for with upper bound for various k between 10 and 425 using $kh = 0.625$.



To illustrate the full pollution effect, we continue by plotting some solutions for several values of k. We have plotted the results for $k = 50$ and $k = 150$ in Figure 5.3 and 8.20. Note that 20 grid points per wavelength are used, which results in $kh = 0.3125$. On the $x-$ and $y-$axis respectively, we have the index i, j corresponding to the gridpoint (x_i, y_j). The colorbar indicates the value of $u(x_i, y_j)$.

Figure 5.3: Exact and numerical solution for MP 2 using second order finite differences and $k = 50$. $kh = 0.3125, n^2 = 25600$.

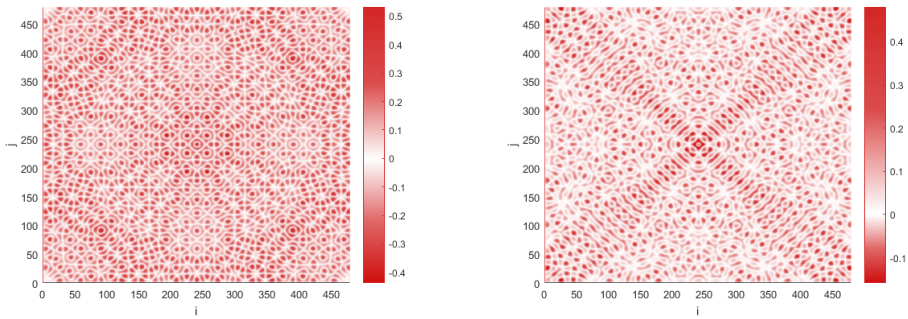


(a) Exact solution

(b) Numerical solution

We can see from Figure 5.3 that for a medium size wavenumber ($k = 50$), the numerical solution is a fair approximation of the exact solution. We can see from the contour of both figures that most of the error does not come from numerical dispersion. If the latter would be the case, the contour of the numerical solution would differ significantly from the exact solution (see Figure 8.20 for example).
We repeat the analysis for a larger wavenumber; $k = 150$. From Figure 8.20 (b) we can see that the accuracy deteriorates rapidly as k increases. Fixing the resolution at $kh = 0.3125$ does not suffice in keeping both the phase and amplitude differences under control. We can see from Figure 8.20 (a) that the exact and numerical solution do not coincide, forcing the conclusion that severe differences between the exact and numerical wavenumber are present. It furthermore supports the observation that increasing the number of grid points mainly results in a substantial resolve of the amplitude differences, rather than the phase differences.

Figure 5.4: Exact and numerical solution for MP 2 using second order finite differences and $k = 150$. $kh = 0.3125, n^2 = 230400$.



(a) Exact solution

(b) Numerical solution

TWO-DIMENSIONAL DISPERSION CORRECTION

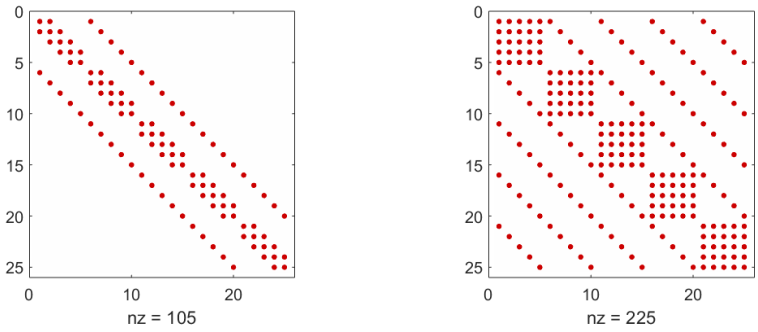
We now investigate the effect of applying a dispersion correction using the eigenvalues for the 2D MP 2. Note that for the 2D case it will not suffice to simply add the constant

$$c = -\hat{\lambda}^{i_{\min}, j_{\min}} + \sum_{n=2}^{10} \frac{(-1)^n (i_{\min}\pi^{2n} + j_{\min}\pi^{2n}) h^{2(n-1)}}{(2n)!}.$$

There may be multiple locations $(i_{\min}, j_{\min})$ where the smallest eigenvalue is located and thus there may be more than one value for c. If the algebraic multiplicity of the smallest eigenvalue is exactly two, then adding the constant c will still reduce the overall error. However, in the 2D case, the algebraic multiplicity may often be larger than two. Therefore, we follow the steps described in Algorithm 7. Given that we are solving for the underlying Green's function and general solution, the property that separation of variables can be applied, results in the fact that we can start correcting the eigenvalues already in the 1D case and use those to construct the new coefficient matrix $\hat{A}$. As this leads to a correction which is independent of the true analytical wavenumber and pre-specified propagation angles,

the resulting coefficient matrix will become more dense and subjected to a different sparsity pattern. In Figure 5.5 we have plotted the sparsity pattern of the corrected coefficient matrix $\hat{A}$ for $k = 10$, using $kh = 1.5$. It is apparent that many diagonals are added to the matrix. Additionally, we can see the formation of clear blocks in the center of the adjusted matrix. For smaller kh, the new coefficient matrix $\hat{A}$ will contain many diagonals and larger blocks, being much more dense yet sparse compared to the original coefficient matrix A.

Figure 5.5: Sparsity pattern for $k = 10$ using $kh = 1.5$



(a) Original matrix A (b) Corrected matrix $\hat{A}$

Before we solve the linear systems explicitly, we verify the 2D dispersion correction. Irrespective of the solution method, we can use the series representation of the discrete solution using the dispersion correction, to establish whether the resulting solution will indeed be dispersion free. Thus, in Table 5.2 we report the results for various k and kh using the dispersion correction on the numerical eigenvalues which we construct from the 1D case. Note that we do not need to compute the 2D eigenvalues and eigenvectors in Algorithm 7 and proceed until step 6 in the algorithm. We note that in almost all cases the true relative error is always larger than 1 without the dispersion correction. Using the new correction for this model problem, the error is reduced significantly and shows relative independence as regards kh. Even when we move to very coarse grids, which will allow for solving the corresponding linear systems accurately and iteratively, the error stays almost constant despite being in the high-frequency range, which to our current knowledge, is a novel theoretical result. For $kh = 2$, $\oslash$ represents a case where the numerical smallest eigenvalue without correction becomes zero and we have resonance. This shows the severity of the dispersion causing the pollution, as the actual analytical eigenvalue is still far away from zero.

Table 5.2: Relative (RE) and corrected relative error (CRE) for various k and kh. ⊘ represents the case where the numerical smallest eigenvalue becomes zero.

k	RE	CRE	RE	CRE	RE	CRE	RE	CRE
	$kh = 0.625$		$kh = 1$		$kh = 1.5$		$kh = 2$	
50	0.599	4.573e-14	0.934	8.270e-12	1.073	1.679e-13	2.367	1.368e-13
100	2.989	2.065e-13	3.408	3.772e-13	2.219	4.949e-13	⊘	1.358e-13
150	2.118	3.878e-14	4.197	7.067e-13	1.518	3.413e-12	4.109	8.293e-13
200	1.525	1.060e-14	6.960	2.805e-13	1.200	2.475e-13	⊘	8.236e-13
250	6.086	1.642e-13	1.758	3.035e-12	1.630	6.911e-13	10.121	1.870e-13
300	1.619	9.541e-13	8.905	3.559e-13	1.529	4.010e-12	⊘	1.000e-13
350	1.005	2.687e-13	1.083	7.340e-13	1.853	9.138e-12	2.176	2.000e-13
400	1.167	2.525e-13	1.058	1.769e-13	8.380	2.353e-13	⊘	7.000e-12
450	2.115	1.975e-13	3.576	3.300e-13	2.073	3.063e-13	5.293	9.104e-13

We now assess the performance in terms of computation time and iterations. In order to make a fair comparison, we solve the linear systems using second-order finite differences using the rule $k^3h^2 = 5$, as this should reduce the pollution error to some extent. We then increase k and report the relative error and number of iterations. From Table 5.2 we observe that we can use coarser grids to solve for the same wavenumber k and we compare the differences. We use GMRES as the iterative solver and apply the standard Complex Shifted Laplacian (CSL) with a complex shift set to 1 using multigrid. We use one V-cycle with one pre- and post-smoothing step. Some important remarks are in place. First of all, the accuracy achieved from the iterative solver will depend on the stopping criterion and we set the relative tolerance at 10^{-6}. Second of all, higher accuracy could have been received of order 10^{-2} by taking k^3h^2 smaller. However, that would lead to large linear systems and thus we report up to $N = 320^2$. Finally, the number of iterations needed to reach convergence for GMRES remains unaffected by the increased accuracy and a detailed study on the convergence behavior lies beyond the scope of this work. For normal matrices in general, GMRES convergence is governed by the smallest eigenvalues in terms of magnitude, in particular the ratio between the smallest and largest eigenvalue. Thus, while the resulting eigenvalues may be more accurate, they may still be small leading to bad convergence.

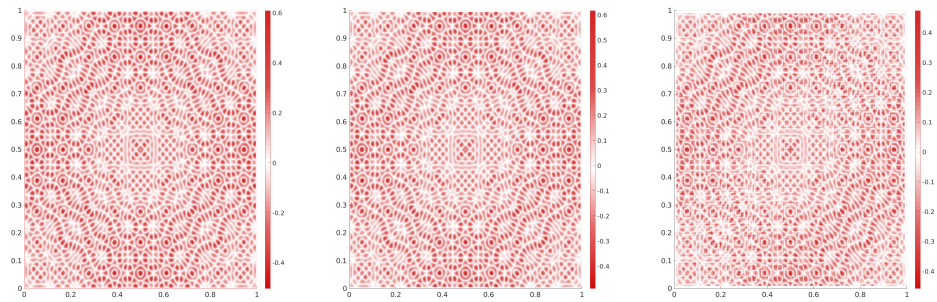
Table 5.3: Exact and numerical solutions for $k = 200$. Exact solution on a fine-grid $kh = 0.625, n^2 = 101761$ and numerical solution on coarse-grids using the eigenvalue dispersion correction. For $kh = 2$, we have $n^2 = 9801$.

	$(A_{2D},\ k^3h^2 \approx 5)$				$(\hat{A}_{2D}\ kh \approx 1)$				$(\hat{A}_{2D}\ kh \approx 2)$			
k	n	RE	Its	CPU(s)	n	RE	Its	CPU(s)	n	RE	Its	CPU(s)
10	15	0.046	18	0.094	10	1.571e-08	13	0.066	5	1.010e-09	6	0.052
20	40	0.083	53	0.247	20	4.842e-07	64	0.178	10	5.192e-08	15	0.072
40	114	0.291	111	6.726	40	8.060e-09	225	2.763	20	2.685e-08	217	0.589
60	208	0.522	377	113.888	60	4.991e-07	480	35.072	10	3.981e-07	464	3.653
80	320	1.861	654	1386.827	80	3.823e-07	712	151.486	40	6.123e-07	901	18.845

Table 5.3 sheds light on some interesting observations made previously. Using the dispersion correction, we can solve for the same wavenumber k while using coarser grids which lead to smaller linear systems. This is beneficial as this implies that the theoretical study of the pollution error can now be studied from all angles simultaneously in higher-dimensions using coarser systems. If, for example, we look at $k = 80$, we note that even with $N = 320^2$

(equivalent to using 27 grid points per wavelength ($k^3h^2 \approx 5$)), the error keeps increasing and even finer grids are required to obtain accurate solutions. Moreover, the standard iterative solver needs 654 iterations and approximately 1386 seconds to reach convergence. On the contrary, using $N = 40^2$, which is equivalent to using 3 grid points per wavelength ($kh \approx 2$), the error is reduced to 10^{-7}.
In Figure 5.6 we have plotted the exact solution for $k = 200$ on a fine grid and compare it to the numerical solution computed on a very coarse grids using the eigenvalue based dispersion correction. We can see that the accuracy and resolution for such a high wavenumber computed on a very coarse grid ($kh = 2$) are still satisfactory. The figures illustrate what we observed for $k = 200$ in Table 5.2; the error, after introducing the dispersion correction, at its best is of order 10^{-14} and at its worse of order 10^{-13}. Even for a simple model problem such as ours, achieving an explicit dispersion correction independent of the propagation angle in higher-dimensions is unprecedented.

Figure 5.6: Exact and numerical solutions for $k = 200$. Exact on a fine-grid $kh = 0.625, n^2 = 101761$ and the numerical on coarse-grids using the eigenvalue dispersion correction. For $kh = 2$, we have $n^2 = 9801$.



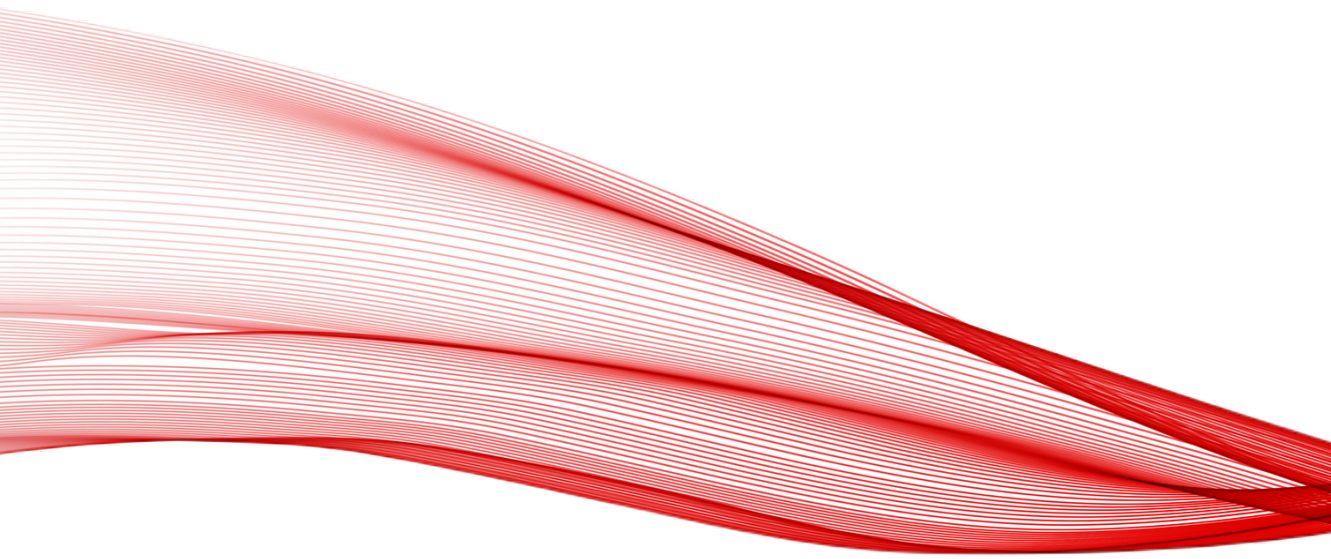
(a) Exact solution for $kh = 0.625$ (b) Numerical solution for $kh = 1$ (c) Numerical solution $kh = 2$

5.7. CONCLUSION

In this chapter we researched the pollution error due to numerical dispersion for the Helmholtz problem using Dirichlet conditions from an unconventional and novel perspective; the eigenvalues. We have sought to provide the first theoretical basis for defining the pollution error in terms of the eigenvalues. This can allow further study of the relation between iterative solvers and the accuracy of numerical solutions now that both have been expressed in terms of a common denominator; the near-zero eigenvalues. This is especially interesting due to the fact that these near-zero eigenvalues, which are generally responsible for hampering the convergence of iterative solvers, are in fact indicators for the pollution effect. Furthermore, by examining the behavior of the eigenvalues, we proposed an upper bound for the relative error. In particular, we showed that if the near-zero eigenvalues and eigenvectors are approximated with high accuracy, then the dispersion part of the pollution error can be minimized considerably. The results also illustrate that the error grows in an oscillatory manner, and the error bound is able to capture and reveal this effect. We additionally constructed a theoretical framework where the pollution error can be brought to approxi-

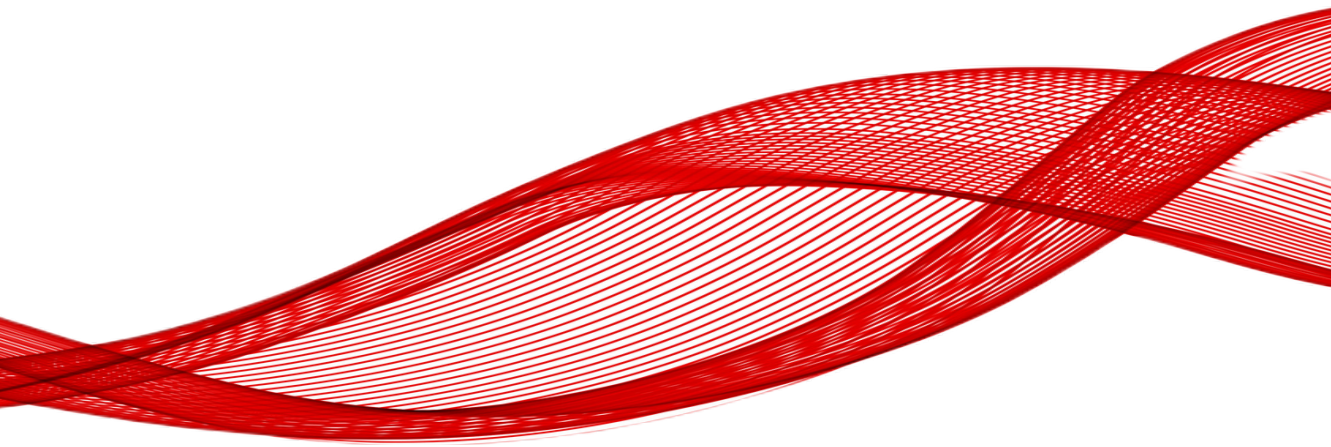
mately zero for very large wavenumbers, irrespective of the grid resolution (kh). The basis of this approach lies in correcting the respective eigenvalues with the remainder, which depends on the order of the truncation error of the finite difference scheme. Consequently, it is possible to obtain pollution-free and therefore accurate one- and 2D solutions using coarser grids. The solutions obtained account for all propagation angles simultaneously and do not rely on pre-determined angles for plane-wave propagation, which promotes a detailed study of the pollution effect.

5



6

Error Minimization



Parts of this chapter have been published in Computer Methods in Applied Mechanics and Engineering **377**, (2021) [64].

In the previous chapter we have discussed the pollution error. We showed a novel method to investigate the pollution error in higher dimensions using information from the eigenvalues. While we are able to enforce a correction for simple theoretical problems, the question remains how to tackle more difficult problems and geometries.
One potential way to mitigate the pollution error for these types of problems is to adopt Isogeometric Analysis (IgA) [65] as a discretization technique. IgA can be considered as the natural extension of the finite element method (FEM) to higher-order B-splines and has become widely accepted as a viable alternative to standard FEM. The use of high-order B-splines or Non-Uniform Rational B-splines (NURBS) enables a highly accurate representation of complex geometries and bridges the gap between computer-aided design (CAD) and computer-aided engineering (CAE) tools.

6.1. ISOGEOMETRIC ANALYSIS

We first provide a brief introduction to IgA and an overview of the literature on this topic. We then apply this to our model problem and investigate the behavior of the pollution error using this novel discretization technique.

6.1.1. VARIATIONAL FORMULATION

To illustrate the variational formulation, we consider the inhomogeneous Helmholtz equation in two dimensions adopting inhomogeneous Robin boundary conditions:

$$-\Delta u(x,y) - k^2(x,y)u(x,y) = f(x,y), \quad (x,y) \in \Omega \subset \mathbb{R}^2, \tag{6.1}$$

$$\left(\frac{\partial}{\partial \mathbf{n}} - ik(x,y)\right) u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega. \tag{6.2}$$

Here, Ω is a connected Lipschitz domain, $f \in L^2(\Omega)$, $g \in L^2(\partial\Omega)$ and $k(x,y)$ a non-constant wavenumber. Let us define $\mathcal{V}$ as the first order Sobolev space $H^1(\Omega)$. The variational formulation of (6.1) is obtained by multiplication with a test function $v \in \mathcal{V}$ and application of integration by parts: Find $u \in \mathcal{V}$ such that

$$a(u,v) = (f,v), \; \forall v \in \mathcal{V}, \tag{6.3}$$

where

$$a(u,v) = \int_\Omega \nabla u \cdot \overline{\nabla v}\, \mathrm{d}\Omega - \int_\Omega k^2 u\overline{v}\, \mathrm{d}\Omega - i\int_{\partial\Omega} ku\overline{v}\, \mathrm{d}\Gamma \qquad (f,v) = \int_\Omega f\overline{v}\, \mathrm{d}\Omega + \int_{\partial\Omega} g\overline{v}\, \mathrm{d}\Gamma. \tag{6.4}$$

A geometry function $\mathbf{F}$ is then defined to parameterize the physical domain Ω by describing an invertible mapping to connect the parameter domain $\Omega_0 = (0,1)^2$ with the physical domain Ω.

$$\mathbf{F} = \Omega_0 \to \Omega, \quad \mathbf{F}(\xi,\eta) = (x,y). \tag{6.5}$$

The considered geometries throughout this paper can be described by a single geometry function $\mathbf{F}$, that is, the physical domain Ω is topologically equivalent to the unit square. In case of more complex geometries, a family of functions $\mathbf{F}^{(m)}$ $(m = 1,\ldots,K)$ is defined and we refer to Ω as a multipatch geometry consisting of m patches. For a more detailed description of multipatch constructions, the authors refer to chapter 2 of [66].

6.1.2. B-SPLINE BASIS FUNCTIONS

To discretize Equation (6.1), univariate B-spline basis functions are defined on the parameter domain Ω_0 by an underlying knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{N+p}, \xi_{N+p+1}\}$, where the knots are located at the interval boundaries. Here, N denotes the number and p the order of the B-spline basis functions. Based on this knot vector, the basis functions are defined recursively by the Cox-de Boor formula [67], starting from the constant ones

$$\phi_{j,0}(\xi) = \begin{cases} 1 & \text{if } \xi_j \leqslant \xi < \xi_{j+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.6}$$

Higher-order B-spline basis functions of order $p > 0$ are then defined recursively

$$\phi_{j,p}(\xi) = \frac{\xi - \xi_j}{\xi_{j+p} - \xi_j}\phi_{j,p-1}(\xi) + \frac{\xi_{j+p+1} - \xi}{\xi_{j+p+1} - \xi_{j+1}}\phi_{j+1,p-1}(\xi). \tag{6.7}$$

The resulting B-spline basis functions $\phi_{j,p}$ are non-zero on the interval $[\xi_j, \xi_{j+p+1})$ and possess the partition of unity property. Furthermore, the basis functions are C^{p-m_j}-continuous, where m_j denotes the multiplicity of knot ξ_j. Throughout this paper, we consider a uniform knot vector with knot span size h, where the first and last knot are repeated $p+1$ times. As a consequence, the resulting B-spline basis functions are C^{p-1} continuous and interpolatory at both end points.
Figure 6.1 illustrates both linear and quadratic B-spline basis functions based on such a knot vector.

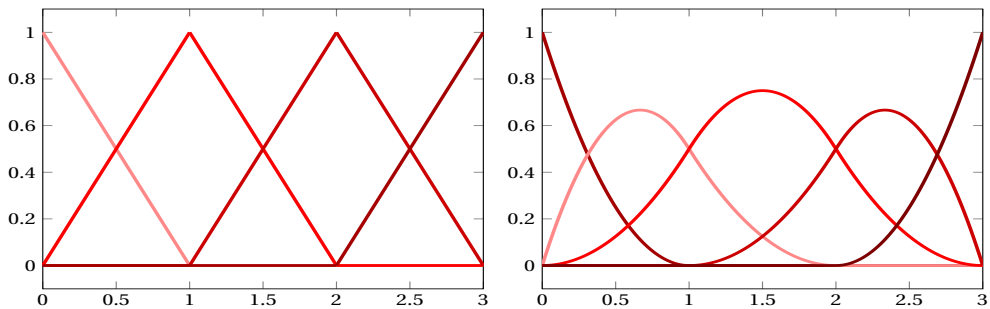


Figure 6.1: Linear and quadratic B-spline basis functions based on the knot vectors $\Xi_1 = \{0,0,1,2,3,3\}$ and $\Xi_2 = \{0,0,0,1,2,3,3,3\}$, respectively.

6.1.3. LINEAR SYSTEM FORMULATION

For the multi-dimensional case, the tensor product of univariate B-spline basis functions is adopted for the spatial discretization. Let N_{dof} denote the total number of multivariate basis functions $\Phi_{j,p}$. The spline space $\mathcal{V}_{h,p}$ can then be written as follows

$$\mathcal{V}_{h,p} = \text{span}\{\Phi_{j,p} \circ \mathbf{F}^{-1}\}_{j=1,\ldots,N_{\text{dof}}}. \tag{6.8}$$

The Galerkin formulation of (6.3) now becomes: Find $u_{h,p} \in \mathcal{V}_{h,p}$ such that

$$a(u_{h,p}, v_{h,p}) = (f_{h,p}, v_{h,p}),\ \forall v_{h,p} \in \mathcal{V}_{h,p}. \tag{6.9}$$

The discretized problem in (6.9) can be written as a linear system

$$\left(K_{h,p} + M_{h,p} + R_{h,p}\right) u_{h,p} = f_{h,p}, \tag{6.10}$$

where we have

$$K_{h,p} = \left[\int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}\Omega\right]_{1\leqslant i,j\leqslant N_{\text{dof}}}, \quad M_{h,p} = -k^2\left[\int_\Omega \varphi_i\varphi_j \, \mathrm{d}\Omega\right]_{1\leqslant i,j\leqslant N_{\text{dof}}}, \tag{6.11}$$

and

$$R = -ik\left[\int_\Gamma \varphi_i\varphi_j \, \mathrm{d}\Gamma\right]_{1\leqslant i,j\leqslant N_{\text{dof}}}. \tag{6.12}$$

Next, by defining $A_{h,p} = K_{h,p} + M_{h,p} + R_{h,p}$ we can write

$$A_{h,p}\, u_{h,p} = f_{h,p}. \tag{6.13}$$

For the ease of notation, we proceed with the notation $Au = f$, and drop the subscript (h,p).

6.1.4. LITERATURE OVERVIEW

IgA can be considered as the natural extension of FEM to higher-order B-splines and has become widely accepted as a viable alternative to standard FEM. The use of high-order B-splines or NURBS enables a highly accurate representation of complex geometries and bridges the gap between computer-aided design (CAD) and computer-aided engineering (CAE) tools. Furthermore, a higher accuracy per degree of freedom can be achieved compared to standard FEM [68]. A new branch of studies has demonstrated that IgA furthermore helps to control the pollution error while keeping the size of the resulting linear system moderate [69–73]. In [74], the authors investigated the obtained accuracy for several Helmholtz-type problems using a non-constant wavenumber and documented increased accuracy. Thus, while the use of IgA for Helmholtz-type problems becomes more established, the process of solving the underlying discretized systems remained fairly untouched. Until recently, a study by Diwan et al. [75] covered this for the Helmholtz equation and researched the use of IgA together with an iterative solver. In this section we mainly focus on the use of IgA to reduce the pollution error. For iterative solvers, see Part III.

6.1.5. RELATION TO FEM

In this section we briefly mention some similarities and differences between both methods. The basic concept behind IgA is to use same basis functions to model the exact geometry and the solution field. As a result, the exact geometry is employed at all levels of discretization, while in FEM, a piecewise polynomial approximation is utilized.

In both IgA and FEM, the solution of the weak form is a linear combination of the basis functions. In IgA, the coefficients are the control variables, while in FEM they are the nodal variables. In IgA, control points and control variables are generally not interpolated, unlike the nodal points and variables in FEM.

In both IgA and FEM, the bases being used forms a partition of unity and the bandwidth of matrices corresponds to the given polynomial order and are equal.

In FEM, the degrees of freedom are located at the nodes, while in IgA they are located at the control points. In FEM the continuity of the basis functions are fixed, while in IgA the

continuity can be controlled. Fig. 6.2 sums up some of the differences and similarities. For more details about IgA, we refer to [65, 68].

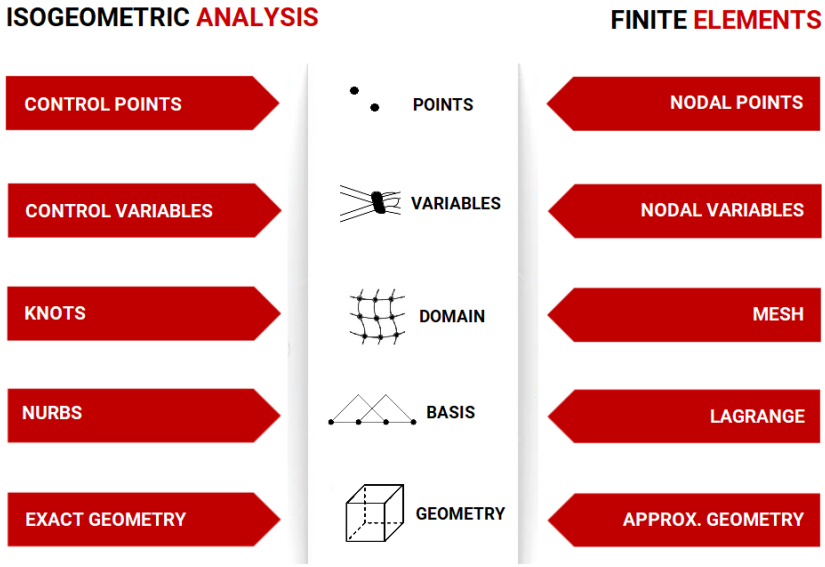


Figure 6.2: Comparison of IgA and FEM.

6.2. PROBLEM DEFINITION

To illustrate the effect of using IgA with respect to the pollution error, we use a 1D model problem, denoted by MP 1-A, from [76].

MP 1-A

$$\begin{aligned} -\frac{d^2u(x)}{dx^2} - k^2\,u(x) &= 1, \quad x \in \Omega = (0,1), \\ u(x) &= 0, \quad x = 0, \\ u'(x) - iku(x) &= 0, \quad x = 1. \end{aligned} \tag{6.14}$$

Here, homogeneous Dirichlet and Sommerfeld boundary conditions are applied on the left and right boundary, respectively. The exact solution for MP1-A is given by $u(x) = e^{ikx}$. Model problem MP 1-A will be adopted to investigate the pollution error for various values of the approximation order p of the B-spline basis functions.

6.2.1. POLLUTION ERROR

In this section we briefly discuss the effects of using IgA on the pollution error for the Helmholtz equation. As mentioned previously, the h-version of the error studies have shown that as the wavenumber k increases, the numerical solution suffers from dispersion errors [38, 39]. While in 1D, one can define an exact modified wavenumber which is able to minimize and

bound the pollution error, this is not possible in 2D and 3D as this relies on the direction of the waves [38, 39]. Thus, instead of resorting to very fine meshes, it has been shown that higher-order methods suffer from less dispersion error and provide a viable alternative to obtain accurate solutions while keeping the problem size economical [76, 77]. In particular, Corollary 4.6 of [76] provides us with the following h-error estimate (given $||u||_{L^2} \sim 1$)

$$||u_{ex} - u_h||_{L^2} \leqslant C\left(h^p + k^{-1}(kh)^p\right) h\left(1 + k(kh)^{p-1}\right). \tag{6.15}$$

Note that for $p > 1$, the error decreases asymptotically faster compared to $p = 1$ where the error scales at best with k. In order to illustrate these properties, we plot the L^2-error under mesh refinement for MP 1-A. Figure 6.3 shows the L^2-error under mesh refinement for different values of k obtained for $p = 1$ (left) and $p = 2$ (right). Note that, the k-dependence for $p = 1$ significantly differs from $p = 2$, as predicted in Corollary 4.6 in [76]. In fact, the numerical results presented in [76] (see Figure 2), showing the relative L^2-error under mesh refinement, are in agreement with the results presented in Figure 6.3.
While the use of IgA significantly reduces the pollution error, they do not remain pollution-free as the wavenumber becomes very large [77]. We illustrate this using the 'rule of thumb', where the waves are resolved using 10 degrees of freedom per wavelength. Note that this has been used widely in practice and lies within the pre-asymptotic range for $p = 1$. In Figure 6.4 we observe that, using $kh = 0.625$ for $p = 2$ to $p = 5$, the L^2-error with respect to the analytical solution decreases. While this leads to significant more accurate solutions, we do observe that as the wavenumber increases, the L^2-error increases accordingly. Moreover, as k increases the advantage of using $p = 5$ over $p = 4$ decreases as both lead to similar accuracy. For standard FEM, this was already observed [59]. Furthermore, decreasing the number of degrees of freedom per wavelength from 10 (solid line) to 7.5 (dashed line) already results in lower accuracy. In fact, the achieved accuracy for $p = 4$ and $p = 5$ with 7.5 degrees of freedom per wavelength is similar to the obtained accuracy for $p = 3$ when 10 degrees of freedom per wavelength are used. We proceed by keeping $kh = 0.625$ and increasing the order p as we want to examine the extent of the iterative solver within this pre-asymptotic range. However, note that for engineering practices, the error can be bounded in the 0.1 to 1% range, where IgA can provide more accurate solutions using smaller linear systems [77].

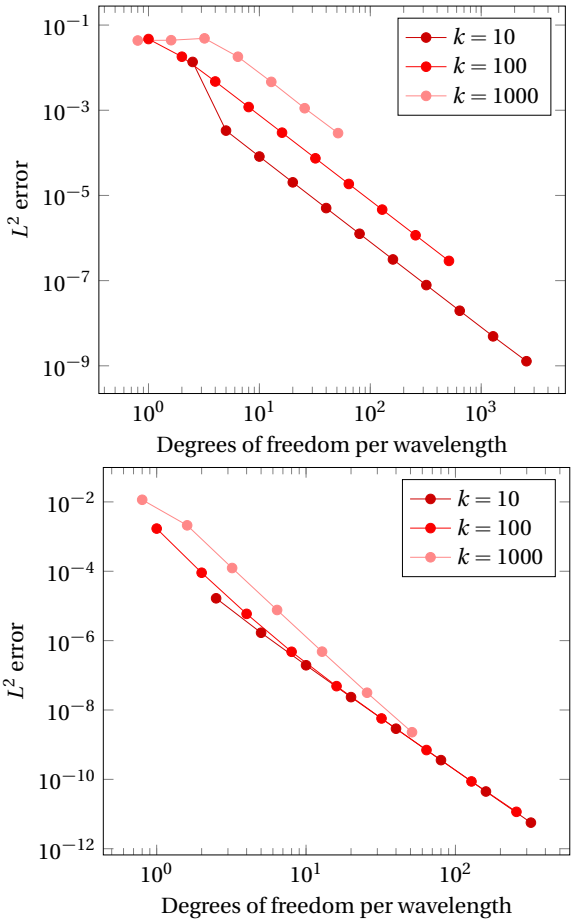


Figure 6.3: L^2-error under h-refinement for MP 1-A using $p = 1$ (top) and $p = 2$ (bottom) for different wavenumbers.

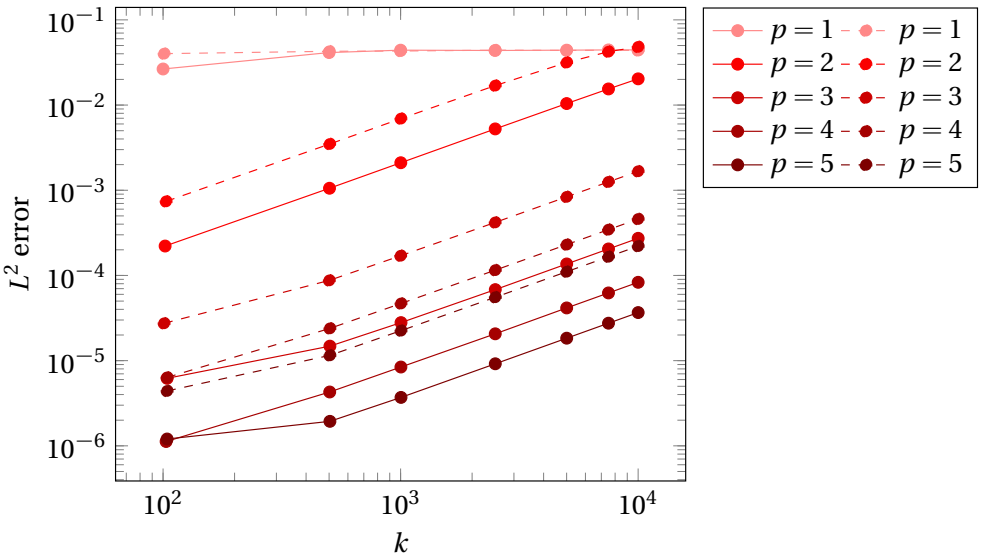


Figure 6.4: L^2-error for MP 1-A using $p = 1$ to $p = 5$ for various wavenumbers k. The solid line uses 10 degrees of freedom per wavelength ($kh = 0.625$) and the dashed line uses 7.5 degrees of freedom per wavelength ($kh = 0.825$).

6.3. CONCLUSION

In this chapter, we studied the combination of IgA discretized linear systems for the Helmholtz equation. In particular, we showed that the use of IgA reduces the pollution error significantly compared to $p-$order FEM. However, the pollution error can not be removed completely and continues to grow with the wavenumber k, unless more degrees of freedom are used. Additionally, obtaining better accuracy by increasing the order p comes at the cost of more dense matrices. Depending on the application and the required level of accuracy, IgA can provide more accurate solutions using smaller linear systems compared to $p-$order FEM.

III

Numerical Iterative Solvers

7

DEFLATION

Parts of this chapter have been published in SIAM Journal on Scientific Computing **42**, (2020) [78].

In this chapter we start by introducing the deflation method and how it's going to be used as a preconditioner to accelerate the solvers from Chapter 3. There we discussed the commonly used preconditioner for the Helmholtz equation, the Complex Shifted Laplacian (CSL). We noted that while it leads to convergence which is linear in the number of iterations relative to the wavenumber, the number of iterations starts increasing rapidly for very large wavenumbers. This manifests itself in near-zero eigenvalues clustering up near the origin, causing the convergence to slack. The main reason for using deflation as a preconditioning strategy is that it allows for the near-zero eigenvalues to be removed, which are hampering the convergence of the Krylov solver. We again start by an introduction to the deflation technique, followed by an overview of the literature, where we discuss deflation based preconditioning strategies which have been studied previously. Next, we provide an overview of the model problems which will be studied in this chapter. We proceed by investigating theoretically what the main reason is behind this inscalability and introduce a novel method which relies on higher-order deflation spaces. Finally, we conclude this chapter with some numerical experiments.

7.1. DEFLATED KRYLOV METHODS

In Part I of this dissertation, we have shown that discretization of the Helmholtz equation leads to an indefinite matrix. This indefiniteness narrows the choice of potential Krylov-based solvers due to the Conjugate Gradient type methods being ineffective.

While the application of the CSL preconditioner was successful in confining the eigenvalues between 0 and 1 in the complex plane, the Krylov solver remains defenseless against the hampering convergence behavior caused by the small eigenvalues for large k, which is why deflation was introduced to boost the convergence behavior of the Krylov solver.

Deflation is a technique which aims to move near-zero eigenvalues to zero by using an orthogonal projection. It can also be used to move these unwanted eigenvalues to 1 or the largest eigenvalue. In both cases, the eigenvalues are mapped to the desired value if the exact eigenvectors are utilized. Due to practical considerations within the context of Krylov solvers, it is possible to alleviate the adverse effects of near-zero eigenvalues using deflation by either explicitly modifying the operator of the linear system [79] or by finding approximations to the eigenvectors corresponding to the troublesome eigenvalues. For example, such approximations are used in [11] and [80], where harmonic Ritz vectors serve as eigenvector approximations to augment the Krylov subspace in order to gain faster convergence. Deflation for large scale problems relies on multiplying the linear system by a projection matrix P and applying the Krylov subspace method to the projected system PA, rendering the projection matrix P to act as a preconditioner at the same time.

7.1.1. DEFLATION BASED PRECONDITIONING FOR GMRES

Consider a general real valued linear system. The projection matrix $\widehat{P}$ and its complementary projection P can be defined as

$$\begin{aligned} &\widehat{P} = AQ \text{ where } Q = ZE^{-1}Z^T \text{ and } E = Z^TAZ \\ &A \in \mathbb{R}^{n\times n},\ Z \in \mathbb{R}^{m\times n} \\ &P = I - AQ \end{aligned} \tag{7.1}$$

where Z functions as the deflation matrix whose $m < n$ columns are considered the deflation vectors and I is the $n \times n$ identity matrix. Additionally, the coarse-grid coefficient matrix E is assumed to be invertible. Matrix P is also known as the projection preconditioner. In Algorithm 8 we present the Preconditioned Deflated GMRES algorithm, which will be used for numerical testing in Section 8.6 and includes a preconditioner matrix M. The preconditioner M is added to improve the convergence.

Algorithm 8: Preconditioned Deflated GMRES for system $Au = b$

Initialization:
Choose u_0 and compute $r_0 = b_0 - Au_0$ and $v_1 = r_0/||r_0||$
for $j = 1, 2, ...k$ **do**
  $\tilde{v}_j := Pv_j$
  $w = M^{-1}A\tilde{v}_j$
  for $i := 1, 2, ..., j$ **do**
    $h_{i,j} := w^T v_i$
    $w := w - h_{i,j} v_i$
  end
  $h_{j+1,j} := ||w||$
  $v_{j+1} := w/h_{j+1,j}$
end
Store:
$V_k = \left[\tilde{v}_1, ..., \tilde{v}_k\right]$;
$H_k = \{h_{i,j}\}$, $1 \leqslant i \leqslant j+1$, $1 \leqslant j \leqslant m$.
Compute:
$h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.
The entries of upper $k+1, k$ Hessenberg Matrix H_k are the scalars $h_{i,j}$.
Form approximate solution:
$\mathbf{u_k = Qb + P^T u_k}$
Restart:
If satisfied stop, else set $u_0 \leftarrow u_k$ and repeat process.

7.1.2. THE DEFLATION PRECONDITIONER (DEF)

Based on the above, the DEF-preconditioner has been defined by taking the prolongation operator I_{2h}^h from a multigrid setting as the deflation subspace Z in equation Eq. (7.1), see Section 4.1.1.
I_{2h}^h can be interpreted as interpolating from grid Ω_{2h} to grid Ω_h. As a result, the DEF-preconditioner is commonly referred to as a two-level method and we obtain

$$\widehat{P} = A_h Q \text{ where } Q = Z {A_{2h}}^{-1} Z^T \text{ and } A_{2h} = Z^T A_h Z \qquad (7.2)$$
$$P = I_h - A_h Q \text{ where } Z = I_{2h}^h$$

In the literature a distinction is made with respect to the two-level deflation operator. On the one hand we have the DEF-preconditioner as defined above. On the other hand we have the ADEF-preconditioner, which is defined by taking $P_{ADEF} = P + \gamma Q$. The inclusion of the shift γ ensures that the coarse-grid solve with respect to A_{2h} can be approximated, for example by considering a multi-level implementation ([81], [28]). When considering approximate inversion, γ is generally either set to 1 or the largest eigenvalue of the original

coefficient matrix. In this work we solely focus on the DEF-preconditioner in a two-level setting, and thus we take $\gamma = 0$. This implies that on the coarsest level, the solution for the system involving A_{2h} is solved with a direct method.
As for the preconditioner M given in Algorithm 8, we use the CSLP-preconditioner, which is defined by

$$M = -\Delta - (\beta_1 + \beta_2 i)k^2 I,$$

where $i = \sqrt{-1}$ and $(\beta_1, \beta_2) \in [0,1]$. The CSL preconditioner is included in order to obtain a more favourable spectrum. Unless stated otherwise, we use one $V(1,1)$-multigrid cycle to obtain an approximate inverse of the CSLP-preconditioner.

7.2. PROBLEM DESCRIPTION

In this section we define the model problems which are used to both theoretically and numerically study the deflation based solver.As mentioned previously, using Dirichlet boundary conditions, the resulting coefficient matrix is normal and hence GMRES-convergence after preconditioning is completely determined by the spectrum. While this allows for extensive analysis of the convergence behavior, no true wavenumber independent convergence has been reported for this model problem unless the shift in the CSLP-preconditioner is kept very small and exact inversion is utilized [32]. This motivates to start with the study of this simple model problem in order to create a foundation for obtaining wavenumber independent convergence.

7.2.1. ONE-DIMENSIONAL CONSTANT WAVENUMBER MODEL

We start by focusing on a one-dimensional mathematical model using a constant wavenumber $k > 0$.

MP 1-A

$$\begin{aligned} -\frac{d^2u}{dx^2} - k^2\,u(x) &= \delta(x-x'), \quad x \in \Omega = [0,L], \\ u(x) &= 0, \quad x = 0, \\ u(x) &= 0, \quad x = L. \end{aligned} \tag{7.3}$$

We refer to this model problem as MP 1-A. Next, we introduc MP 1-B as the model problem where Sommerfeld radiation conditions have been implemented instead of Dirichlet conditions.

MP 1-B

$$\begin{aligned} -\frac{d^2u}{dx^2} - k^2\,u(x) &= \delta(x-x'), \quad x \in \Omega = [0,L], \\ u(x) &= 0, \quad x = 0, \\ u'(x) - iku(x) &= 0, \quad x = L, \end{aligned} \tag{7.4}$$

DISCRETIZATION

For both model problems, discretization using second order finite differences with stepsize $h = \frac{1}{n}$ leads to

$$\frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} - k^2 u_j = f_j, j = 1, 2, \ldots, n.$$

The linear system can be formulated exclusively on the internal grid points due to the homogeneous Dirichlet boundary conditions. We obtain the following linear system and eigenvalues with indices $j = 1, 2, \ldots n$

$$\begin{aligned} Au &= \frac{1}{h^2}\text{tridiag}[-1 \;\; 2 - k^2h^2 \;\; -1]u = f, \\ \hat{\lambda}^j &= \frac{1}{h^2}\left(2 - 2\cos(j\pi h)\right) - k^2. \end{aligned} \tag{7.5}$$

7.2.2. TWO- AND THREE-DIMENSIONAL CONSTANT WAVENUMBER MODEL

In order to investigate the scalability of the convergence in higher dimensions (Section 8.6), we define MP 2 and MP 3 to be the 2-D and 3-D versions of the 1D model problem MP 1-A defined above Section 8.3.0.2. The discretization using second order finite differences with a lexicographic ordering goes accordingly for higher dimensions. The resulting linear system matrices are penta- and hepta-diagonal for 2D and 3D respectively.

7.2.3. MARMOUSI MODEL

The final test problem is a representation of an industrial problem and is widely referred to as the 2D Marmousi Problem, which we denote by MP 4. We consider an adapted version of the original Marmousi problem developed in [28]. The original domain has been truncated to $\Omega = [0, 8192] \times [0, 2048]$ in order to allow for efficient geometric coarsening of the discrete velocity profiles given that the domain remains a power of 2. The original velocity $c(x, y)$ is also adapted by letting $2587.5 \leqslant c \leqslant 3325$. We use the adjusted domain in order to benchmark against the results from [28]. In the adjusted domain Ω, we define

MP 4

$$\begin{aligned} -\Delta u(x,y) - k^2(x,y)u(x,y) &= \delta(x - 4000, y - y'), \quad (x,y) \in \Omega \subset \mathbb{R}^2, \\ \left(\frac{\partial}{\partial \mathbf{n}} - ik(x,y)\right) u(x,y) &= 0, \quad (x,y) \in \partial\Omega, \end{aligned}$$

where n denotes the outward normal unit vector. The discretization has been performed using the same second order finite difference scheme. Note that we now have a non-constant wavenumber $k(x, y) = \frac{2\pi freq}{c(x,y)}$, where the frequency is given in Hertz.

7.3. LITERATURE OVERVIEW

For the Helmholtz equation in particular, the first strategy to consider a deflation preconditioner was proposed in [27]. After this work, several subsequent works have studied the combination of applying deflation and the CSL as a preconditioner for the Helmholtz equation, see [28–30]. Compared to the convergence performance of the CSLP, the inclusion of a

deflation based preconditioner allowed for improvement and reduced the numbers of iterations significantly. However, for very large wavenumbers and for 2D and 3D problems, the near-zero eigenvalues kept reappearing. This in its turn was immediately translated into an increased number of iterations to reach convergence.

Moreover, a distinction is made with respect to the two-level deflation operator. In practice, the preconditioner P is adapted by considering $P + \gamma Q$. The inclusion of the shift γ ensures that the coarse-grid solve with respect to E can be approximated, for example by considering a multi-level implementation [28, 81, 82] or using an iterative solver to obtain a solution to the term containing E^{-1}. When considering approximate inversion, γ is generally either set to 1 or the largest eigenvalue of the original coefficient matrix.

We mentioned previously that the deflated GMRES algorithm also includes a preconditioner M, which is applied to further accelerate convergence. In the literature a distinction is made between 'first precondition, then deflate' and 'first deflate, then precondition'. Spectral analysis conducted in [29, 83] has shown that for the Helmholtz problem, the performance is the same. It must be noted that in case of 'first deflate, then precondition', the deflation preconditioner P should include γQ in order to ensure stability when inexact solves for E^{-1} are performed.

More recent preconditioners use polynomial smoothing techniques to accelerate convergence [84]. A different approach can be found by using preconditioning techniques based on domain decomposition methods applied to the corresponding (shifted) problem, which is largely based on the work in [25]. These methods split the computational domain in subdomains and solve a local subproblem of smaller dimension using a direct method [85–91]. The performance of these preconditioners depends on the accuracy of the transmission conditions, which currently is robust for constant wavenumber model problems [92, 93]. While this resulted in a reduced number of iterations, the number of iterations still mildly grows with the constant wavenumber k.

7.3.1. EFFECT OF NON-NORMALITY

By assuming Dirichlet boundary conditions for our first model problem, we are able to simplify the analysis and perform rigorous Fourier analysis, which shows that the new scheme is able to align the near-zero eigenvalues of the fine- and coarse-grid coefficient matrix. Having a higher-order approximation scheme for the deflation vectors enables us to reach wavenumber independent convergence in 1D and close to wavenumber independent convergence in 2D and 3D for very large wavenumbers. The difficulty in using Sommerfeld radiation conditions is that the resulting coefficient matrix becomes complex and non-normal. Therefore, there are no closed-form expressions for the eigenvalues. Additionally, it has been noted that in case of non-normal matrices, spectral analysis becomes less meaningful in order to assess convergence properties [94, 95]. If A is normal, then the condition number of the eigenvector matrix is one. In case of a non-normal diagonalizable matrix, the condition number of the eigenvector matrix is larger than one. As a result it has been shown that arbitrary matrices can be created with arbitrary eigenvalues and right-hand sides which give the same GMRES residual norms [94]. While this often has been interpreted as 'spectral analysis for a non-normal matrix is insufficient', the original authors also mentioned that even for a matrix which is far from normal, GMRES can converge very well and the eigenvalues can still primarily govern its convergence in some specific cases. For example it may be the case that the eigenvector matrix is well conditioned, A is close to Hermitian

despite having ill-conditioned eigenvectors or zero is outside the Field of Values (FOV) of A. While the latter approach has received great attention in the past years to explain convergence behavior of the Helmholtz equation, its use is very sensitive to having zero inside the FOV, which often seems to be the case for indefinite systems [96]. A more recent and detailed analysis showed that the dependence on the condition number of the eigenvectors is often a large overestimation of the actual error [7]. In fact, it has been shown that for diagonalizable matrices, eigenvalues close to the origin indeed hamper GMRES-convergence and GMRES-convergence does not explicitly depend on the condition number of the eigenvector matrix [10]. While the latter may be large, convergence is still predominantly governed by the eigenvalues if the eigenvector matrix is not too far from unitary. Similarly for non-diagonalizable matrices such as a highly non-normal single, plain Jordan block, GMRES-convergence can still be strongly governed by an eigenvalue with large modulus [7, 10, 96–98]. An important implication of this for a diagonalizable matrix is that convergence for a non-normal A can behave as convergence for a normal A. While the literature does not quantify terms as a 'small' condition number or 'not too far from normality/unitary' for this particular application, there exist vast numerical evidence showing that clustering the spectrum leads to better GMRES-convergence. This corroborates the acceleration of GMRES-convergence using deflation preconditioning techniques [11, 99–101]. In fact, in [101] the authors state that "deflated GMRES can be effective even when the eigenvectors are poorly defined .. and for highly non-normal matrices", where convergence is boosted after removing small (pseudo)eigenvalues. Therefore, in order to fully understand the efficiency of our proposed deflation preconditioner, we start conducting spectral and convergence behavior analysis of the proposed preconditioner for the normal case. We then provide numerical evidence to investigate the performance of the preconditioner for non-normal problems.

7.4. INSCALABILITY AND SPECTRAL ANALYSIS

We now start shifting our focus towards the spectral analysis by studying the eigenvalues of the DEF-operator without inclusion of CSLP. To study the eigenvalues, we use the analytical derivations and expressions for the spectrum of the DEF-operator applied to the coefficient matrix A from [99]. The authors have provided concise analytical expressions for the eigenvalues of the standard two-level DEF-operator. We use these expressions to perform a preliminary analysis of the spectrum.

7.4.1. SPECTRAL ANALYSIS

For $j = 1, 2, \ldots, \frac{n}{2}$, the eigenvalues of the system PA are given by

$$\lambda^j(PA) = \lambda^j(A)\left(1 - \frac{\lambda^j(A)\cos(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})}\right) + \lambda^{n+1-j}(A)\left(1 - \frac{\lambda^{n+1-j}(A)\sin(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})}\right). \tag{7.6}$$

Inspection of Eq. (7.6) leads to the observation that the eigenvalues of the deflation operator P are given by

$$\lambda^j(P) = \left(1 - \frac{\lambda^j(A)\cos(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})}\right) + \left(1 - \frac{\lambda^{n+1-j}(A)\sin(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})}\right). \tag{7.7}$$

By introducing the following coefficients, we can rewrite Eq. (7.6) as

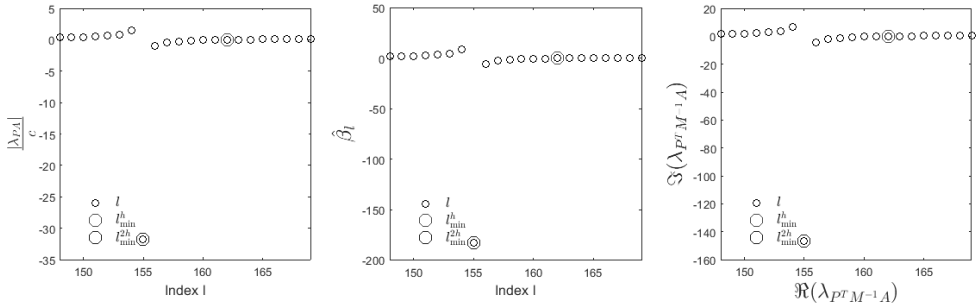
$$\alpha^j = \left(1 - \frac{\lambda^j(A)\cos(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})}\right) = \frac{\lambda^{n+1-j}(A)\sin(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})},$$
$$\beta^j = \left(1 - \frac{\lambda^{n+1-j}(A)\sin(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})}\right) = \frac{\lambda^j(A)\cos(j\pi\frac{h}{2})^4}{\lambda^j(A_{2h})},$$
$$\lambda^j(PA) = \lambda^j(A)\alpha^j + \lambda^{n+1-j}(A)\beta^j, j = 1,2,\ldots,\frac{n}{2} \tag{7.8}$$

Since the sine and cosine terms are always strictly less than 1, the eigenvalues of the system PA are essentially the product of eigenvalues of A multiplied by the scaled ratio of the eigenvalues of A and A_{2h}. In order to simplify the analysis, we therefore proceed by analyzing

$$\hat{\beta}^j = \left|\frac{\lambda^j(A)}{\lambda^j(A_{2h})}\right|, j = 1,2,\ldots,\frac{n}{2}, \tag{7.9}$$

which provides an upperbound to the previously defined coefficients. It is easy to see that the eigenvalues of PA will approach the origin if the factor $\hat{\beta}^j$ becomes small for some j. If we define the constant c to be the magnitude of the largest eigenvalue of A, then we can scale the eigenvalues of PA by c and compare them to the eigenvalues of $P^TM^{-1}A$ and $\hat{\beta}^j$.

Figure 7.1: $kh = 0.625, k = 500$. *Left: eigenvalues of PA scaled by magnitude of the largest eigenvalue* (c). *Center: Ratio between eigenvalues of the fine-grid and coarse-grid operator* ($\hat{\beta}$ *from equation Eq.* (7.9)). *Right: real part of eigenvalues* $P^TM^{-1}A$.



In Fig. 7.1 we have plotted a selected range of eigenvalues of PA scaled by c and compared these to the eigenvalues of $P^TM^{-1}A$ (right) and $\hat{\beta}^j$ (center). On the x-axis we have the relevant indices j corresponding to the respective close to zero eigenvalues. The figure provides affirmative support for our remark that the behaviour of the eigenvalues of both PA and $P^TM^{-1}A$ are, apart from a scaling factor, determined by the behaviour of $\hat{\beta}^j$ as all three figures exhibit the same shape and pattern. $\hat{\beta}^j$ approaches the origin whenever $\left|\lambda^j(A)\right|$ becomes small, which is at $j = j_{\min,h}$ (marker). If $j_{\min,h} \neq j_{\min,2h}$ and $j_{\min,2h} < j_{\min,h}$, then we are dividing a relatively small number $\left|\lambda^{j_{\min,h}}(A)\right|$ by a larger number $\left|\lambda^{j_{\min,h}}(A_{2h})\right|$, which brings the resulting fraction closer to zero. The further apart $j_{\min,h}$ and $j_{\min,2h}$ are, the closer

to zero the resulting term will be. The outlier appointed by the marker, is the result of exactly the opposite effect. At $j = j_{\min,2h}$, $\left|\lambda^j(A_{2h})\right|$ will be at its smallest, while the magnitude of $\left|\lambda^j(A)\right|$ will still be large. In similar manner, we get a large term, which explains the typical outliers we often encounter when the spectra of the operators PA and $P^T M^{-1} A$ are plotted.

7.4.2. EIGENVECTOR PERTURBATIONS

The next question which needs to be answered is what is causing the near-zero eigenvalues of the coarse grid operator to shift. It has been reported that interpolating coarse-grid functions always introduces high-frequency modes, which can be interpreted as an aliasing phenomenon [102], [35]. These high-frequency modes are the main cause for interpolation errors [102]. The effect becomes more severe as index j increases. If the high-frequency eigenmodes are activated by interpolating from a coarse to a fine grid, then the coarse-grid eigenvectors will not be approximated accurately. This affects the eigenvalues of A_{2h} as A_{2h} is obtained by first restricting the fine-grid elements onto the coarse-grid and then transferring the result back onto the fine-grid.
To measure the extent of this effect, we make use of Lemma 11.2 and Corollary 11.2.

Lemma 11.2: Intergrid Transfer - I

Let B be the $\frac{n}{2} \times \frac{n}{2}$ matrix given by $B = Z^T Z$, where $Z = I_{2h}^h$ is the prolongation matrix and let $j_{\min,h}$ be the index of smallest eigenvalue of A in terms of magnitude. Then there exist a constant C_h, depending on h such that

$$B v_{2h}^{j_{\min}} = C_h v_{2h}^{j_{\min}} \text{ and } \lim_{h\to 0} C_h = \lambda^{j_{\min}}(B) = 2,$$

where v_h^j is the $j-$th eigenvector on the fine-grid of A and $\lambda^j(B)$ is the $j-$th eigenvalue of B.

Proof. We use the method from [35]. For $i = 1, 2, \dots n$ we have

$$\begin{aligned}
[Z^T v_h^{j_{\min}}]_i &= \frac{1}{2}\left(\sin((2i-1)h\pi j_{\min,h}) + 2\sin(2ih\pi j_{\min,h}) + \sin((2i+1)h\pi j_{\min,h})\right), \\
&= \frac{1}{2}\left(2\sin(2ih\pi j_{\min,h}) + 2\cos(2ih\pi j_{\min,h})\right)\sin(2ih\pi j_{\min,h}), \\
&= \left(1+\cos(j_{\min,h}\pi h)\right)\sin(2ih\pi j_{\min,h}), \\
&= C_1(h) v_{2h}^{j_{\min}}.
\end{aligned}$$

Now taking the limit as h goes to zero of the coefficient C_h gives $\lim_{h\to 0} C_1(h) = 2$. For $i = 1, 2, \dots, n$ we distinguish two cases; i is odd and i is even. We start with the first case

$$\begin{aligned}
[Z v_{2h}^{j_{\min}}]_i &= \frac{1}{2}\left(\sin(\frac{(i-1)h\pi j_{\min,h}}{2}) + \sin(\frac{(i+1)h\pi j_{\min,h}}{2})\right), \\
&= \frac{1}{2}\left(\sin((i-1)h\pi j_{\min,h}) + \sin((i+1)h\pi j_{\min,h})\right), \\
&= \cos(j_{\min,h}\pi h)\sin(ih\pi j_{\min,h}), \\
&= C_2(h) v_h^{j_{\min}}.
\end{aligned}$$

7

Again, taking the limit as h goes to zero of the coefficient $C_2(h)$ gives $\lim_{h\to 0} C_2(h) = 1$. For i is even, we obtain $Z v_{2h}^{j_{\min}} = \sin(\frac{ih\pi j_{\min,h}}{2}) = \sin(ih\pi j_{\min,h}) = v_h^{j_{\min}}$. We can combine both results to obtain $B v_{2h}^{j_{\min}} = Z^T Z v_{2h}^{j_{\min}} = Z^T(C_2(h) v_h^{j_{\min}}) = C_1(h)C_2(h) v_{2h}^{j_{\min}} = \hat{\lambda}^{j_{\min,h}}(B) v_{2h}^{j_{\min}}$, where $\hat{\lambda}^{j_{\min,h}}(B)$ represents the perturbed eigenvalue of B at index $j_{\min,h}$ due to the approximation error.
Taking the limit as h goes to zero provides $\lim_{h\to 0} \hat{\lambda}^{j_{\min,h}}(B) = \lim_{h\to 0} C_1(h)C_2(h) = 2 = \lambda^{j_{\min,h}}(B) = \lambda^{j_{\min,2h}}(B)$. ■

Lemma 11.2 shows that in the absence of interpolation errors, the location of the smallest eigenvalue of B, which we denote by $j_{\min,2h}$, is located at exactly index $j_{\min,h}$, i.e. $j_{\min,h} = j_{\min,2h}$.

Corollary 11.2: Coarse-grid kernel

Let A_{2h} be the $\frac{n}{2} \times \frac{n}{2}$ matrix given by $A_{2h} = Z^T A Z$, where $Z = I_{2h}^h$ is the prolongation matrix and let $j_{\min,h}$ be the index of smallest eigenvalue of A in terms of magnitude. Then

$$A_{2h} v_{2h}^{j_{\min}} = C_h \lambda^{j_{\min,h}}(A) v_{2h}^{j_{\min}}, \text{ and } \lim_{h\to 0} C_h = \lambda^{j_{\min,h}}(B).$$

where v_{2h}^j is the $j-$th eigenvector on the coarse-grid of A_{2h} and $\lambda^j(A_{2h})$ is the $j-$th eigenvalue of A_{2h}.

7

Proof. Using Lemma 11.2 and its proof, we have

$$\begin{aligned}
A_{2h} v_{2h}^{j_{\min}} &= \left(Z^T A Z\right) v_{2h}^{j_{\min}}, \\
&= Z^T A \left(Z v_{2h}^{j_{\min}}\right), \\
&= Z^T A (C_2(h) v_h^{j_{\min}}), \\
&= C_1(h) Z^T A v_h^{j_{\min}}, \\
&= C_1(h) Z^T \lambda^{j_{\min,h}}(A) v_h^{j_{\min}}, \\
&= \lambda^{j_{\min,h}}(A) C_1(h) \left(Z^T v_h^{j_{\min}}\right), \\
&= \lambda^{j_{\min,h}}(A) C_1(h) C_2(h) v_{2h}^{j_{\min}}.
\end{aligned}$$

Using Lemma 11.2 it is easy to see that after taking the limit the eigenvalues of A_{2h} can be written as a product of the eigenvalues of A and the eigenvalues of B. ■

From Lemma 11.2 and Corollary 11.2 it is clear that for $j_{\min,h}$, which is within the smooth-frequency range, the near-kernel coarse-grid eigenvalues $\lambda^{j_{\min,2h}}(A_{2h})$ are equal to the product of $\lambda^{j_{\min,h}}(A)$ and $\lambda^{j_{\min,2h}}(B) = \lambda^{j_{\min,h}}(B)$ when h goes to zero. Consequently, in the limiting case the coarse-grid kernel and the fine-grid kernel will be aligned proportionally and both A and A_{2h} will reach its smallest absolute eigenvalues at the same index $j_{\min}$.

Recall that the behavior of the eigenvalues of PA can be represented by

$$\hat{\beta}^j = \left| \frac{\lambda^j(A)}{\lambda^j(A_{2h})} \right| \text{ for } j = 1,2,\ldots,\frac{n}{2},$$

where we found that this ratio becomes very small by a mismatch of the smallest absolute eigenvalue of A and A_{2h} respectively. As in the limit, $\lambda^{j_{\min,h}}(A_{2h}) = \lambda^{j_{\min,h}}(B)\lambda^{j_{\min,h}}(A_h)$, perturbations up to $\lambda^{j_{\min,h}}(B)$ will propagate throughout the low-frequency part of the spectrum for $j \in \{1,2,\ldots,j_{\min,h}\}$, eventually resulting in the errors related to $\lambda^j(A_{2h})$ for $j = j_{\min,h}$.

7.4.3. PROJECTION ERROR

To measure to what extent these perturbations to $\lambda(B)$ lead to errors, we examine the projection error to quantify the error we make when projecting the eigenvector onto the subspace spanned by the column of Z.

Theorem 12: Projection Error - I

Let X be the deflation space spanned by column vectors of Z and let the eigenvector corresponding to the smallest eigenvalue of A be denoted by $v_h^{j_{\min}} \notin X$. Let $P = ZB^{-1}Z^T$ with $B = Z^TZ$ be the orthogonal projector onto X. Then the projection error E is given by

$$E = \|(I-P)v_h^{j_{\min}}\|^2 = {v_h^{j_{\min}}}^T v_h^{j_{\min}} - {v_h^{j_{\min}}}^T ZB^{-1}Z^T v_h^{j_{\min}}.$$

Proof. By idempotency of the orthogonal projector, we have

$$\begin{aligned}
\|(I-P)v_h^{j_{\min}}\|^2 &= {v_h^{j_{\min}}}^T (I-P)(I-P)v_h^{j_{\min}}, \\
&= {v_h^{j_{\min}}}^T (I-P)v_h^{j_{\min}}, \\
&= {v_h^{j_{\min}}}^T v_h^{j_{\min}} - {v_h^{j_{\min}}}^T ZB^{-1}Z^T v_h^{j_{\min}}.
\end{aligned}$$

■

We proceed by rewriting the projection error in terms of a perturbation to the eigenvalues of the operator B.

Corollary 12.1: Projection Error - II

Let X be the deflation space spanned by the column vectors of Z and let the eigenvector corresponding to the smallest eigenvalue of A be denoted by $v_h^{j_{\min}} \notin X$. Let $P = ZB^{-1}Z^T$ with $B = Z^TZ$ be the orthogonal projector onto X. Then the projection error E is given by

$$E = \|(I-P)v_h^{j_{\min}}\|^2 = \left(1 - \frac{\lambda^{j_{\min},h}(B) - \delta_1}{\lambda^{j_{\min},h}(B) - \delta_2}\right) {v_h^{j_{\min}}}^T v_h^{j_{\min}},$$

where $\delta_1 = \lambda^{j_{\min},h}(B) - \frac{{v_h^{j_{\min}}}^T \hat{B} v_h^{j_{\min}}}{{v_h^{j_{\min}}}^T v_h^{j_{\min}}}$ and $\delta_2 = \lambda^{j_{\min},h}(B) - \frac{{v_h^{j_{\min}}}^T \hat{B} v_h^{j_{\min}}}{{v_h^{j_{\min}}}^T Z\left(B^{-1}Z^T v_h^{j_{\min}}\right)}$.

Proof. Using Lemma 11.2 and its proof we know that in the limit $Z^T v_h^{j_{\min}}$ is an eigenvector of B. We would thus have

$$\begin{aligned}
\|(I-P)v_h^{j_{\min}}\|^2 &= {v_h^{j_{\min}}}^T v_h^{j_{\min}} - {v_h^{j_{\min}}}^T Z\left(B^{-1}Z^T v_h^{j_{\min}}\right), \\
&= {v_h^{j_{\min}}}^T v_h^{j_{\min}} - \frac{{v_h^{j_{\min}}}^T ZZ^T v_h^{j_{\min}}}{\lambda^{j_{\min},h}(B)}, \\
&= {v_h^{j_{\min}}}^T v_h^{j_{\min}} - \frac{{v_h^{j_{\min}}}^T \left(\hat{B} v_h^{j_{\min}}\right)}{\lambda^{j_{\min},h}(B)}.
\end{aligned}$$

Note that $\hat{B}$ has dimension $n \times n$ and has $\frac{n}{2}$ eigenvalues equal to the eigenvalues of B and $\frac{n}{2}$ zero eigenvalues. By Lemma 11.2 and its proof, we also have that $v_h^{j_{\min}}$ is an eigenvector of $\hat{B}$, which leads to

$$\|(I-P)v_h^{j_{\min}}\|^2 = \lim_{h\to 0}\left({v_h^{j_{\min}}}^T v_h^{j_{\min}} - \frac{{v_h^{j_{\min}}}^T \left(\lambda^{j_{\min},h}(\hat{B}) v_h^{j_{\min}}\right)}{\lambda^{j_{\min},h}(B)}\right) = 0. \tag{7.10}$$

Now, in the non-limiting case, we have two sources of errors; the factor containing $\lambda^{j_{\min},h}(B)$ both in the numerator and denominator will be subjected to perturbations. Starting with the denominator, if we let $\tilde{\lambda}_{j_{\min},h}(B)$ denote the perturbed eigenvalue of B, we have

$${v_h^{j_{\min}}}^T Z\left(B^{-1}Z^T v_h^{j_{\min}}\right) = {v_h^{j_{\min}}}^T Z\left(\frac{Z^T v_h^{j_{\min}}}{\tilde{\lambda}_{j_{\min},h}(B)}\right) \neq {v_h^{j_{\min}}}^T Z\left(\frac{Z^T v_h^{j_{\min}}}{\lambda^{j_{\min},h}(B)}\right).$$

Reordering leads to

$$\tilde{\lambda}_{j_{\min},h}(B) = \frac{{v_h^{j_{\min}}}^T ZZ^T v_h^{j_{\min}}}{{v_h^{j_{\min}}}^T Z\left(B^{-1}Z^T v_h^{j_{\min}}\right)} = \frac{{v_h^{j_{\min}}}^T \hat{B} v_h^{j_{\min}}}{{v_h^{j_{\min}}}^T Z\left(B^{-1}Z^T v_h^{j_{\min}}\right)}.$$

The perturbation to $\lambda^{j_{\min,h}}(B)$ can now be written as

$$\delta_2 = \lambda^{j_{\min,h}}(B) - \tilde{\lambda}_{j_{\min},h}(B) = \lambda^{j_{\min,h}}(B) - \frac{{v_h^{j_{\min}}}^T \hat{B} v_h^{j_{\min}}}{{v_h^{j_{\min}}}^T Z \left(B^{-1} Z^T v_h^{j_{\min}}\right)}.$$

For the numerator, if we let η denote the error, i.e. $\eta = \hat{B} v_h^{j_{\min}} - \lambda^{j_{\min,h}}(B) v_h^{j_{\min}}$, then $\hat{B} v_h^{j_{\min}} = \lambda^{j_{\min,h}}(B) v_h^{j_{\min}} + \eta$ and substitution gives

$$\begin{aligned}\tilde{\lambda}_{j_{\min},h}(B) {v_h^{j_{\min}}}^T Z \left(B^{-1} Z^T v_h^{j_{\min}}\right) &= {v_h^{j_{\min}}}^T \hat{B} v_h^{j_{\min}}, \\ &= {v_h^{j_{\min}}}^T \left(\lambda^{j_{\min,h}}(B) v_h^{j_{\min}} + \eta\right).\end{aligned}$$

Letting $\delta_1 = -\frac{{v_h^{j_{\min}}}^T \eta}{{v_h^{j_{\min}}}^T v_h^{j_{\min}}}$, we obtain

$$\tilde{\lambda}_{j_{\min},h}(B) {v_h^{j_{\min}}}^T Z \left(B^{-1} Z^T v_h^{j_{\min}}\right) = \left(\lambda^{j_{\min,h}}(B) - \delta_1\right) {v_h^{j_{\min}}}^T v_h^{j_{\min}}.$$

Finally, we now rewrite the projection error E in terms of perturbations to the eigenvalues of B;

$$\begin{aligned}\left\|(I-P) v_h^{j_{\min}}\right\|^2 &= {v_h^{j_{\min}}}^T v_h^{j_{\min}} - {v_h^{j_{\min}}}^T Z \left(B^{-1} Z^T v_h^{j_{\min}}\right), \\ &= \left(1 - \frac{\lambda^{j_{\min,h}}(B) - \delta_1}{\lambda^{j_{\min,h}}(B) - \delta_2}\right) {v_h^{j_{\min}}}^T v_h^{j_{\min}},\end{aligned}$$

which gives the statement. ∎

7

POLLUTION ERROR

We can proof an additional statement with respect to the pollution error. We know that the pollution error is minimized when we keep the step size $h = k^{-\frac{3}{2}}$, see Chapter 5 Section 5.3.1. We can study the behavior of the projection error by letting k go to infinity.

Corollary 12.2: Pollution error

Let $h = k^{-\frac{3}{2}}$. Let X be the deflation space spanned by column vectors of Z and let the eigenvector corresponding to the smallest eigenvalue of A be denoted by $v_h^{j_{\min}} \notin X$. Let $P = ZB^{-1}Z^T$ with $B = Z^T Z$ be the orthogonal projector onto X. Then the projection error E goes to zero

$$E = \lim_{k \to \infty} \left\|(I-P) v_h^{j_{\min}}\right\|^2 = 0.$$

Proof. Using Lemma 11.2 and Corollary 11.2 we have

$$\begin{aligned}\left[Z^T v_h^{j_{\min}}\right]_i &= (1 + \cos j_{min,h} \pi h) v_{2h}^{j_{\min}}, \\ &= (1 + \cos j_{min,h} \frac{\pi}{k^{\frac{3}{2}}}) v_{2h}^{j_{\min}}.\end{aligned}$$

Now taking $k \to \infty$ gives $\lim_{k\to\infty}[Z^T \phi_{j_{min},h,i}] = 2v_{2h}^{j_{\min}}$. Similarly,

$$\begin{aligned}[Z\phi j_{min,2h}]_i &= \cos(j_{min,2h}\pi h)\, v_h^{j_{\min}}, \\ &= \cos(j_{min,h}\frac{\pi}{k^{\frac{3}{2}}})\, v_h^{j_{\min}}.\end{aligned}$$

Again, taking $k \to \infty$ gives $\lim_{k\to\infty}[Z\phi_{j_{min},2h,i}] = v_h^{j_{\min}}$. Now, substituting these expressions into the projection error E gives

$$\begin{aligned}E = \lim_{k\to\infty} ||(I-P)v_h^{j_{\min}}||_2^2 &= \lim_{k\to\infty} v_h^{j_{\min}T} v_h^{j_{\min}} - v_h^{j_{\min}T} Z(B^{-1}Z^T v_h^{j_{\min}}) \\ &= \lim_{k\to\infty} \left(v_h^{j_{\min}T} v_h^{j_{\min}} - v_h^{j_{\min}T} ZB^{-1}(2v_{2h}^{j_{\min}}) \right), \\ &= \lim_{k\to\infty} \left(v_h^{j_{\min}T} v_h^{j_{\min}} - 2v_h^{j_{\min}T} Z(B^{-1}v_{2h}^{j_{\min}}) \right), \\ &= \lim_{k\to\infty} \left(v_h^{j_{\min}T} v_h^{j_{\min}} - \frac{2}{\lambda^{j_{\min,h}}(B)} v_h^{j_{\min}T}(Zv_{2h}^{j_{\min}}) \right), \\ &= \lim_{k\to\infty} \left(v_h^{j_{\min}T} v_h^{j_{\min}} - \frac{2}{\lambda^{j_{\min,h}}(B)} v_h^{j_{\min}T} v_h^{j_{\min}} \right), \\ &= \lim_{k\to\infty} \left(1 - \frac{2}{\lambda^{j_{\min,h}}(B)} \right).\end{aligned}$$

We know from Corollary 11.2 that $\lambda^{j_{\min}}(B) \to 2$ when h goes to zero. And thus we obtain the statement. ■

7

Corollary 12.1 reveals that the projection error due to the inaccurate approximations of the eigenvectors can be represented by deviations from $\lambda^{j_{\min,h}}(B)$. In Table 7.1 we present the projection error for various k. The results illustrate that the projection error increases linearly with k. Along with the projection error, the misalignment between $j_{\min,h}$ and $j_{\min,2h}$ increases, shifting the near-zero eigenvalue of A and A_{2h}. If we let $kh = 0.3125$, the projection error is reduced. However, already for $k = 1000$, the error regains magnitude, which explains why, despite resorting to a finer grid, the near-zero eigenvalues reappear when k increases. The results for $k^3h^2 = 1$ are in line with Corollary 12.2. As the step-size h gets smaller, the error of the interpolation and restriction operations from the fine to the coarse grid and vice versa reduces. This explains why the projection error decreases as the wavenumber k increases. This can also be noticed from the last two columns of Table 7.1. Note that the location of the smallest eigenvalue in terms of magnitude of A and A_{2h} are always located at the same index.

Table 7.1: Projection Error for $v_h^{j_{\min}}$ for various values of k. $j_{\min,h}$ and $j_{\min,2h}$ denote the index for the smallest absolute eigenvalue of A and A_{2h} respectively.

k	E	$j_{\min,h}$	$j_{\min,2h}$	E	$j_{\min,h}$	$j_{\min,2h}$	E	$j_{\min,h}$	$j_{\min,2h}$
	$kh = 0.625$			$kh = 0.3125$			$k^3h^2 = 1$		
10	0.0672	3	3	0.0077	3	3	0.0077	3	3
50	0.4409	16	15	0.0503	16	16	0.0045	16	16
100	0.8818	32	31	0.0503	32	32	0.0032	32	32
500	4.670	162	155	0.5031	162	158	0.0013	162	162
1000	9.2941	324	310	1.0062	324	316	0.0009	324	324

INSCALABILITY

In Section Section 7.4 we have shown that the spectrum of PA and $PM^{-1}A$ is (apart from a scaling factor) equivalent to

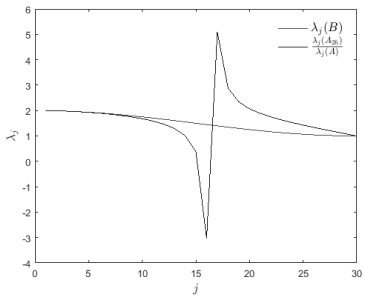
$$\hat{\beta}^j = \left| \frac{\lambda^j(A)}{\lambda^j(A_{2h})} \right|, j = 1,2,\ldots,\frac{n}{2}.$$

From Lemma 11.2 and Corollary 11.2 we additionally found that in the limit near $j = j_{\min,h}$ we can express the eigenvalues of the coarse-grid operator A_{2h} in terms of $\lambda^{j_{\min,h}}(B)$ by $\lambda^{j_{\min,h}}(A_{2h}) = \lambda^{j_{\min,h}}(A)\lambda^{j_{\min,h}}(B)$. Thus in the vicinity of the smallest eigenvalue, we can write

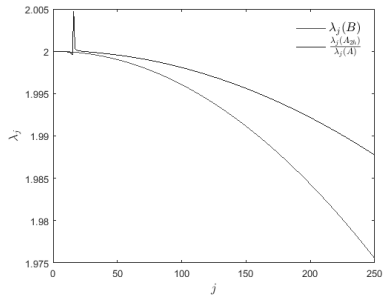
$$\hat{\beta}^j = \left| \frac{\lambda^j(A)}{\lambda^j(A_{2h})} \right| = \frac{1}{\lambda^j(B)}. \tag{7.11}$$

Corollary 12.1 reflects that errors in projecting the eigenvectors onto the coarse-grid lead to errors in the eigenvalues of the operator B. These errors accumulate and increase as index j increases, due to the eigenvectors becoming more oscillatory. If we account for these errors, then Eq. (7.11) becomes $\hat{\beta}^j = \left| \frac{\lambda^j(A)}{\lambda^j(A_{2h})} \right| = \frac{1}{\hat{\lambda}^j(B)}$, for some perturbed $\hat{\lambda}^j(B)$. These perturbations to the eigenvalues of B cause inaccurate scaling of the eigenvalues of A, eventually leading to the smallest eigenvalue of A_{2h} being located at a different index $j_{\min,2h} \neq j_{\min,h}$. In Fig. 7.3(a) and Fig. 7.3(b) we have plotted the eigenvalues of B and the ratio between the eigenvalues of A_{2h} and A according to equation Eq. (7.11). Note that the latter essentially represents the perturbed $\lambda^j(B)$ due to errors accumulated during prolongating and restricting the eigenvectors of A. It can be noted that as h becomes smaller, the ratio slowly converges to $\lambda^j(B)$. This observation is also in line with the projection error decreasing.

Figure 7.2: $k = 50$. *Plot of the ratio between the fine-grid and coarse-grid eigenvalues (equation (Eq. (7.11))) and the eigenvalues of* B. $j_{\min,h} = 16$ *and* $j_{\min,2h} = 15$ *for* $kh = 0.825$. *For* $kh = 0.01$, $j_{\min,h} = j_{\min,2h} = 16$.



(a) $kh = 0.825$,



(b) $kh = 0.01$

7.5. HIGHER-ORDER DEFLATION

In this section we start defining the higher order approximation techniques to construct the deflation space. We start by with a general representation of the linear interpolation scheme and work our way up towards higher-order schemes.

7.5.1. QUADRATIC APPROXIMATION

Recall that the grid transfer functions $u_{2h} = [u_{2h_1}, \ldots, u_{2h_n}]$ from Ω_{2h} to the fine grid Ω_h using standard linear interpolation are given by

$$I_{2h}^{h} : \Omega_{2h} \to \Omega_h, \quad u_{2h} \to I_{2h}^{h}\, u_{2h} \tag{7.12}$$

such that

$$\begin{cases} [u_{2h}]_{i/2} & \text{if } i \text{ is even,} \\ \frac{1}{2}\left([u_{2h}]_{(i-1)/2} + [u_{2h}]_{(i+1)/2}\right) & \text{if } i \text{ is odd,} \end{cases} \quad i = 1, \ldots, n-1 \tag{7.13}$$

A closer look reveals that the current transfer functions are only reinforced at the odd components, leaving the even components unchanged. In fact, these components are mapped to a linear combination of their fine-grid counterparts v_h^j and a complimentary mode v_h^{n+1-j} with first order accuracy [102].

Bézier CURVES

A more general representation of the linear interpolation operator for the even components can be given by using rational *Bézier* curves, which are defined in Definition 3, Definition 4 and Definition 5. The use of these curves within the context of multigrid methods has been studied in [103] and [104]. Using these vectors as vectors for the input of the prolongation and restriction matrices in a multigrid setting is referred to as a *monotone multigrid method*. The monotonicity comes from the construction of the coarse-grid approximations, which ensures that the coarse-grid functions approximate the fine-grid functions monotonically [104], [105]. The higher order approximation schemes are defined in Definition 6.

Definition 3 (*Bézier* curve). *A Bézier curve of degree n is a parametric curve defined by*

$$B(t) = \sum_{j=0}^{n} b_{j,n}(t) P_j, \quad 0 \leqslant t \leqslant 1, \quad \text{where the polynomials}$$

$$b_{j,n}(t) = \binom{n}{j} t^j (1-t)^{n-j}, \quad j = 0, 1, \ldots, n,$$

are known as the Bernstein basis polynomials of order n. The points P_j are called control points for the Bézier curve.

Definition 4 (Rational *Bézier* curve). *A rational Bézier curve of degree n with control points $P_0, P_1, \ldots, P_n$ and scalar weights $w_0, w_1, \ldots, w_n$ is defined as*

$$C(t) = \frac{\sum_{j=0}^{n} w_j b_{j,n}(t) P_j}{\sum_{j=0}^{n} w_j b_{j,n}(t)}.$$

Definition 5 (Linear Interpolation). *Let $[u_{2h}]_{(j-1)/2}$ and $[u_{2h}]_{(j+1)/2}$, be the end points within a component span defined on the coarse grid. Then the prolongation scheme for the even nodes can be characterized by a Rational Bézier curve of degree 1 with polynomials*

$$b_{0,1}(t) = 1 - t,$$
$$b_{1,1}(t) = t,$$

whenever j is odd by taking the weights $w_0 = w_1 = 1$ and $t = \frac{1}{2}$. Note that in case $w_0 = w_1$ and non-rational we obtain the original Bézier curve.

$$C(\frac{1}{2}) = \frac{\frac{1}{2}[u_{2h}]_{(j-1)/2} + (1-\frac{1}{2})[u_{2h}]_{(j+1)/2}}{\frac{1}{2} + (1-\frac{1}{2})}, \tag{7.14}$$

$$= \frac{1}{2}\left([u_{2h}]_{(j-1)/2} + [u_{2h}]_{(j+1)/2}\right). \tag{7.15}$$

When j is even, we take the middle component $[u_{2h}]_{j/2}$, which itself gets mapped onto the fine grid.

For large k, the prolongation operator working on the even components is not sufficiently accurate to map the near kernels to adjacent modes on Ω_{2h} and Ω_h. Consequently, we wish to find a higher order approximation scheme, which takes the even components into account. We thus consider a quadratic rational *Bézier* curve in order to find appropriate coefficients to yield a higher order approximation of the fine-grid functions by the coarse grid functions.

Definition 6 (Quadratic Approximation). *Let $[u_{2h}]_{(j-2)/2}$ and $[u_{2h}]_{(j+2)/2}$, be the end points within a component span defined on the coarse grid. Then the prolongation operator can be characterized by a Rational Bézier curve of degree 2 with polynomials*

$$b_{0,2}(t) = (1-t)^2,$$
$$b_{1,2}(t) = 2t(1-t),$$
$$b_{2,2}(t) = t^2,$$

and control point $[u_{2h}]_{j/2}$*, whenever* j *is even. Because we wish to add more weight to the center value, we take weights* $w_0 = w_2 = \frac{1}{2}$*,* $w_1 = \frac{3}{2}$ *and* $t = \frac{1}{2}$ *to obtain*

$$\begin{aligned} C(t) &= \frac{\frac{1}{2}(1-t)^2[u_{2h}]_{j-1} + \frac{3}{2}2t(1-t)[u_{2h}]_j + \frac{1}{2}(t)^2[u_{2h}]_{j+1}}{\frac{1}{2}(1-t)^2 + \frac{3}{2}2t(1-t) + \frac{1}{2}(t)^2} \\ &= \frac{\frac{1}{2}(1-\frac{1}{2})^2[u_{2h}]_{j-1} + \frac{3}{2}(2)(\frac{1}{2})(1-\frac{1}{2})[u_{2h}]_j + \frac{1}{2}(\frac{1}{2})^2[u_{2h}]_{j+1}}{\frac{1}{2}(1-\frac{1}{2})^2 + \frac{1}{2}(2)(\frac{1}{2})(1-\frac{1}{2}) + \frac{1}{2}(\frac{1}{2})^2} \\ &= \frac{\frac{1}{8}[u_{2h}]_{j-1} + \frac{3}{4}[u_{2h}]_j + \frac{1}{8}[u_{2h}]_{j+1}}{1} \\ &= \frac{1}{8}\left([u_{2h}]_{j-1} + 6[u_{2h}]_j + [u_{2h}]_{j+1}\right). \end{aligned} \tag{7.16}$$

When j *is odd,* $[u_{2h}]_{(j-1)/2}$ *and* $[u_{2h}]_{(j+1)/2}$ *have an even component and we are in the same scenario as is the case with linear interpolation.*

7.5.2. ADAPTED DEFLATION PRECONDITIONER

Using these higher-order approximations, we redefine the prolongation and restriction operator which are used to construct the deflation preconditioner P. We start by noting that the new restriction and prolongation operators become

$$I_{2h}^{h}[u_{2h}]_i = \begin{Bmatrix} \frac{1}{8}\left([u_{2h}]_{(i-2)/2} + 6[u_{2h}]_{(i)/2} + [u_{2h}]_{(i+2)/2}\right) & \text{if } i \text{ is even,} \\ \frac{1}{2}\left([u_{2h}]_{(i-1)/2} + [u_{2h}]_{(i+1)/2}\right) & \text{if } i \text{ is odd,} \end{Bmatrix}, \tag{7.17}$$

for $i = 1, \ldots, n-1$ and

$$I_h^{2h}[u_h]_i = \frac{1}{8}\left([u_h]_{(2i-2)} + 4[u_h]_{(2i+1)} + 6[u_h]_{(2i)} + 4[u_h]_{(2i+1)} + [u_h]_{(2i+2)}\right),$$

for $i = 1, \ldots, \frac{n}{2}$.

BLOCK-DIAGONALIZATION

Using the new matrices I_{2h}^h and I_h^{2h}, we now construct similar analytical expressions for the eigenvalues of A_{2h}, PA and $P^T M^{-1} A$, where we follow the same approach as [102], [35] and [99]. Here, the basis consisting of eigenvectors is re-ordered and the projection operator P is block-diagonalized. This allows thorough spectral analysis of each eigenvalue of PA for MP 1-A as each block now contains the non-zero analytical eigenvalues. We therefore start by following a similar approach with respect to the block-diagonalization by reordering the basis consisting of the eigenvectors as follows

$$V = \left\{ v_h^1, v_h^{(n+1)-1}, v_h^2, v_h^{(n+1)-2}, \ldots, v_h^{\frac{n}{2}}, v_h^{(n+1)-\frac{n}{2}} \right\}.$$

Here the fine-grid eigenvector are given by $v_h^j = \sin(j\pi h)$ and the coarse-grid eigenvectors are obtained by substituting $2h$ for h. The prolongation operator maps the coarse-grid eigenvectors for indices $i, j = 1, 2, \ldots \frac{n}{2}$ to

$$\begin{aligned} [I_h^{2h} v_h]_i^j &= \frac{1}{8}\left[\sin((i-2)/2)j\pi 2h) + 6\sin((i)/2)j\pi 2h) + \sin((i+2)/2)j\pi 2h)\right], \\ &= \left[\frac{1}{4}\cos(2j\pi h) + \frac{3}{4}\right]\sin(ij\pi h), \end{aligned}$$

for i is even and

$$\begin{aligned}[I_h^{2h} v_h]_i^j &= \frac{1}{8}\left[4\sin((i-1)/2)\,j\pi 2h) + 4\sin((i+1)/2)\,j\pi 2h)\right],\\ &= \left[\cos(j\pi h)\right]\sin(i\,j\pi h),\end{aligned}$$

for i is odd. With respect to the remaining part of the index set containing j, we use that

$$\begin{aligned}\left[v_h^{n+1-j}\right]_i &= -(-1)^i \sin(i\,j\pi h),\\ i = 1,2,\ldots n-1, &\text{and } j = 1,2,\ldots \frac{n}{2}.\end{aligned} \tag{7.18}$$

Note that Eq. (7.18) is only positive when i is odd. Consequently for even i such that $i \in \left\{\frac{n}{2},\ldots,n-1\right\}$ is even, we obtain

$$\begin{aligned}[I_h^{2h} v_h]_i^j &= \frac{1}{8}\left[-\sin((i-2)/2)\,j\pi 2h) - 6\sin((i)/2)\,j\pi 2h) - \sin((i+2)/2)\,j\pi 2h)\right],\\ &= \left[-\frac{1}{4}\cos(2\,j\pi h) - \frac{3}{4}\right]\sin(i\,j\pi h),\end{aligned}$$

whereas for i is odd, we now have

$$\begin{aligned}[I_h^{2h} v_h]_i^j &= \frac{1}{8}\left[4\sin((i-1)/2)\,j\pi 2h) + 4\sin((i+1)/2)\,j\pi 2h)\right],\\ &= \left[\cos(j\pi h)\right]\sin(i\,j\pi h).\end{aligned}$$

With respect to our basis, we therefore obtain the following 2×1 block for the prolongation operator

$$[I_{2h}^h]^j = \begin{bmatrix}\cos(j\pi h) + \frac{1}{4}\cos(2\,j\pi h) + \frac{3}{4}\\ \cos(j\pi h) - \frac{1}{4}\cos(2\,j\pi h) - \frac{3}{4}\end{bmatrix}.$$

Similarly, the restriction operator is defined by taking $[I_{2h}^h]^{j\,T}$ and thus we obtain a 1×2 block. For ease of notation, we now define

$$\begin{aligned}v^j &= \cos(j\pi h) + \frac{1}{4}\cos(2\,j\pi h) + \frac{3}{4},\\ v^{n+1-j} &= \cos(j\pi h) - \frac{1}{4}\cos(2\,j\pi h) - \frac{3}{4}.\end{aligned}$$

Using these expressions, we now compute the eigenvalue of the Galerkin coarse grid operator, which is given by the 1×1 diagonal block

$$\lambda^j(A_{2h}) = [I_{2h}^h]^j A^j [I_h^{2h}]^j = \left(v^j\right)^2 \lambda^j(A) + \left(v^{n+1-j}\right)^2 \lambda^{n+1-j}(A). \tag{7.19}$$

In order to obtain the eigenvalues of PA, we have to compute the 2×2 diagonal blocks of the projection operator P first. Recall that P is defined by

$$P^j = I - (I_{2h}^h)^j (A_{2h}^j)^{-1} (I_h^{2h})^j A^j.$$

We thus obtain the following block system

$$P^j = \begin{bmatrix} 1-\frac{(v^j)^2}{\lambda^j(A_{2h})} & \frac{v^j v^{n+1-j}}{\lambda^j(A_{2h})} \\ \frac{v^{n+1-j}v^j}{\lambda^j(A_{2h})} & 1-\frac{(v^{n+1-j})^2}{\lambda^j(A_{2h})} \end{bmatrix} \begin{bmatrix} \lambda^j(A) & 0 \\ 0 & \lambda^{n+1-j}(A) \end{bmatrix},$$
$$= \begin{bmatrix} \lambda^j(A)\left(1-\frac{(v^j)^2}{\lambda^j(A_{2h})}\right) & \lambda^{n+1-j}(A)\left(\frac{v^j v^{n+1-j}}{\lambda^j(A_{2h})}\right) \\ \lambda^j(A)\left(\frac{v^{n+1-j}v^j}{\lambda^j(A_{2h})}\right) & \lambda^{n+1-j}(A)\left(1-\frac{(v^{n+1-j})^2}{\lambda^j(A_{2h})}\right) \end{bmatrix}. \tag{7.20}$$

From here, we retrieve the eigenvalues of PA by multiplying Eq. (7.20) again with the 2×2 diagonal block containing the eigenvalues of A with respect to index j on our defined basis.

$$[PA]^j = \begin{bmatrix} (\lambda^j(A))^2\left(1-\frac{(v^j)^2}{\lambda^j(A_{2h})}\right) & (\lambda^{n+1-j}(A))^2\left(\frac{v^j v^{n+1-j}}{\lambda^j(A_{2h})}\right) \\ (\lambda^j(A))^2\left(\frac{v^{n+1-j}v^j}{\lambda^j(A_{2h})}\right) & (\lambda^{n+1-j}(A))^2\left(1-\frac{(v^{n+1-j})^2}{\lambda^j(A_{2h})}\right) \end{bmatrix}. \tag{7.21}$$

Similarly, the eigenvalues of $P^T M^{-1}A$ are obtained by simply multiplying Eq. (7.20) with the 2×2 block containing the eigenvalues of $M^{-1}A$ instead of A and computing the trace. This operation leads to the following analytical expressions for the eigenvalues of $P^T M^{-1}A$ for $j = 1,2,\ldots,\frac{n}{2}$

$$\lambda^j(P^T M^{-1}A) = \frac{(\lambda^j(A))^2}{\lambda^j(M)}\left(1-\frac{(v^j)^2}{\lambda^j(A_{2h})}\right) + \frac{(\lambda^{n+1-j}(A))^2}{\lambda^j(M)}\left(1-\frac{(v^{n+1-j})^2}{\lambda^j(A_{2h})}\right). \tag{7.22}$$

Using Eq. (8.26), we proceed with the spectral analysis of the DEF-preconditioner for MP 1-A.

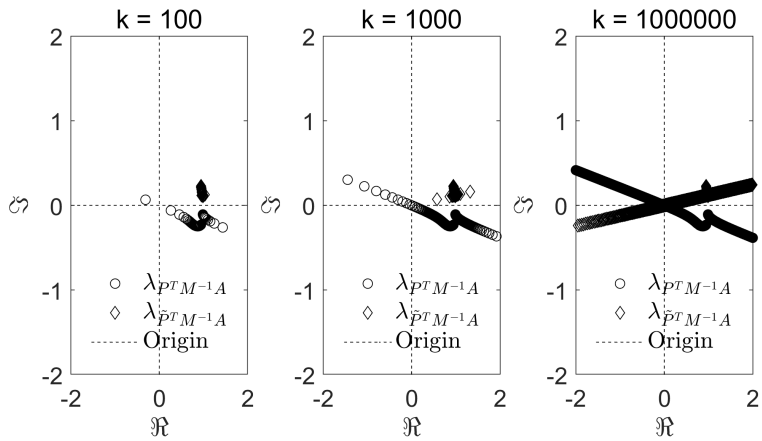
7.5.3. SPECTRAL ANALYSIS

In order to keep track of both (original and adapted) deflation based preconditioned systems, we use the $\sim$-notation to denote the adapted system. We now compare the spectrum of the DEF + CSL preconditioned matrix ($P^T M^{-1}A$), with the adapted Deflation + CSL precondtioned matrix ($\tilde{P}^T M^{-1}A$) for MP 1-A. In Fig. 7.3 we have plotted the spectrum of both $P^T M^{-1}A$ (dot marker) and $\tilde{P}^T M^{-1}A$ (diamond marker) for very large wavenumbers. Starting with the results for $kh = 0.625$, we note that incorporating the new deflation scheme leads to a remarkable reduction in the near-zero eigenvalues. Compared to the original DEF-scheme, the spectrum of the adapted scheme is more densely located near the point $(1,0)$. As a result, the spectrum of the adapted scheme has shorter tails. For example, for $k = 10^3$, there are almost no near-zero eigenvalues. However, as k increases to 10^6, we see the near-zero eigenvalues reappearing. If we switch to a finer grid using $kh = 0.3125$ in Fig. 7.3 (b), we observe an even greater improvement. For $k = 10^6$ a few eigenvalues are slightly moving towards the origin, however these results are negligible compared to the magnitude of the wavenumber. Table 7.2 contains the projection error according to Corollary 12.1 for both schemes. The projection error for the new scheme is reduced significantly. However, as k increases we observe that the projection error increases accordingly, which is in line with the spectral analysis.

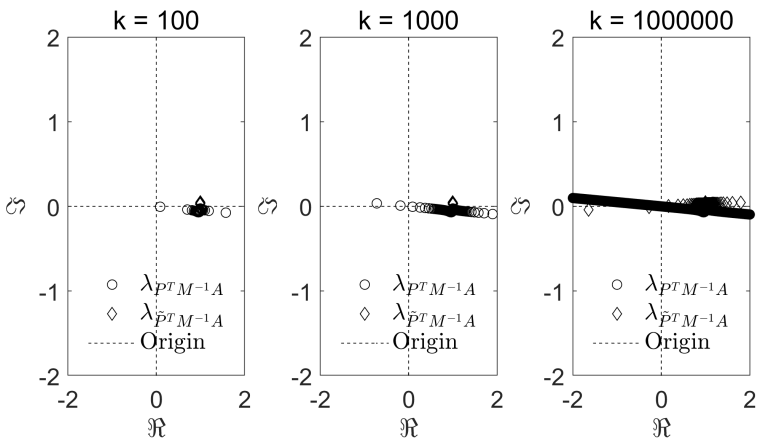
Table 7.2: Projection error for the old scheme E and the adapted scheme $\tilde{E}$.

k	E	$\tilde{E}$	E	$\tilde{E}$
	$kh = 0.625$		$kh = 0.3125$	
10^1	0.0672	0.0049	0.0077	0.0006
10^2	0.8818	0.0154	0.1006	0.0008
10^3	9.2941	0.1163	1.0062	0.0014
10^4	92.5772	1.1021	10.0113	0.007
10^5	926.135	10.9784	100.1382	0.0635
10^6	9261.7129	109.7413	1001.3818	0.6282

Figure 7.3: Eigenvalues of $P^T M^{-1} A$ and $\tilde{P}^T M^{-1} A$. The top row contains the spectrum of $P^T M^{-1} A$ and $\tilde{P}^T M^{-1} A$ for $kh = 0.625$. The bottom row contains the eigenvalues for $kh = 0.3125$.



(a) $kh = 0.625$



(b) $kh = 0.3125$

7.5.4. PARAMETER SENSITIVITY

We have seen that for very large k such as $k = 10^6$, the adapted scheme using $\tilde{P}$ still has a small number of near-zero eigenvalues. This result is supported by the increasing projection error for $kh = 0.625$ (see Table 7.2), One explanation is that for these large wavenumbers, the low-frequency eigenmode corresponding to $j_{\min,h}$ for A and $j_{\min,2h}$ for $\tilde{A}_{2h}$ are still very oscillatory vectors. Furthermore, apart from these eigenmodes themselves being relatively oscillatory, the high-frequency modes which get activated are again a source for approximation errors when prolonging the coarse-grid eigenvectors. Necessarily, at some point, the scheme based on the adapted deflation vectors will again suffer from accumulation errors as their approximation power reduces when k increases.

One of the characteristics of Bézier curves implies that at systematic intervals some discontinuities appear as sharp corners at certain points [106]. If the eigenvectors become oscillatory due to the wavenumber being very large, then keeping the grid resolution constant, these discontinuities become a source of approximation error. Instead of diverting to higher-order approximation schemes, the use of rational Bézier curves allow simple modifications which can alter the shape and movement of the utilized curve segments. In fact, the weights of the rational Bézier curve are shape parameters, which allow control over the curve segments. For example, increasing the weight corresponding to a control point forces the curvature to move more closely and sharply to that control point. Decreasing the weight of a control point, on the other hand, results in the curve flattening and expanding more towards its endpoints. In our case, the quadratic approximation using the rational Bézier curve has one control point per segment. This would lead to the following redefinition

$$I_{2h}^{h}[u_{2h}]_i = \begin{cases} \left(\frac{1}{8}[u_{2h}]_{(i-2)/2} + (\frac{3}{4}-\varepsilon)[u_{2h}]_{(i)/2} + \frac{1}{8}[u_{2h}]_{(i+2)/2}\right) & \text{if } i \text{ is even,} \\ \frac{1}{2}\left([u_{2h}]_{(i-1)/2} + [u_{2h}]_{(i+1)/2}\right) & \text{if } i \text{ is odd,} \end{cases},$$

for $i = 1,\ldots,n-1$, and $\varepsilon > 0$ The new scheme alters the expressions for the eigenvalues of $\tilde{P}^T M^{-1} A$ according to

$$\begin{aligned} \tilde{v}^j &= \cos(j\pi h) + \frac{1}{4}\cos(2j\pi h) + (\frac{3}{4}-\varepsilon), \\ \tilde{v}^{n+1-j} &= \cos(j\pi h) - \frac{1}{4}\cos(2j\pi h) - (\frac{3}{4}-\varepsilon). \end{aligned}$$

Straightforward substitutions of the altered expressions for $\tilde{v}^j$ and $\tilde{v}^{n+1-j}$ into Eq. (7.21) renders the analytical expressions for the eigenvalues of $\tilde{P}^T M^{-1} A$. The next question which needs to be answered is, given a fixed kh, how do we find ε? ε should be chosen such that the projection error E is minimized. In order to find this value, we can use two approaches. The first approach is straightforward; our ultimate aim is to have the eigenvalue of $\lambda^j(\tilde{P}^T M^{-1} A)$ at index $j_{\min,h}$ to be equal to 1. Recall from the proof of Corollary 11.2 that in the absence of errors the eigenvalues of A_{2h} can be written as a product of the eigenvalues of A and the eigenvalues of B. Thus, using Eq. (7.19), we can write

$$\begin{aligned} \lambda^j(A_{2h}) &= [I_{2h}^h]^j A^j [I_h^{2h}]^j, \\ &= \left(v^j\right)^2 \lambda^j(A) + \left(v^{n+1-j}\right)^2 \lambda^{n+1-j}(A) = \lambda^j(A)\lambda^j(B). \end{aligned} \tag{7.23}$$

Note that the sum of $\left(v^j\right)^2$ and $\left(v^{n+1-j}\right)^2$ in expression Eq. (7.23) are exactly equal to $\lambda^j(B)$. If we want Eq. (7.23) to hold at index $j_{\min,h}$ in the presence of errors, we need to pick ε such that $\left(v^{n+1-j}\right)^2 = 0$, which is equivalent to

$$\varepsilon = 0.75 - (\cos(j\pi h) - \frac{1}{4}\cos(2j\pi h)). \tag{7.24}$$

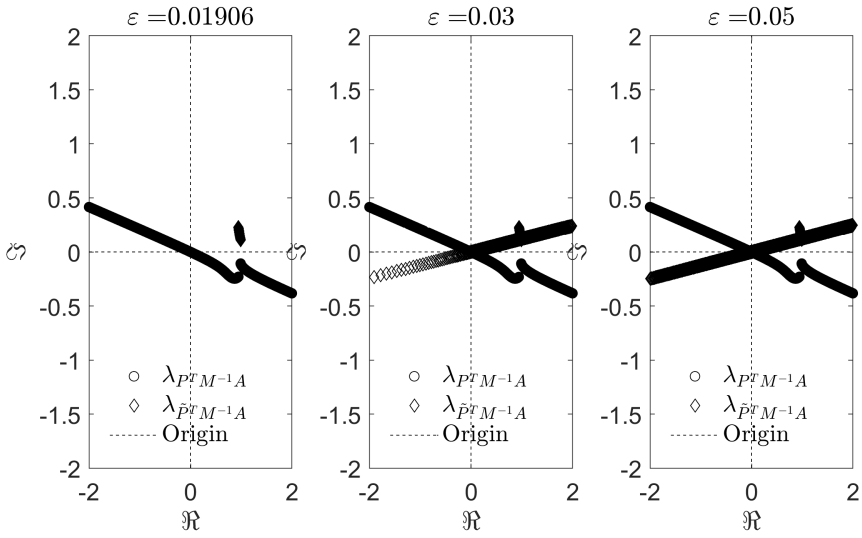
This way the near-zero eigenvalue of A_{2h} will always be proportional to the near-zero eigenvalue of A. Fortunately, the eigenvalues of B containing the term ε are independent of the eigenvalues of A. Therefore, finding ε primarily depends on the approximation scheme which determines the eigenvalues of B. An interesting observation is that ε is completely determined by the step-size h and therefore by the grid resolution kh.
We can take advantage of this k-independence, as it enables us to determine a ε without having to account for the wavenumber. Also, once we find an ε which works for some kh, then it will work for all k as long as kh remains constant. Thus, especially for practical applications of higher-dimensional problems, instead of computing the exact smallest eigenvalues of the fine- and coarse-grid operator, we can find the ε by performing a grid search for some small k. A similar strategy was used in [107] for the open cavity problem in order to find the optimal parameter for a given k and a given paritition in the context of optimized Schwarz methods (with overlap). There the best parameter was chosen to be the one which resulted in the smallest GMRES residual. In our case, the best parameter ε is the one which minimizes the projection error for some fixed h. Therefore, for MP 2 and MP 3, we use the heuristic in Algorithm 9. This provides a practical alternative to computing the analytical expressions for the eigenvalues of B.

Algorithm 9: Projection Error Minimizer

Initialization:
Initialize k small, $v_h^{j_{\min}}$, $\varepsilon_0 = 0.0001$, tol = 10^{-4}
for $c = 1, 2, ..m$ **do**
  Compute $E_{c\varepsilon_0}$ using $c\varepsilon_0$ to construct Z
  $y1 = Z^T v_h^{j_{\min}}$, $w = v_h^{j_{\min}\,T} Z, B = Z^T Z$
  $By2 = y1$, solve for $y1$ ▷ direct or iteratively
  $E_{c\varepsilon_0} = v_h^{j_{\min}\,T} v_h^{j_{\min}} - wy2$
  while $\bar{E}_{c\varepsilon_0} > tol$ **do**
    Compute $E_{(c+1)\varepsilon_0}$ and repeat until $\bar{E}_{(c+1)\varepsilon} <$ tol
  end
end
Update ε**:**
Set $\varepsilon = \tilde{c}\varepsilon_0$ for some $\tilde{c} \in [1, m]$.

We proceed by re-examining the spectrum of MP 1-A for $k = 10^6$ after introducing the weight-parameter. We have plotted the eigenvalues for $kh = 0.625$ for $\varepsilon = 0.01906$ (left), $\varepsilon = 0.03$ (center) and $\varepsilon = 0.05$ (right) in Fig. 7.4. It immediately becomes apparent that using the right parameter to minimize the projection error completely shifts the spectrum. Particularly, the left column contains the results where the optimal ε has been used and it can be noted that the spectrum stays clustered near $(1, 0)$ independent of the wavenumber k.

Figure 7.4: Eigenvalues of $P^T M^{-1} A$ and $\tilde{P}^T M^{-1} A$ using $kh = 0.625$ for various weight-parameters ε. The wavenumber k has been set to 10^6.



In the next section, we provide numerical experiments with these parameters for MP 1-A in order to test whether we obtain a reduced number of iterations as theorized.

7

7.6. NUMERICAL EXPERIMENTS

In this section, we examine the convergence behavior of the adapted solver using various kh. Unless stated otherwise, we deploy the CSL preconditioner with $(\beta_1, \beta_2) = (1,1)$ as we approximate the inverse of M using one $V(1,1)$-multigrid iteration. The tolerance level for the relative residual has been set to 10^{-7}.

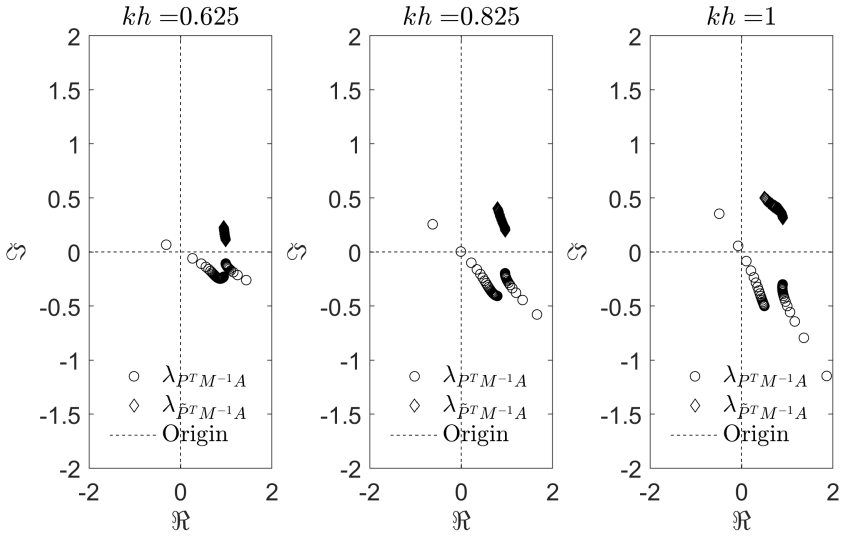
7.6.1. ONE-DIMENSIONAL MODELS

We start by collecting the numerical results for the one-dimensional constant wavenumber model problems using Dirichlet and Sommerfeld boundary conditions respectively.

MP1-A

For MP 1-A the results are presented in Table 7.3 and Table 7.4. Table 7.3 gives the number of iterations and Table 7.4 provides the projection error for increasing k. The numerical results presented are in line with with the theoretical results from Section 7.4.2 and the spectral analysis from Fig. 7.5. The consistently clustered spectrum near $(1,0)$ is reflected in a significant reduction in the number of iterations. On coarser levels, the number of iterations is still constant yet higher. In particular, compare the 6 iterations for $kh = 1$ with the 5 iterations for $kh = 0.825$. Even for such a simple model problem as MP1-A, these results present the first numerical evidence of obtaining true wavenumber independent convergence for very high wavenumbers without having to resort to keeping the shift in the CSL preconditioner small and inverting the preconditioner exactly.

Figure 7.5: Eigenvalues for $k = 10^6$ of $P^T M^{-1} A$ and $\tilde{P}^T M^{-1} A$ using various kh. The weight-parameter ε has been determined using equation Eq. (7.24).



If we keep the grid resolution practical $kh = C$, where $C \in [0.3125, 1]$, we observe that, unlike the previous deflation scheme using linear interpolation, the adapted scheme has an almost constant projection error as the wavenumber increases Table 7.4. With respect to the pollution error, it is necessary to keep the grid resolution $(k^3h^2 \approx 1)$. The last column of Table 7.3 contains the number of iterations using $(k^3h^2 \approx 1)$. These results are in line with the theory from Section 7.4.2, Corollary 12.2 and corroborate that an increasing wavenumber in fact leads to a lower projection error (Table 7.4) and hence a decreasing number of iterations (Table 7.3). This brings us to the final observation. The use of the weight-parameter ε becomes redundant in case we let $k^3h^2 = 1$. Recall that the weight-parameter is necessary in order to capture the perturbations which arise in mapping the eigenvectors as the wavenumber increases. Corollary 12.2 shows why this becomes unneccessary as the mappings naturally become more accurate as we let h go to zero.
Finally, compared to the CSL preconditioner whom shows $h-$independent convergence behavior, the use of the APD-preconditioner could allow for more accurate solutions while keeping the number of iterations constant and small. For example, one could use a higher-order finite difference scheme, combined with a coarser grid resolution in order to solve large scale problems more accurately without being penalized by an increased number of iterations.

Table 7.3: Number of GMRES-iterations for MP 1-A using the Adapted Preconditioned Deflation scheme APD(ε). ε has been determined using Eq. (7.24). APD(0) is the adapted deflation scheme without the projection error minimizer ε. The shift in CSLP has been set to $(1,1)$ and the preconditioner has been inverted inexactly.

k	APD(0.1250)	APD(0.0575)	APD(0.01906)	**APD(0)**	APD(0.00125)	**APD(0)**
	$kh = 1$	$kh = 0.825$	$kh = 0.625$	$kh = 0.625$	$kh = 0.3125$	$k^3h^2 = 1$
10^1	2	3	4	**4**	3	**4**
10^2	6	5	4	**4**	3	**4**
10^3	6	5	4	**6**	3	**4**
10^4	6	5	4	**12**	3	**4**
10^5	6	5	4	**59**	3	**4**
10^6	6	5	4	**509**	3	**4**

Table 7.4: Projection error for MP 1-A E(ε) for various kh. ε has been determined using Eq. (7.24).

k	E(0.1250)	E(0.0575)	E(0.01906)	E(0.00125)
	$kh = 1$	$kh = 0.825$	$kh = 0.625$	$kh = 0.3125$
10^1	0.0127	0.0075	0.0031	0.0006
10^2	0.0233	0.0095	0.0036	0.0007
10^3	0.0245	0.0095	0.0038	0.0007
10^4	0.0246	0.0095	0.0038	0.0007
10^5	0.0246	0.0095	0.0038	0.0007
10^6	0.0246	0.0095	0.0368	0.0007

MP1-B

Table 7.5 contains the results for MP1-B. We observe that including Sommerfeld radation conditions does not lead to deviating conclusions. While the results of the RFA for MP1-A are not analogously applicable to the case where we use Sommerfeld radiation conditions, we have used the same values for ε determined for MP1-A and observe that the convergence behavior is very similar. This provides numerical evidence for the notion that the convergence behavior for MP1-A and MP1-B are very similar and in both cases we obtain pure wavenumber independent convergence.

Table 7.5: Number of GMRES-iterations for MP 1-B using APD(ε) and **Sommerfeld** radiation conditions. ε has been determined using Eq. (7.24). The shift in CSLP has been set to $(1,1)$ and has been inverted inexactly.

k	APD(0.1250)	APD(0.0575)	APD(0.01906)	APD(0.00125)	**APD(0)**
	$kh = 1$	$kh = 0.825$	$kh = 0.625$	$kh = 0.3125$	$k^3h^2 = 1$
10^1	2	3	5	4	**5**
10^2	8	6	5	4	**5**
10^3	8	6	5	4	**5**
10^4	8	6	5	4	**5**
10^5	8	6	5	4	**5**
10^6	8	6	5	4	**5**

7.6.2. TWO-DIMENSIONAL MODELS

In this section we perform numerical experiments for the two-dimensional model problem using a constant wavenumber k and Dirichlet boundary conditions. The weight-parameter

ε has been optimized using Algorithm 9.

MP 2

Table 7.6 contains the number of iterations for $kh = 0.625$. We start with the case where we use the APD-scheme without using the weight-parameter ε. In this case, the third column shows that we can solve for $k = 1000$ in 53 iterations. To see the effect of the deflation technique without the influence of the CSL preconditioner, the fourth column contains the number of iterations for the AD-scheme including the weight-parameter. Remarkably, we can solve for $k = 1000$ in 18 iterations. Finally, combining both the weight-parameter and the approximate inversion of the CSL preconditioner, it takes 9 iterations to solve for $k = 1000$. If we would have inverted the CSL preconditioner exactly using a small shift to compensate for the use of no weight-parameter, it would take the solver 8 iterations to solve for $k = 1000$. These results are almost similar, but the use of the weight-parameter and approximate inversion is less computationally expensive compared to exact inversion of the CSL preconditioner. This is very promising as this implies that we can include a powerful preconditioner without having to pay the price of exact inversion at the finest level. While we do see a slight increase in the number of iterations throughout Table 7.6, these are the lowest reported number of iterations for a sequential implementation using such high wavenumbers. Without the use of the deflation preconditioner, CSLP-preconditioned GMRES would need 45 iterations to converge despite using a small shift of order $k^{-1} = 10^{-3}$.

Table 7.6: Number of iterations for MP 2 using $kh = 0.625$ using $APD(\varepsilon)$. ε has been optimized using Algorithm 9. Approximate CSLP inversion using one V(1,1)-cycle. Exact inversion includes the CSLP-shift $(1, k^{-1})$. AD contains no CSL preconditioner.

k	n^2	APD(0)	AD(0.01906)	APD(0.01906)	APD(0)	CSLP
		Approximate inversion			Exact inversion	
50	6400	4	13	5	3	9
100	25600	5	13	6	3	12
250	160000	10	13	6	5	20
500	640000	15	14	8	5	28
750	1440000	37	16	9	7	36
1000	2560000	53	18	9	8	45

We now repeat the same analysis for $kh = 0.3125$, with results reported in Table 7.7. Note that in this case we do not include an adjusted weight coefficient parameter, i.e. we set $\varepsilon = 0$. The inclusion of ε may in particular be more useful when using coarser grids. The reason behind this is that increasing the problem size already results in more accuracy and faster convergence (see Corollary 12.2). We also compare the performance of the adapted scheme with and without the inclusion of the CSL preconditioner. Results are reported in Table 7.7. If we compare these results to the ones obtained from Table 7.6, we note that, with the inclusion of the CSL preconditioner, increasing the problem size leads to faster convergence as theorized. Two important remarks can be made with respect to letting $kh = 0.3125$. First of all, in case we set $\varepsilon = 0$, we go from 53 iterations for $kh = 0.625$ to 8 iterations for $kh = 0.3125$ when $k = 1000$. However, once we include the weight-parameter (Table 7.6, column 5), we obtain 9 iterations for $kh = 0.625$ and 8 iterations for $kh = 0.3125$ and the convergence behavior becomes very similar irrespective of using a finer grid resolution. Second of all, the number of iterations with and without the CSL preconditioner is almost

the same for all reported values of k in Table 7.7. It may be argued that for fine grid resolutions, some computational time can be saved by excluding the CSL preconditioner as we need one multigrid iteration to approximate the inverse. The numerical results from the previous and current section show that there are plenty of optimization strategies to exploit when it comes to balancing a small and fixed number of iterations and a fine-grid resolution. The latter is equally important to obtain accurate solutions.

Table 7.7: Number of iterations for MP 2 using $kh = 0.3125$ using $APD(\varepsilon)$. Approximate CSLP inversion using one V(1,1)-cycle. AD contains no CSL preconditioner.

k	n^2	AD(0)	APD(0)
		Iterations	Iterations
25	6400	4	4
50	25600	4	4
100	102400	3	4
250	640000	4	4
500	2560000	5	5
750	5760000	5	5
1000	10240000	7	8

MP 4

In this section we present the numerical results for the industrial two-dimensional Marmousi problem (MP 4) (Section 7.2.3.1). Results are reported in Table 7.8 and Table 7.9. Starting with Table 7.8 we implement no correction using ε given that the grid for this model problem has been resolved such that $kh \leqslant 0.39$ on average and the maximum wavenumber is approximately 400. [1] Table 7.8 contains the results for frequencies $f = 1, 10, 20$ and 40 using 10 grids points per wavelength for the largest wavenumber k. The results show that even for this challenging problem, the APD-scheme leads to very satisfactory results. If we compare the results between DEF (which uses linear interpolation) and APD, we note an improved performance in terms of both metrics; solve time and iterations. For $f = 1$, the number of iterations for APD are larger than DEF. The latter method takes 6 iterations, while the former takes 3 iterations, which is clearly reflected in the lower solve time. Once we start increasing the frequency, we note that the APD scheme quickly catches up in terms of both iterations and solve time. For example for $f = 40$, we obtain 5 iterations and a total solve time of 111.78 seconds.

[1] If we use the dimensionless model we obtain a wavenumber of $\sqrt{\left(\frac{2\pi 40}{2587.5}\right)^2 \times 2048 \times 8192} \approx 398$.

Table 7.8: Results for the Marmousi problem using 10 gpw. All solvers are combined with the inexact inversion of the CSL preconditioner using shifts (1,1). TL denotes two-level.

f	DEF	APD	DEF	APD
	Iterations		Solve Time (s)	
1	3	6	1.72	4.08
10	16	5	7.30	3.94
20	31	5	77.34	19.85
40	77	5	1175.99	111.78

Table 7.9 repeats the same simulation without the use of the CSL preconditioner. We observe very similar behavior as compared to the results obtained for the constant wavenumber problem (Table 7.6). Excluding the CSL preconditioner and solely using the deflation preconditioner results in a constant number of iterations and a significant reduction in sequential solve time. If we use the old deflation preconditioner (DEF) based on the linear interpolation scheme, then a similar effect can not be observed. For example for $f = 40Hz$, we obtain 82 iterations versus 12 for the adapted scheme. These results provide a promising basis for future research where the coarse-grid solve can be optimized and balanced with respect to the number of iterations and time scalability of the overarching solver.

Table 7.9: Results for the Marmousi problem using 10 gpw using no CSL preconditioner. TL denotes two-level.

f	DEF	APD	DEF	APD
	Iterations		Solve Time (s)	
1	10	12	1.41	2.76
10	20	12	2.44	2.80
20	35	12	17.15	15.15
40	82	12	219.39	85.87

7.6.3. THREE-DIMENSIONAL MODELS

In this section we present some three-dimensional numerical results for MP 3. We have used the same weight-parameter ε from the two-dimensional test problem MP 2.

MP 3

From Table 7.10 we see that even without the weight-parameter ε, the 3D-results show promising features for scalability with respect to the number of iterations. These results are in line with the previous results obtained for the one- and two-dimensional constant wavenumber model. We similarly expect the importance of ε to decrease along with kh.

Table 7.10: Number of iterations for MP 3 using $kh = 0.625$. *AD* contains no CSL preconditioner. APD contains the CSLP with shift $(1,1)$, which has been inverted inexactly.

k	n^3	APD(0)	APD(0.00125)
		Iterations	Iterations
5	512	4	4
10	4096	4	4
25	64000	5	4
50	512000	5	4
75	1728000	6	4

7.7. CONCLUSION

We have shown that the near-zero eigenvalues for deflation based preconditioners are related to the near-kernel eigenmodes of the fine-grid operator A and coarse-grid operator A_{2h} being misaligned. This effect can be attributed to the interpolation scheme not being able to sufficiently approximate the transferring of the grid functions at very large wavenumbers.

We analytically measure the effect of these errors on the construction of the projection preconditioner by means of the projection error. The quality of the deflation vectors determine whether the projection error dominates. To minimize the projection error, we proposed the implementation of a higher order approximation scheme to construct the deflation vectors. Incorporating a weight-parameter within the approximation scheme provides sufficient counterbalance to mitigate the re-appearance of the near-zero eigenvalues. Two options are available for determining the weight-parameter. The first is to use the analytical eigenvalues of B at the smallest index $j_{\min,h}$ and solve for ε. This approach is straightforward to use as it primarily depends on the eigenvalues of B, which can be computed independently of the eigenvalues of A. The second approach is to use the projection error minimizing algorithm, which finds the ε which minimizes the error on average.

Even without adjusting the weight-parameter, the spectrum of the proposed operator is still the most favourable compared to other preconditioning operators based on deflation. The numerical results are in line with the theoretical results as the number of iterations for both the one-, two- and three-dimensional constant wavenumber model problems are more or less wavenumber independent. Numerical evidence furthermore supports the notion that the proposed method also works for non-selfadjoint and heterogeneous problems, even when the CSL preconditioner is excluded. The latter allows for a substantial speed up.

8

MULTI-LEVEL DEFLATION

Parts of this chapter have been **accepted** for publication in Journal of Computational Physics (2022).

8.1. MULTI-LEVEL DEFLATION METHODS

In the previous chapter, we discussed the two-level deflation preconditioner, which we denoted by ADP (Adapted Deflation Preconditioner). While the method resulted in close to wavenumber independent convergence, it still relies on the exact solution on the second level. Consequently, for the 3D model problem, we were only able to test up to $k = 75$ as the memory requirement for the exact solve starts to dominate the numerical costs in 3D. As the focus of this dissertation is on sequential methods, apart from using parallelization techniques, another way to expand the reach of the preconditioner is to consider a multi-level deflation method. This is essentially a recursive application of the two-level deflation preconditioner from [78]. A natural question which arises is whether we can extend the wavenumber independent convergence to a multi-level setting, thereby combining both the gain in computational efficiency with our previous scalability results. The structure of this chapter is as follows. We start by introducing our model problems in Section 8.3. We then discuss the deflated Krylov methods and the multi-level algorithm in Section 8.4. We proceed by extensively developing theory for the multi-level deflation operator in Section 8.5. We perform Rigorous Fourier Analysis (RFA) by block-diagonalizing the resulting operators and inspecting the spectral properties. Finally we present numerical results for benchmark problems in Section 8.6.

8.2. LITERATURE OVERVIEW

A large branch within this research has focused on developing preconditioners, such as the (Complex) Shifted Laplacian (CSL) [32, 33, 108, 109]. In order to apply the preconditioner, one multigrid cycle is used to approximate its inverse. The latter serves as an alternative to using multigrid as a stand-alone solver as the method is generally known to diverge for the Helmholtz equation once coarser levels are reached [102]. Some works have focused on obtaining a stand-alone multigrid solver [35, 110–112], with success for either practical wavenumbers and/or one-dimensional model problems.

A recent and promising branch of research has combined its efforts towards preconditioning techniques based on domain decomposition methods applied to the corresponding (shifted) problem [25]. These methods split the computational domain in subdomains and solve a local subproblem of smaller dimension using a direct method [85, 87–90]. The performance of these preconditioners depends on the accuracy of the transmission conditions, which currently is robust for constant wavenumber model problems [92, 93]. While the domain decomposition preconditioners have resulted in a reduced number of iterations and higher computational efficiency by exploiting parallelization strategies, the number of iterations still grows with the wavenumber k.

As a result, some have studied the use of deflation techniques (combined with the CSL preconditioner) in order to accelerate the convergence of the Krylov subspace method, which we denote DEF [28–30]. Incorporating the deflation preconditioner has improved the convergence, but taxed the efficiency in terms of memory and computational cost. For a two-level deflation preconditioner, the direct solve on the second level takes up most of the computational power and memory. Consequently, multi-level variants of the two-level method have been proposed in order to counter this effect [27, 29]. A multi-level extension replaces the direct solve in the two-level method by applying a similar two-level extension recursively combined with an outer Flexible GMRES (FGMRES) solver. In both variants, however,

the number of iterations still grows with the wavenumber k.

8.3. PROBLEM DEFINITION

We now continue by defining the model problem which we use to develop the theory. In Section 8.6 we elaborate on more sophisticated model problems for numerical experimentation purposes. For now, we start by focusing on a one-dimensional mathematical model using a constant wavenumber $k > 0$, which we denote by MP 1-A.

MP 1-A

$$\begin{aligned} -\frac{d^2u(x)}{dx^2} - k^2\,u(x) &= 1, \quad x \in \Omega = (0,1), \\ u(x) &= 0, \quad x = 0, \\ u'(x) - iku(x) &= 0, \quad x = 1. \end{aligned} \tag{8.1}$$

We refer to this model problem as MP 1-A. If we define $h = \frac{1}{n}$, where n is chosen according to $kh = \frac{2\pi}{c}$, where c is the number of grid points per wavelength, then discretization on the unit interval using second order finite differences leads to

$$\frac{-u_{j-1} + 2u_j - u_{j+1}}{h^2} - k^2 u_j = f_j, j = 1,2,\ldots,n.$$

Lexicographic ordering leads to the following linear system and eigenvalues for MP 1-A with indices $j = 1,2,\ldots n$

$$\begin{aligned} Au &= \frac{1}{h^2}\text{tridiag}[-1 \;\; 2-k^2h^2 \;\; -1]u = f, \\ \hat{\lambda}^j &= \frac{1}{h^2}\left(2 - 2\cos(j\pi h)\right) - k^2. \end{aligned} \tag{8.2}$$

This simple model problem will allow us to develop the theory for the constant wavenumber case, as finding robust multi-level solvers for this case is still an active and current research area. To allow for more practical examples, we introduce MP 1-B as the model problem where Sommerfeld radiation conditions have been implemented.

MP 1-B

$$\begin{aligned} -\frac{d^2u(x)}{dx^2} - k^2\,u(x) &= 1, \quad x \in \Omega = (0,1), \\ u'(x) - iku(x) &= 0, \quad x = 0, \\ u'(x) - iku(x) &= 0, \quad x = 1. \end{aligned} \tag{8.3}$$

8.4. DEFLATED KRYLOV METHODS

We start by briefly explaining the two-level deflation preconditioning technique to solve the resulting linear system. We then proceed by extending the two-level method recursively to a multi-level Krylov method.

8.4.1. TWO-LEVEL DEFLATION

For a linear system $Au = f$ we construct the deflation preconditioner P where the column space of Z is used as the deflation subspace. The aim of including a deflation preconditioner is to project the unwanted near-zero eigenvalues to zero such that the convergence of the underlying Krylov subspace method can be accelerated. For the two-level method, the preconditioner P in fact is a projection operator. As for the deflation matrix, Z can be interpreted as interpolating from the coarse grid to the fine grid.

$$P = I - AQ + Q \text{ where } Q = ZE^{-1}Z^T \text{ and } E = Z^T AZ$$

The inexact inversion which will be used for a multi-level approach requires the addition of an extra term Q in order to prevent synthetic close-to-zero eigenvalues from obstructing the convergence of the Krylov solver [27, 82, 113]. Without the addition of Q, the deflation operator is sensitive to rounding errors stemming from the inexact inversion of E. In [78], we used higher-order Bezier curves to construct Z. Using these higher-order polynomials, the prolongation and restriction operator act on a grid function as follows

$$Z[u_{2h}]_i = \begin{cases} \frac{1}{8}\left([u_{2h}]_{(i-2)/2} + 6[u_{2h}]_{(i)/2} + [u_{2h}]_{(i+2)/2}\right) & i \text{ is even,} \\ \frac{1}{2}\left([u_{2h}]_{(i-1)/2} + [u_{2h}]_{(i+1)/2}\right) & i \text{ is odd,} \end{cases}, \tag{8.4}$$

for $i = 1, \ldots, n-1$ and for $i = 1, \ldots, \frac{n}{2}$. To obtain even better convergence, the CSL preconditioner was included, which is given by

$$M = L - (\beta_1 + \sqrt{-1}\beta_2)k^2 I,$$

where $(\beta_1, \beta_2) \in [0, 1]$ and L is the discretized Poisson equation. The system to be solved becomes $M^{-1}PAu = M^{-1}Pf$. By allowing higher-order interpolation schemes, the near-zero eigenspace of the fine- and coarse-grid coefficient matrix remains perfectly aligned. As a result, the smallest eigenvalue in magnitude of both A and E is located at the same index. This prevents the eigenvalues of the deflated system from shifting towards the origin. While the method provides close to wavenumber independent convergence in one- and two-dimensions for fairly large wavenumbers $k = 10^6$ (1D) and $k = 10^3$ (2D), it requires the exact solve of the coarse-grid coefficient matrix E, adding to the computational complexity in 3D.

In Algorithm 1, we present the two-level deflated FGMRES algorithm, where we use the following abbreviations: MV (matrix vector product), MM (matrix matrix product), ES (exact solve), VU (vector update), AS (approximate solve), DP (dot product). We moreover let C_{it} denote the number of constant iterations. The motivation for using FGMRES lies in the recursive process which will be applied in order to avoid having to solve the coarse-grid system on the second level exactly. We have split the pseudo-code into two parts. The blue section contains the part where the deflation preconditioner is applied. All matrix vector multiplications within the blue section are sparse. The pink section is the general GMRES-process. Furthermore, Table 8.1 contains the number of non-zero elements per column of the operators involved. These can be used to quantify the flops for sparse matrix-matrix and sparse matrix-vector products. Note that in the context of the multi-level algorithm, the number of non-zeros after the second level remains the same. We include the largest dimension-dependent constants for the leading order time complexity term. For example,

for the 2D matrix-vector product we take $7n$ as the leading term instead of $5n$. As a result, we obtain a strict upperbound for the costs involved, even for 1D and 2D problems and the costs in practice will be less.

Table 8.1: Upper bound to number of non-zero elements per column of $A \in \mathbb{R}^{n\times n}, E \in \mathbb{R}^{m\times m}, Z \in \mathbb{R}^{n\times m}$ and $Z^T \in \mathbb{R}^{m\times n}$.

Operator	Linear			Quadratic		
	1D	2D	3D	1D	2D	3D
A	3	5	7	3	5	7
E	3	3^2	3^3	7	7^2	7^3
Z	3	3^2	3^3	5	5^2	5^3
Z^T	2	2^2	2^3	3	3^2	3^3
AZ	5	5^2	5^3	9	9^2	9^3

Algorithm 10: Two-level Deflation FGMRES

Initialization:
Choose u_0 initial guess and dimension k of the Krylov subspaces.
Define $(k+1)\times k$ $\bar{H}_k$ and initialize to zero.
Arnoldi process:
$r_0 = f - Au_0$, $\beta = \|r_0\|_2$, $v_1 = r_0/\beta$.
for $j = 1,2,...k$ **do**
  $\hat{v} = Z^T v_j$ ▷ MVP - $5^d n$
  $\tilde{v} = E^{-1}\hat{v}$ ▷ ES - m^2
  $t = Z\tilde{v}$ ▷ MVP - $3^d m$
  $s = At$ ▷ MVP - $7n$
  $\tilde{r} = v_j - s$ ▷ VU - n
  $r = M^{-1}\tilde{r}$ ▷ AS - $C_{it}n$
  $x_j = r + t$ ▷ VU - n
  $w = Ax_j$ ▷ MVP - $7n$
  for $i = 1,2,...j$ **do**
    $h_{i,j} = (w, v_j)$ ▷ DP - jn
    $w = w - h_{i,j}v_i$ ▷ VU - jn
  end
  Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$. ▷ MVP - n
  Define $X_k = [x_1, x_2, \ldots, x_k]$ and $\bar{H}_k = \{h_{i,j}\}_{1\leqslant i\leqslant j+1, 1\leqslant j\leqslant k}$ ▷ MVP - n
end
Form approximate solution:
Compute $u_k = u_0 + X_k y_k$ where $y_k =_y \|\beta e_1 - \bar{H}_k y\|_2$.
Restart:
If satisfied stop, else set $u_0 \leftarrow u_k$ and repeat Arnoldi process.

Considering Algorithm 10, the vector update $x_j = r + t$ is split in two parts. r contains the application of $I - AQ$ to the vector v_j and then lastly applies the preconditioner M to obtain r. The vector t contains the part where we add Q to $I - AQ$ in order to prevent synthetic near-zero eigenvalues due to rounding errors. Analyzing the costs of Algorithm

1, confirms that the dominant factor is $\mathcal{O}(m^2)$. Furthermore, in order to mitigate the cost of the preconditioning step, one ($C_{it} = 1$) multigrid F-cycle is generally applied in order to approximate the solution of the system $Mr = \tilde{r}$ [29]. When opting for this configuration, the shift β_2 has to be kept large enough for multigrid to converge [33, 35, 102]. Another option is by allowing a few GMRES-iterations to approximate the preconditioner. For example in the context of using multigrid as a preconditioner, the standard relaxation step is replaced by 10-40 GMRES-iterations, acting as a polynomial smoother. On each level the unstable Jacobi and Gauss-Seidel smoother are replaced by Krylov iterations [110, 114, 115].

8.4.2. MULTI-LEVEL DEFLATION

As mentioned previously, apart from the standard computational costs associated with the FGMRES-algorithm, the largest additional cost comes from solving the coarse-grid system exactly, which dominates with the factor $\mathcal{O}(m^2)$. In order to circumvent this, we apply the two-level cycle recursively. Before we expand the two-level algorithm to the multi-level algorithm, a few remarks are in place. We deploy five changes, apart from using Bezier interpolation polynomials as a basis for the deflation vectors. First, application of the CSL preconditioner to the Helmholtz operator shifts the spectrum towards the complex plane and resolves the indefiniteness. On levels where the matrix becomes negative definite, we apply a Jacobi iteration using the diagonal matrix of the CSLP as the preconditioner M. Second, the multi-level preconditioner is applied to A rather than AM^{-1}. This saves one matrix-vector product per level. Third, while the use of the CSLP preconditioner together with a geometric multigrid method for approximate inversion works well for homogeneous problems, it is not suitable for heterogeneous problems with high contrasts [27, 29]. As we are interested in heterogeneous problems and require only an approximate application of the preconditioner, we perform Krylov subspace iterations to approximately invert the CSLP. As mentioned previously, this can be considered as applying a polynomial smoother in the context of multigrid, which damps both ends of the spectrum. We let C_{it} denote the constant for the maximum number of iterations. The number of Krylov subspace iterations as a smoother ranges from 5-40 for two-dimensional constant wavenumber model problems, where the stopping criterion results in the residual to be scaled with kh on each level [110, 115, 116]. We use Bi-CGSTAB as the computational costs and memory do not grow with the number of iterations such as is the case with non-restarted GMRES. We moreover do not require convergence or set any tolerance dependent on the level. However, we set the maximum number of Bi-CGSTAB iterations at a constant times $\lceil n^{(l)} \rceil^{\frac{1}{4}}$, where $n^{(l)}$ denotes the problem size on level l where the linear system is still indefinite. Our motivation for doing so is twofold. Primarily we want to have the number of outer FGMRES iterations as small as possible while the wavenumber increases, as FGMRES becomes more computationally expensive when more iterations are needed. Second of all, we do not require the residual to remain orthogonal to all previous components, we can use Bi-CGSTAB to achieve a smaller residual within the multi-level hierarchy without necessarily imposing that it is in fact the minimized residual. Fourth, given that we are no longer using multigrid for the approximate inversion, the restrictions for choosing the complex shift can be lifted. Thus, we can take advantage of using a small shift which makes the preconditioner more similar to the original Helmholtz operator and keeps the property of lifting the indefiniteness at certain levels. As a result, we will be able to test our algorithm on heterogeneous models with highly varying contrast profiles. Fifth, instead of allowing many iterations on each coarse level, we

only allow one iteration on the coarser levels. Consequently, we obtain a V-cycle, which leads to a similar V-cycle structure from multigrid when taking $\gamma = 1$, see Section 8.4.2. The multi-level deflation algorithm is given below, where we used the number of non-zero elements from Table 8.1 to account for the dimension dependent constants for the sparse matrix-matrix and sparse matrix-vector products on subsequent levels.

Algorithm 11: Multi-level ADP Implementation

Initialization

Set $A^{(1)} = A, M^{(1)} = M, n^{(1)} = n$

for $l = 1, 2, \ldots m$ *the coarsest level* **do**

  Construct $Z^{(l,l+1)}$ and $Z^{(l,l+1)T}$

  Construct $A^{(l+1)} = {Z^{(l,l+1)}}^T A^{(l)} Z^{(l,l+1)}$ ▷ MMP - $45^d \cdot n^{(l+1)}$

  Construct $M^{(l+1)} = {Z^{(l,l+1)}}^T M^{(l)} Z^{(l,l+1)}$ ▷ MMP - $45^d \cdot n^{(l+1)}$

end

Iterative stage

$l = 1$, $u_0^{(1)} = 0$

Solve $A^{(1)} u^{(1)} = b^{(1)}$ using Two-level Deflated FGMRES

$\hat{v}^{(2)} = {Z^{(1,2)}}^T v^{(1)}$ ▷ MVP - $5^d \cdot n^{(1)}$

if $l + 1 = m$ **then**

  Solve $\tilde{v}^{(2)} = {A^{(2)}}^{-1} \hat{v}^{(2)}$ exactly

else

  $l = l + 1$, $\tilde{v}_0^{(2)} = 0$

  Solve $A^{(2)} \tilde{v}^{(2)} = \hat{v}^{(2)}$ using Two-level Deflated FGMRES

  $\hat{v}^{(3)} = {Z^{(2,3)}}^T v^{(2)}$ ▷ MVP - $5^d \cdot n^{(2)}$

  if $l + 1 = m$ **then**

    Solve $\tilde{v}^{(3)} = {A^{(3)}}^{-1} \hat{v}^{(3)}$ exactly

  else

    $l = l + 1$, $\tilde{v}_0^{(3)} = 0$

    Solve $A^{(3)} \tilde{v}^{(3)} = \hat{v}^{(3)}$ using Two-level Deflated FGMRES

    $\hat{v}^{(4)} = {Z^{(3,4)}}^T v^{(3)}$ ▷ MVP - $5^d \cdot n^{(3)}$

    ⋮

    if $l + 1 = m$ **then**

      Solve $\tilde{v}^{(m)} = {A^{(m)}}^{-1} \hat{v}^{(m)}$ exactly ▷ ES - $\mathcal{O}(1)$

    end

    $t^{(m-1)} = Z^{(m-1,m)} \tilde{v}^{(m)}$ ▷ MVP - $3^d \cdot n^{(m)}$

    $s^{(m-1)} = A^{(m-1)} t^{(m-1)}$ ▷ MVP - $7^d \cdot n^{(m-1)}$

    $\tilde{r}^{(m-1)} = v^{m-1} - s^{(m-1)}$ ▷ VU - $n^{(m-1)}$

    $r^{(m-1)} = {M^{(m-1)}}^{-1} \tilde{r}^{(m-1)}$ ▷ AS - $n^{(m-1)}$

    $x^{(m-1)} = r^{(m-1)} + t^{(m-1)}$ ▷ VU - $n^{(m-1)}$

    $w^{(m-1)} = A^{(m-1)} x^{(m-1)}$ ▷ MVP - $7^d \cdot n^{(m-1)}$

    ⋮

  end

  $t^{(2)} = Z^{(2,3)} \tilde{v}^{(3)}$ ▷ MVP - $3^d \cdot n^{(3)}$

  $s^{(2)} = A^{(2)} t^{(2)}$ ▷ MVP - $7^d \cdot n^{(2)}$

  $\tilde{r}^{(2)} = v^3 - s^{(2)}$ ▷ VU - $n^{(2)}$

  $r^{(2)} = {M^{(2)}}^{-1} \tilde{r}^{(2)}$ ▷ AS - $21(C_{it} {n^{(2)}}^{\frac{1}{4}}) n^{(2)}$

  $x^{(2)} = r^{(2)} + t^{(2)}$ ▷ VU - $n^{(2)}$

  $w^{(2)} = A^{(2)} x^{(2)}$ ▷ MVP - $7^d \cdot n^{(2)}$

end

Algorithm 12: Multi-level ADP Implementation cont'd.

Initialization
Set $A^{(1)} = A, M^{(1)} = M, n^{(1)} = n$
for $l = 1, 2, \ldots m$ *the coarsest level* **do**
  Construct $Z^{(l,l+1)}$ and $Z^{(l,l+1)\,T}$
  Construct $A^{(l+1)} = Z^{(l,l+1)^T} A^{(l)} Z^{(l,l+1)}$ ▷ MMP - $45^d \cdot n^{(l+1)}$
  Construct $M^{(l+1)} = Z^{(l,l+1)^T} M^{(l)} Z^{(l,l+1)}$ ▷ MMP - $45^d \cdot n^{(l+1)}$
end
Iterative stage
$l = 1$, $u_0^{(1)} = 0$
Solve $A^{(1)} u^{(1)} = b^{(1)}$ using Two-level Deflated FGMRES
$\hat{v}^{(2)} = Z^{(1,2)^T} v^{(1)}$ ▷ MVP - $5^d \cdot n^{(1)}$
if $l + 1 = m$ **then**
  Solve $\tilde{v}^{(2)} = A^{(2)^{-1}} \hat{v}^{(2)}$ exactly
else
  ⋮
end
$t^{(1)} = Z^{(1,2)} \tilde{v}^{(2)}$ ▷ MVP - $3^d \cdot n^{(2)}$
$s^{(1)} = A^{(1)} t^{(1)}$ ▷ MVP - $7^d \cdot n^{(1)}$
$\tilde{r}^{(1)} = v^1 - s^{(1)}$ ▷ VU - $n^{(1)}$
$r^{(1)} = M^{(1)^{-1}} \tilde{r}^{(1)}$ ▷ AS - $21(C_{it} n^{(1)^{\frac{1}{4}}}) n^{(1)}$
$x^{(1)} = r^{(1)} + t^{(1)}$ ▷ VU - $n^{(1)}$
$w^{(1)} = A^{(1)} x^{(1)}$ ▷ MVP - $7^d \cdot n^{(1)}$

A schematic representation is given below.

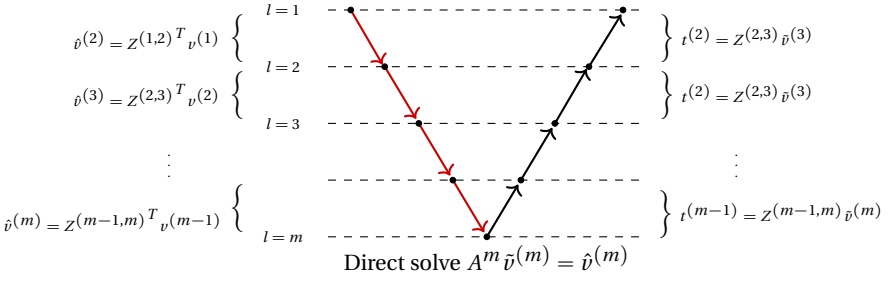


Figure 8.1: V-cycle Deflated FGMRES. The pink arrows represent the coarsening. The blue arrows represent the prolongation.

Using the above, we can formulate upper bounds in terms of FLOPs for the complete algorithm.

Theorem 13: Multi-level deflation upper bound number of operations

For $l = 1$, set $n^{(1)} = n$ and let $n^{(l)} = 2^{-ld} n$ for $l > 1$, where d denotes the dimension. Assume the following holds for $l \geqslant 1$

- Restriction $\hat{v}^{(l+1)} := Z^{(l,l+1)^T} \hat{v}^{(l)}$
$$\text{flops} \leqslant c_r 5^d n^{(l)}$$
- Prolongation $t^{(l)} := Z^{(l,l+1)} \tilde{v}^{(l+1)}$
$$\text{flops} \leqslant c_p 3^d n^{(l+1)}$$
- Krylov Smoothing $r^{(l)} := {M^{(l)}}^{-1} \tilde{r}^{(l)}$ when $l < 3$
$$\text{flops} \leqslant c_k 7^d {n^{(l)}}^{1+\frac{1}{4}}$$
- Jacobi Smoothing $r^{(l)} := {M^{(l)}}^{-1} \tilde{r}^{(l)}$ when $l \geqslant 3$
$$\text{flops} \leqslant c_j 7^d n^{(l)}$$
- Matrix vector product:= $w^{(l)} = A^{(l)} x^{(l)}$
$$\text{flops} \leqslant c_v 7^d n^{(l)}$$
- Coarse-grid solve:= $A^{(m)-1} \hat{v}^{(m)}$
$$\text{flops} \leqslant c_0$$

Then
$$\text{total flops} \leqslant C_d O(n^{1+\frac{1}{4}}),$$
where C_d is a constant which only depends on the dimension d.

8

Proof. At each level l, after we have obtained $w^{(l)}$, we proceed with the Arnoldi process (see pink section, Algorithm 10), which is already $\mathcal{O}(n^l)$ given that the maximum number of FGMRES iterations at each level is set at one. We thus obtain the following upper bound

of the additional costs occurred with the preconditioning step

$$
\begin{aligned}
\text{flops} &\leqslant c_0 + \left(c_r 5^d + C_k 7^d n^{(l)^{\frac{1}{4}}} + c_v 7^d \right) n^{(l)} + c_p 3^d n^{(l+1)}, \\
&+ \left(c_r 5^d + C_k 7^d n^{(l+1)^{\frac{1}{4}}} + c_v 7^d \right) n^{(l+1)} + c_p 3^d n^{(l+2)}, \\
&+ \ldots \left(c_r 5^d + C_k 7^d + c_v 7^d \right) n^{(m-1)} + c_p 3^d n^{(m)}, \\
&= c_0 + \left(c_r 5^d + C_k 7^d n^{(1)^{\frac{1}{4}}} + c_v 7^d \right) n^{(1)} + C_k 7^d n^{(2)^{\frac{1}{4}}} n^{(2)}, \\
&+ \left(c_r 5^d + C_j 7^d + c_v 7^d + c_p 3^d \right) \sum_{l=2}^{m-1} 2^{-dl} n^{(1)} + c_p 3^d n^{(m)}, \\
&\leqslant c_0 + \left(c_r 5^d + C_k 7^d n^{(1)^{\frac{1}{4}}} + c_v 7^d \right) n^{(1)} + C_k 7^d n^{(2)^{\frac{1}{4}}} n^{(2)}, \\
&+ \left(c_r 5^d + C_j 7^d + c_v 7^d + c_p 3^d \right) \sum_{l=2}^{m-1} 2^{-l} n^{(1)} + c_p 3^d n^{(m)}, \\
&< c_0 + \left(c_r 5^d + C_k 7^d n^{(1)^{\frac{1}{4}}} + \frac{1}{2} C_k 7^d n^{(1)^{\frac{1}{4}}} + c_v 7^d \right) n^{(1)} + c_p 3^d n^{(m)}, \\
&+ \frac{1}{2} \left(c_r 5^d + C_a 7^d + c_v 7^d \right) n^{(1)}, \\
&< C 7^d n^{(1+\frac{1}{4})} = \mathcal{O}(n^{(1+\frac{1}{4})}),
\end{aligned}
$$

where we used that $n^{(1)} = n$ and the fact that $\sum_{l=2}^{m-1} 2^{-l} < \frac{1}{2}$. ■

The upper bound to the total computational costs is constructed with respect to the iterative stage and already accounts for the costs of the matrix vector multiplication in FGMRES. It in fact bounds the total cost of the multi-level extension of the blue section in Algorithm 1. Overall, the algorithm runs in $\mathcal{O}(n^{(1)^{1+\frac{1}{4}}})$ time complexity. However, a few points need further explanation and discussion.

The construction of the coarse-grid systems on each level l requires two sparse matrix-matrix multiplications. While the maximum number of non-zeros along each column remains constant with respect to $n^{(1)}$ (see Table 8.1), it results in a constant of 45^d for the matrix-matrix multiplication to construct $A^{(l>1)}$. While this may seem expensive, it already pays-off for very large and highly indefinite 2D and 3D problems, as the constant is independent of the fine-grid problem size $n^{(1)}$. We illustrate this through our numerical experiments in Section 8.6.

Moreover, as mentioned previously, the multi-level preconditioner is applied to A rather than AM^{-1}. By using the 'First Deflate, then Precondition' method, we save one extra matrix-matrix product, one matrix-vector product and one extra application of the preconditioning step [83].

Finally, we restrict the number of FGMRES iterations on each level to one, whereas a sequence of $(8,2,1)$ and $(6,2,2)$ iterations are used [27, 29, 83]. For example $(8,2,1)$ denotes, 8 iterations on level $l = 2$, 2 iterations on $l = 3$ and 1 iteration on all levels $l > 3$. Thus, on the finest level $n^{(1)}$, the largest cost related to the matrix-vector product during the iterative

stage is $\mathcal{O}(8 \cdot 3^d n^{(1)})$ compared to $\mathcal{O}(7^d n^{(1)})$. While the dimension-dependent constant 7^d is approximately 1.5 times larger than $8 \cdot 3^d$ for $d = 3$, the significant reduction in the number of outer iterations provides an advantageous leverage.

8.5. INSCALABILITY

In this section we extend the theoretical results of the two-level ADP-scheme to a multi-level setting for MP 1-A. Given that the coefficient matrix remains normal, spectral analysis can be performed to assess the convergence behavior. We have provided a detailed summary of the literature as regards the role of the eigenvalues when the matrix is non-normal in [78].

8.5.1. MULTI-LEVEL MAPPING

In order to develop theory for the multi-level ADP-scheme from the two-level ADP-scheme, we need expressions for the nested or composite mappings between the fine and coarse spaces. Similar to our approach for the two-level method in [78], we start with the linear case and extend it to the quadratic case. In Theorem 14 we start by deriving analytical expressions for the actions of the intergrid transfer operators on eigenvectors of each respective coarse space for the linear case, whereas Corollary 14.1 contains the expressions for the quadratic case.

Theorem 14: Multi-level Prolongation and Restriction (linear)

Let Z_m be the $n_{m-1} \times n_m$ prolongation matrix based on linear interpolation for $m = 1,2,\dots m_{\max}$, with $n_m = \frac{n}{2^m}$. If we define $v_m^j = \sin(2^m hi\pi j)$, and $v_m^{j'} = \sin(2^m hi\pi(n_m + 1 - j))$, where on the finest level we have $m = 0$, then there exist constants C_1^j and C_2^j depending on h such that restriction operator maps the eigenvectors to

$$\prod_{l=m}^{1} Z_l^T v_0^j = C_1^j v_m^j,\ j = 1,2,\dots,n_m \text{ and } \prod_{l=m}^{1} Z_l^T v_0^{j'} = C_2^j v_m^j,\ j = 1,2,\dots,n_m.$$

where $C_1^j = \left(\frac{1}{2}\right)^m \prod_{l=1}^{m} \left(1 + \cos(j\pi 2^{l-1} h)\right)$ and $C_2^j = \left(\frac{1}{2}\right)^m \prod_{l=1}^{m} \left(\cos(j\pi 2^{l-1} h) - 1\right)$. Similarly, the prolongation operator maps the eigenvectors to

$$\prod_{l=1}^{l} Z_l [v_m]_i = C_1^j [v_0^j]_i, \text{ for } i \text{ is odd. and } \prod_{l=1}^{l} Z_l [v_m]_i = C_2^j [v_0^j]_i, \text{ for } i \text{ is even..}$$

Finally, if we let $B_m = \prod_{l=1}^{m} Z_l \prod_{l=m}^{1} Z_l^T$ and $\hat{B}_m = Z_m Z_m^T$ for $m = 1,2,\dots,m_{\max}$, then B_m has dimension n_0 with n_m non-zero eigenvalues.

Proof. We start with a brief outline of the proof. We start by defining the mapping operators and the respective vector spaces and their bases to which they are applied. This allows us to move between fine and coarse spaces. Then we continue by showing the action of the restriction operator on the basis for these respective vector spaces. To keep an overview

of what is happening between the vector spaces on an abstract level, we use both the analytical operator and their matrix representations in the proof. We then repeat this for the prolongation operator. Once we have analytical expressions for these nested operators, we show that the kernel and range of the composite mapping consisting of the restriction and prolongation operator span a subspace containing the eigenvectors. We use this to show that the eigenvalues of B_m are related to the eigenvalues of $\hat{B}_m$.

Basis and ordering

We start by defining $n_m = \frac{n}{2^m}$ and rearranging the space spanned by the eigenvectors at each level such that we obtain the following subspace

$$\mathcal{V}_m^j = \text{span}\{v_m^j, v_m^{n_m+1-j}\},$$

for $j = 1,2,\ldots,n_{m+1}$. Moreover let

$$V_{m+1} = \bigoplus_{j=1}^{n_{m+1}} \text{span}\{v_{m+1}^j\},$$

denote the space spanned by the eigenvectors at a coarser level $m+1$. Note that the basis spans $\mathbb{C}^{n_m}$ and $\mathbb{C}^{n_{m+1}}$ as we can write

$$\mathbb{C}^{n_m} = \bigoplus_{j=1}^{n_{m+1}} \mathcal{V}_m^j \text{ and } \mathbb{C}^{n_{m+1}} = V_{m+1}^j,$$

and at each subsequent level $m+1$ we re-order the basis to obtain $\mathcal{V}_{m+1}$. Thus, on each level we define the automorphism such that we can bring the basis of V_m in to the order of $\mathcal{V}_m$

$$\alpha_{\pi(j)}^m : V_m \to V_m : j \mapsto n_m + 1 - (j-1) \text{ for } j \text{ is even.}$$

For $m = 0,1,2\ldots m_{\max}$, the linear interpolation and restriction operator maps between subsequent vector spaces

$$\mathcal{I}_m^{m+1} : \mathcal{V}_m \to V_{m+1}, \text{ such that } \mathcal{V}_m^j \mapsto \mathcal{I}_m^{m+1} V_m^j$$
$$\mathcal{I}_{m+1}^m : V_{m+1} \to \mathcal{V}_m, \text{ such that } v_{m+1}^j \mapsto \mathcal{I}_{m+1}^m v_{m+1}^j.$$

Restriction operator

We now apply the corresponding matrices to the respective eigenvectors on each level, where we let $\mathcal{I}_m^{m+1} = Z_{m+1}$. We start by taking $m = 0$. Using the basis of eigenvectors for $\mathcal{V}_0$ we have for index j

$$\begin{aligned}\left[Z_1^T v_0^j\right]_i &= \frac{1}{4}\left(\sin((2i-1)h\pi j) + 2\sin(2ih\pi j) + \sin((2i+1)h\pi j)\right),\\ &= \frac{1}{2}\left(1+\cos(j\pi h)\right)\sin(2hi\pi j),\\ &= C_{1,h}^j \left[v_1^j\right]_i.\end{aligned}$$

Now, for the complementary mode on level $m=0$ corresponding to index j we define $j' = n_0+1-j$. Note that we can write

$$[v_0^{j'}]_i = -(-1)^j \sin(ihj\pi), \quad i = 1,2,\dots n_m, \text{and } j = 1,2,\dots n_{m+1}. \tag{8.5}$$

Applying the restriction operator to the complementary eigenvector gives

$$\begin{aligned}\left[Z_1^T v_0^{j'}\right]_i &= \frac{1}{4}\left(\cos(j\pi h)\sin(2hi\pi j) - (-1)^{2i}\sin(2hi\pi j)\right),\\ &= \frac{1}{4}\left(\cos(j\pi h)-1\right)\sin(2hi\pi j),\\ &= C_{2,h}^j \left[v_1^j\right]_i.\end{aligned}$$

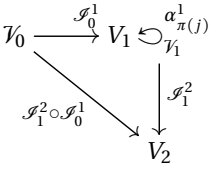
We thus have that at level $m=1$, the fine-grid eigenvectors from level $m=0$ are mapped by the restriction operator Z_1^T according to

$$Z_1^T v_0^j = C_{1,h}^j v_1^j,\ j = 1,2,\dots,n_1, \tag{8.6}$$

$$Z_1^T v_0^{n_0+1-j} = C_{2,h}^j v_1^j,\ j = 1,2,\dots,n_1. \tag{8.7}$$

Note that $v_1^j \in V_1\ \forall j$. Additionally, note that n_1 vectors from $\mathcal{V}_0$ are mapped to zero which implies that the nullspace of Z_1^T has $\dim\mathcal{N}(Z_1^T) = n_1$. In order to move from $m=1$ to $m=2$, which maps $\mathcal{V}_1 \to V_2$, we apply Z_2^T. The mapping trajectory is given by the following diagram

$\mathscr{I}_1^2 \circ \mathscr{I}_0^1 : \mathcal{V}_0 \xrightarrow{\mathscr{I}_0^1} \mathcal{V}_1 \xrightarrow{\mathscr{I}_1^2} V_2$, where



We obtain $\mathcal{V}_0$ by first applying $\alpha^0_{\pi(j)}$ such that we get the ordering of the basis in pairs j, j'. The restriction operator $\mathscr{I}_0^1$ maps these basis vectors to V_1. Then in order to move to the second coarse space V_2, we again have to reorder the basis on V_1 by applying the automorphism $\alpha^1_{\pi(j)}$. After permuting the elements of the basis, we can apply $\mathscr{I}_1^2$. Consequently, the range of $\mathscr{I}_1^2$ is V_2. This is equivalent to having a composition of the linear transformations $\mathscr{I}_1^2 \circ \mathscr{I}_0^1$. Thus, in terms of the matrix representations, applying Z_2^T gives

$$\begin{aligned}\left[Z_2^T\left[Z_1^T v_0^j\right]\right]_i &= C_{1,h}^j\left(Z_2^T\left[v_1^j\right]_i\right),\\ &= \frac{1}{2}\left(1+\cos(j\pi h)\right)\left(Z_2^T\sin(2hi\pi j)\right),\\ &= \frac{1}{2}\left(1+\cos(j\pi h)\right)\left(\frac{1}{4}\sin((2i-1)2h\pi j) + 2\sin((2i)2h\pi j) + \sin((2i+1)2h\pi j)\right),\\ &= \left(\frac{1}{2}\left(1+\cos(j\pi h)\right)\right)\left(\frac{1}{2}\left(1+\cos(j\pi 2h)\right)\right)\sin(4hi\pi j),\\ &= C_{1,h}^j C_{1,2h}^j \left[v_2^j\right]_i.\end{aligned}$$

As regards the complementary modes on level $m=1$ note that $\alpha^1_{\pi(j)} : V_1 \mapsto \mathcal{V}_1$ enables us to redefine $j' = n_1 + 1 - j$, where

$$
\begin{aligned}
&[v_1^{j'}]_i = -(-1)^j \sin(i2hj\pi), \\
&i = 1,2,\ldots n_1, \text{and } j = 1,2,\ldots n_2.
\end{aligned}
\tag{8.8}
$$

Thus, applying the restriction operator to the complementary modes on $m=1$ gives

$$
\begin{aligned}
\left[Z_2^T\left[Z_1^T v_0^{j'}\right]\right]_i &= C^j_{2,h}\left(Z_2^T\left[v_1^j\right]_i\right), \\
&= \frac{1}{2}\left(\cos(j\pi h)-1\right)\left(Z_2^T\left[v_1^j\right]_i\right), \\
&= \frac{1}{2}\left(\cos(j\pi h)-1\right)\left(\frac{1}{4}\left(\cos(j\pi h)\sin(2hi\pi j)-(-1)^{2i}\sin(2hi\pi j)\right)\right), \\
&= \left(\frac{1}{2}\left(\cos(j\pi h)-1\right)\right)\left(\frac{1}{2}\left(\cos(j\pi 2h)-1\right)\right)\sin(4hi\pi j), \\
&= C^j_{2,h}C^j_{2,2h}\left[v_2^j\right]_i.
\end{aligned}
$$

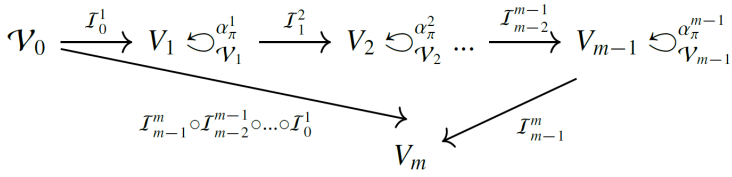
Note that $v_2^j \in V_2\ \forall j$. Consequently, using Z_1^T to map from level $m=0$ to $m=1$ and Z_2^T to map from level $m=1$ to $m=2$, results in the fine-grid eigenvectors being mapped in a nested application according to

$$
\begin{aligned}
Z_2^T\left(Z_1^T v_0^j\right) &= C_1^j v_2^j,\ j = 1,2,\ldots,n_2, \\
Z_2^T\left(Z_1^T v_0^{n+1-j}\right) &= C_2^j v_2^j,\ j = 1,2,\ldots,n_2, \text{ where,} \\
C_1^j &= \left(\frac{1}{2}\right)^m \prod_{l=1}^{m}\left(1+\cos(j\pi 2^{l-1}h)\right) \text{ and,} \\
C_2^j &= \left(\frac{1}{2}\right)^m \prod_{l=1}^{m}\left(\cos(j\pi 2^{l-1}h)-1\right).
\end{aligned}
$$

In this case, n_2 vectors from $\mathcal{V}_1$ are mapped to zero which implies that the nullspace of Z_2^T has $\dim\mathcal{N}(Z_2^T) = n_2$. Consequently, in order to move to $m=3$ which maps $\mathcal{V}_2 \to V_3$, we can continue applying Z_3^T. From here, it is easy to see that for each subsequent level $m>2$, consecutive application of the matrices Z_m^T is equivalent to the following linear mapping between the vector spaces $\mathcal{V}_m$

$$
\mathcal{I}^m_{m-1} \circ \mathcal{I}^{m-1}_{m-2} \circ \ldots \circ \mathcal{I}^1_0 : \mathcal{V}_0 \overset{\mathcal{I}^1_0}{\to} \mathcal{V}_1 \overset{\mathcal{I}^2_1}{\to} \mathcal{V}_2 \ldots \mathcal{V}_{m-1} \overset{\mathcal{I}^m_{m-1}}{\to} \mathcal{V}_m,
$$

which can be represented by the following diagram



We thus have $v_m^j \in V_m \; \forall j$, and in terms of the matrices, we therefore obtain

$$\begin{aligned}\left[\prod_{l=m}^{1} Z_l^T v_0^j\right]_i &= \left[Z_m^T Z_{m-1}^T \ldots \left[Z_2^T \frac{1}{2}\left(1+\cos(j\pi h)\right) v_1\right]\right]_i, \\ &= \left[Z_m^T Z_{m-1}^T \ldots \left[Z_3^T \frac{1}{4}\left(1+\cos(j\pi h)\right)\left(1+\cos(j\pi 2h)\right) v_2\right]\right]_i, \\ &= \left(\frac{1}{2}\right)^m \prod_{l=1}^{m}\left(1+\cos(j\pi 2^{l-1}h)\right)[v_m]_i = C_1^j [v_m^j]_i,\end{aligned}$$

for $j = 1,2,\ldots,n_m$. Similarly, for the complementary part corresponding to $j' = n_{m-1}+1-j$ we obtain

$$\left[\prod_{l=m}^{1} Z_l^T v_0^{j'}\right]_i = \left(\frac{1}{2}\right)^m \prod_{l=1}^{m}\left(\cos(j\pi 2^{l-1}h)-1\right)[v_m]_i = C_2^j [v_m^j]_i.$$

To conclude, we obtain

$$\prod_{l=m}^{1} Z_l^T v_0^j = C_1^j v_m^j, \; j = 1,2,\ldots,n_m, \tag{8.9}$$

$$\prod_{l=m}^{1} Z_l^T v_0^{j'} = C_2^j v_m^j, \; j = 1,2,\ldots,n_m. \tag{8.10}$$

where $C_1^j = \left(\frac{1}{2}\right)^m \prod_{l=1}^m \left(1+\cos(j\pi 2^{l-1}h)\right)$ and $C_2^j = \left(\frac{1}{2}\right)^m \prod_{l=1}^m \left(\cos(j\pi 2^{l-1}h)-1\right)$.

Prolongation operator

The restriction operator was defined as the transpose of $\mathcal{I}_{m+1}^m$, and thus we have that the matrix representation of the prolongation operator is given by Z_m. For the prolongation operator, we again start with $m = 1$ and take the basis V_1 as the prolongation operator works on a coarse-grid eigenvector on level m and maps it to a fine-grid counterpart on level $m-1$. We distinguish two cases; i is odd and i is even. We start with the first case

$$\begin{aligned}[Z_1 v_1^j]_i &= \frac{1}{4}\left(\sin(\frac{(i-1)2h\pi j}{2}) + \sin(\frac{(i+1)2h\pi j}{2})\right), \\ &= \frac{1}{4}\left(\sin((i-1)h\pi j) + \sin((i+1)h\pi j)\right), \\ &= \frac{1}{2}\cos(j\pi h)\sin(ih\pi j),\end{aligned} \tag{8.11}$$

8

for $j = 1,2,\ldots,n_1$. If i is even, we obtain

$$[Z_1 v_1^j]_i = \frac{1}{2}\sin(\frac{2hi\pi j}{2}) = \frac{1}{2}\sin(hi\pi j) = \frac{1}{2}[v_0^j]_i. \tag{8.12}$$

Using Eq. (8.8), if we define $j' = n_{m-1} + 1 - j$, we can write Eq. (8.12) as

$$[Z_1 v_1^j]_i = \sin(hi\pi j) = -(-1)^i \sin(j\pi hi) = [v_0^{j'}]_i, \tag{8.13}$$

if i = odd. Thus, if i is odd, combining Eq. (8.8) and Eq. (8.13), gives

$$[Z_1 v_1^j]_i = \frac{1}{2}[v_0^{j'}]_i + \frac{1}{2}\cos(j\pi h)[v_0^j]_i = C_{1,h}^j [v_0^j, v_0^{j'}]_i,$$

for $j = 1,2,\ldots,n_1$. Similarly, if i is even, we obtain

$$[Z_1 v_1^j]_i = -\frac{1}{2}[v_0^{j'}]_i + \frac{1}{2}\cos(j\pi h)[v_0^j]_i = C_{2,h}^j [v_0^j, v_0^{j'}]_i,$$

for $j = 1,2,\ldots,n_1$. Note that $[v_0^j, v_0^{j'}]_i$ is an element of $\mathcal{V}_0$ and the coarse-grid eigenvectors are mapped by the interpolation operator Z_1 according to

$$\mathcal{I}_1^0 : V_1 \overset{\mathcal{I}_1^0}{\to} \mathcal{V}_0.$$

Also note that $\mathcal{R}(Z_1) \subset V_0$, and we have $V_0 = \mathcal{N}(Z_1^T) \bigoplus \mathcal{R}(Z_1)$. We now take $m = 2$, using the basis V_2. From the above, it follows that

$$[Z_2 v_2^j]_i = \frac{1}{2}[v_1^{j'}]_i + \frac{1}{2}\cos(j\pi 2h)[v_1^j]_i = C_{1,2h}^j [v_1^j, v_1^{j'}]_i, \ i \text{ is odd} \tag{8.14}$$

$$[Z_2 v_2^j]_i = -\frac{1}{2}[v_1^{j'}]_i + \frac{1}{2}\cos(j\pi 2h)[v_1^j]_i = C_{2,2h}^j [v_1^j, v_1^{j'}]_i, \ i \text{ is even}, \tag{8.15}$$

for $j = 1,2,\ldots,n_2$ and $j' = n_1 + 1 - 1$. As the v_1^j's are the eigenvectors on level $m = 1$, we can rewrite the complementary indices j' in terms of j again by using

$$\begin{aligned} &[v_1^{j'}]_i = -(-1)^i \sin(i2hj\pi), \\ &i = 1,2,\ldots n_1, \text{and } j = 1,2,\ldots n_2. \end{aligned} \tag{8.16}$$

Substituting Eq. (8.16) into Eq. (8.14) and Eq. (8.15) gives

$$[Z_2 v_2^j]_i = \frac{1}{2}[v_1^j]_i + \frac{1}{2}\cos(j\pi 2h)[v_1^j]_i = C_{1,2h}^j [v_1^j]_i, \ i \text{ is odd} \tag{8.17}$$

$$[Z_2 v_2^j]_i = -\frac{1}{2}[v_1^j]_i + \frac{1}{2}\cos(j\pi 2h)[v_1^j]_i = C_{2,2h}^j [v_1^j]_i, \ i \text{ is even}, \tag{8.18}$$

and $\mathcal{R}(Z_2) \subset V_1$, and we have $V_1 = \mathcal{N}(Z_2^T) \bigoplus \mathcal{R}(Z_2)$. Moving from $m = 1$ to $m = 0$ by left-multiplying Eq. (8.17) and Eq. (8.18) with Z_1 is now straightforward as we get the coefficient $C_{1,h}^j$ and $C_{2,h}^j$ times $[Z_1 v_1^j]_i$ from above. This corresponds to a composition of the linear transformations where at $\mathcal{V}_1$ we reorder the basis to V_1 using Eq. (8.16)

$$\mathcal{I}_1^0 \circ \mathcal{I}_2^1 : V_2 \overset{\mathcal{I}_2^1}{\to} \mathcal{V}_1 \overset{\mathcal{I}_1^0}{\to} \mathcal{V}_0, \text{ where } \begin{array}{ccc} V_2 & \xrightarrow{\mathcal{I}_2^1} & \mathcal{V}_1 \circlearrowleft_{V_1} \\ & \searrow^{\mathcal{I}_1^0 \circ \mathcal{I}_2^1} & \downarrow{\scriptstyle \mathcal{I}_1^0} \\ & & \mathcal{V}_0 \end{array}.$$

8

From here it is easy to see that for $m > 2$ successive application gives

$$\begin{aligned}\left[\prod_{l=1}^{l} Z_l v_m\right]_i &= \left[Z_1 Z_2 \ldots \frac{1}{2}\left(1+\cos(j\pi 2^m h)\right)\left[Z_{m-1} v_{m-1}^j\right]\right]_i, \\ &= \left[Z_1 Z_2 \ldots \frac{1}{4}\left(1+\cos(j\pi 2^m h)\right)\left(1+\cos(j\pi 2^{m-1} h)\right)\left[Z_{m-2} v_{m-2}^j\right]\right]_i, \\ &= \left(\frac{1}{2}\right)^m \prod_{l=m}^{1}\left(1+\cos(j\pi 2^l h)\right)[v_0^j]_i = C_1^j [v_0^j]_i, \text{ for } i \text{ is odd.} \end{aligned} \tag{8.19}$$

Finally, if i is even we get $\left[\prod_{l=1}^{l} Z_l v_m\right]_i = \left(\frac{1}{2}\right)^m \prod_{l=m}^{1}\left(\cos(j\pi 2^l h) - 1\right)[v_0^j]_i = C_2^j [v_0^j]_i$ and $\mathscr{R}(Z_{m+1}) \subset V_m$, and we have $V_m = \mathscr{N}(Z_{m+1}^T) \bigoplus \mathscr{R}(Z_{m+1})$.

Composite mapping subspaces

Let us now take $B_m = \prod_{l=1}^{m-1} Z_l \prod_{l=m-1}^{1} Z_l^T$, and $\hat{B}_m = Z_m Z_m^T$. We furthermore let

$$\begin{aligned} {}^t f^m &: \mathscr{V}_0 \to V_m : \mathscr{I}_{m-1}^m \circ \mathscr{I}_{m-2}^{m-1} \circ \ldots \circ \mathscr{I}_0^1, \text{ and} \\ f^m &: V_m \to \mathscr{V}_0, \text{ and} \\ g^m &: \mathscr{V}_{m-1} \to \mathscr{V}_{m-1} : \mathscr{I}_m^{m-1} \circ \mathscr{I}_{m-1}^m \end{aligned}$$

where ${}^t f^m$ is the transpose of the linear map f^m. Note that g^m is a automorphism. We can define

$$h^m : \mathscr{V}_0 \to \mathscr{V}_0 : f^m \circ {}^t f^m,\ f^m \in V_m,$$

to denote the composite linear mapping along the m-vectors spaces. Here ${}^t f^m$ maps elements of $\mathscr{V}_0$ to V_m and we can write $h^m : f^{m-1} \circ (g^m \circ {}^t f^{m-1})$. This gives

$$\begin{aligned} \ker g^m &= \{v_0^{j'} \in \mathscr{V}_0, : {}^t f^{m-1} v_0^j = 0\} \subset V_{m-1}, \text{ and} \\ \operatorname{Im} g^m &= \{v_0^j \in \mathscr{V}_0 : {}^t f^{m-1} v_0^j \neq 0\} = V_{m-1}/\ker g^m \subset V_{m-1}, \end{aligned}$$

where j' are the complementary indices corresponding to $n_0 + 1 - j$. But then by definition and the fact that g^m is an automorphism, ${}^t f^{m-1} v_0^j$ must be an eigenvector of g^m. Given that we can write $V_{m-1} = \ker g^m \bigoplus \operatorname{Im} g^m$, the rank-nullity theorem furthermore tells us that $\dim(V_{m-1}) = \dim(\ker g^m) + \dim(\operatorname{Im} g^m) = n_m + n_m = n_{m-1}$. Thus, g^m must have n_m zero eigenvalues and n_m non-zero eigenvalues as the kernel of g^m is non-trivial. This leads to

$$\begin{aligned} (g^m \circ {}^t f^{m-1}) v_0^j &= g^m({}^t f^{m-1} v_0^j), \\ &= \lambda(g^m)({}^t f^{m-1} v_0^j) = \lambda(g^m) v_{m-1}^j, \end{aligned}$$

where $\lambda(g^m)$ denotes the scalar eigenvalue corresponding to g^m. Applying f^{m-1}, finally gives

$$\begin{aligned} f^{m-1} \circ (g^m \circ {}^t f^{m-1}) v_0^j &= f^{m-1}\left(g^m({}^t f^{m-1})\, v_0^j\right), \\ &= \lambda(g^m) f^{m-1}({}^t f^{m-1} v_0^j) = \lambda(g^m)\lambda(h^{m-1}) v_{m-1}^j. \end{aligned}$$

Eigendecomposition of B_m

If B_{m-1} and $\hat{B}_m$ are the matrix representations of h^{m-1} and g^m respectively, then $\dim(\ker g^m) = \dim(\mathcal{N}(\hat{B}_m)) = n_m$, and $\dim(\operatorname{Im} g^m) = \dim(\mathcal{R}(\hat{B}_m)) = n_m$, and thus $\hat{B}_m$ has only n_m non-zero eigenvalues. But then B_m must also have n_m non-zero eigenvalues as well. ■

We similarly extend the multi-level operators for the higher-order deflation vectors. This is given in Corollary 14.1 and follows naturally from the linear case.

Corollary 14.1: Multi-level Prolongation and Restriction (quadratic)

Let Z_m be the $n_{m-1} \times n_m$ prolongation matrix based on rational Bezier curves for $m = 1,2,\dots m_{\max}$, with $n_m = \frac{n}{2^m}$. If we define $v_m^j = \sin(2^m hi\pi j)$, and $v_m^{j'} = \sin(2^m hi\pi(n_m + 1 - j))$, where on the finest level we have $m = 0$. Then there exist constants C_1^j and C_2^j depending on h such that the restriction operator maps the eigenvectors to

$$\prod_{l=m}^{1} Z_l^T v_0^j = C_1^j v_m^j,\ j = 1,2,\dots,n_m, \text{ and } \prod_{l=m}^{1} Z_l^T v_0^{j'} = C_2^j v_m^j,\ j = 1,2,\dots,n_m.$$

where $C_1^j = \left(\frac{1}{2}\right)^m \prod_{l=1}^m C_{1,lh}^j$ and $C_2^j = \left(\frac{1}{2}\right)^m \prod_{l=1}^m C_{2,lh}^j$. Similarly, the prolongation operator maps the eigenvectors to

$$\prod_{l=1}^{l} Z_l[v_m]_i = C_1^j[v_0^j]_i,\ i \text{ is odd. and } \prod_{l=1}^{l} Z_l[v_m]_i = C_2^j[v_0^j]_i,\ i \text{ is even..}$$

Finally, if we let $B_m = \prod_{l=1}^m Z_l \prod_{l=m}^1 Z_l^T$ and $\hat{B}_m = Z_m Z_m^T$ for $m = 1,2,\dots,m_{\max}$, then B_m has dimension n_0 with n_m non-zero eigenvalues.

8

Proof. The proof is exactly the same as the proof of Theorem 14, however we now have

$$C_{1,mh}^j = \left(\cos(j\pi 2^m h) + \cos(j\pi 2^{m+1} h)\frac{1}{4} + \frac{3}{4}\right),$$
$$C_{2,mh}^j = \left(\cos(j\pi 2^m h) - \cos(j\pi 2^{m+1} h)\frac{1}{4} - \frac{3}{4}\right).$$

For a detailed proof of deriving $C_{1,mh}^j$ and $C_{2,mh}^j$ see [78]. The statement is obtained by substituting these coefficients into the proof of Theorem 14. ■

Using this result we can approximate the location where the near-zero eigenvalues of the coarse-grid matrices are located. This is important as we only want to apply the smoother on levels where it is needed. We start by denoting the coarse grid linear systems by E_m and we set $E_0 = A$, where A is the fine grid linear matrix. Analytical expressions for the location of the smallest eigenvalue are found in the following corollary.

Corollary 14.2: Coarse near-zero eigenvalues

Let Z_m be the $n_{m-1} \times n_m$ prolongation matrix for $m = 0, 1, 2, \dots m_{\max}$, with $n_m = \frac{n}{2^m}$. We define the symmetric coarse-grid coefficient matrix $E_m = \prod_{l=m}^{1} Z_l^T A \prod_{l=1}^{m} Z_m$. If we let $[v_m^j]_i = \sin(2^m h i \pi j)$ be the eigenvectors of E_m, where for $m = 0$ we have the finest level, then $\exists \tilde{m}$: for $m > \tilde{m}$ E_m is negative definite. For $m \leqslant \tilde{m}$ E_m is indefinite.

Proof. Let $\Lambda(A)$ denotes the $n_0 \times n_0$ diagonal matrix containing the eigenvalues of A, then using Theorem 14 for each i, either odd or even, we have

$$\lim_{h\to 0} \left| E_m [v_m^j]_i \right| \leqslant \lim_{h\to 0} \left| \prod_{l=m}^{1} Z_l^T \Lambda(A) \prod_{l=1}^{m} Z_l [v_m^j]_i \right| \leqslant \lim_{h\to 0} \left| \lambda_A^j (C_1^j)^2 [v_m^j]_i \right| \leqslant 4^m \left| \lambda_A^j [v_m^j]_i \right|,$$

where we used that by definition of C_1^j and C_2^j, for all j we have $\left| C_1^j C_2^j \right| \leqslant \left| (C_1^j)^2 \right| \leqslant 4^m$. Note that in case of i is even, we would have $C_1^j C_2^j$ instead of $C_1^{j^2}$. Thus, in the limit as h goes to zero, we can bound the expression for $\lambda_{E_m}^j$ from above by $\left| \lambda_{E_m}^j \right| \leqslant 4^m \lambda_A^j$ for each j. Now to find a bound for the smallest eigenvalue in magnitude of E_m, we need to minimize the right-hand side of the upper-inequality over all indices j. This is achieved at $j = j_{\min}$, corresponding to the smallest eigenvalue in magnitude of A as this eigenvalue is the closest eigenvalue to zero. We thus have $\left| \lambda_{E_m}^{j_{\min}} \right| \leqslant 4^m \lambda_A^{j_{\min}}$. We now need to find the level m at which the matrix E_m becomes negative definite. Recall that

$$j_{\min} = \left\lfloor \frac{\cos^{-1}\left(\frac{1-k^2h^2}{2}\right)}{\pi h} \right\rfloor = \left\lfloor \frac{n \cos^{-1}\left(\frac{1-k^2h^2}{2}\right)}{\pi} \right\rfloor.$$

8

Therefore, to find the level $\tilde{m}$ which still contains index $j_{\min}$, for $j = 1, 2, \dots n_m$, we have to find $m : n_m = \frac{n}{2^m} > j_{\min}$. Note $j_{\min}$ is unaffected by h as h goes to zero and thus we can assess how many times $j_{\min}$ fits into n. Additionally, coarsening leads to the problem size being halved for each m, and thus need to divide by 2 as well.

$$\left\lfloor \frac{n}{2 j_{\min}} \right\rfloor = \left\lfloor \frac{\cos^{-1}\left(\frac{1-k^2h^2}{2}\right)}{2\pi} \right\rfloor = \tilde{m}.$$

Consequently, for $m > \tilde{m}$, $j_{\min}$ is no longer within the range of n_m. Therefore, all eigenvalues of $E_{m>\tilde{m}}$ for $j = 1, 2, \dots n_{m>\tilde{m}} \leqslant j_{\min}$ must have the same sign, due to the fact that $\lambda_A^{j_{\min}}$ is an upper bound and the only eigenvalue of A where a sign-change can occur. ∎

Corollary 14.2 shows that for $m \leqslant \tilde{m}$, the resulting coarse-grid coefficient matrices E_m are indefinite. Thus, on these subsequent levels, it is important that the near-zero eigenvalues are reduced and aligned in coherence with the fine-grid level. In order to analytically assert this, we proceed by defining the multi-level deflation operator and block-diagonalizing it using a similar basis as we used for the two-level ADP scheme. This will allow us to perform spectral analysis of the multi-level deflation operator as the latter reduces to applying the two-level ADP scheme recursively.

8.5.2. BLOCK-DIAGONAL SYSTEMS

Using the matrices Z_m and Z_m^T to denote the prolongation and restriction operator on level m, and using the theory developed so far, we can construct similar analytical expressions for the eigenvalues of the preconditioner applied to the coefficient matrix. We perform the analysis for MP 1-A. Taking $E_0 = A$, we define the $n \times n$ projection operator $P_{h,m}$ to be

$$P_{h,m} = I - AQ_m, \text{ where } Q_m = \prod_{l=1}^{m} Z_l E_m^{-1} \prod_{l=m}^{1} Z_l^T \text{ and } E_m = Z_m^T E_{m-1} Z_m, \tag{8.20}$$

$$P_m = I_m - E_m Q_m, \text{ where } Q_m = Z_m E_m^{-1} Z_m^T \text{ and } E_m = Z_m^T E_{m-1} Z_m \tag{8.21}$$

Note that this is equivalent to constructing P by solving E_m directly on the m-th level and then prolonging the inverse back to the fine grid in order to proxy the effect of having an approximate inversion of E_1 in the two-level method. We will refer to $P_{h,m}$ as the **global** multi-level deflation preconditioner and P_m as the **local** level deflation preconditioner.

GLOBAL SYSTEM BLOCK-DIAGONALIZATION

In order to extend the spectral analysis of the two-level ADP-scheme to a multi-level setting, we use the bases and operators defined in the first part of the proof of Theorem 14. To assist the reading of the proofs below, we briefly recall the basis and its reordering. For $n_m = \frac{n}{2^m}$, we rearranged the space spanned by the eigenvectors at each level m such that we obtain the following subspace

$$\mathcal{V}_m^j = \text{span}\{v_m^j, v_m^{n_m+1-j}\} \text{ and } V_{m+1}^j = \text{span}\{v_{m+1}^j\}$$

for $j = 1, 2, \ldots, n_{m+1}$. Note that the subspace $\mathcal{V}_m^j$ consists of two vectors and the subspace $\mathcal{V}_{m+1}^j$ consists of one vector. We furthermore have that both bases $\mathbb{C}^{n_m}$ and $\mathbb{C}^{n_{m+1}}$ respectively as we can write

$$\mathbb{C}^{n_m} = \bigoplus_{j=1}^{n_{m+1}} \mathcal{V}_m^j \text{ and } \mathbb{C}^{n_{m+1}} = \bigoplus_{j=1}^{n_{m+1}} V_{m+1}^j,$$

and at each subsequent level $m+1$ we can always define an automorphism to re-order the basis V_m to obtain $\mathcal{V}_{m+1}$.

We start with Lemma 14.1, which will provide the building blocks to block-diagonalize Q_m by first block-diagonalizing B_m. This is equivalent to using the operators and expressions from Theorem 14 and writing them in matrix form using 2×2 blocks by evaluating their action on the underlying basis of eigenvectors.

Lemma 14.1: Block-diagonalization B_m

Let Z_m be the $n_{m-1} \times n_m$ interpolation matrix with $n_m = \frac{n}{2^m}$ for $m = 0, 1, 2, \ldots, m_{\max}$. Let $B_m = \prod_{l=m}^{1} Z_l \prod_{l=1}^{m} Z_l^T$ and $\hat{B}_m = Z_m Z_m^T$ for $m = 1, 2, \ldots, m_{\max}$. Defining the rearranged basis

$$\mathcal{V}_m = \bigoplus_{j=1}^{n_{m+1}} \operatorname{span}\{v_m^j, v_m^{n_{m+1}+1-j}\},$$

where $v_m^j = [\sin(j\pi hi2^m)]_{i=1}^{n_m}$, the eigenvalues of B_m are given by

$$\lambda_{B_m}^j = \left(\frac{1}{2}\right)^m \prod_{l=m}^{1} \left((r_l^j)^2 + (p_l^j)^2\right),$$

where $r_l^j = C_{1,mh}^j$ and $p_l^j = C_{1,lh}^j$. Here $C_{2,lh}^j$ and $C_{2,mh}^j$ are either the linear or quadratic coefficients.

Proof. We continue by using the results from Theorem 14. To keep the notation compact we let $r_m^j = C_{1,mh}^j$ and $p_m^j = C_{2,mh}^j$. We start with the case where $m = 1$. Using the basis $\mathcal{V}_0, V_1$, Z_1 and Z_1^T have the block form

$$[Z_1]_{V_1}^j = \begin{bmatrix} r_1^j \\ p_1^j \end{bmatrix}, \tag{8.22}$$

$$[Z_1^T]_{\mathcal{V}_0}^j = \begin{bmatrix} r_1^j & p_1^j \end{bmatrix}, \tag{8.23}$$

for $j = 1, 2, \ldots, n_1$. In block-diagonal form on we can write Z_1 as

$$\begin{bmatrix} \boxed{\begin{matrix} r_1^1 \\ p_1^1 \end{matrix}} & & & \mathbf{0} \\ & \boxed{\begin{matrix} r_1^2 \\ p_1^2 \end{matrix}} & & \\ & & \ddots & \\ \mathbf{0} & & & \boxed{\begin{matrix} r_1^{n_1} \\ p_1^{n_1} \end{matrix}} \end{bmatrix}$$

To block-diagonalize $\hat{B}_1$, we therefore multiply the respective blocks for each j

$$[Z_1[Z_1^T]_{\mathcal{V}_0}^j]_{V_1}^j = \begin{bmatrix} r_1^j \\ p_1^j \end{bmatrix} \begin{bmatrix} r_1^j & p_1^j \end{bmatrix} = \begin{bmatrix} (r_1^j)^2 & (r_1^j p_1^j) \\ (r_1^j p_1^j) & (p_1^j)^2 \end{bmatrix}.$$

Now, $\hat{B}_1$ has n_1 non-zero eigenvalues given by the trace of each respective block and n_1 zero eigenvalues, which was also discussed in the proof of Theorem 14. The non-zero eigenvalues are thus given by the 1×1 block $\lambda_{\hat{B}_1}^j = (r_1^j)^2 + (p_1^j)^2$ for $j = 1, 2, \ldots, n_1$ and $\hat{B}_1 = B_1$ has

the block-diagonal form

$$[B_1]_{\mathcal{Y}_0} = \left[\begin{array}{ccc|c} \boxed{\lambda^1_{\hat{B}_1}} & & & \\ & \ddots & & \mathbf{0} \\ & & \boxed{\lambda^{n_1}_{\hat{B}_1}} & \\ \hline & \mathbf{0} & & \begin{array}{ccc} 0 & & \\ & \ddots & \\ & & 0 \end{array} \end{array}\right].$$

We now take $m = 2$ and block-diagonalize $\hat{B}_2$. Using the same steps as above we have

$$[Z_2 Z_2^T]^j_{\mathcal{Y}_1} = \begin{bmatrix} r_2^j \\ p_2^j \end{bmatrix} \begin{bmatrix} r_2^j & p_2^j \end{bmatrix} = \begin{bmatrix} (r_2^j)^2 & (r_2^j p_2^j) \\ (r_2^j p_2^j) & (p_2^j)^2 \end{bmatrix},$$

for $j = 1, 2, \ldots, n_2$. Computing the trace of each block gives $\lambda^j_{\hat{B}_2} = (r_2^j)^2 + (p_2^j)^2$ with block-diagonal form

$$[\Lambda(\hat{B}_2)]_{\mathcal{Y}_1} = \left[\begin{array}{ccc|c} \boxed{\lambda^1_{\hat{B}_2}} & & & \\ & \ddots & & \mathbf{0} \\ & & \boxed{\lambda^{n_2}_{\hat{B}_2}} & \\ \hline & \mathbf{0} & & \begin{array}{ccc} 0 & & \\ & \ddots & \\ & & 0 \end{array} \end{array}\right]. \tag{8.24}$$

Note that we have $n_2 = \frac{n}{4}$ zero and non-zero eigenvalues and the dimension of $\hat{B}_2$ is $n_1 \times n_1$. This is equivalent to having n_2 blocks of dimension 1×1 containing the non-zero eigenvalues and n_2 blocks, also with dimension 1×1 containing the zero eigenvalues. We now apply Z_1 to the left and Z_1^T to the right of Eq. (8.24), where we use the block-diagonal form of Z_1 and Z_1^T given by Eq. (8.22) and Eq. (8.23) respectively. Z_1 has n_1 blocks of dimension 2×1 and Z_1^T has n_1 blocks of dimension 1×2. Thus, Z_1 works on each non-zero 1×1 block of $\hat{B}_2$, and then Z_1^T is applied to the resulting 2×1 block. However, only the first n_2 blocks of $\Lambda(\hat{B}_2)$ contain non-zero terms as we can see from Eq. (8.24) and thus only the indices $j = 1, 2, \ldots n_2$ in Z_1 and Z_1^T lead to non-zero terms. Thus, for $j = 1, 2, \ldots, n_2$ we obtain $[\Lambda(B_2)]_{\mathcal{Y}_0} = [\Lambda(Z_1 \hat{B}_2 Z_1^T)]_{\mathcal{Y}_0}$, which is given by the following matrix representation

$$\left[\begin{array}{ccc} \boxed{\begin{matrix} r_1^1 \\ p_1^1 \end{matrix}} & & \mathbf{0} \\ & \boxed{\begin{matrix} r_1^2 \\ p_1^2 \end{matrix}} & \\ & \ddots & \\ \mathbf{0} & & \boxed{\begin{matrix} r_1^{n_1} \\ p_1^{n_1} \end{matrix}} \end{array}\right] \left[\begin{array}{ccc|c} \boxed{\lambda^1_{\hat{B}_2}} & & & \\ & \ddots & & \mathbf{0} \\ & & \boxed{\lambda^{n_2}_{\hat{B}_2}} & \\ \hline & \mathbf{0} & & \begin{array}{ccc} 0 & & \\ & \ddots & \\ & & 0 \end{array} \end{array}\right] \left[\begin{array}{ccc} \boxed{\begin{matrix} r_1^1 & p_1^1 \end{matrix}} & & \mathbf{0} \\ & \boxed{\begin{matrix} r_1^2 & p_1^2 \end{matrix}} & \\ & \ddots & \\ \mathbf{0} & & \boxed{\begin{matrix} r_1^{n_1} & p_1^{n_1} \end{matrix}} \end{array}\right]$$

Thus, at the level of each respective j-th block we have

$$[\Lambda(B_2)]^j_{\mathcal{V}_0} = \begin{bmatrix} r_1^j \\ p_1^j \end{bmatrix} \lambda^j_{\hat{B}_2} \begin{bmatrix} r_1^j & p_1^j \end{bmatrix} = \lambda^j_{\hat{B}_2} \begin{bmatrix} (r_1^j)^2 & (r_1^j p_1^j) \\ (r_1^j p_1^j) & (p_2^j)^2 \end{bmatrix},$$

for $j = 1,2,\ldots,n_2$. Computing the trace of each respective block gives

$$\lambda^j_{B_2} = \left((r_1^j)^2 + (p_1^j)^2\right)(\lambda^j_{\hat{B}_2}) = \left((r_1^j)^2 + (p_1^j)^2\right)\left((r_2^j)^2 + (p_2^j)^2\right). \tag{8.25}$$

Thus, we obtain the following block-diagonal form

$$[B_2]_{\mathcal{V}_0} = \left[\begin{array}{ccc|cc} \boxed{\lambda^1_{B_2}} & & & & \\ & \ddots & & \mathbf{0} & \\ & & \boxed{\lambda^{n_2}_{B_2}} & & \\ \hline & & & 0 & \\ & \mathbf{0} & & \ddots & \\ & & & & 0 \end{array}\right],$$

where $\lambda^j_{B_2}$ is given by Eq. (8.25). From here it is easy to see that successive application of Z_m and Z_m^T for $m > 2$ gives

$$[\Lambda(B_m)]^j_{\mathcal{V}_0} = \begin{bmatrix} \prod_{l=m-1}^1 r_l^j \\ \prod_{l=m-1}^1 p_l^j \end{bmatrix} \lambda^j_{\hat{B}_m} \begin{bmatrix} \prod_{l=m-1}^1 r_l^j & \prod_{l=m-1}^1 p_l^j \end{bmatrix},$$

for $j = 1,2,\ldots,n_m$ with $\lambda^j_{B_m} = \prod_{l=m}^1 \left((r_l^j)^2 + (p_l^j)^2\right)$. ■

8

Using the results from Lemma 14.1, we can block-diagonalize the operator Q_m, where m again denotes the level.

Theorem 15: Block-diagonalization Q_m

Let Z_m be the $n_{m-1} \times n_m$ interpolation matrix with $n_m = \frac{n}{2^m}$ for $m = 0,1,2,\ldots,m_{\max}$. We define the coarse linear system $E_m = Z_m^T E_{m-1} Z_m$ with $E_0 = A$. Let $B_m = \prod_{l=m}^1 Z_l \prod_{l=1}^m Z_l^T$ and $\hat{B}_m = Z_m Z_m^T$ for $m = 1,2,\ldots,m_{\max}$. Then using the basis $\mathcal{V}_0$ from Lemma 14.1, the eigenvalues of Q_m are given by

$$[\Lambda(Q_m)]^j_{\mathcal{V}_0} = [\Lambda(\prod_{l=1}^m Z_l E_m^{-1} \prod_{l=m}^1 Z_l^T)]^j_{\mathcal{V}_0} = \lambda^{-1}_{E_m{}^j}[\Lambda(B_m)]^j_{\mathcal{V}_0} = \lambda^{-1}_{E_m{}^j} \prod_{l=m}^1 \left((r_l^j)^2 + (p_l^j)^2\right),$$

with $\lambda^j_{E_m} = (r_m^j)^2 \lambda^j_{E_{m-1}} + (p_m^j)^2 \lambda^{j'}_{E_{m-1}}$ for $j = 1,2,\ldots,n_m$ and $j' = n_{m-1} + 1 - j$.

Proof. The proof is very similar to the one for Lemma 14.1. We again start with a brief outline of the proof. We start by block-diagonalizing the fine grid linear system A. Consequently, we recursively multiply the block-diagonal version of A with the matrix containing

the 2×2 blocks representing $Z_1, Z_2 \dots Z_m$ and ${Z_1}^T, {Z_2}^T \dots {Z_m}^T$ respectively to obtain E_m. Finally, we rewrite Q_m in terms of B_m, and use Lemma 14.1 to obtain the final analytical expressions.

On the basis $\mathcal{V}_0$ defined with respect to the finest level $m = 0$, we can block-diagonalize the coefficient matrix A in terms of a total of n_1 blocks with size 2×2. If we define the complementary index $j' = n_m + 1 - j = n_0 + 1 - j$, then each j-th respective block has the form

$$[\Lambda(A)]^j_{\mathcal{V}_0} = \begin{bmatrix} \lambda^j_A & 0 \\ 0 & \lambda^{j'}_A \end{bmatrix},$$

for $j = 1, 2, \dots, n_1$. Moving to $m = 1$, we now start using $\mathcal{V}_1$ as E_1 resides in the coarse-space. After applying Z_1^T and Z_1, we obtain, for $j = 1, 2, \dots n_1$, the 1×1 block

$$[\Lambda(E_1)]^j_{\mathcal{V}_1} = \left[Z_1^T A_0 Z_1\right]^j_{\mathcal{V}_1} = \begin{bmatrix} r_1^j & p_1^j \end{bmatrix} \begin{bmatrix} \lambda^j_A & 0 \\ 0 & \lambda^{j'}_A \end{bmatrix} \begin{bmatrix} r_1^j \\ p_1^j \end{bmatrix} = \left(r_1^j\right)^2 \lambda^j_A + \left(p_1^j\right)^2 \lambda^{j'}_A.$$

Thus, if we define $\lambda^j_{E_1} = \left(r_1^j\right)^2 \lambda^j_A + \left(p_1^j\right)^2 \lambda^{j'}_A$ for $j = 1, 2, \dots, n_1$, then E_1 has block-diagonal form.

$$[\Lambda(E_1)]_{V_1} = \begin{bmatrix} \boxed{\lambda^1_{E_1}} & & & \mathbf{0} \\ & \boxed{\lambda^2_{E_1}} & & \\ & & \ddots & \\ \mathbf{0} & & & \boxed{\lambda^{n_1}_{E_1}} \end{bmatrix}.$$

Note that E_1 has no zero eigenvalues and dimension $n_1 \times n_1$. Consequently, we have a total of n_1 blocks with size 1×1 corresponding to each index j at level $m = 1$. To apply Z_2^T and Z_2 to E_1, we now need the 2×2 blocks. We apply the permutation matrix corresponding to α_π with respect to V_1 such that we get the ordered basis $\mathcal{V}_1$. On this basis the block-diagonal form of E_1 is form

$$[\Lambda(E_1)]_{\mathcal{V}_1} = \begin{bmatrix} \boxed{\begin{matrix} \lambda^1_{E_1} & 0 \\ 0 & \lambda^{1'}_{E_1} \end{matrix}} & & & \mathbf{0} \\ & \boxed{\begin{matrix} \lambda^2_{E_1} & 0 \\ 0 & \lambda^{2'}_{E_1} \end{matrix}} & & \\ & & \ddots & \\ \mathbf{0} & & & \boxed{\begin{matrix} \lambda^{n_2}_{E_1} & 0 \\ 0 & \lambda^{n'_2}_{E_1} \end{matrix}} \end{bmatrix}.$$

for $j = 1,2,\dots n_2$. Now, applying the block-diagonal form of Z_2^T and Z_2 to $[\Lambda(E_1)]_{\mathcal{V}_1}$ gives

$$\begin{bmatrix} \boxed{\begin{matrix} r_2^1 & p_2^1 \end{matrix}} & & & \mathbf{0} \\ & \boxed{\begin{matrix} r_2^2 & p_2^2 \end{matrix}} & & \\ & & \ddots & \\ \mathbf{0} & & & \boxed{\begin{matrix} r_2^{n_2} & p_2^{n_2} \end{matrix}} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} \lambda_{E_1}^1 & 0 \\ 0 & \lambda_{E_1}^{1'} \end{matrix}} & & & \mathbf{0} \\ & \ddots & & \\ & & \boxed{\begin{matrix} \lambda_{E_1}^{n_2} & 0 \\ 0 & \lambda_{E_1}^{n_2'} \end{matrix}} & \\ \mathbf{0} & & & \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} r_2^1 \\ p_2^1 \end{matrix}} & & & \mathbf{0} \\ & \boxed{\begin{matrix} r_2^2 \\ p_2^2 \end{matrix}} & & \\ & & \ddots & \\ \mathbf{0} & & & \boxed{\begin{matrix} r_2^{n_2} \\ p_2^{n_2} \end{matrix}} \end{bmatrix}.$$

Note that $[\Lambda(E_1)]_{\mathcal{V}_1}$ has size $(n_1 \times n_1)$ and Z_2^T has size $(n_2 \times n_1)$. Thus, for $j = 1,2,\dots,n_2$ and $j' = n_1 + 1 - j$, each respective j-th block leads to the (1×1) block containing

$$[\Lambda(E_2)]_{\mathcal{V}_1}^j = \begin{bmatrix} r_2^j & p_2^j \end{bmatrix} \begin{bmatrix} \lambda_{E_1}^j & 0 \\ 0 & \lambda_{E_1}^{j'} \end{bmatrix} \begin{bmatrix} r_2^j \\ p_2^j \end{bmatrix} = (r_2^j)^2 \lambda_{E_1}^j + (p_2^j)^2 \lambda_{E_1}^{j'}.$$

From here it is easy to see that for $m > 2$, application of Z_m^T and Z_m recursively gives a $j-$th (1×1) block with $\lambda_{E_m}^j = (r_m^j)^2 \lambda_{E_{m-1}}^j + (p_m^j)^2 \lambda_{E_{m-1}}^{j'}$ for $j = 1,2,\dots,n_m$ and $j' = n_{m-1}+1-j$, where each j-th block has the form

$$[\Lambda(E_m)]_{\mathcal{V}_m}^j = \begin{bmatrix} \lambda_{E_m}^j & 0 \\ 0 & \lambda_{E_m}^{j'} \end{bmatrix}.$$

We now combine Lemma 14.1 and the previous expression for the eigenvalues of E_m to block-diagonalize Q_m. We can now use the result from Lemma 14.1. This gives

$$[\Lambda(Q_m)]_{\mathcal{V}_0}^j = [\Lambda(\prod_{l=1}^{m} Z_l E_m^{-1} \prod_{l=m}^{1} Z_l^T)]_{\mathcal{V}_0}^j = \lambda_{E_m{}^j}^{-1} [\Lambda(B_m)]_{\mathcal{V}_0}^j = \lambda_{E_m{}^j}^{-1} \prod_{l=m}^{1} \left((r_l^j)^2 + (p_l^j)^2 \right),$$

for $j = 1,2,\dots,n_m$.

8

■

We can now easily block-diagonalize P_m as follows. We start by writing P_m in block-diagonal form using Theorem 15 and our rearranged basis ${\mathcal{V}_0}^j$.

$$[\Lambda(P_m)]_{V_0}^j = [I - AQ_m]_{\mathcal{V}_0}^j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{\mathcal{V}_0}^j - \frac{\lambda_{B_m}^j}{\lambda_{E_m}^j} \begin{bmatrix} \lambda_A^j & 0 \\ 0 & \lambda_A^{j'} \end{bmatrix}_{\mathcal{V}_0}^j = \begin{bmatrix} 1 - \frac{\lambda_A^j \lambda_{B_m}^j}{\lambda_{E_m}^j} & \frac{\lambda_A^j \lambda_{B_m}^j}{\lambda_{E_m}^j} \\ \frac{\lambda_A^{j'} \lambda_{B_m}^j}{\lambda_{E_m}^j} & 1 - \frac{\lambda_A^{j'} \lambda_{B_m}^j}{\lambda_{E_m}^j} \end{bmatrix}_{\mathcal{V}_0}^j$$

Including the CSL preconditioner M^{-1} and applying the multi-level deflation preconditioner P_m to the coefficient matrix A finally gives the block-diagonal expressions of the preconditioned system

$$[\Lambda(P_m M^{-1} A)]_{\mathcal{V}_0}^j = \frac{\lambda_A^j}{\lambda_M^j} \begin{bmatrix} 1 - \frac{\lambda_A^j \lambda_{B_m}^j}{\lambda_{E_m}^j} & \frac{\lambda_A^j \lambda_{B_m}^j}{\lambda_{E_m}^j} \\ \frac{\lambda_A^{j'} \lambda_{B_m}^j}{\lambda_{E_m}^j} & 1 - \frac{\lambda_A^{j'} \lambda_{B_m}^j}{\lambda_{E_m}^j} \end{bmatrix}_{\mathcal{V}_0}^j.$$

At last, we obtain the eigenvalues of $P_m M^{-1} A$ for $j = 1,2,\ldots,n_1$ and $j' = n_0 + 1 - j$, by computing the trace of each respective block

$$\lambda^j(P_m M^{-1} A) = \frac{\lambda_A^j}{\lambda_M^j}\left(1 - \frac{\lambda_A^j \lambda_{B_m}^j}{\lambda_{E_m}^j}\right) + \frac{\lambda_A^{j'}}{\lambda_M^j}\left(1 - \frac{\lambda_A^{j'} \lambda_{B_m}^j}{\lambda_{E_m}^j}\right), \tag{8.26}$$

with $\lambda_{B_m}^j = \prod_{l=m}^{1}\left((r_l^j)^2 + (p_l^j)^2\right)$.

8.5.3. SPECTRAL ANALYSIS

Using these expressions, we proceed by analyzing the various operators involved in the multi-level deflation operator. For the purpose of this section, we choose the shift in the CSL preconditioner to be large ($\beta_2 = 1$) in order to emphasize the effect of the deflation method. In this section, we plot the expressions from Eq. (8.20) and Eq. (8.21), which are the global and local multi-level deflation preconditioner respectively. The global operator is obtained by inverting E_m at level m exactly and prolongating back to the fine grid until we obtain $P_{h,m}$. The local operator is obtained by applying two-level deflation locally at level m, which gives us P_m.

GLOBAL NEAR-ZERO EIGENVALUES

We start with by denoting the global deflation operator by $P_{h,m}$, where m indicates the level. We analyze the spectrum up to the level where the coefficient matrix becomes negative definite, which according to Corollary 14.2 is at $\tilde{m} = 3$. Before we start with the spectral analysis, several remarks are in place. The eigenvalues of the preconditioned systems can be retrieved analytically in case we have Dirichlet boundary conditions. In case of Sommerfeld boundary conditions, the analytical eigenvalues can not be determined and we are forced to compute them numerically. For the sake of completeness, we include them in the spectral analysis for the one-dimensional model problems.

For $k = 1000$ we define $P_{h,1}, P_{h,2}$ according to Eq. (8.20). We use 10 grid points per wavelength (gpw) for this part of the analysis. Fig. 2 and Fig. 3 contain the results using linear interpolation for both Dirichlet and Sommerfeld boundary conditions respectively. Similarly, Fig. 4 and Fig. 5 contain the results using high-order deflation vectors.

When we use Dirichlet boundary conditions and compare Fig. 2 to Fig. 4, we immediately observe that there are less near-zero eigenvalues on the first and second level when using higher-order deflation vectors. Especially for the first level (blue, moving from n to $\frac{n}{2}$), the difference seems to be significant.

Using Sommerfeld conditions, the conditions for Fig. 3 and Fig. 5 the conclusion is similar. The use of these boundary conditions for the linear interpolation case seems to be more prevalent at the first level (blue). Here, Fig. 3 shows a slightly different angle away from the zero, compared to Fig. 2. At the second level, there appears to be no difference. If we move to higher-order deflation vectors in Fig. 4 and Fig. 5 for both the Dirichlet and Sommerfeld case, the eigenvalues at the first level remain clustered near the point $(1,0)$ in the complex plane. The eigenvalues start dispersing once we move to the second level (red) (from $\frac{n}{2}$ to $\frac{n}{4}$). An important distinction is visible for the higher-order case. Using Sommerfeld conditions in Fig. 5 keeps the eigenvalues of $P_{h,2}$ away from zero relative to Fig. 4.

Note that for $m \geqslant 3$, we have proved that the resulting coarse-grid coefficient matrix E_3 is completely negative definite. Consequently, the problem of the near-zero eigenvalues of $E_{m\geqslant 3}$ resolves itself at these levels given that the location of the smallest eigenvalue in terms of magnitude is now fixed away from zero due to the matrix being negative-definite. Moreover, the further down the levels we move, the smaller the number of eigenvalues become which get projected away.

Spectrum of the global deflation + CSLP preconditioned system using $kh = 0.625$ or equivalently 10 gpw. Blue uses a two-level scheme and red uses a three-level scheme.

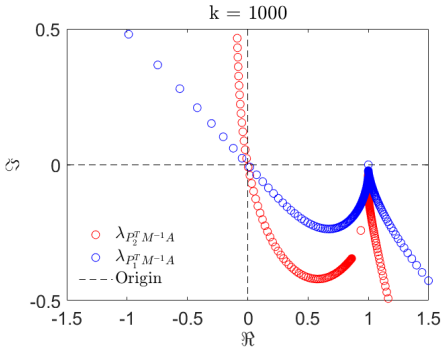


Figure 8.2: Linear interpolation (Dirichlet).

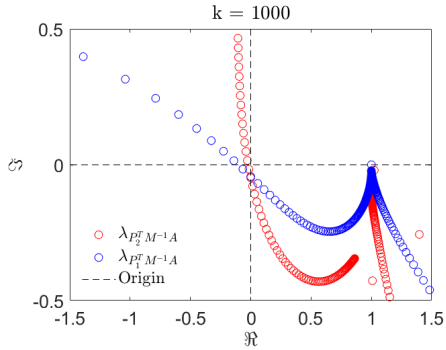


Figure 8.3: Linear interpolation (Sommerf.).

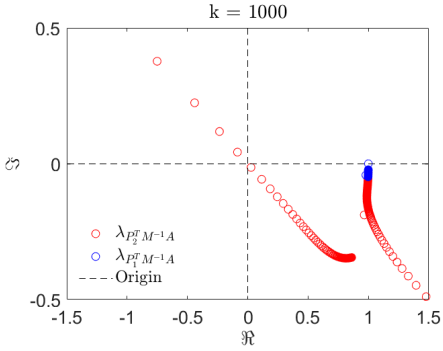


Figure 8.4: Quadratic rational Bezier (Dirichlet).

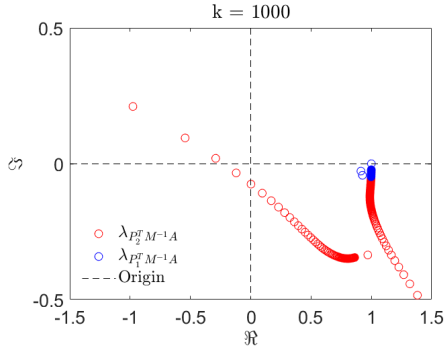


Figure 8.5: Quadratic rational Bezier (Sommerf.).

Next, we repeat the analysis for $k = 1000$, but this time we use 20 gpw. We define $P_{h,1}, P_{h,2}$ and $P_{h,3}$ according to Eq. (8.20). When we use Dirichlet boundary conditions, comparing Fig. 6 and Fig. 8 immediately shows that there are more near-zero eigenvalues when using linear interpolation. Overall, for the first and second level, the spectrum remains tightly clustered when using higher-order deflation. Thus, for the linear interpolation case in Fig. 6, the first level (black) appears to benefit the most from using a finer grid.

Moving on to the Sommerfeld boundary conditions, comparing Fig. 7 and Fig. 9 shows a large difference in the clustering of the eigenvalues at the first and second level (black and blue). Using a finer grid seems to affect the first and second level, i.e. the spectrum of $P_{h,1}$ and $P_{h,2}$ of the higher-order case more. If we compare Fig. 7 and Fig. 3 we only observe a

significant difference at the first level. In all cases it shows that the largest clustering gain can be achieved at the levels where the matrix remains highly indefinite.

Spectrum of the global deflation + CSLP preconditioned system using $kh = 0.3125$ or equivalently 20 gpw. Black uses a two-level scheme, blue uses a three-level scheme and red uses a four-level scheme.

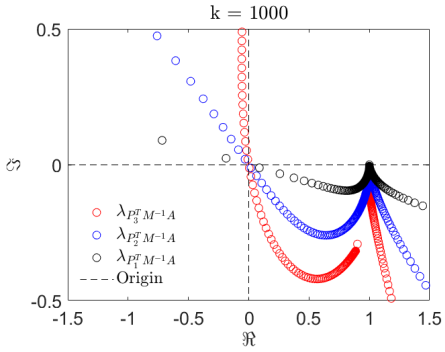


Figure 8.6: Linear interpolation (Dirichlet).

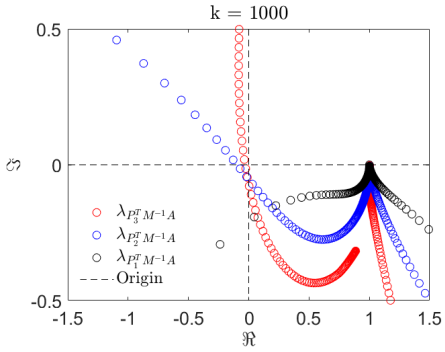


Figure 8.7: Linear interpolation (Sommerf.).

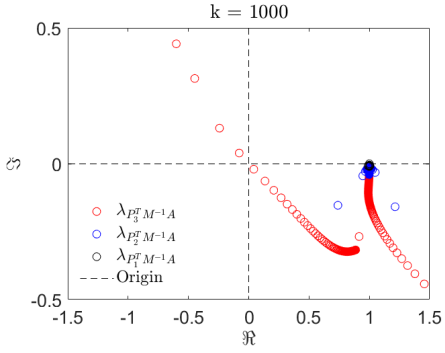


Figure 8.8: Quadratic rational Bezier (Dirichlet).

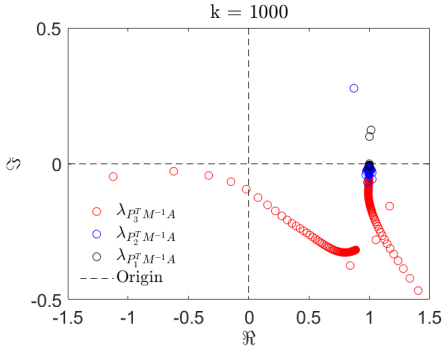


Figure 8.9: Quadratic rational Bezier (Sommerf.).

LOCAL DEFLATED NEAR-ZERO EIGENVALUES

Here we start by plotting the local near-zero eigenvalues for $k = 1000$ of P_2 and P_3 and compare them to $P_{h,2}$ and $P_{h,3}$ respectively. For this part of the analysis, we only use the case with Dirichlet boundary conditions. So far we have observed that the inclusion of Dirichlet boundary conditions leads to a spectrum which appears to be less favourably clustered compared to when we include Sommerfeld boundary conditions.

Starting with 10 gpw, for all cases irrespective of linear interpolation or higher-order deflation vectors, the eigenvalues of the local and global operator are similar. If we use a higher-order scheme the largest gain in terms of removing the near-zero eigenvalues is realized at level $m \leqslant 2$. At these levels, comparing Fig. 10 and Fig. 12, we observe that we have less near-zero eigenvalues both globally and locally. As soon as the matrix becomes negative definite, the spectrum is fully determined by the spectrum of CSLP applied to the global

and/or local coefficient matrix. Comparing Fig. 11 and Fig. 12 shows no difference irrespective of the underlying basis functions used to construct the deflation vectors.

Spectrum of global and local deflation + CSLP preconditioned system using $kh = 0.625$ or equivalently 10 gpw. Red represents the global spectrum and blue represents the local spectrum.

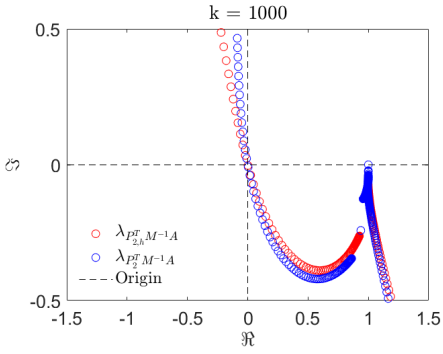


Figure 8.10: Linear interpolation ($m = 2$).

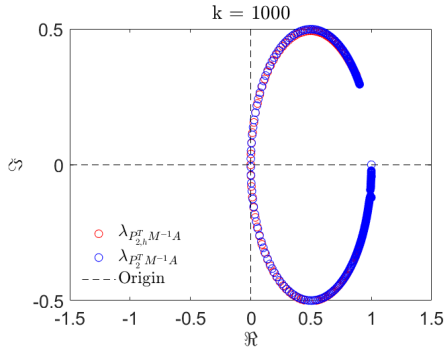


Figure 8.11: Linear interpolation ($m = 3$).

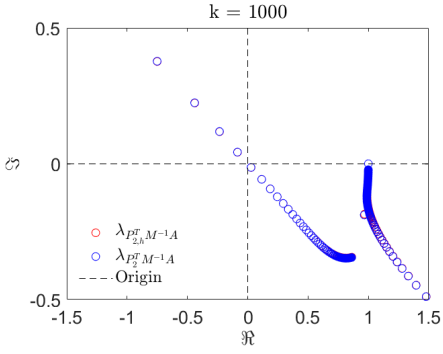


Figure 8.12: Quadratic rational Bezier ($m = 2$).

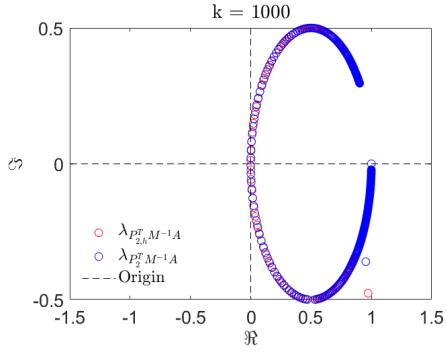


Figure 8.13: Quadratic rational Bezier ($m = 3$).

We repeat the analysis for $k = 1000$, again using 20 gpw. For both linear interpolation and quadratic rational Bezier, the global and local preconditioned spectra appear similar. We again have plotted $m = 2$ in Fig 14. and Fig. 16 and $m = 3$ in Fig. 15 and Fig. 17. Note that when using 20 gpw the resulting underlying coarse linear system does not become negative definite at $m = 3$. Instead, the linear systems become negative definite at $m \geqslant 4$. As for the levels discussed here, we clearly observe a significant difference in clustering at both levels, when we use higher-order deflation vectors.

At the coarsest level where the matrix is still indefinite, in this case $m = 3$, we observe in Fig. 17 that the spectrum is slowly starting to disperse for the higher-order scheme. In terms of magnitude, it is easy to see that the near-zero eigenvalues in Fig. 15 are smaller. This observation supports the notion that the largest effect of using a deflation strategy with higher-order basis function can be realized when the matrices at the finer level are highly indefinite. These are also the linear systems which are the largest in terms of the problem size.

Spectrum of global and local deflation + CSLP preconditioned system using $kh = 0.3125$ or equivalently 20 gpw. Red represents the global spectrum and blue represents the local spectrum.

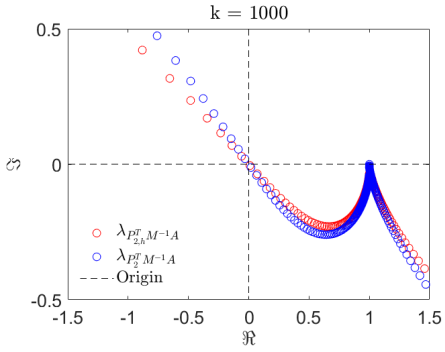


Figure 8.14: Linear interpolation ($m = 2$).

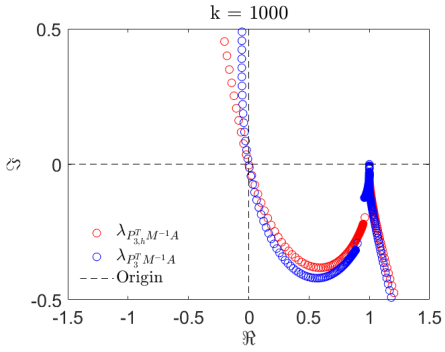


Figure 8.15: Linear interpolation ($m = 3$).

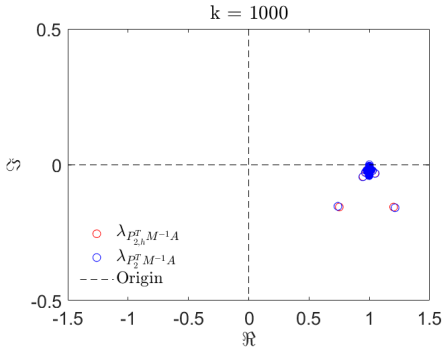


Figure 8.16: Quadratic rational Bezier ($m = 2$).

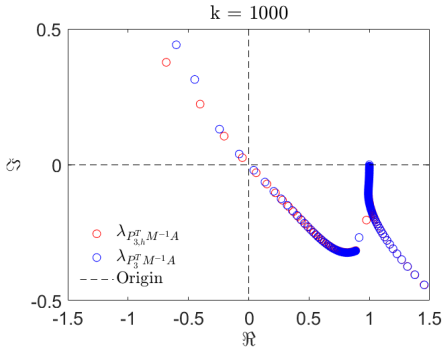


Figure 8.17: Quadratic rational Bezier ($m = 3$).

8

LOCAL NEAR-ZERO EIGENVALUES

Here we proceed by plotting the eigenvalues of the coarse-grid systems for levels $m \leqslant 3$. We take $k = 100$ as for smaller k, the plot containing the complete spectrum and the near-zero eigenvalues is better visible. The results are comparable to the ones obtained for the two-level ADP preconditioner. The near-zero eigenvalues for all levels where the coefficient matrices are indefinite remain aligned, see Fig. 8.19. Comparing this to Fig. 8.18 for the linear interpolation case, the near-zero eigenvalues start shifting as we move from $m = 0$ to $m = 2$. Note that at $m = 3$ all eigenvalues are negative, which follows from Corollary 14.2.

Spectrum of the coarse linear systems for $k = 100$ and $m \leqslant 3$.

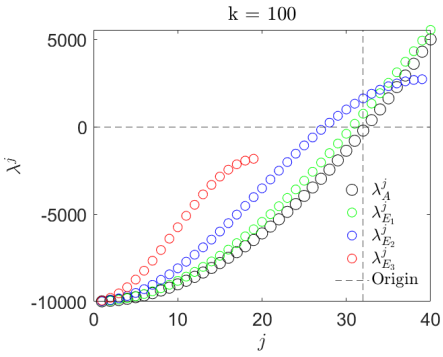


Figure 8.18: Linear Interpolation

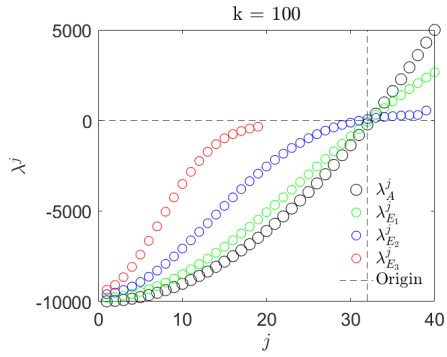


Figure 8.19: Quadratic Rational Bezier

8.6. NUMERICAL EXPERIMENTS

In this section we provide numerical experiments to study the convergence behavior of our multi-level preconditioner. All experiments are implemented sequentially on a Dell laptop using 8GB RAM and a i7-8665U processor. An exact solve is performed at the coarsest level with problem size $n < 10$. Moreover, we allow one FGMRES-iteration on each level to retain the V-cycle structure ($\gamma = 1$) and 10 grid points per wavelength, unless stated otherwise.
At levels where the matrix is indefinite, Bi-CGSTAB is used as smoother. For constant wavenumber model problems we need more Bi-CGSTAB iterations, as for heterogeneous problems the grid has been resolved with respect to the largest wavenumber. Thus for smaller values, there is in fact more accuracy than the required 10 grid points per wavelength. In all cases the number of inner iterations is determined by a constant times $n^{(l)\frac{1}{4}}$, where $n^{(l)}$ is the problem size of the linear system on level l, given that we do not want more iterations than necessary on the coarser levels. We use the following test models and report the number of iterations.

- 2D constant wavenumber (Sommerfeld) - MP2-A
- 2D constant wavenumber (Dirichlet + Sommerfeld) - MP2-B
- 2D wedge (Sommerfeld) - MP2-C
- 2D full Marmousi (Sommerfeld) - MP2-D
- 3D sine model (Dirichlet) - MP3-A
- 3D time-harmonic Elastic Wave equation (Dirichlet + Sommerfeld) - MP3-B

If timings are reported they will include the CPU-time in seconds using Matlab 2019Rb. The timings are for indicative purposes, please see our detailed complexity analysis above. The timings include all costs associated with the algorithm, including setting up the coarse-grid linear systems through matrix-matrix multiplications.

8.6.1. TWO-DIMENSIONAL CONSTANT WAVENUMBER MODEL PROBLEMS

We start by presenting the numerical results in Table 8.2 for the constant wavenumber model problem MP2-A and MP2-B using Sommerfeld and Sommerfeld and Dirichlet boundary conditions respectively. Given that we are looking for outer scalability of the FGMRES iterations in terms of k, we stop the simulations once more than 150 iterations are performed without having reached the desired tolerance level. The results indicate that we obtain a solver which is weakly dependent on the wavenumber. If we allow for Dirichlet conditions on one boundary, then the number of iterations increases.

Table 8.2: Number of outer FGMRES-iterations for MP2-A and MP2-B. ⊘ indicates that more than 150 iterations were needed to reach convergence. CPU time in seconds is rounded above and given in brackets. Max. iterations for inner Bi-CGSTAB has been set at $6n^{(l)\,\frac{1}{4}}$, for $l = 1,2$.

		MP2-A		MP2-B	
k	n	ADP-ML	DEF-ML	ADP-ML	DEF-ML
50	6.561	8 (1)	14 (1)	13 (1)	50 (2)
100	25.921	9 (2)	18 (4)	19 (6)	37 (8)
200	102.400	9 (7)	48 (43)	22 (42)	62 (95)
400	410.881	11 (68)	46 (285)	24 (201)	⊘
800	1.638.400	15 (342)	⊘	32 (955)	⊘

To put these results into perspective, if we were to use industry standard configuration with the CSLP inverted approximately using one multigrid V-cycle, then for MP2-B and $k = 200$ we would need 296 Bi-CGSTAB iterations which take 99.96 seconds to reach convergence. For MP2-A we would need 160 iterations which take approximately 40 seconds. Without a preconditioner and if no outer FGMRES with multi-level deflation were to be used, we would need 6188 Bi-CGSTAB iterations with a total time of 70.986 seconds for MP2-B and 2797 with 22.541 seconds for MP2-A. For $k = 200$, the maximum number of inner Bi-CGSTAB iterations is set at approximately 102. Using the multi-level deflation solver, we need 9 iterations to reach convergence which takes up roughly 7 seconds. While the inner iterations may appear to be a lot, note that, if we were to apply 9 times 102 iterations on a stand-alone basis, it is still less than the number of iterations of Bi-CGSTAB without any preconditioner (6188 and 2797 for MP2-B and MP2-A respectively). Moreover, both GMRES and Bi-CGSTAB used with the approximated inverse of the CSLP, require much more computation time. What we thus observe is that the synergy of using outer FGMRES to create the hierarchy of coarse-grid levels with inner Bi-CGSTAB leads to both lower computation times and lower iteration counts for highly indefinite systems.

Finally, we observe that use of the old deflation scheme (DEF-ML) based on linear interpolation provides less scalability in terms of k. This resonates with the theory from Section 8.5, where we concluded that the underlying spectrum of the local deflation operators remain aligned along the levels where the corresponding linear systems are indefinite.

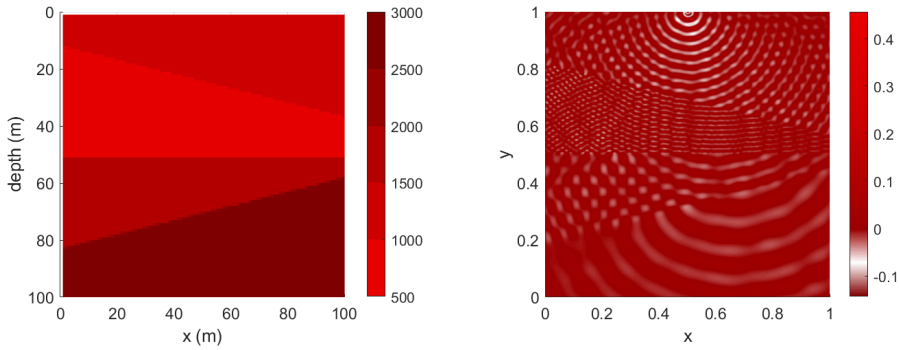
8.6.2. TWO-DIMENSIONAL HETEROGENEOUS MODEL PROBLEMS

In this subsection we provide the results to the numerical experiments for the Wedge model (MP2-C) and the Marmousi model (MP2-D).

WEDGE

Starting with MP2-C, Fig. 8.20 illustrates the underlying geometry of the wedge and the numerical solution. We divide the numerical domain into four sections containing a wedge.

Figure 8.20: Velocity profile and numerical solution for MP2-C for $f = 60$



(a) Velocity $c(x,y)$ (b) Pressure field

Table 8.3: Number of outer FGMRES-iterations for MP2B (Wedge). The largest wavenumber is resolved using 10 gpw. Max. iterations for inner Bi-CGSTAB has been set at $6n^{(l)\frac{1}{4}}$, for $l = 1,2$.

$k = 2\pi f$		$c(x,y) \in [500,3000]$ m/s		$c(x,y) \in [1000,6000]$ m/s	
f (Hz)	n	Iterations	CPU(s)	Iterations	CPU(s)
10	10.201	9	0.428	9	0.598
20	41.209	11	2.112	14	11.148
40	162.409	17	47.080	19	86.171
60	366.025	21	157.143	22	325.960
80	648.025	23	459.561	25	774.926

From Table 8.3 we again observe that for both velocity profiles reported, the number of iterations weakly depends on the frequency. In particular for the larger frequencies, we observe that if the problem size is doubled, the increase in computation time is four-fold. Additionally, it is apparent that the use of a variable wavenumber leads to more iterations and requires more computing time than the use of a constant wavenumber.

MARMOUSI

Next we consider an adapted version of the original Marmousi problem developed in [28]. The original domain has been truncated to $\Omega = [0,8192] \times [0,2048]$ in order to allow for efficient geometric coarsening of the discrete velocity profiles. Similar to some experiments in the literature, the coarsening keeps the proportions of the original velocity the same but

lets $c(x,y)$ vary between $2587.5 \leqslant c \leqslant 3325$. On Ω, we define

$$-\Delta u(x,y) - k(x,y)^2 u(x,y) = \delta(x-4000,y), (x,y) \in \Omega \backslash \partial\Omega \subset \mathbb{R}^2, \tag{8.27}$$
$$\left(\frac{\partial}{\partial \mathbf{n}} - ik\right) u(x,y) = 0, (x,y) \in \partial\Omega,$$

where $\mathbf{n}$ denotes the outward normal unit vector. The wavenumber is given by $k(x,y) = \frac{2\pi f}{c(x,y)}$, where the frequency f is given in Hertz.

Table 8.4: Number of outer FGMRES-iterations for the Marmousi problem MP2C, where f denotes the frequency in Hertz. The largest wavenumber has been resolved using 10 gpw. Max. iterations for inner Bi-CGSTAB has been set at $6n^{(l)\frac{1}{4}}$, for $l = 1,2$.

	n	$\gamma = 1$		$\gamma = 2$	
f (Hz)	n	Iterations	CPU(s)	Iterations	CPU(s)
10	66.177	18	18.113	13	18.551
20	263.425	21	117.677	14	75.177
40	1.051.137	30	810.90	20	914.297

The results from Table 8.4 show that the number of iterations again weakly depends on the wavenumber. Here we experiment with using a W-cycle instead of a V-cycle to construct the multi-level hierarchy. For the Marmousi problem we observe that it leads to a lower number of iterations for all the reported frequencies. However, for the largest test case, while the number of iterations are lower (20 instead of 30), the computation time increases. This can be explained by noting that for the W-cycle, more work is performed within each level.

8.6.3. THREE-DIMENSIONAL HETEROGENEOUS MODEL PROBLEMS

In this subsection we provide the results to the numerical experiments for the Sine model (MP3-A) and the Elastic wave model (MP3-B). Note that in the elastic wave equation, both force and displacement are vector quantities.

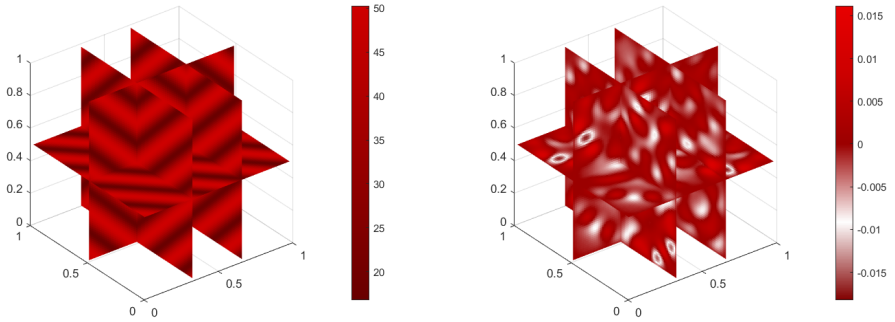
SINE MODEL

In this model we artificially construct a variant of the Helmholtz equation with sharply changes within a range of wavenumbers across the entire numerical domain. We therefore define the following

$$-\Delta u(x,y,z) - \tilde{k}(x,y,z)^2 u(x,y,z) = \delta(x-\frac{1}{2},y,z), (x,y,z) \in \Omega = [0,1]^3 \subset \mathbb{R}^3,$$
$$\tilde{k}(x,y,z) = \frac{k_1^2 + k_2^2}{2} + \left|\frac{k_2^2 - k_1^2}{2}\right| \sin(8\pi(x+y+z)), k_1 \in \mathbb{N}, k_2 = \frac{k_1}{3}$$
$$u(x,y,z) = 0, (x,y,z) \in \partial\Omega.$$

An illustration of the wavenumber profile is given in Fig. 8.21 (a).
The results are reported in Table 8.5. We observe that for this wavenumber model problem, where the wavenumber switches rapidly from low- to high-contrast, the dependency of the

Figure 8.21: wavenumber and numerical solution for MP3B (Sine) for $f = 8$



(a) wavenumber $\tilde{k}(x, y, z)$

(b) Numerical solution

iteration count on $k(x, y, z)$ is more pronounced. Experimenting with the W-cycle instead of the V-cycle does lead to a lower iteration count. However, for the largest test case ($f = 12$), we again observe an increase in the computation time.

Table 8.5: Number of outer FGMRES-iterations for sine-problem (MP3C), where f denotes the frequency in Hertz. Max. iterations for inner Bi-CGSTAB has been set at $6n^{(l)^{\frac{1}{4}}}$, for $l = 1,2$.

		8π			
$k = 2\pi f$		$\gamma = 1$		$\gamma = 2$	
$f(Hz)$	n	Iterations	CPU(s)	Iterations	CPU(s)
4	68.921	8	3.041	6	4.026
8	531.441	26	133.688	15	123.218
12	1.771.561	49	1259.185	28	1359.926

ELASTIC WAVE

For the time-harmonic elastic wave equation in a three-dimensional wedge we use the model from [117]. No splitting has been performed and the global system is solved. The results are given in Table 8.6. We again experiment with the V-cycle and the W-cycle. For the frequencies reported, the number of iterations slowly increases with the wavenumber. When comparing the computation time, once the frequency increases and the problem becomes large, the V-cycle is preferred. While the W-cycle leads to less iterations, it requires more computational work.

Table 8.6: Number of outer FGMRES-iterations for the time-harmonic elastic wave equation (MP3-B), where f denotes the frequency in Hertz using 20 gpw. Max. iterations for inner Bi-CGSTAB has been set at $7n^{(l)^{\frac{1}{4}}}$, for $l = 1,2$.

$k = 2\pi f$	n	$\gamma = 1$		$\gamma = 2$	
$f(Hz)$		Iterations	CPU(s)	Iterations	CPU(s)
1	19.968	8	2.871	8	3.598
2	147.033	11	87.214	9	77.971
4	1.127.463	15	1665.686	13	1735.294

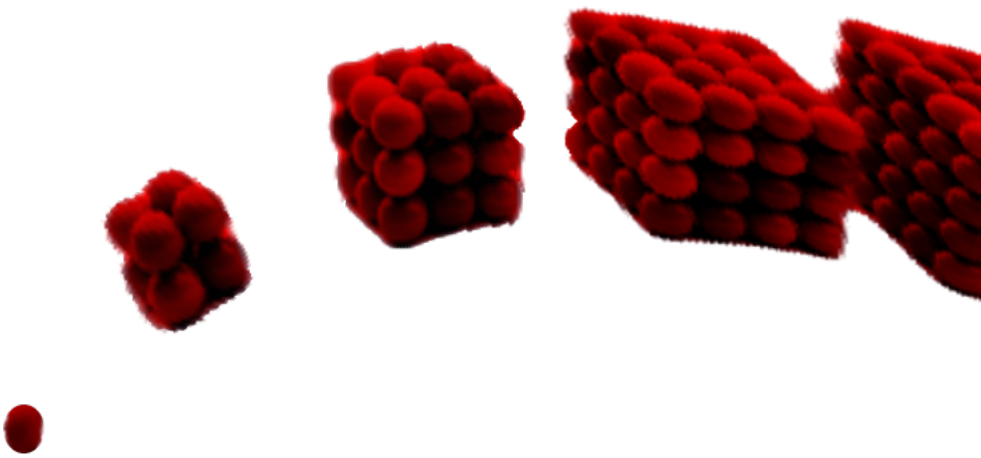
8.7. CONCLUSION

In this chapter we extend the two-level deflation preconditioner using higher-order deflation vectors to a multi-level deflation preconditioner [78]. We provide theoretical and numerical evidence to show that up to a certain level, the coefficient matrices are indefinite. These levels are of paramount importance as the near-zero eigenvalues at these levels can effectively be removed by the multi-level deflation preconditioner. If the near-zero eigenvalues are aligned, then the eigenvalues cluster near the point $(1,0)$ in the complex plane, accelerating the convergence of the underlying Krylov solver.

After this level, the subsequent coarse coefficient matrices become negative definite and its spectrum resembles the spectrum of the CSLP-preconditioned system. Thus, we implement $\mathcal{O}(n^{\frac{1}{4}})$ inner Bi-CGSTAB-iterations on the indefinite levels to approximate the CSLP using the inverse of the wavenumber k as the shift ($\beta_2 = k^{-1}$). This circumvents the difficulty of multigrid approximations, where the shift β_2 has to be kept large. The proposed configuration leads to scalable results as we obtain close to wavenumber independent convergence in terms of a fixed number of iterations. It furthermore, extends the results for

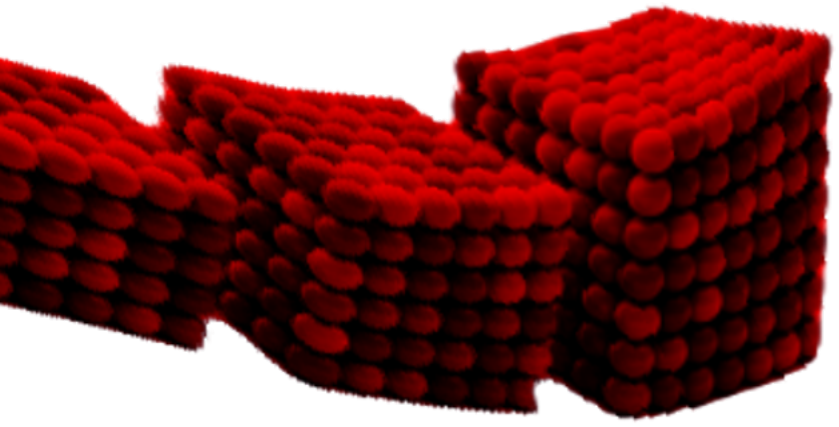
both a constant and non-constant wavenumber model problem, such as the two-dimensional industrial Marmousi model problem and the three-dimensional elastic wave equation. Additionally, sequential implementation of the method leads to scalable timing results for the model problems, which has been demonstrated using numerical experiments and a complexity analysis.

8



9

MULTIGRID METHODS



Parts of this chapter have been submitted to the SIAM Journal on Scientific Computing.

In this chapter we present the development of a convergent multigrid scheme for the highly indefinite Helmholtz equation. For over three decades, researchers have studied ways to design convergent and efficient multigrid schemes for the indefinite Helmholtz equation, which to this day remains an open problem in applied mathematics. Multigrid methods are widely known to be efficient and scalable numerical solvers for elliptic PDEs. In Chapter 4 we introduced the multigrid method and showed that it has linear complexity. It provided a state-of-the-art numerical approach, where the amount of computational work scales with the number of unknowns. In essence, the method combines solutions on coarser grids with relaxation (smoothing) techniques to arrive at a solution on the fine grid.
While multigrid works well for the positive-definite variant of the Helmholtz equation or when the wavenumber is small, the method generally fails to converge as a stand-alone solver or requires too many iterations for the indefinite version. This break-down can be attributed to two things. First of all, the coarse-grid generally needs to be fine enough in order to resolve the waves. A difference between the wavenumber on the fine vs. coarse-grid(s) can cause a phase error leading to instabilities. Moreover, eigenvalues approaching zero also hamper the convergence as they lead to near-singularity within the coarse-grid operator.
To address these concerns (in this chapter), we again start by introducing the general multigrid technique, followed by an overview of the literature, where we discuss existing multigrid strategies for the Helmholtz equation. Consequently, we define the model problems which will be the used in this chapter. Once we have defined the model problems, we start developing the theory towards a convergent standard multigrid scheme. By using similar techniques from the higher-order deflation method discussed in Chapter 7 and Chapter 8, we investigate the effect of introducing more accuracy on construction of the coarse levels with respect to the convergence. We conclude this chapter with some numerical experiments.

9.1. MULTIGRID METHODS

Recall from Chapter 4 that the multigrid method as a stand-alone solver combines two ingredients in a recursive way: a coarse grid correction and a smoother. If we let P and P' denote the matrix variant of the prolongation and restriction operators I_h^H nd I_H^h respectively from Section 4.1.1, then we equivalently define the coarse grid correction matrix C from Eq. (4.14) as

$$C = I - P' {A_c}^{-1} PA, \quad A_c = PAP'.$$

Note that we here denote the coarse grid version of A by A_c instead of A_H as the matrix A is now complex due to the Sommerfeld boundary conditions and we reserve the $H-$script to denote the conjugate transpose.
As for the smoother, we work with the standard weighted-Jacobi smoother, which is defined as

$$S_{JAC} = I - X^{-1} A, \quad X = \omega \Lambda_A,$$

where Λ_A denotes the matrix containing the diagonal elements of our matrix A.
To develop the theory, we start with the assumption that we are only using a post-smoother.

In this case, the two-grid iteration matrix is given by

$$T_0 = \left(I - P' {A_c}^{-1} PA\right)\left(I - X^{-1}A\right). \tag{9.1}$$

Generally the multigrid iteration scheme converges if the spectral radius of the two-grid iteration matrix is less than 1. Thus, the straightforward way to prove convergence is by showing that the spectral radius of the two-grid iteration matrix is strictly less than zero. If Dirichlet conditions are used, this naturally boils down to using Rigorous Fourier Analysis (RFA) to find analytical expressions for the eigenvalues of T_0. However, once we use Sommerfeld boundary conditions, this is no longer an option and we have to find another way to assess the boundedness of the spectral radius. We discuss this in detail in the next sections, but first give an overview of the literature related to using standard multigrid as a stand-alone solver for the Helmholtz equation, and strong indefinite systems in general.

9.2. LITERATURE OVERVIEW

Several strategies have been proposed to study the convergence behavior of indefinite systems [36, 118–120]. This boils down to prove convergence of the two-grid method for the positive definite variant (see [36], chapter 11.6). Another approach finding upper and lower bounds to the field of values of the two-grid operator [121–124]. This requires that the Hermitian part of the linear system $\frac{1}{2}\left(A + A^T\right)$ is positive definite, which is not the case for the indefinite Helmholtz equation. While we can still use the condition that the spectral radius should be smaller than one, the computation of the eigenvalues becomes difficult in case of Sommerfeld boundary conditions.

9.2.1. OPTIMALITY

For positive-definite and complex Hermitian problems, Geometric and/or Algebraic Multigrid can deal with these difficulties by constructing coarser linear systems based on the information provided by the fine-grid linear system. For complex Hermitian matrices, the coarse linear systems can be build using the Galerkin condition, which has been studied in [125, 126]. If the use of a simple smoother such as weighted-Jacobi or Gauss-seidel is preferred, the coarse grid correction scheme should balance the simplicity of the smoother by accurately transferring smooth errors from the coarse grid to the fine grid [126]. If the linear system is complex symmetric, linear interpolation can be used to construct the prolongation operator. The restriction operator can then be chosen as the transpose of the prolongation operator. In fact, for real matrices optimality studies have been conducted in order to construct the interpolation and prolongation operators [127–130]. In [131], a similar optimality condition from [129, 130] has been extended to the Hermitian case. These studies show that the optimal prolongation and restriction operator are based on the eigenvectors of the system matrix and several approximations are given which could function as a satisfactory proxy in the case of SPD and Hermitian systems respectively.

9.2.2. INDEFINITENESS

Including a complex shift, leads to the Complex Shifted Laplacian (CSL), which converges if the multigrid method is used as a preconditioner. Consequently, multigrid has also been used as a preconditioner to solve the original indefinite Helmholtz problem, where the CSL is inverted approximately [27, 33, 109, 132]. While this works well for homogeneous problems, it is not suitable for heterogeneous problems with high contrasts [27, 29]. When it

comes to using multigrid as a stand-alone solver for original Helmholtz operator, several studies have focused on getting multigrid to comply with the underlying characteristics of the indefinite system. One study uses a smoothed aggregation approach, where an effective prolongation operator is constructed by minimizing the energy norm defined using the normal equations $A^H A$. This is achieved by running the CGNR algorithm to solve the equation $A^H AP = 0$, where P is the tentative prolongation operator [133, 134]. Two other methods are the wave-ray approach [112] and the first-order least squares (FOLS) approach [135]. Both compute multiple coarse spaces to approximate the plane waves in the near null space. In both works, the development of more efficient approaches for such linear systems has been stressed. Moreover, for variable wavenumbers, it remains challenging to choose an effective prolongation and restriction operator, especially if the aim is to keep the algorithm as light as possible.

9.2.3. POLYNOMIAL SMOOTHING

One important study finds that replacing the weighted-Jacobi smoother with a few GMRES iterations could lead to improved convergence [110]. In [102], it was shown that for the 1D indefinite Helmholtz equation, convergence can be achieved by considering a two-step Jacobi smoother. Scheduling the smoothing steps on all levels as a function of the wavenumber provides a uniform error reduction in the high-frequency modes. More recently a 2D dispersion correction has been developed for the original Helmholtz operator with a constant wavenumber. Here, a larger number of smoothing steps combined with at least 4.5 grid points per wavelength at the coarsest level leads to a convergent multigrid scheme. [51]. A full multigrid hierarchy, without any restrictions on the grid resolution at the coarsest level, is given in [136]. Here, a re-discretization technique is used where a level-dependent complex shift is incorporated, except the finest level. The authors show that the altered two-grid correction scheme, combined with 3 GMRES iterations as a smoother, remains robust and does not lead to agitation of smooth modes. The method works well on structured grids and in cases where the behavior of the wavenumber on the fine and coarse(r) grids can be defined a-priori. Complex geometries, high-contrast and irregularly varying wavenumber would pose difficulties in constructing the level-dependent coarser linear systems when using re-discretization.

9

9.3. PROBLEM DESCRIPTION

We continue by defining the model problems which will be studied in this chapter. We focus on the 2D constant wavenumber model using $k > 0$, which we call MP 2-A.

MP 2-A

$$\begin{aligned} -\Delta u(x,y) - k^2 u(x,y)\, u(x,y) &= \delta(x-\frac{1}{2}, y-\frac{1}{2}), \quad (x,y) \in \Omega = [0,1]^2, \\ \left(\frac{\partial}{\partial \mathbf{n}} - ik\right) u(x,y) &= 0, \quad (x,y) \in \partial\Omega. \end{aligned} \tag{9.2}$$

The final test problem uses a non-constant wavenumber $k(x,y)$. This gives us MP 2-B as defined below

MP 2-B

$$
\begin{aligned}
-\Delta u(x,y) - k(x,y)^2 u(x,y)\, u(x,y) &= \delta(x-\frac{1}{2}, y-\frac{1}{2}), \quad (x,y)\in\Omega=[0,1]^2, \\
\left(\frac{\partial}{\partial \mathbf{n}} - ik(x,y)\right) u(x,y) &= 0, \quad (x,y)\in\partial\Omega, \\
k(x,y) &= k_1^2 + \chi(x,y)\left|k_1^2 - k_2^2\right|,
\end{aligned} \tag{9.3}
$$

where $\chi(x,y)$ is a random real function in the range of $[0,1]$ and k_1, k_2 are positive real numbers. Note that in this case k varies exactly between k_1 and k_2. We use second-order finite differences to discretize the model problems and define the step-size $h = \frac{1}{n}$, where n is chosen such that for each k_1, k_2 we have $kh = 0.625$. This is equivalent to having 10 grid points per wavelength.

9.4. MULTIGRID METHODS

In this section, we prove two properties which we need in order to construct a robust solver: convergence and an optimal smoother. We start with convergence and then work our way towards deriving optimality conditions for the $\omega-$Jacobi smoother. The $\omega-$Jacobi smoother is known to diverge for Helmholtz problems. However, we show that with the right conditions, we can obtain a convergent solver.

9.4.1. CONVERGENCE

For convergence of the two-grid method we require that $\|T_0\|_2 < 1$ independent of h. We can write T_0 as $T_0 = I - DA$, where D represents an approximation to A^{-1}. This is shown in Lemma 15.1. We use this to show that if the two-grid operator T_0 can be written in this form and the two-norm is bounded by one, then the multigrid method will converge.

Lemma 15.1: Convergence - I

Let $A \in \mathbb{C}^{n\times n}$ be a non-singular and non-defective matrix, and let the two-grid operator be given by $T_0 = (I - P'{A_c}^{-1}PA)(I - X^{-1}A)$. Then, T_0 can be written as $T_0 = I - DA$, with

$$D = X^{-1} + P'{A_c}^{-1}P - P'{A_c}^{-1}PX^{-1} \tag{9.4}$$

Moreover we have ${T_0}^H = I - A^H D^H$.

Proof. Expanding T_0, and factoring in terms of A we have

$$
\begin{aligned}
T_0 &= I - X^{-1}A - P'{A_c}^{-1}PA + P'{A_c}^{-1}PX^{-1}A, && (9.5)\\
&= I - \left(X^{-1} + P'{A_c}^{-1}P - P'{A_c}^{-1}PX^{-1}\right)A, && (9.6)\\
&= I - DA, && (9.7)
\end{aligned}
$$

9

where $D = X^{-1} + P' {A_c}^{-1} P - P' {A_c}^{-1} P X^{-1}$. We similarly obtain

$$
\begin{aligned}
{T_0}^H &= \left(\left(I - P' {A_c}^{-1} P A \right) \left(I - X^{-1} A \right) \right)^H, && (9.8)\\
&= \left(I - X^{-1} A \right)^H \left(I - P' {A_c}^{-1} P A \right)^H, && (9.9)\\
&= \left(I - A^H X^{-1} \right) \left(I - A^H P' \left({A_c}^H \right)^{-1} P \right), && (9.10)\\
&= I - A^H X^{-1} - A^H P' \left({A_c}^H \right)^{-1} P + A^H X^{-1} A^H P' \left({A_c}^H \right)^{-1} P, && (9.11)\\
&= I - A^H \left(X^{-1} + P' \left({A_c}^H \right)^{-1} P - X^{-1} A^H P' \left({A_c}^H \right)^{-1} P \right), && (9.12)\\
&= I - A^H D^H, && (9.13)
\end{aligned}
$$

where $D^H = X^{-1} + P' \left({A_c}^H \right)^{-1} P - X^{-1} P' \left({A_c}^H \right)^{-1} P$. ■

It has been shown that convergence of the two-grid operator implies multigrid convergence, see [36]. The proof is based on an induction argument, where for the $W-$cycle, it is shown that the multigrid iteration matrix can be written as the two-grid iteration matrix plus a perturbation term. The perturbation term is bounded if the prolongation operator is a bounded linear operator and the smoother can be bounded by a constant. Note that a convergent smoothing scheme is not required. In fact, the smoothing property also holds for non-convergent iterations [36]. It is only required that the error is smoothed up to a certain ν, where ν is the number of smoothing steps. In other words, there exists a ν such that the error is reduced. It is not imperative that the smoothing property holds for $\nu = \infty$. We now work towards conditions for which $\| T_0 \|_2 < 1$ holds.

Theorem 16: Convergence - I

Let $A \in \mathbb{C}^{n \times n}$ be a non-singular and non-defective matrix. Let the two-grid operator be given by $T_0 = (I - P' {A_c}^{-1} P A)(I - X^{-1} A)$, where X is the $\omega-$Jacobi smoother, $P, P' \in \mathbb{R}^{n \times m}$ are the prolongation and restriction operator and $A_c = P' A P$. Let

$$\Gamma = A^H D^H + DA - A^H D^H DA, \tag{9.14}$$

then ${T_0}^H T_0 = I - \Gamma$. If Γ is positive definite, then $\lambda_j(\Gamma) \in (0,2)$ and $\| T_0 \|_2 < 1$ independent of h.

Proof. Using equation Lemma 15.1, equation Eq. (9.7) and Eq. (9.13), we can write

$$
\begin{aligned}
{T_0}^H T_0 &= \left(I - A^H D^H \right) (I - DA), && (9.15)\\
&= I - \left(DA + A^H D^H - A^H D^H DA \right), && (9.16)\\
&= I - \Gamma. && (9.17)
\end{aligned}
$$

We can show a stronger result than just positive definiteness since Γ is also Hermitian. To

see this, note

$$\Gamma^H = \left(A^H D^H + DA - A^H D^H DA\right)^H, \tag{9.18}$$
$$= \left(DA + A^H D^H\right)^H - \left(A^H D^H DA\right)^H, \tag{9.19}$$
$$= (DA)^H + \left(A^H D^H\right)^H - \left(A^H D^H DA\right)^H, \tag{9.20}$$
$$= A^H D^H + DA - A^H D^H DA = \Gamma. \tag{9.21}$$

Given that Γ is HPD, the singular value decomposition (SVD) and eigendecomposition coincide, so we can find a unitary matrix $U \in \mathbb{C}^{n\times n}$ such that $\Gamma = U\Sigma_\Gamma U^H$, where Σ is the diagonal matrix containing the singular resp. eigenvalues (σ_j), where we have $\sigma_1 \geqslant \sigma_2 \geqslant \sigma_3 \ldots \geqslant \sigma_n > 0$. We have

$$\Gamma = A^H D^H + DA - A^H D^H DA, \tag{9.22}$$

which is equivalent to

$$\Sigma_\Gamma = UA^H D^H U^H + UDAU^H - UA^H D^H DAU^H. \tag{9.23}$$

By our assumption on Γ being positive definite, we have that for all $x \in \mathbb{C}^n \backslash \{0\}$,

$$0 < x^H \Gamma x \Leftrightarrow 0 < x^H A^H D^H x + x^H DAx - x^H A^H D^H DAx. \tag{9.24}$$

Similarly, by using Eq. (9.23), for all $x \in \mathbb{C}^n \backslash \{0\}$,

$$0 < x^H \Sigma_\Gamma x \Leftrightarrow 0 < x^H UA^H D^H U^H x + x^H UDAU^H x - x^H UA^H D^H DAU^H x, \tag{9.25}$$
$$\Leftrightarrow 0 < x^H UA^H D^H DAU^H x < x^H UA^H D^H U^H x + x^H UDAU^H x. \tag{9.26}$$

Using the unitary invariance of the SVD, Eq. (9.26) leads to

$$0 < x^H UA^H D^H DAU^H x < x^H UA^H D^H U^H x + x^H UDAU^H x, \tag{9.27}$$
$$\Rightarrow 0 < \sigma_j(DA)^2 < \sigma_j(A^H D^H + DA) \leqslant 2\sigma_j(DA), \tag{9.28}$$
$$\Rightarrow \sigma_j(DA) < 2. \tag{9.29}$$

This gives an upperbound for $\sigma_j(DA)$. We now develop the upper bound for the largest singular value of Γ, which is denoted by $\sigma_1(\Gamma)$. We know that $0 \leqslant \sigma_j(DA) < 2$. The case where $0 \leqslant \sigma_j(DA) < 1$ is trivial as that automatically renders $\|T_0\|_2 < 1$. We focus on the case where $1 \leqslant \sigma_j(DA) < 2$. Recall, we assumed Γ is HPD, but DA does not necessarily have to be. As a result, the singular value decomposition for Γ with unitary matrix U might not coincide with the singular value decomposition for DA.
Suppose, we can write DA as $DA = W\Gamma_{DA}V^H$, where $WV^H \neq I$. We then still have

$$A^H D^H DA = V{\Gamma_{DA}}^H W^H W\Gamma_{DA}V^H, \tag{9.30}$$
$$V^H A^H D^H DAV = {\Gamma_{DA}}^H \Gamma_{DA} = (\Gamma_{DA})^2. \tag{9.31}$$

Moreover, we also have for $v_j \in V$

$$DAv_j = \sigma_j(DA)u_j \text{ and} \tag{9.32}$$
$$(DAv_j)^H = {v_j}^H A^H D^H = {u_j}^H \sigma_j(DA). \tag{9.33}$$

For $WV^H \neq I$ we will always have that for some index j and $v_j \in V$ and $w_j \in W$

$$v_j{}^H w_j + w_j{}^H v_j < 2, \tag{9.34}$$

where we used that the complex parts cancel out. We therefore have

$$v^H DAv + v^H A^H D^H v = v_j{}^H \sigma_j(DA) u_j + w_j{}^H \sigma_j(DA) v_j < 2\sigma_j(DA). \tag{9.35}$$

Taking $x = v_j$ in Eq. (9.24) and using Eq. (9.35), $x^H \Gamma x$ can be bounded by

$$\begin{aligned}
0 &< v_j{}^H A^H D^H v_j + v_j{}^H DAv_j - v_j{}^H A^H D^H DAv_j, && (9.36)\\
&= \sigma_j(DA)\left(v_j{}^H w_j + w_j{}^H v_j\right) - \sigma_j(DA)^2, && (9.37)\\
&< 2\sigma_j(DA) - \sigma_j(DA)^2 = \sigma_j(DA)\left(2 - \sigma_j(DA)\right), && (9.38)\\
&< \sigma_j(DA) < 2. && (9.39)
\end{aligned}$$

As Γ is positive definite, the eigenvalues and singular values coincide and we have

$$\lambda_{\min}(\Gamma) > 0 \text{ and } \lambda_{\max}(\Gamma) < 2. \tag{9.40}$$

We are now ready to bound the two-grid operator T_0 using the 2-norm over $\mathbb{C}^{n\times n}$

$$\begin{aligned}
\|T_0{}^H T_0\|_2 &= \sup_{x\in\mathbb{C}^n, \|x\|_2=1} \|\left(T_0{}^H T_0\right) x\|_2 = \sqrt{\rho\left(T_0{}^H T_0\right)},\\
&= \sqrt{\rho\left(I - \left(A^H D^H + DA - A^H D^H DA\right)\right)}, && (9.41)\\
&= \sqrt{\max|\lambda_j\left(I - \left(A^H D^H + DA - A^H D^H DA\right)\right)|}, && (9.42)\\
&\leqslant \sqrt{|1 - \min\lambda_j\left(A^H D^H + DA - A^H D^H DA\right)|}, && (9.43)\\
&< 1,
\end{aligned}$$

where we used that trivially $0 < \min\lambda_j\left(A^H D^H + DA - A^H D^H DA\right)$ by positive definiteness of Γ. As regards, the largest eigenvalue, we obtain

$$\begin{aligned}
\|T_0{}^H T_0\|_2 &= \sup_{x\in\mathbb{C}^n, \|x\|_2=1} \|\left(T_0{}^H T_0\right) x\|_2 = \sqrt{\rho\left(T_0{}^H T_0\right)},\\
&= \sqrt{\rho\left(I - \left(A^H D^H + DA - A^H D^H DA\right)\right)}, && (9.44)\\
&= \sqrt{\max|\lambda_j\left(I - \left(A^H D^H + DA - A^H D^H DA\right)\right)|}, && (9.45)\\
&\leqslant \sqrt{|1 - \max\lambda_j\left(A^H D^H + DA - A^H D^H DA\right)|}, && (9.46)\\
&< \sqrt{|1-2|} = 1, \text{ using Eq. (9.40)}. && (9.47)
\end{aligned}$$

Thus, if Γ is positive definite then $\lambda_j(\Gamma) \in (0,2)$ and consequently $\lambda_j(T_0) \in (0,1)$. ■

Theorem 16 shows that if Γ is positive definite, then the eigenvalues of T_0 will lie in the interval $(0,1)$ which automatically leads to a bound for the 2-norm of the two-grid operator. Note that this is independent of h, as no constant appears in Eq. (9.40) which depends on h. Now that we know that a sufficient conditions for convergence is to have Γ be positive definite, we need a way to practically assess this. One approach is by simplifying the expression for Γ and finding another sufficient condition for positive definiteness, which is more easy to analyze. This will be constructed in Corollary 16.1. We again start by defining the simplified operator in Definition 7.

Definition 7 (Simplified operator)**.** *Let $A \in \mathbb{C}^{n\times n}$ be a non-singular and non-defective matrix. $P, P' \in \mathbb{R}^{n\times m}$ are real valued prolongation and restriction operators and $A_c = P'AP$. The $\omega-$Jacobi smoother is given by $X = \omega\Lambda_A$, where Λ denotes the diagonal matrix containing the diagonal entries of A. Then we define $\tilde{D} = X^{-1} + P'{A_c}^{-1}P$ such that $\tilde{\Gamma} = A^H\tilde{D}^H + \tilde{D}A - A^H\tilde{D}^H\tilde{D}A$.*

Note that in the definition given above, $\tilde{D}$ does not contain the term $P'{A_c}^{-1}PX^{-1}A$, which is part of the original operator D.

Corollary 16.1: Convergence - II

Let $A \in \mathbb{C}^{n\times n}$ be a non-singular and non-defective matrix. Let $\tilde{\Gamma}$ be defined as in Definition 7. If $\tilde{\Gamma}$ is positive definite, then Γ from Theorem 16 is positive definite.

Proof. To see this, we write Γ as

$$\Gamma = A^HD^H + DA - A^HD^HDA, \tag{9.48}$$
$$= \left(A^H\tilde{D}^H + \tilde{D}A - A^H\tilde{D}^H\tilde{D}A\right) + A^HX^{-1}P'\left({A_c}^H\right)^{-1}PP'{A_c}^{-1}PX^{-1}A \tag{9.49}$$
$$- A^H\tilde{D}^HP'{A_c}^{-1}PX^{-1}A - A^HX^{-1}P'\left({A_c}^H\right)^{-1}P\tilde{D}A. \tag{9.50}$$

We assume $\tilde{\Gamma}$ is positive definite and HPD respectively. Now, assuming the opposite, i.e. Γ is not HPD implies that $\exists x \in \mathbb{C}^n \setminus \{0\}$ such that

$$x^H\tilde{\Gamma}x + x^HA^HX^{-1}P'\left({A_c}^H\right)^{-1}PP'{A_c}^{-1}PX^{-1}Ax \quad (*)$$
$$- x^HA^H\tilde{D}^HP'{A_c}^{-1}PX^{-1}Ax - x^HA^HX^{-1}P'\left({A_c}^H\right)^{-1}P\tilde{D}Ax \leqslant 0. \quad (**)$$

By assumption and the fact that the second term in $(*)$ contains a quadratic form, $(*)$ is always positive for any $x \in \mathbb{C}^n \setminus \{0\}$. But if Γ is not positive definite then using $(**)$ we must also have

$$0 < x^H\tilde{\Gamma}x + x^HA^HX^{-1}P'\left({A_c}^H\right)^{-1}PP'{A_c}^{-1}PX^{-1}Ax, \tag{9.51}$$
$$\leqslant x^HA^H\tilde{D}^HP'{A_c}^{-1}PX^{-1}Ax + x^HA^HX^{-1}P'\left({A_c}^H\right)^{-1}P\tilde{D}Ax \tag{9.52}$$

We show that $P'{A_c}^{-1}PX^{-1}A$ has singular values equal to zero. Observe that the rank of $P'{A_c}^{-1}PX^{-1}A$ is $\frac{n}{2}$. Suppose we can write the singular value decomposition as $\hat{\Gamma} = W^H\left(P'{A_c}^{-1}PX^{-1}A\right)V$. In this case we have V and W to represent two unitary complex matrices as we have not assumed that $P'{A_c}^{-1}PX^{-1}A$ is HPD. We know that $P'{A_c}^{-1}P$ is singular and half of all singular

values are equal to zero. We thus need to show that $P' {A_c}^{-1} P X^{-1} A$ has zero singular values as well. In order to do this, we use an analogous version of Ostrowski's theorem. Here instead of using eigenvalues, we use singular values. According to the theorem [66], for $K \in \mathbb{C}^{n\times n}$ and $Y \in \mathbb{C}^{n\times n}$ with Y is non-singular, we have

$$\sigma_j(KY) \leqslant \sigma_1(Y)\sigma_j(A), \quad j = 1,2,\dots n, \tag{9.53}$$

where $\sigma_1 \geqslant \sigma_2 \geqslant \dots \sigma_n$. Applying Eq. (9.53) to our case by taking $K = P' {A_c}^{-1} P$ and $Y = X^{-1}A$, we obtain

$$\sigma_j\left(P' {A_c}^{-1} P X^{-1} A\right) \leqslant \sigma_1(X^{-1}A)\sigma_j(P' {A_c}^{-1} P), \quad j = 1,2,\dots n. \tag{9.54}$$

We know that $\sigma_1(X^{-1}A) > 0$ given that both X and A are non-singular. However, we know that $P' {A_c}^{-1} P$ is singular and has zero singular values. Thus there exists indices $\tilde{j}$ such that

$$\sigma_{\tilde{j}}\left(P' {A_c}^{-1} P X^{-1} A\right) \leqslant 0, \quad \tilde{j} \in \{1,2,\dots n\}. \tag{9.55}$$

Trivially $0 \leqslant \sigma_{\tilde{j}}(P' {A_c}^{-1} P X^{-1} A)$ is a lower bound for the smallest singular value. Combining this with Eq. (9.55) leads to

$$0 \leqslant \sigma_{\tilde{j}}\left(P' {A_c}^{-1} P X^{-1} A\right) \leqslant 0, \quad \exists \tilde{j} \in \{1,2,\dots n\}, \tag{9.56}$$

and so by the interlacing theorem, we must have that there exist at least one singular value at an index $\tilde{j}$ such that $\sigma_{\tilde{j}}(P' {A_c}^{-1} P X^{-1} A) = 0$. Naturally, the eigenvalue of $P' {A_c}^{-1} P X^{-1} A$ corresponding to the index $\tilde{j}$ must be zero as well, given that the singular value is zero.
We know that for any complex matrix, there exists a unitary matrix such that we can write it in the Schur decomposition. Without the assumption of normality, it solves a general eigenvalue problem of the form $Kv = \lambda Bv$ for some matrix B.
We take $x = v_{\tilde{j}}$, corresponding to the zero eigenvalue, which we denote by $\lambda^0_{\tilde{j}}$ such that

$$P' {A_c}^{-1} P X^{-1} A v_{\tilde{j}} = \lambda_{\tilde{j}}(P' {A_c}^{-1} P X^{-1} A) B v_{\tilde{j}} = 0 B v_{\tilde{j}}. \tag{9.57}$$

Substituting Eq. (9.57) into Eq. (9.52) gives

$$\begin{aligned} 0 &< {v_{\tilde{j}}}^H \tilde{\Gamma} v_{\tilde{j}} + {v_{\tilde{j}}}^H A^H X^{-1} P' \left({A_c}^H\right)^{-1} P P' {A_c}^{-1} P X^{-1} A v_{\tilde{j}}, && (9.58)\\ &\leqslant {v_{\tilde{j}}}^H A^H \tilde{D}^H P' {A_c}^{-1} P X^{-1} A v_{\tilde{j}} + {v_{\tilde{j}}}^H A^H X^{-1} P' \left({A_c}^H\right)^{-1} P \tilde{D} A v_{\tilde{j}}, && (9.59)\\ &= \lambda^0_{\tilde{j}} {v_{\tilde{j}}}^H A^H \tilde{D}^H B v_{\tilde{j}} + \lambda^0_{\tilde{j}} {v_{\tilde{j}}}^H B^H \tilde{D} A v_{\tilde{j}} = 0. && (9.60) \end{aligned}$$

This gives a contradiction as $\tilde{\Gamma}$ is assumed to be positive definite and must always be larger than zero for any $x \in \mathbb{C}^n \setminus \{0\}$. We thus have that if $\tilde{\Gamma}$ is positive definite, then $x^H \Gamma x > 0 \; \forall x \in \mathbb{C}^n \setminus \{0\}$, i.e. Γ must be positive definite as well. ■

Using the above, we now have that it is sufficient to check if $\tilde{\Gamma}$ is positive definite. We therefore test for convergence by verifying whether $\tilde{\Gamma}$ is positive definite. We use the Cholesky decomposition to determine whether $\tilde{\Gamma}$ is positive definite, as any HPD system matrix B can be written as $B = LL^*$, where $L \in \mathbb{C}^{n\times n}$ is a lower-triangular matrix. Another method to check for positive definiteness without having to compute the Cholesky factors is by checking all of the following

1. $b_{ii} > 0 \forall i$
2. $b_{ii} + b_{jj} > 2|\Re[b_{ij}]|$ for $i \neq j$
3. The element with the largest modulus lies on the diagonal
4. $\det(B) > 0$

CONVERGENCE FOR HELMHOLTZ

We now proceed by evaluating these properties for the Helmholtz operator. We use MP2-A, which uses a constant wavenumber k. We numerically investigate the changes we can make to the design of the coarse-grid system in order to achieve convergence. In this work, we consider two options; we either use higher-order interpolation schemes to construct the prolongation and restriction operator and/or we use the CSL, which will be denoted by C, as the system on which we apply the coarsening operations. We use the higher-order interpolation scheme based on the quadratic rational Beziér curve from [78]. There, the scheme was used to construct the deflation vectors for a two-level deflation preconditioner, which showed wavenumber independent convergence. The main motivation for studying these type of higher-order schemes is that they have been shown to reduce the projection error. Using this scheme, the prolongation operator acts on a grid function as follows

$$P[u_{2h}]_i = \left\{ \begin{array}{cc} \frac{1}{8}\left([u_{2h}]_{(i-2)/2} + 6[u_{2h}]_{(i)/2} + [u_{2h}]_{(i+2)/2}\right) & \text{if } i \text{ is even,} \\ \frac{1}{2}\left([u_{2h}]_{(i-1)/2} + [u_{2h}]_{(i+1)/2}\right) & \text{if } i \text{ is odd,} \end{array} \right\}, \tag{9.61}$$

for $i = 1, \ldots, N$ and for $i = 1, \ldots, \frac{N}{2}$, where N denotes the size of the fine-level linear system. Note that the second line (i = odd) is in fact the linear interpolation scheme. Thus, we now have a combination of both a higher-order and linear interpolation scheme for the nodes i within the numerical domain.

The inclusion of the complex shift was first applied to multigrid schemes in [136]. There, the main motivation relies on spectral analysis of the two-grid operator and the observation that the coarse-grid eigenvalues can never approach zero due to the complex part. We proceed by a different yet general theoretical argument, which is independent of the boundary conditions. Moreover, we include a constant complex shift compared to a level-dependent one.

In Table 9.1, we show that the use of the CSL in combination with a higher-order interpolation scheme in fact leads to a HPD system. Note that while our conditions do not require the actual computation of the spectral radius, we will do so for sake of illustration and completeness.

	Linear				Beziér			
k	A		C		A		C	
5	✗	2.284	✗	1.304	✓	1.009	✓	0.936
10	✗	5.888	✗	1.351	✓	1.105	✓	0.943
20	✗	8.786	✗	1.328	✓	1.306	✓	0.968
30	✗	10.660	✗	1.325	✗	1.504	✗	0.990

Table 9.1: At the right of each entry, the spectral radius of the two-grid operator is given. Linear uses linear interpolation to construct P, P'. Beziér uses rational quadratic Beziér interpolation. A represents $A_c = P'AP$. C represents $A_c = P'CP$, where C denotes the CSL. In all cases, one post-smoothing step is used. Left of each entry, ✓ denotes that Γ is HPD and ✗ denotes it is not.

Several interesting observations can be made. First of all, in almost call cases when using a combination of both higher-order interpolation and the CSL for coarsening, we obtain an HPD matrix and consequently a spectral radius which is bounded by one. However, in the last entry we observe that while the matrix ceases to be HPD, we still have that the spectral radius of the two-grid operator is bounded by 1. If we increase the number of post-smoothing steps i.e., replace X^{-1} with $\tilde{X}^{-1}$, where $\tilde{X}^{-1} = 2X^{-1} - X^{-1}AX^{-1}$ in the theorems above, then the matrix becomes HPD again (see Table 9.2). While we observe that the two-grid method can converge given the right parameter choices, the current spectral radii indicate that convergence will be slow.
Finally, we report the positive definiteness of $\tilde{\Gamma}$ in Table 9.2, which will be easier to verify compared to Γ. From the previous, we know that in order to have an HPD matrix we need both coarsening on C and higher-order prolongation and restriction. Therefore, we only report the results for Γ and $\tilde{\Gamma}$ using these adjustments. The results reported in Table 9.2 agree with Corollary 16.1.

	$\nu = 1$		$\nu = 2$	
k	$\tilde{\Gamma}$	Γ	$\tilde{\Gamma}$	Γ
5	✓	✓	✓	✓
10	✓	✓	✓	✓
20	✓	✓	✓	✓
30	✗	✗	✓	✓

Table 9.2: ✓ denotes whether Γ and $\tilde{\Gamma}$ are HPD respectively. ✗ denotes if they're not. ν denotes the number of post-smoothing steps.

9.4.2. OPTIMALITY

Apart from having a condition to check for convergence, we would like to have a measure of how fast we can expect the two-grid method to converge. By having a positive definite Γ, we know that the minimal and maximal eigenvalues are bounded by zero and two respectively. To obtain a more accurate estimate, we study the condition number of Γ and $\tilde{\Gamma}$. Using the condition numbers we obtain a sharper bound in Theorem 16. This entails that the smallest eigenvalue of $\tilde{\Gamma}$ will be a lower bound to the smallest eigenvalue of Γ.

Corollary 16.2: Convergence Optimality Condition - I

Let $A \in \mathbb{C}^{n\times n}$ be a non-singular and non-defective matrix and let $\tilde{D} = X^{-1} + P' {A_c}^{-1} P$. If $\tilde{\Gamma} = A^H \tilde{D}^H + \tilde{D}A - A^H D^H DA$ is positive definite and $\|\Gamma\| \leqslant \|\tilde{\Gamma}\|$ together with $\kappa(\tilde{\Gamma}) \leqslant \kappa(\Gamma)$ then $\lambda_{\min}(\tilde{\Gamma})$ is given and bounded by

$$\frac{\|\tilde{\Gamma}\|}{\kappa(\tilde{\Gamma})} \leqslant \lambda_{\min}(\Gamma). \tag{9.62}$$

Proof. Note that $\tilde{\Gamma}$ and Γ are HPD and so all eigenvalues are real. We can therefore consider $\|\Gamma^{-1}\|$ and $\|\tilde{\Gamma}^{-1}\|$. Moreover, $\forall$ matrices $B \in \mathbb{C}^{n\times n}, \rho(B) = \max_{1\leqslant j\leqslant n} |\lambda_j(B)| \leqslant \|B\|$ in any associative norm. If $\kappa(\tilde{\Gamma}) \leqslant \kappa(\Gamma)$, then we obtain

$$\|\Gamma\| \leqslant \|\tilde{\Gamma}\| \Rightarrow \frac{1}{\|\tilde{\Gamma}\|} \leqslant \frac{1}{\|\Gamma\|} \Rightarrow \frac{\kappa(\tilde{\Gamma})}{\|\tilde{\Gamma}\|} \leqslant \frac{\kappa(\Gamma)}{\|\Gamma\|} \Rightarrow \|\tilde{\Gamma}^{-1}\| \leqslant \|\Gamma^{-1}\|. \tag{9.63}$$

Using that the systems are HPD, we have

$$\rho(\Gamma^{-1}) = \max_{1\leqslant j\leqslant n} |\lambda_j(\Gamma^{-1})| = \min_{1\leqslant j\leqslant n} |\lambda_j(\Gamma)|^{-1} \leqslant \|\Gamma^{-1}\|. \tag{9.64}$$

Note that this gives the reciprocal of the smallest eigenvalue of Γ. To get the actual smallest eigenvalue of Γ, we have to take the reciprocal, which provides us with inequality Eq. (9.62) and a lower bound to $\lambda_{\min}(\Gamma)$. ■

We can use Corollary 16.2 to construct a sharper bound in Theorem 16.

Corollary 16.3: Convergence Optimality Condition - II

Let $A \in \mathbb{C}^{n\times n}$ be a non-singular and non-defective indefinite matrix. Let T_0 be such that we can write T_0 as $T_0 = I - DA$ with $D = X^{-1} + P' {A_c}^{-1} P - P' {A_c}^{-1} P X^{-1}$. Let $\Gamma = A^H D^H + DA - A^H D^H DA$. If $\tilde{\Gamma} = A^H \tilde{D}^H + \tilde{D}A - A^H \tilde{D}^H \tilde{D}A$ with $\tilde{D} = X^{-1} + P' {A_c}^{-1} P$, is positive definite and $\|\tilde{\Gamma}\| \leqslant \|\Gamma\|$ together with $\kappa(\Gamma) \leqslant \kappa(\tilde{\Gamma})$, then

$$\|{T_0}^H T_0\|_2 < \sqrt{\left|1 - \frac{\|\tilde{\Gamma}\|}{\kappa(\tilde{\Gamma})}\right|} < 1.$$

Using Theorem 16 we can write ${T_0}^H T_0 = I - \Gamma$. Using Eq. (9.43) from Theorem 16 and substituting Eq. (9.62) from Corollary 16.3 have

$$\begin{aligned}
\|{T_0}^H T_0\|_2 &= \sup_{x\in\mathbb{C}^n, \|x\|_2=1} \|\left({T_0}^H T_0\right) x\|_2 = \sqrt{\rho\left({T_0}^H T_0\right)}, \\
&= \sqrt{|1-\min\lambda_j(\Gamma)|}, && (9.65)\\
&< \sqrt{|1-\min\lambda_j(\tilde{\Gamma})|}, && (9.66)\\
&< \sqrt{\left|1-\frac{\|\tilde{\Gamma}\|}{\kappa(\tilde{\Gamma})}\right|} < 1. && (9.67)
\end{aligned}$$

■

OPTIMAL CONVERGENCE FOR HELMHOLTZ

From Section 9.4, we have observed that we need a clever combination of coarsening using the CSL, higher-order interpolation schemes and the right amount of smoothing steps in order to obtain HPD systems. The results from Table 9.1, however, indicate that convergence will be slow as the spectral radius is still close to one. We can use Corollary 16.3 to obtain some insights into how we can optimize the convergence. In this work we focus on the remaining parameter we can work with; the relaxation parameter ω in the $\omega-$Jacobi smoother. We let ω vary and report the value of Eq. (9.62) from Corollary 16.2 in Table 9.3. Note that smaller these number, the closer to one the two-grid spectral radius will be, given Eq. (9.67).

	$\omega=1.5$		$\omega=2$		$\omega=2.5$		$\omega=4.5$		$\omega=7$	
k	$\nu=1$	$\nu=2$	$\nu=1$	$\nu=2$	$\nu=1$	$\nu=2$	$\nu=1$	$\nu=2$	$\nu=1$	$\nu=2$
5	0.250	0.100	0.250	0.251	0.200	0.242	0.083	0.200	0.001	0.125
10	0.142	0.071	0.125	0.142	0.111	0.142	0.023	0.111	0.001	0.038
20	0.030	0.030	0.030	0.029	0.023	0.024	0.007	0.024	0.001	0.016
30	0.009	0.006	0.008	0.008	0.007	0.008	0.001	0.007	0.001	0.003

Table 9.3: We report the value of $\|\tilde{\Gamma}\|\kappa(\tilde{\Gamma})^{-1}$ in the $p=1$ norm, with $\tilde{\Gamma}$ from Corollary 16.2. ν denotes the number of ω-Jacobi smoothing steps.

The results from Table 9.3 reveal that a lower ω is in favor when we use one post-smoothing steps. For $\omega = 1.5$, we observe that increasing k and ν leads to an decrease in $\|\tilde{\Gamma}\|\kappa(\tilde{\Gamma})^{-1}$. This means that as we perform more smoothing steps, the two-grid spectral radius moves closer and closer to one. For faster convergence, we require the opposite: the addition of a few more smoothing steps should lead to an increase rather than a decrease in $\|\tilde{\Gamma}\|\kappa(\tilde{\Gamma})^{-1}$. Based on these results, the optimal estimate for ω lies between $[2, 4.5]$ for the reported values of k. Moreover, increasing the number of post-smoothing steps can help repair the positive definiteness of $\tilde{\Gamma}$ and therefore Γ. While in general the application of the $\omega-$Jacobi smoother diverges for highly indefinite Helmholtz problems, choosing the right ω combined with a higher-order interpolation scheme and alternative coarsening strategies can lead to a simple yet convergent two-grid scheme.

9.5. NUMERICAL EXPERIMENTS

In this section we examine the convergence behavior of the multigrid solver. We use the model problems from Section 9.3. Unless stated otherwise, we use 10 grid points per wavelength, which is equivalent to $kh = 0.625$. All experiments are implemented sequentially on a Dell laptop using 8GB RAM and a i7-8665U processor. In all experiments, we set the relative tolerance to 10^{-5}. We allow for coarsening until the dimension of the underlying linear system N_c is smaller than 10. The maximum size of the linear system on the coarsest grid is therefore 10×10. We use the $\omega-$Jacobi smoother with the approximate optimal relaxation parameter from Section 9.4.2.1, i.e. $\omega = 4.5$ as we want to test for large k. Moreover, we perform the coarse-grid corrections using the higher-order prolongation and restriction operator, together with the CSL. For the latter, the complex shift is set at $\beta_2 = 0.7$ unless stated otherwise.

9.5.1. 2D CONSTANT k

For the constant wavenumber problem (MP 2-A), we first start by confirming that we have $h-$independent convergence. Recall that the constant from Theorem 16 does not depend on h, which for the proposed setup should lead to $h-$independent convergence. Thus, while the convergence bounds can be shown to be independent of h, the practical convergence speed may still depend on h.

$h-$INDEPENDENCE

We report the results for various k using $h = 2^{-p}$, with $p = 5$ up to $p = 9$. We observe that

	$k=15$			$k=30$		
h	$\nu=1$	$\nu=2$	$\nu=4$	$\nu=1$	$\nu=2$	$\nu=4$
2^{-5}	45	24	18	$\odot$	$\odot$	$\odot$
2^{-6}	34	22	18	66	37	28
2^{-7}	36	22	18	52	33	27
2^{-8}	40	24	18	54	34	27
2^{-9}	42	23	18	58	36	27

Table 9.4: Number of V-cycles for $k = 15$ and $k = 30$. ν denotes the number of ω-Jacobi smoothing steps using $\omega = 4.5$ $\odot$ denotes the case where $kh > 0.625$, which is excluded.

the solver indeed performs independently of h. In fact, it appears that with 4 smoothing steps, the solver converges in about $\tilde{C}k$ iterations, where $\tilde{C} \approx 1$ is a constant. The latter holds irrespective of the problem size.

GENERAL RESULTS

Now that we have established, both theoretically and numerically, that the convergence is $h-$independent once we take 4 smoothing steps, we revert back to letting $kh = 0.625$. Note that in order to obtain improved accuracy, kh can be decreased without affecting the convergence behavior due to the $h-$independence. In Table 9.5 we report the results for MP 2-A using a constant wavenumber and Sommerfeld boundary conditions. From Table 9.5 we observe that for the V-cycle the convergence improves if we use more smoothing steps. Note that these additional smoothing steps are computationally cheap, as we use the $\omega-$Jacobi smoother. In general, we observe that we again need approximately $\tilde{C}k$

Table 9.5: Number of V- ($\gamma = 1$) and W-cycles ($\gamma = 2$) for constant k (MP 2-A) using tol. 10^{-5}. ν denotes the number of ω-Jacobi smoothing steps. N_D is the size of the coarsest system.

	$k = 50$		$k = 100$		$k = 150$		$k = 200$		$k = 250$	
	$N = 6724$		$N = 26244$		$N = 57600$		$N = 102400$		$N = 160000$	
	$N_D = 8$		$N_D = 8$		$N_D = 4$		$N_D = 8$		$N_D = 4$	
γ	1	2	1	2	1	2	1	2	1	2
$\nu = 4$	58	58	104	108	155	159	209	213	267	271
$\nu = 5$	58	58	104	104	150	166	194	229	238	287
$\nu = 6$	55	58	99	102	139	167	183	222	226	283
$\nu = 7$	53	60	97	101	136	163	179	219	221	280
$\nu = 8$	53	60	95	104	131	161	178	212	218	277

iterations with $\tilde{C} \approx 1$, in order to reach convergence. While the convergence behavior is promising, the results do provide some further insights into the behavior of the smoother. For example, we observe that moving from a V-cycle to a W-cycle does not seem to improve the performance. In fact, we observe that we need more iterations instead of less. One potential explanation could be that we know that in general the $\omega-$Jacobi smoother diverges for Helmholtz-type of problems. While this in itself does not have to lead to divergence (the smoothing property is still satisfied for a divergent smoother, see [36]), the use of the W-cycle requires more smoothing iterations than the V-cycle. At some point we expect the efficiency of these smoothing steps to reduce, which could be reflected in the higher number of iterations.

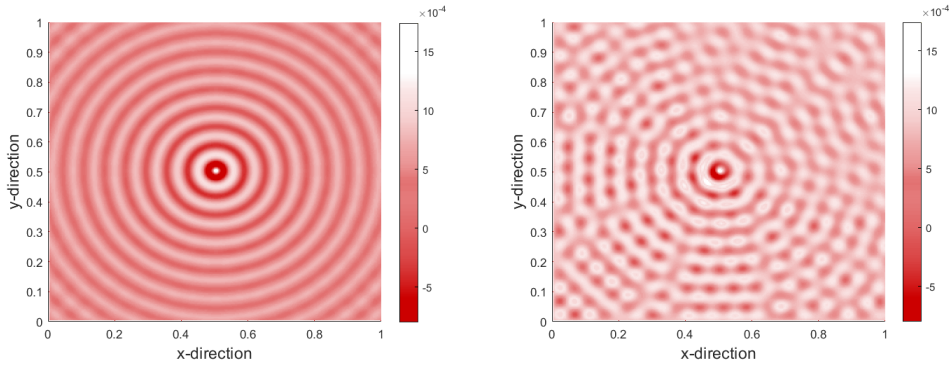


Figure 9.1: Real part of the 2D solution for MP2-A $k = 100$ (left) and $k = 150$ (right)

9.5.2. 2D RESULTS NON-CONSTANT $k(x, y)$

In this subsection we report on the numerical results for the non-constant wavenumber model problem. We distinguish two cases; a medium-varying and a highly-varying wavenumber profile problem. In this respect the profile contains either smooth or sharp changes between a range of wavenumbers. For an illustration of the wavenumber profile, see Fig. 9.2 for the smooth changing wavenumber profile and Fig. 9.3 for the profile where the wavenumber changes sharply. In all cases we make sure that the largest value of $k(x, y)$ still obeys the

rule of thumb $kh \approx 0.625$.

SMOOTH CHANGES

Fig. 9.2 illustrates how the wavenumber varies between 10 and 75 for the medium-varying problem. Note that the transition from a low to high wavenumber goes gradually.

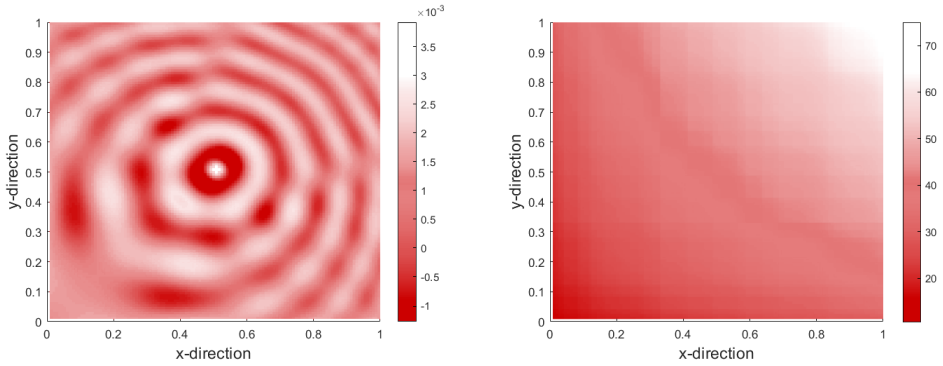


Figure 9.2: Left: real part of the 2D numerical solution. Right: $k(x,y)$

Table 9.6 contains the number of V-cycles needed to reach convergence for $k(x,y)$ varying between $(10,50)$ and $(10,75)$ respectively. We again observe that the number of iterations scales linearly with k. The iteration count follows approximately $\tilde{C}k$, with $\tilde{C} \approx 1$, where k in this case denotes the largest wavenumber. In this event, the W-cycle does lead to a smaller number of iterations, compared to the case where we have a constant wavenumber.

Table 9.6: Number of V- $(\gamma = 1)$ and W-cycles $(\gamma = 2)$ for MP 2-B (medium variation). ν denotes the number of ω-Jacobi smoothing steps.

	$(k_1,k_2) = (10,50)$		$(k_1,k_2) = (10,75)$	
γ	1	2	1	2
$\nu = 4$	65	60	90	88
$\nu = 5$	62	59	86	86
$\nu = 6$	61	58	85	85
$\nu = 7$	60	57	84	84
$\nu = 8$	59	57	83	83

SHARP CHANGES

In this subsection we report on the results for the highly-varying profile model problem, containing sharp changes between the wavenumbers $k = 10$ and $k = 75$. Fig. 9.3 illustrates how the wavenumber varies between 10 and 75 for the medium-varying problem. In this case, we let $k(x,y)$ vary randomly, which gives a wavenumber which varies highly across the entire numerical domain.

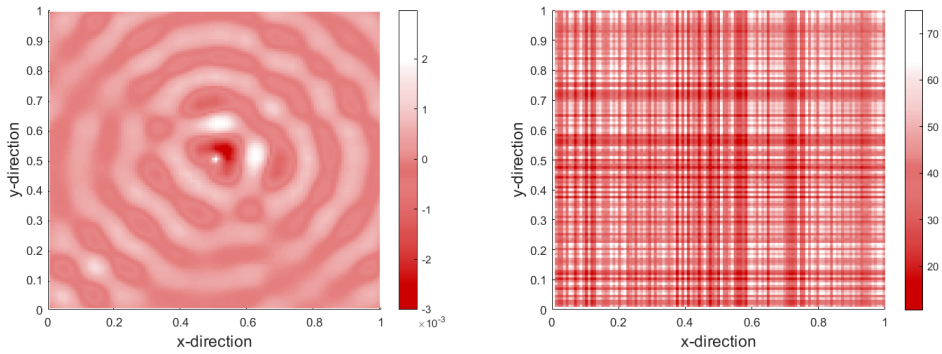


Figure 9.3: Left: real part of the 2D numerical solution. Right: $k(x,y)$

Table 9.7 contains the number of V-cycles needed to reach convergence for $k(x,y)$ varying between $(10,50)$ and $(10,75)$ respectively. We immediately observe that the number of iterations goes up by a factor of approximately 1.5 once we allow for the sharp and rapid changes in the wavenumber. For example for the medium-varying case containing more smooth transition between the wavenumbers, using 8 smoothing steps, we had 59 and 57 iterations respectively for the V- and W-cycle. In this case, we have 94 iterations for both the V- and W-cycle. Another interesting observation is that for the W-cycle, $k(x,y)$ varying between 10 and 75 leads to a larger number of iterations when $\nu > 4$. This could again be an indication of the smoother losing some of its efficacy when we allow for more smoothing operations using the W-cycle. Also for this model problem, we conclude that as the wavenumber grows, the V-cycle is preferred over the W-cycle for this choice of smoothing scheme. In the reported cases, the method converges in approximately $1.5k$ iterations, where k is the largest admissible wavenumber.

	$(k_1,k_2)=(10,50)$		$(k_1,k_2)=(10,75)$	
γ	1	2	1	2
$\nu=4$	102	96	111	107
$\nu=5$	97	95	103	105
$\nu=6$	95	95	101	104
$\nu=7$	94	94	102	104
$\nu=8$	94	94	102	104

Table 9.7: Number of V- $(\gamma=1)$ and W-cycles $(\gamma=2)$ for MP 2-B (high variation). ν denotes the number of ω-Jacobi smoothing steps.

9.5.3. GMRES-SMOOTHING

In this section we explore how the solver performs when we use GMRES(3) smoothing. In [136], where the authors propose coarsening on the CSL instead of the original Helmholtz operator, GMRES(3) is used as the only smoother. The motivation for this lies in the fact that indeed the $\omega-$Jacobi diverges when combined with the original and standard configuration of the multigrid method. By using a few GMRES iterations, the smoother acts as a polyno-

mial smoother. Already in [110], the use of GMRES(3) was proposed as a smoother. There, the authors manually optimized the smoothing schedule, where the number of smoothing steps ranges between 13 and 39 smoothing steps for a constant k ranging from 4π to 32π. More generally in the literature, on each level where the Jacobi smoother becomes unstable, the smoother is replaced by the Krylov iterations ranging between 5-40 iterations per smoothing step [110, 114, 115]. In this work we let the number of smoothing steps using GMRES(3) vary between 1 and 5. One advantage of this approach is that by allowing for only 3 GMRES iterations per smoothing step, the cost of applying GMRES does not increase significantly with respect to memory and computation time. We perform the same numerical experiments using the model problems mentioned previously.

2D RESULTS CONSTANT k

In this subsection we report the results for the constant wavenumber model problem (MP 2-A). As we are coarsening using the CSL, we want to distinguish between the cases where the complex shift is large versus small. We therefore start by keeping the shift $\beta_2 = 0.7$, similar to the one used in all previous experiments. Results are reported in Table 9.8 for 1 up to 5 smoothing steps. We immediately observe that the number of iterations for both the V- and W-cycle are drastically reduced. If we compare these results to the ones obtained when using the $\omega-$Jacobi smoother in Table 9.5, the reduction in the number of iterations is approximately 2.5 times. For example, when using 5 smoothing steps within a V-cycle, the $\omega-$Jacobi smoother requires 238 iterations to reach convergence. When GMRES(3) is used as a smoother with a similar amount of smoothing steps, 88 iterations are required to reach convergence. Note that there appears to be no difference in the number of iterations when using the V-cycle and W-cycle.

Table 9.8: Number of V- ($\gamma = 1$) and W-cycles ($\gamma = 2$) for constant k (MP 2-A) using tol. 10^{-5}. ν denotes the number of GMRES(3) smoothing steps with $\beta_2 = 0.7$

	$k=50$		$k=100$		$k=150$		$k=200$		$k=250$	
	$N=6724$		$N=26244$		$N=57600$		$N=102400$		$N=160000$	
	$N_D=8$		$N_D=8$		$N_D=4$		$N_D=8$		$N_D=4$	
γ	1	2	1	2	1	2	1	2	1	2
$\nu=1$	37	36	68	67	99	98	132	131	162	161
$\nu=2$	29	29	53	53	78	78	104	104	128	128
$\nu=3$	24	24	45	45	67	67	89	89	112	112
$\nu=4$	22	22	40	40	59	59	78	78	98	98
$\nu=5$	20	20	36	36	53	53	71	71	88	88

Next, we repeat the same numerical experiments, however this time we set the complex shift to $\beta_2 = k^{-1}$. Note that this is a very small shift, in fact the CSL matrix starts being a close resemblance of the original Helmholtz operator. Results are reported in Table 9.9. For all cases, we report that there is a drastic improvement in the number of iterations. We obtain a solver which is close to $k-$ independent when we use $\nu > 2$ and $\gamma = 2$. Unlike the previous case where $\beta_2 = 0.7$ in Table 9.8, we now do observe a lower number of iterations for the W-cycle.

In fact, additional numerical experiments confirm that the results are similar if we let $\beta_2 = 0$, which results in the original Helmholtz operator. Thus, when using GMRES(3) as a smoother,

the original Helmholtz operator can in fact be used instead of the CSL for coarsening within the multigrid hierarchy. However, the fast convergence is only observed when combined with the high-order prolongation and restriction operators.
Given that we obtained the best numerical results with $\beta_2 = k^{-1}$ without any additional costs, we continue with this shift for the CSL in the upcoming sections.

	$k=50$		$k=100$		$k=150$		$k=200$		$k=250$	
	$N=6724$		$N=26244$		$N=57600$		$N=102400$		$N=160000$	
	$N_D=8$		$N_D=8$		$N_D=4$		$N_D=8$		$N_D=4$	
γ	1	2	1	2	1	2	1	2	1	2
$\nu=1$	14	7	24	10	39	19	51	24	64	29
$\nu=2$	8	5	13	7	22	10	28	13	34	16
$\nu=3$	6	5	10	6	16	9	20	10	24	12
$\nu=4$	6	5	8	5	12	7	15	9	18	10
$\nu=5$	5	5	7	5	11	7	13	8	15	9

Table 9.9: Number of V- ($\gamma=1$) and W-cycles ($\gamma=2$) for constant k (MP 2-A) using tol. 10^{-5}. ν denotes the number of GMRES(3) smoothing steps with $\beta_2 = k^{-1}$.

2D RESULTS NON-CONSTANT $k(x,y)$

Finally, we investigate the convergence behavior using the GMRES(3) smoother on the highly-varying problem (MP 2-B2). In Section 9.5.2, we observed that this was the hardest problem to solve in terms of heterogeneous problem, due to the wavenumber varying highly between $k=10$ and $k=75$ across the numerical domain. Thus, we only test for this case in this subsection.

SHARP CHANGES

Results are reported in Table 9.10 and again indicate a drastic improvement compared to the case where we use the $\omega-$Jacobi smoother. In fact, even for this highly-varying model containing sharp changes between the wavenumbers $k=10$ and $k=75$, where the wavenumber is allowed to vary randomly between 10 and 75 across the domain, we obtain a $k-$ and $h-$independent multigrid solver for $\nu \geqslant 3$. Similar to the previous case, even when using the original Helmholtz operator, we reach convergence when using GMRES(3) as a smoother. However, the $k-$independence is only observed when the higher-order prolongation and restriction scheme is used. These results are promising and provide a solid framework for future research, given that the current industry standard (CSL preconditioner with multigrid inversion) works well for homogeneous problems, but is less suitable for heterogeneous problems with sharp changes between wavenumbers due to the instability of the inexact inversion using geometric multigrid [27, 29].

	$(k_1,k_2)=(10,50)$		$(k_1,k_2)=(10,75)$	
γ	1	2	1	2
$\nu=1$	28	12	31	12
$\nu=2$	16	8	17	7
$\nu=3$	12	7	12	6
$\nu=4$	10	6	10	6
$\nu=5$	9	6	9	6

Table 9.10: Number of V- ($\gamma=1$) and W-cycles ($\gamma=2$) for MP 2-B (high variation). ν denotes the number of GMRES(3) smoothing steps and $\beta_2 = {k_{\max}}^{-1}$.

9.6. CONCLUSION

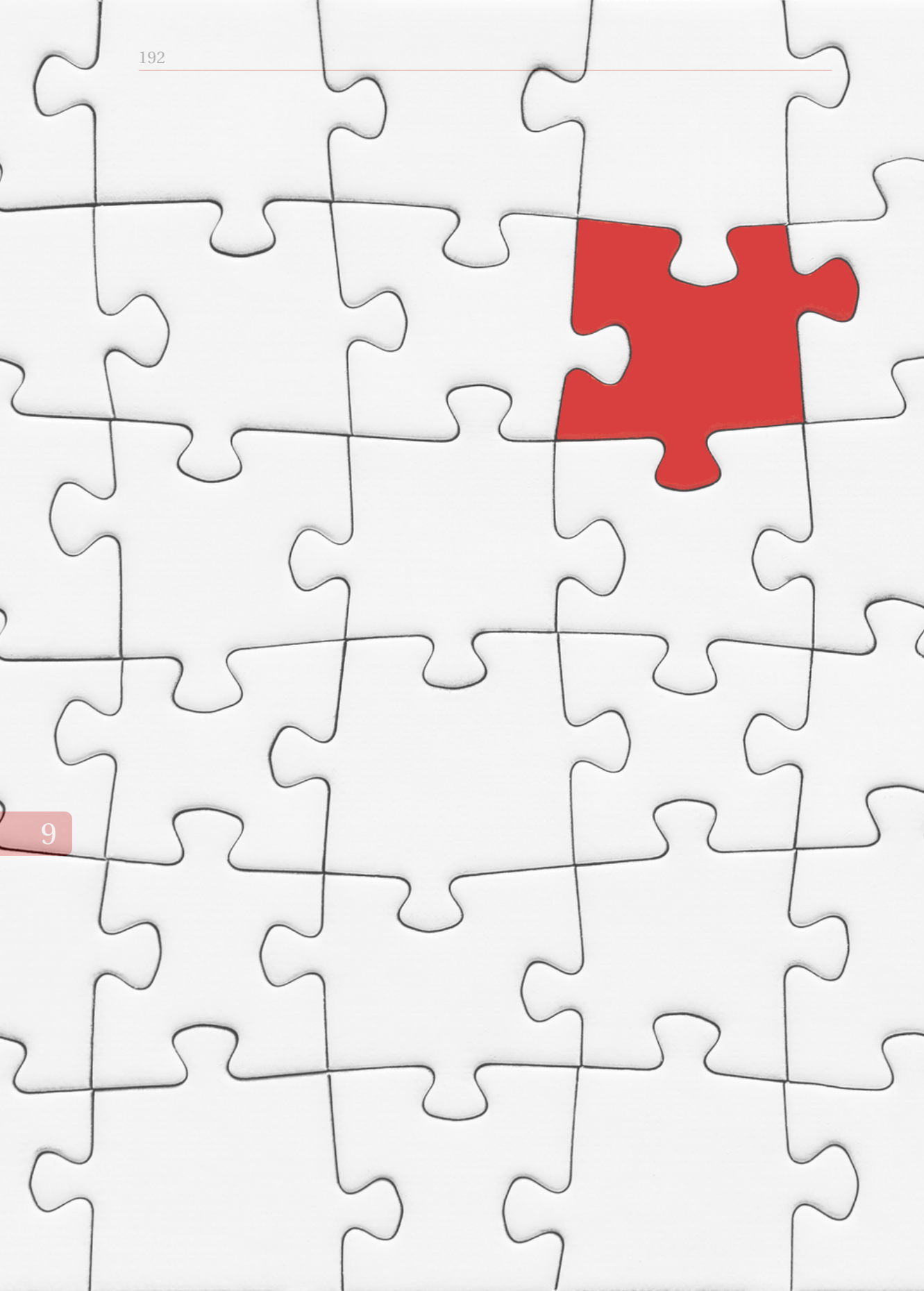
In this chapter, we developed a novel stand-alone multigrid solver for the indefinite Helmholtz equation using standard-components, such as the weighted Jacobi smoother. The resulting algorithm additionally shows $h-$independent convergence and thus adheres to the classic multigrid features. Two novel and striking features should be mentioned. First of all, no restriction is imposed on the number of grid points on the coarsest grid. As a result, we construct a full multigrid hierarchy for both the V- and W-cycles. Second of all, no level-dependent parameters are needed.

Apart from the numerical results, we provide a new theory to assess convergence of highly indefinite linear systems. Where most proofs require some adjointness and/or symmetry assumptions on the underlying linear system, we have constructed theoretical notions which do not require such assumptions. In fact, we have found that the addition of a complex shift, solely for the purpose of coarsening in the multigrid hierarchy, combined with higher-order interpolation schemes for the inter-grid transfer operators leads to a Hermitian positive definite (HPD) system. The positive definiteness can be verified for medium-sized wavenumbers by assessing properties of the matrix. As a result, it can be shown that the convergence is independent of h. Thus, the computation of the spectral radius can be circumvented, especially since the analytical eigenvalues can only be computed in case of Dirichlet boundary conditions. We also study the behavior of the solver once we use GMRES(3) as a smoother instead of $\omega-$Jacobi. The convergence significantly improves and less smoothing steps are needed. Second of all, if we use GMRES(3) as a smoother, we can keep the complex shift very small without paying a penalty in terms of additional costs. As a result, we obtain close to $k-$ and $h-$independent convergence, both for the constant wavenumber model problem and the non-constant wavenumber model problem. In fact, we can even use the original Helmholtz operator for coarsening in this configuration without losing efficiency and effectiveness as the GMRES iterations provide polynomial smoothing.

This work is one of the few to demonstrate, both theoretically and numerically, the existence and potential of multigrid methods for such linear systems. However, a lot of future work remains to be done. For example, future research could focus on exploring different combinations of interpolation schemes and smoothers and extending proofs. We have shown convergence for $\omega-$Jacobi, mainly because of its simplicity and parallelizability, but many other options remain to be examined.

IV

Conclusions



10

Findings and Discussion

This dissertation prolongs a long tradition of research in trying to find fast and robust numerical methods for the Helmholtz equation. A fast and robust numerical solver should lead to accurate solutions while being scalable in terms of the computational complexity and the number of iterations to reach convergence. To study this both numerical and theoretically, we constructed three research pillars: wavenumber independent convergence, linear complexity and accuracy. We outline our findings and results with respect to these pillars below.

WAVENUMBER INDEPENDENT CONVERGENCE

The building block of the results related to convergence of the solver can be retraced to the development of the two-level deflation preconditioner in Chapter 7. We theoretically show if the near-zero eigenvalues of the indefinite fine-grid and coarse-grid operator at the second level are misaligned, the number of iterations to reach convergence increases. This effect can be attributed to the interpolation scheme not being able to sufficiently approximate the transferring of the oscillatory grid functions at very large wavenumbers.

We can analytically measure the effect of these errors on the construction of the projection preconditioner by means of the projection error. The quality of the deflation vectors determine whether the projection error dominates. To minimize the projection error, we propose the implementation of a higher order approximation scheme to construct the deflation vectors. Here we used quadratic rational Bezier polynomials, as these provide us with control points and respective weights to cater to our accuracy requirements.

The direct improvement of the preconditioned spectrum is visible in terms of clustering and the absence of outliers and near-zero eigenvalues. Numerical experiments corroborate these findings as the number of iterations to reach convergence for one-, two- and three-dimensional constant wavenumber model problems are now more or less wavenumber independent. Similar results are obtained for heterogeneous problems.

The GMRES-based solver benefits significantly from having a fixed or close to wavenumber independent number of iterations to reach convergence given that the computational cost in terms of matrix-vector products and memory increases with each additional iteration. However, several remarks are in place.

First of all, using a higher-order scheme to construct the deflation vectors results in a less sparse coarse-grid matrix at the second level. The coarse-grid operator thus contains more non-zero elements compared to the fine-grid operator. The two-level deflation preconditioner requires the exact solution on the second level. While the problem size on the second level is significantly smaller, this can become a problem for large wavenumber problems in 3D. Consequently, for sequential implementations, the need for multilevel methods which balance the number of iterations to reach convergence and computational complexity is of paramount importance when trying to solve large 3D problems.

LINEAR COMPLEXITY

Extending the two-level deflation preconditioner to a multilevel preconditioner comes naturally once we aim to circumvent the memory bottleneck for large problem sizes. Similar

to the two-level method, we use higher-order deflation vectors and construct a V-shaped hierarchy in Chapter 8. We achieve this by recursively applying the two-level method. The higher-order deflation vectors are also constructed on the basis of quadratic rational Bezier polynomials.

An important theoretical finding in this chapter is that up to a certain level, the coefficient matrices within the hierarchy are indefinite. These levels are of paramount importance as the near-zero eigenvalues at these levels can effectively be removed by the multilevel deflation preconditioner. If the near-zero eigenvalues are aligned, then the eigenvalues cluster and near-zero eigenvalues are significantly minimized, accelerating the convergence. After this level, the subsequent coarse coefficient matrices become negative definite.

The algorithm itself uses FGMRES as the outer solver and within each level of the hierarchy adds either polynomial (on indefinite levels) or weighted-Jacobi 'smoothing' (on negative definite levels). We implement some inner Bi-CGSTAB-iterations as a polynomial smoother to further reduce the smooth components of the error, similar to multigrid methods.

Sequential implementation of the method leads to scalable timing results for higher dimensional model problems, which has been demonstrated using numerical experiments and a complexity analysis. Also here, several remarks are in place. The method uses polynomial inner smoothing by means of Bi-CGSTAB-iterations. For high wavenumbers and thus very oscillatory waves, an inexact solve at the coarser level creates an additional sensitivity with respect to the projected error. This can be counteracted by a few Bi-CGSTAB-iterations. The challenge lies in balancing the number of inner iterations needed to obtain satisfactory outer iterations, while keeping the overall complexity quasi-linear and the inner iterations as small as possible.

A similar remark as regards the density of the coarse-grid systems can be due to the use of the higher-order deflation schemes. The complexity study reveals that we quickly achieve a break-even point as regards the additional memory and computational cost once we apply the solver to 2D and 3D model problems with large wavenumbers.

Finally, the development of the multilevel deflation preconditioner lies very close to the construction of a multigrid hierarchy. While multigrid works well for the positive-definite variant of the Helmholtz equation or when the wavenumber is small, the method generally fails to converge as a stand-alone solver or requires too strict restrictions on the coarsest grid in order to remain feasible and attain linear complexity.

Using the same building blocks from the quadratic rational Bezier polynomials, we now construct the inter-grid transfer operators using this higher-order scheme. Note that this is the same as the deflation matrices used in deflation-based preconditioners. Apart from the numerical results, we discuss a new theory to assess convergence of highly indefinite linear systems. Where most proofs require some adjointness and/or symmetry assumptions on the underlying linear system, we have constructed theoretical notions which do not require such assumptions. In fact, we find that the addition of a complex shift, solely for the purpose of coarsening in the multigrid hierarchy, combined with higher-order interpolation

schemes for the inter-grid transfer operators leads to a Hermitian positive definite (HPD) system. Thus, the computation of the spectral radius can be circumvented.

We study the behavior using two smoothers: weighted-Jacobi and polynomial smoothing using GMRES(3). Numerical experiments on 2D constant and non-constant wavenumber problems show that while the method converges, the number of iterations are not wavenumber independent. Once we allow for polynomial smoothing, the convergence improves significantly and less smoothing steps are needed. This polynomial smoothing is reminiscent of the Bi-CGSTAB-iterations in the multilevel deflation method.

This work is one of the few to demonstrate, both theoretically and numerically, the existence and potential of multigrid methods for such linear systems. Some remarks are in place. The method so far shows a sensitivity to the use of Dirichlet boundary conditions on a part of the boundary, which requires much more iterations to reach convergence compared to the case of Sommerfeld boundary conditions. Also, while the multigrid solver finally converges and adheres to the linear complexity requirement, the number of iterations are still large and we lose the sense of wavenumber independent convergence. This can be remedied by using polynomial smoothers, but can come at a future cost when trying to construct a parallel version of the solver. At last, more theory needs to be developed in order to better understand why the wavenumber independent convergence is lost in the first place.

ACCURACY

Our last research pillar deals with the accuracy of the numerical solutions. In the absence of any numerical errors, the waves modelled by the Helmholtz equation will propagate without any dissipation or dispersion. However, shifting from the continuous problem to its discrete counterpart, gives rise to the pollution error, which can not be removed in 2D and 3D and grows with the wavenumber.

Chapter 5 in this dissertation provides the first theoretical basis for defining the pollution error in terms of the eigenvalues. By examining the behavior of the eigenvalues, we propose an upper bound for the relative error and show that if the near-zero eigenvalues and eigenvectors are approximated with high accuracy, then the dispersion part of the pollution error can be minimized considerably. The results also illustrate that the error grows in an oscillatory manner, and the error bound is able to capture and reveal this effect.

We additionally study a theoretical framework where the pollution error can be brought to approximately zero for very large wavenumbers, irrespective of the grid resolution. The basis of this approach lies in correcting the respective eigenvalues with the remainder, which depends on the order of the truncation error of the finite difference scheme. Consequently, it is possible to obtain pollution-free and therefore accurate one- and 2D solutions using coarser grids. The solutions obtained account for all propagation angles simultaneously and do not rely on pre-determined angles for plane-wave propagation and error correction, which promotes a detailed study of the pollution effect in higher dimensions.

These theoretical results are primarily useful in trying to obtain an in depth understanding of the pollution error in higher dimensions. For more practical results, we study the

use of IgA discretized linear systems for the Helmholtz equation in Chapter 6. In particular, we show that the use of IgA reduces the pollution error significantly compared to $p-$order FEM. However, the pollution error can not be removed completely and continues to grow with the wavenumber k, unless more degrees of freedom are used. Additionally, obtaining better accuracy by increasing the order p comes at the cost of more dense matrices. Depending on the application and the required level of accuracy, IgA can provide more accurate solutions using smaller linear systems compared to $p-$order FEM.

10

11

Outlook

In this chapter we are going to discuss the outlook and future work related to the content of this dissertation. We again distinguish between the three research pillars and also briefly provide an additional section discussing novel opportunities and applications.

WAVENUMBER INDEPENDENT CONVERGENCE

In the previous chapter we mentioned that the two-level deflation solver can run into memory bottlenecks for large 3D problems due to the more dense coarse-grid linear system at the second level which needs to be solved exactly. One important step forward would be to investigate parallelization techniques. In case we can parallelize the two-level deflation preconditioner efficiently, we can benefit from the wavenumber independent convergence while moving towards better time complexity. Another challenging research aspect would be to investigate whether we can implement a parallel version of the deflation preconditioner in a matrix-free way. Here, the use of different Krylov subspace methods, such as IDR(s), could also be explored.

LINEAR COMPLEXITY

Entering the high-performance computing realm, will allow for more large scale testing of 3D model problems. It would be interesting to observe whether the multilevel deflation and multigrid methods can be parallelized efficiently and where the most computational gains can be realized.

As for the multilevel deflation method, some research can be dedicated to developing more theory in order to understand how the remaining error behaves once subjected to the polynomial smoothing using the inner BICG-stab iterations.

Regarding the multigrid solver, apart from considering it as a stand-alone solver, it would be interesting to assess its application in the classical sense of being used as a preconditioner. In fact, in practice the CSL preconditioner is always inverted inexactly using one or two multigrid iterations. If the use of the novel multigrid scheme could be used in a similar fashion, however at the cost of less iterations than this could potentially provide a new way of applying multigrid as a preconditioner for highly indefinite operators.

When it comes to perpetuating its use as a stand-alone solver, a lot of theory remains to be developed and results to be analyzed. As this is one of the first cases where convergence is reached for highly indefinite 2D constant and non-constant wavenumber problems, the method is still in its infancy. For example, more theory could be developed related to how large the optimal shift should be for coarsening purposes. Similarly, the inclusion of different smoothers remains to be investigated as well as testing the solver on 3D model problems.

11

ACCURACY

Apart from obtaining faster solutions, we always aim to obtain accurate solutions as well. In relation to this dissertation, several topics remain to be explored. For example, the theory from the pollution error in higher-dimensions could be extended to include the 3D case. Here, the Green's function for the model problem should be derived analytically and the proofs have to be extended to obtain similar explicit bounds in terms of the eigenvalues.

The application of IgA to the Helmholtz equation and wave problems in general is quite novel and various directions remain to be explored as well. One area of investigation could be how the pollution error responds to local or adaptive refinement techniques. It would also be important to test for more industrial model problems. Both $h-$ and $p-$refinement methods could be studied and related to the pollution error. Here, some results from our work on the theoretical properties of the pollution error could be combined, where we applied a similar local $p-$refinement correction through the eigenvalues which significantly reduced the pollution error while keeping the size of the linear systems fixed.

In the extended version of our IgA work, we furthermore observed that using IgA for the Helmholtz equation goes well with the two-level deflation based solver compared to the industry standard CSL preconditioner. Combining these future refinement techniques and balancing the sparsity of the linear system coefficient matrices, could provide an all-round numerical solver tailored for wave propagation problems respecting both the accuracy and scalability requirements, without necessarily increasing the problem sizes.

Additional Applications and Methods

Many methods developed in this section can also be applied to the time-harmonic Maxwell equation. This remains to be investigated and could provide additional insights into the behavior of the solver in case of non-scalar equations. Preliminary results using the time-harmonic elastic wave equation have showed similar convergence properties. However, more interesting would be to allow these numerical methods to be studied in conjunction with time-stepping methods to explore the time-dependent variants of the PDEs studied in this dissertation.

The extension to these time-dependent variants will allow for broader reception in industrial applications. For example, a lot of interest has been shown for the CSL preconditioner. However, it still remains unclear how to apply this to the Maxwell equation and what robustness guarantees can be deduced.

Finally, the deflation preconditioner and domain decomposition based preconditioners used in parallel solvers contain a lot of similarities. In the construction of these domain decomposition preconditioners, a coarse space is also added to obtain better scalability with the number of sub-domains and the number of iterations to reach convergence. The quadratic rational Bezier polynomials, similar to the ones used to construct the deflation and multigrid inter-grid transfer vectors, could be used to construct these coarse spaces within the domain decomposition preconditioner. Its influence on the convergence and the scalability with respect to the wavenumber and number of sub-domains could then become another potential topic for future research.

REFERENCES

[1] F.-C. Lin and M. H. Ritzwoller, *Helmholtz surface wave tomography for isotropic and azimuthally anisotropic structure,* Geophysical Journal International **186**, 1104 (2011).

[2] A. Gaul, *Recycling Krylov subspace methods for sequences of linear systems: analysis and applications,* (2014).

[3] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem,* Quarterly of Applied Mathematics **9**, 17 (1951).

[4] C. Lanczos, *Solution of systems of linear equations by minimized iterations,* J. Res. Nat. Bur. Standards **49**, 33 (1952).

[5] C. Vuik and D. Lahaye, *Scientific computing (wi4201),* Lecture notes for wi4201 (2012).

[6] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, Vol. 66 (SIAM, 2011).

[7] J. Liesen and Z. Strakos, *Convergence of GMRES for tridiagonal Toeplitz matrices,* SIAM Journal on Matrix Analysis and Applications **26**, 233 (2004).

[8] M. Eiermann and O. G. Ernst, *Geometric aspects of the theory of Krylov subspace methods,* Acta Numerica 2001 **10**, 251 (2001).

[9] A. Hannukainen, *Convergence analysis of GMRES for the Helmholtz equation via pseudospectrum,* arXiv preprint arXiv:1505.08072 (2015).

[10] G. Meurant and J. D. Tebbens, *The role eigenvalues play in forming GMRES residual norms with non-normal matrices,* Numerical Algorithms **68**, 143 (2015).

[11] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors,* SIAM Journal on Matrix Analysis and Applications **16**, 1154 (1995).

[12] P. Sonneveld and M. B. Van Gijzen, *Idr (s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations,* SIAM Journal on Scientific Computing **31**, 1035 (2009).

[13] M. Baumann and M. B. Van Gijzen, *Nested Krylov methods for shifted linear systems,* SIAM Journal on Scientific Computing **37**, S90 (2015).

[14] R. Astudillo and M. B. van Gijzen, *Induced dimension reduction method for solving linear matrix equations,* Procedia Computer Science **80**, 222 (2016).

[15] T. P. Collignon and M. B. Van Gijzen, *Minimizing synchronization in IDR (s),* Numerical Linear Algebra with Applications **18**, 805 (2011).

[16] A. Sheikh, C. Vuik, and D. Lahaye, *Fast iterative solution methods for the Helmholtz equation*, Tech. Rep. (Delft University of Technology, Faculty of Electrical and Engineering, Mathematics and Computer Science, Delft Institute of Applied Mathematics, 2009).

[17] Y. A. Erlangga, *A robust and efficient iterative method for the numerical solution of the Helmholtz equation*, Ph.D. thesis (2005).

[18] M. J. Gander and F. Nataf, *Ailu: a preconditioner based on the analytic factorization of the elliptic operator,* Numerical linear algebra with applications **7**, 543 (2000).

[19] F. N. M. Gander, *An incomplete lu preconditioner for problems in acoustics,* Journal of Computational Acoustics **13**, 1 (2005).

[20] M. J. Gander and F. Nataf, *Ailu for Helmholtz problems: a new preconditioner based on the analytic parabolic factorization,* Journal of Computational Acoustics **9**, 1499 (2001).

[21] A. Bayliss, C. I. Goldstein, and E. Turkel, *An iterative method for the Helmholtz equation,* Journal of Computational Physics **49**, 443 (1983).

[22] A. L. Laird and M. Giles, *Preconditioned iterative solution of the 2D Helmholtz equation*, Tech. Rep. (2002).

[23] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems,* SIAM Journal on Scientific Computing **27**, 1471 (2006).

[24] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, *Comparison of multigrid and incomplete LU shifted-laplace preconditioners for the inhomogeneous Helmholtz equation,* Applied numerical mathematics **56**, 648 (2006).

[25] M. J. Gander, F. Magoules, and F. Nataf, *Optimized Schwarz methods without overlap for the Helmholtz equation,* SIAM Journal on Scientific Computing **24**, 38 (2002).

[26] N. Bootland, V. Dwarka, P. Jolivet, V. Dolean, and C. Vuik, *Inexact subdomain solves using deflated GMRES for Helmholtz problems,* arXiv preprint arXiv:2103.17081 (2021).

[27] Y. A. Erlangga and R. Nabben, *On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted laplacian,* Electronic Transactions on Numerical Analysis **31**, 3 (2008).

[28] A. Sheikh, *Development Of The Helmholtz Solver Based On A Shifted Laplace Preconditioner And A Multigrid Deflation Technique* (TU Delft, Delft University of Technology, 2014).

[29] A. Sheikh, D. Lahaye, L. G. Ramos, R. Nabben, and C. Vuik, *Accelerating the shifted laplace preconditioner for the Helmholtz equation by multilevel deflation,* Journal of Computational Physics **322**, 473 (2016).

[30] A. Sheikh, D. Lahaye, and C. Vuik, *On the convergence of shifted Laplace preconditioner combined with multilevel deflation,* Numerical Linear Algebra with Applications **20**, 645 (2013).

[31] Y. Erlangga, C. Vuik, and C. Oosterlee, *On a class of preconditioners for solving the discrete Helmholtz equation,* .

[32] M. J. Gander, I. G. Graham, and E. A. Spence, *Applying GMRES to the Helmholtz equation with shifted laplacian preconditioning: what is the largest shift for which wavenumber independent convergence is guaranteed?* Numerische Mathematik **131**, 567 (2015).

[33] P.-H. Cocquet and M. J. Gander, *How large a shift is needed in the shifted Helmholtz preconditioner for its effective inversion by multigrid?* SIAM Journal on Scientific Computing **39**, A438 (2017).

[34] M. van Gijzen, Y. Erlangga, and C. Vuik, *Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian,* SIAM Journal on Scientific Computing **29**, 1942 (2007).

[35] O. G. Ernst and M. J. Gander, *Multigrid methods for Helmholtz problems: A convergent scheme in 1d using standard components,* .

[36] W. Hackbusch, *Multi-grid methods and applications*, Vol. 4 (Springer Science & Business Media, 2013).

[37] V. Dwarka and C. Vuik, *Pollution and accuracy of solutions of the Helmholtz equation: A novel perspective from the eigenvalues,* Journal of Computational and Applied Mathematics **395**, 113549 (2021).

[38] A. Deraemaeker, I. Babuška, and P. Bouillard, *Dispersion and pollution of the fem solution for the Helmholtz equation in one, two and three dimensions,* International journal for numerical methods in engineering **46**, 471 (1999).

[39] I. M. Babuška and S. A. Sauter, *Is the pollution effect of the fem avoidable for the Helmholtz equation considering high wave numbers?* SIAM Journal on Numerical Analysis **34**, 2392 (1997).

[40] F. Ihlenburg and I. Babuska, *Finite element solution of the Helmholtz equation with high wave number part ii: the hp version of the fem,* SIAM Journal on Numerical Analysis **34**, 315 (1997).

[41] M. Ainsworth, *Discrete dispersion relation for hp-version finite element approximation at high wave number,* SIAM Journal on Numerical Analysis **42**, 553 (2004).

[42] E. Turkel, D. Gordon, R. Gordon, and S. Tsynkov, *Compact 2D and 3D sixth order schemes for the Helmholtz equation with variable wave number,* Journal of Computational Physics **232**, 272 (2013).

[43] K. Wang and Y. S. Wong, *Pollution-free finite difference schemes for non-homogeneous Helmholtz equation,* Int. J. Numer. Anal. Model **11**, 787 (2014).

[44] K. Gerdes and F. Ihlenburg, *On the pollution effect in fe solutions of the 3D-Helmholtz equation,* Computer Methods in Applied Mechanics and Engineering **170**, 155 (1999).

[45] K. Wang, Y. Wong, and J. Huang, *Analysis of pollution-free approaches for multi-dimensional helmholtz equations,* International Journal of Numerical Analysis and Modeling **16**, 412 (2019).

[46] K. Wang and Y. S. Wong, *Is pollution effect of finite difference schemes avoidable for multi-dimensional Helmholtz equations with high wave numbers?* Communications in Computational Physics **21**, 490 (2017).

[47] T. Wu, *A dispersion minimizing compact finite difference scheme for the 2d Helmholtz equation,* Journal of Computational and Applied Mathematics **311**, 497 (2017).

[48] L. L. Thompson and P. M. Pinsky, *Complex wavenumber fourier analysis of the p-version finite element method,* Computational Mechanics **13**, 255 (1994).

[49] J. Galkowski, E. H. Müller, and E. A. Spence, *Wavenumber-explicit analysis for the Helmholtz h-bem: error estimates and iteration counts for the Dirichlet problem,* Numerische Mathematik **142**, 329 (2019).

[50] Y. Du, H. Wu, and Z. Zhang, *Superconvergence analysis of linear fem based on polynomial preserving recovery for Helmholtz equation with high wave number,* Journal of Computational and Applied Mathematics **372**, 112731 (2020).

[51] P.-H. Cocquet, M. J. Gander, and X. Xiang, *Closed form dispersion corrections including a real shifted wavenumber for finite difference discretizations of 2D constant coefficient Helmholtz problems,* SIAM Journal on Scientific Computing **43**, A278 (2021).

[52] W. Read, *Analytical solutions for a Helmholtz equation with Dirichlet boundary conditions and arbitrary boundaries,* Mathematical and computer modelling **24**, 23 (1996).

[53] F. Ihlenburg and I. Babuška, *Dispersion analysis and error estimation of galerkin finite element methods for the Helmholtz equation,* International journal for numerical methods in engineering **38**, 3745 (1995).

[54] C.-H. Jo, C. Shin, and J. H. Suh, *An optimal 9-point, finite-difference, frequency-space, 2-d scalar wave extrapolator,* Geophysics **61**, 529 (1996).

[55] Z. Chen, D. Cheng, W. Feng, and T. Wu, *An optimal 9-point finite difference scheme for the Helmholtz equation with PML.* International Journal of Numerical Analysis & Modeling **10** (2013).

[56] Z. Chen, D. Cheng, and T. Wu, *A dispersion minimizing finite difference scheme and preconditioned solver for the 3D Helmholtz equation,* Journal of Computational Physics **231**, 8152 (2012).

[57] S. Britt, S. Tsynkov, and E. Turkel, *Numerical simulation of time-harmonic waves in inhomogeneous media using compact high order schemes,* Communications in Computational Physics **9**, 520 (2011).

[58] T. Wu and R. Xu, *An optimal compact sixth-order finite difference scheme for the Helmholtz equation,* Computers & Mathematics with Applications **75**, 2520 (2018).

[59] I. Singer and E. Turkel, *High-order finite difference methods for the Helmholtz equation,* Computer Methods in Applied Mechanics and Engineering **163**, 343 (1998).

[60] C. C. Stolk, *A dispersion minimizing scheme for the 3-d Helmholtz equation based on ray theory,* Journal of Computational Physics **314**, 618 (2016).

[61] P.-H. Cocquet, M. J. Gander, and X. Xiang, *A finite difference method with optimized dispersion correction for the Helmholtz equation,* in *International Conference on Domain Decomposition Methods* (Springer, 2017) pp. 205–213.

[62] F. Ihlenburg and I. Babuška, *Solution of Helmholtz problems by knowledge-based fem,* Computer Assisted Mechanics and Engineering Sciences **4**, 397 (1997).

[63] C.-Y. Chiang and M. M. Lin, *The eigenvalue shift technique and its eigenstructure analysis of a matrix,* Journal of Computational and Applied Mathematics **253**, 235 (2013).

[64] V. Dwarka, R. Tielen, M. Möller, and C. Vuik, *Towards accuracy and scalability: Combining isogeometric analysis with deflation to obtain scalable convergence for the Helmholtz equation,* Computer Methods in Applied Mechanics and Engineering **377**, 113694 (2021).

[65] T. Hughes, J. Cottrell, and Y. Bazilevs, *Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement,* Computer Methods in Applied Mechanics and Engineering **194**, 4135 (2005).

[66] J. Cottrell, T. Hughes, and Y. Bazilevs, *Isogeometric analysis: toward integration of CAD and FEA* (Wiley, 2009).

[67] C. De Boor, *A practical guide to splines* (Springer, 1978).

[68] T. Hughes, A. Reali, and G. Sangalli, *Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method nurbs,* Computer Methods in Applied Mechanics and Engineering **197**, 4104 (2007).

[69] A. Buffa, G. Sangalli, and R. Vázquez, *Isogeometric analysis in electromagnetics: B-splines approximation,* Computer Methods in Applied Mechanics and Engineering **199**, 1143 (2010).

[70] A. Buffa and R. Vázquez, *Isogeometric analysis for electromagnetic scattering problems,* in *2014 International Conference on Numerical Electromagnetic Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO)* (International Conference on Numerical Electromagnetic Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO). IEEE, 2014) pp. 1–3.

[71] H. Wu, W. Ye, and W. Jiang, *Isogeometric finite element analysis of interior acoustic problems,* Applied Acoustics **100**, 63 (2015).

[72] L. Coox, E. Deckers, D. Vandepitte, and W. Desmet, *A performance study of nurbs-based isogeometric analysis for interior two-dimensional time-harmonic acoustics,* Computer Methods in Applied Mechanics and Engineering **305**, 441 (2016).

[73] D. Drzisga, B. Keith, and B. Wohlmuth, *The surrogate matrix methodology: Accelerating isogeometric analysis of waves,* Computer Methods in Applied Mechanics and Engineering **372** (2020).

[74] V. H. Mederos, I. A. A. Ugalde, R. M. B. Alfonso, D. Lahaye, and V. G. Ones, *Isogeometric solution of Helmholtz equation with Dirichlet boundary condition: numerical experiences,* arXiv preprint: 2001.07795 (2020).

[75] G. C. Diwan and M. S. Mohamed, *Iterative solution of Helmholtz problem with high-order isogeometric analysis and finite element method at mid-range frequencies,* Computer Methods in Applied Mechanics and Engineering **363** (2020).

[76] S. Esterhazy and J. M. Melenk, *An analysis of discretizations of the Helmholtz equation in l^2 and in negative norms,* Computers & Mathematics with Applications **67**, 830 (2014).

[77] G. C. Diwan and M. S. Mohamed, *Pollution studies for high order isogeometric analysis and finite element for acoustic problems,* Computer Methods in Applied Mechanics and Engineering **350**, 701 (2019).

[78] V. Dwarka and C. Vuik, *Scalable convergence using two-level deflation preconditioning for the Helmholtz equation,* SIAM Journal on Scientific Computing **42**, A901 (2020).

[79] R. A. Nicolaides, *Deflation of conjugate gradients with applications to boundary value problems,* SIAM Journal on Numerical Analysis **24**, 355 (1987).

[80] R. B. Morgan, *GMRES with deflated restarting,* SIAM Journal on Scientific Computing **24**, 20 (2002).

[81] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga, *Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods,* Journal of scientific computing **39**, 340 (2009).

[82] R. Nabben and C. Vuik, *A comparison of deflation and the balancing preconditioner,* SIAM Journal on Scientific Computing **27**, 1742 (2006).

[83] Y. A. Erlangga, L. G. Ramos, and R. Nabben, *The multilevel Krylov-multigrid method for the Helmholtz equation preconditioned by the shifted laplacian,* in *Modern Solvers for Helmholtz Problems* (Springer, 2017) pp. 113–139.

[84] X. Liu, Y. Xi, Y. Saad, and M. V. de Hoop, *Solving the three-dimensional high-frequency Helmholtz equation using contour integration and polynomial preconditioning,* SIAM Journal on Matrix Analysis and Applications **41**, 58 (2020).

[85] L. Conen, V. Dolean, R. Krause, and F. Nataf, *A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator,* Journal of Computational and Applied Mathematics **271**, 83 (2014).

[86] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation* (Society for Industrial and Applied Mathematics, USA, 2015).

[87] I. G. Graham, E. A. Spence, and E. Vainikko, *Recent results on domain decomposition preconditioning for the high-frequency Helmholtz equation using absorption,* in *Modern solvers for Helmholtz problems* (Springer, 2017) pp. 3–26.

[88] I. Graham, E. Spence, and E. Vainikko, *Domain decomposition preconditioning for high-frequency Helmholtz problems with absorption,* Mathematics of Computation **86**, 2089 (2017).

[89] I. Graham, E. Spence, and J. Zou, *Domain decomposition with local impedance conditions for the Helmholtz equation,* arXiv preprint arXiv:1806.03731 (2018).

[90] M. Bonazzoli, V. Dolean, I. Graham, E. Spence, and P.-H. Tournier, *Domain decomposition preconditioning for the high-frequency time-harmonic Maxwell equations with absorption,* Mathematics of Computation **88**, 2559 (2019).

[91] N. Bootland, V. Dolean, P. Jolivet, and P.-H. Tournier, *A comparison of coarse spaces for Helmholtz problems in the high frequency regime,* Computers Mathematics with Applications **98**, 239 (2021).

[92] M. J. Gander and H. Zhang, *A class of iterative solvers for the Helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods,* SIAM Review **61**, 3 (2019).

[93] M. J. Gander and H. Zhang, *Restrictions on the use of sweeping type preconditioners for Helmholtz problems,* in *International Conference on Domain Decomposition Methods* (Springer, 2017) pp. 321–332.

[94] A. Greenbaum and Z. Strakos, *Matrices that generate the same Krylov residual spaces,* in *Recent advances in iterative methods* (Springer, 1994) pp. 95–118.

[95] L. N. Trefethen, *Pseudospectra of linear operators,* SIAM Review **39**, 383 (1997).

[96] M. Embree, *How descriptive are GMRES convergence bounds?* (1999).

[97] I. C. Ipsen, *Expressions and bounds for the GMRES residual,* BIT Numerical Mathematics **40**, 524 (2000).

[98] I. C. Ipsen, *Departure from normality and eigenvalue perturbation bounds,* A+ A **1**, 2 (2003).

[99] L. Garcia Ramos and R. Nabben, *On the spectrum of deflated matrices with applications to the deflated shifted laplace preconditioner for the Helmholtz equation,* SIAM Journal on Matrix Analysis and Applications **39**, 262 (2018).

[100] M. Eiermann, O. G. Ernst, and O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods,* Journal of Computational and Applied Mathematics **123**, 261 (2000).

[101] R. B. Morgan, Z. Yang, and B. Zhong, *Pseudoeigenvector bases and deflated GMRES for highly nonnormal matrices,* Numerical Linear Algebra with Applications **23**, 1032 (2016).

[102] O. G. Ernst and M. J. Gander, *Why it is difficult to solve Helmholtz problems with classical iterative methods,* in *Numerical analysis of multiscale problems* (Springer, 2012) pp. 325–363.

[103] M. Donatelli, *A note on grid transfer operators for multigrid methods,* arXiv preprint arXiv:0807.2565 (2008).

[104] M. Holtz and A. Kunoth, *B-spline-based monotone multigrid methods,* SIAM Journal on Numerical Analysis **45**, 1175 (2007).

[105] A. Pinkus, *On L^1-approximation*, Vol. 93 (Cambridge University Press, 1989).

[106] D. F. Rogers, *An introduction to NURBS: with historical perspective* (Elsevier, 2000).

[107] M. J. Gander and H. Zhang, *Optimized Schwarz methods with overlap for the Helmholtz equation,* SIAM Journal on Scientific Computing **38**, A3195 (2016).

[108] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, *On a class of preconditioners for solving the Helmholtz equation,* Applied Numerical Mathematics **50**, 409 (2004).

[109] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems,* SIAM Journal on Scientific Computing **27**, 1471 (2006).

[110] H. C. Elman, O. G. Ernst, and D. P. O'leary, *A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations,* SIAM Journal on scientific computing **23**, 1291 (2001).

[111] S. Kim and S. Kim, *Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media,* SIAM Journal on Scientific Computing **24**, 684 (2002).

[112] I. Livshits and A. Brandt, *Accuracy properties of the wave-ray multigrid algorithm for Helmholtz equations,* SIAM Journal on Scientific Computing **28**, 1228 (2006).

[113] D. Lahaye and C. Vuik, *How to choose the shift in the shifted laplace preconditioner for the Helmholtz equation combined with deflation,* in *Modern Solvers for Helmholtz Problems* (Springer, 2017) pp. 85–112.

[114] H. Chen, P. Lu, and X. Xu, *A robust multilevel method for hybridizable discontinuous galerkin method for the Helmholtz equation,* Journal of Computational Physics **264**, 133 (2014).

[115] H. Chen, H. Wu, and X. Xu, *Multilevel preconditioner with stable coarse grid corrections for the Helmholtz equation,* SIAM Journal on Scientific Computing **37**, A221 (2015).

[116] C. W. Oosterlee, *A GMRES-based plane smoother in multigrid to solve 3D anisotropic fluid flow problems,* Journal of Computational Physics **130**, 41 (1997).

[117] M. Baumann, R. Astudillo, Y. Qiu, E. Ang, M. Van Gijzen, and R. Plessix, *An Msss-preconditioned matrix equation approach for the time-harmonic elastic wave equation at multiple frequencies,* Computational Geosciences **22**, 43 (2018).

[118] A. Brandt and S. Ta'asan, *Multigrid method for nearly singular and slightly indefinite problems,* in *Multigrid Methods II* (Springer, 1986) pp. 99–121.

[119] J. H. Bramble, J. E. Pasciak, and J. Xu, *The analysis of multigrid algorithms for nonsymmetric and indefinite elliptic problems,* Mathematics of Computation **51**, 389 (1988).

[120] J. Wang, *Convergence analysis of multigrid algorithms for nonselfadjoint and indefinite elliptic problems,* SIAM Journal on Numerical Analysis **30**, 275 (1993).

[121] Y. Notay, *Algebraic analysis of two-grid methods: The nonsymmetric case,* Numerical Linear Algebra with Applications **17**, 73 (2010).

[122] Y. Notay, *Convergence analysis of perturbed two-grid and multigrid methods,* SIAM Journal on Numerical Analysis **45**, 1035 (2007).

[123] Y. Notay, *An aggregation-based algebraic multigrid method,* Electronic transactions on numerical analysis **37**, 123 (2010).

[124] Y. Notay, *Analysis of two-grid methods: The nonnormal case,* Mathematics of Computation **89**, 807 (2020).

[125] D. Lahaye, H. De Gersem, S. Vandewalle, and K. Hameyer, *Algebraic multigrid for complex symmetric systems,* IEEE Transactions on Magnetics **36**, 1535 (2000).

[126] S. P. MacLachlan and C. W. Oosterlee, *Algebraic multigrid solvers for complex-valued matrices,* SIAM Journal on scientific computing **30**, 1548 (2008).

[127] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid based on element interpolation,* SIAM Journal on Scientific Computing **22**, 1570 (2001).

[128] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (α sa) multigrid,* SIAM Review **47**, 317 (2005).

[129] R. D. Falgout and P. S. Vassilevski, *On generalizing the algebraic multigrid framework,* SIAM Journal on Numerical Analysis **42**, 1669 (2004).

[130] J. Brannick, F. Cao, K. Kahl, R. D. Falgout, and X. Hu, *Optimal interpolation and compatible relaxation in classical algebraic multigrid,* SIAM Journal on Scientific Computing **40**, A1473 (2018).

[131] L. G. Ramos and R. Nabben, *On optimal algebraic multigrid methods,* arXiv preprint arXiv:1906.01381 (2019).

[132] Y. Erlangga and E. Turkel, *Iterative schemes for high order compact discretizations to the exterior Helmholtz equation,* ESAIM: Mathematical Modelling and Numerical Analysis **46**, 647 (2012).

[133] L. N. Olson and J. B. Schroder, *Smoothed aggregation for Helmholtz problems,* Numerical Linear Algebra with Applications **17**, 361 (2010).

[134] L. N. Olson, J. B. Schroder, and R. S. Tuminaro, *A general interpolation strategy for algebraic multigrid using energy minimization,* SIAM Journal on Scientific Computing **33**, 966 (2011).

[135] B. Lee, T. A. Manteuffel, S. F. McCormick, and J. Ruge, *First-order system least-squares for the Helmholtz equation,* SIAM Journal on Scientific Computing **21**, 1927 (2000).

[136] S. Cools, B. Reps, and W. Vanroose, *A new level-dependent coarse grid correction scheme for indefinite Helmholtz problems,* Numerical Linear Algebra with Applications **21**, 513 (2014).

Vandana **Dwarka**

MATHEMATICIAN, ECONOMETRICIAN, LAWYER

Amsterdam | @ v.n.s.r.dwarka@tudelft.nl |

Experience

2018–2022	**PhD. Applied Mathematics** TU DELFT · Delft
2012–2013	**Quantitative Analyst** MINISTRY OF FINANCE · The Hague
2010–2012	**Consultant** ERNST & YOUNG · Amsterdam
2009–2010	**Consultant** DELOITTE · Amsterdam

Talks

2021	**SIAM Copper Mountain**	*Boulder*, USA*
2021	**SIAM Geoscience**	*Milan*, Italy*
2021	**SIAM CSE**	*Texas*, USA*
2020	**26th International DDM**	*Hong Kong*, China*
2019	**WSC Spring Meeting Antwerp**	*Antwerp, Belgium*
2019	**SIAM CSE**	*Spokane, USA*
2018	**25th International DDM**	*St. John's, Canada*
2018	**SIAM Copper Mountain**	*Boulder, USA*
2017	**Invited Talk TU Berlin**	*Berlin, Germany*

Certificates, Grants & Prizes

2022	Finalist PhD. Prize, Nederlands Mathematisch Congres
2021	Machine Learning, Stanford Online
2020	Quantum Mechanics & Computing, BerkeleyX
2019	Winner Poster Award, Woudschoten Conference
2012	Duisenberg Thesis Awards

Programming

Matlab	● ● ● ● ●
R/STATA	● ● ● ● ●
Python	● ● ● ● ○
C++	● ● ● ○ ○
MPI/CUDA	● ● ● ○ ○

Supervision

2022–present	*MSc. Student Erik Sieburgh.* Co-supervisor.
2020–present	*PhD. Candidate Jinqiang Chen.* Daily supervisor.

Education

2014–2017	**MSc. Applied Mathematics** TU DELFT
2012–2014	**Minor Applied Mathematics** TU DELFT
2011–2012	**MSc. Finance (Hon.)** VU
2009–2010	**LLM. Fiscal Law (Hon.)** UvA

Journal Papers

2021 V. Dwarka & C. Vuik, *Stand-Alone Multigrid for Helmholtz Revisited: Towards Convergence Using Standard Components.*, SIAM Journal on Scientific Computing, SIAM (SISC). *Under review.*

2021 V. Dwarka & C. Vuik *Scalable Multi-Level Deflation Preconditioning for the Highly Indefinite Helmholtz Equation.*, Journal of Computational Physics, Elsevier (JCP). *Accepted.*

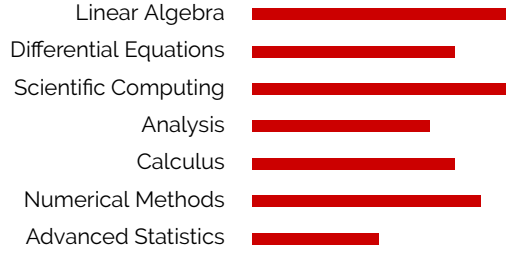
2021 N. Bootland & V. Dwarka et al. *Inexact subdomain solves using deflated GMRES for Helmholtz problems*, Lecture Notes in Computational Science, Springer (Proceeding).

2021 V. Dwarka & C. Vuik *Pollution and Accuracy of Solutions of the Helmholtz Equation: A Novel Perspective From the Eigenvalues*, Journal of Computational & Applied Mathematics, Elsevier (JCAM).

2021 V. Dwarka & C. Vuik *Towards Accuracy and Scalability: Combining Isogeometric Analysis With Deflation to Obtain Scalable Convergence for the Helmholtz Equation*, Computer Methods in Applied Mechanics and Engineering, Elsevier (CMAME).

2020 V. Dwarka & C. Vuik *Scalable Convergence Using Two-Level Deflation Preconditioning for the Helmholtz Equation.*, Journal on Scientific Computing, SIAM (SISC).

Teaching Experience



Delft Institute of Applied Mathematics