Delft University of Technology

Fast Iterative Solution of the Time-Harmonic Elastic Wave Equation at Multiple Frequencies

Baumann, Manuel

DOI
10.4233/uuid:b1024bc5-46ad-450e-a3d3-090a166a67a7

Publication date
2018

Document Version
Final published version

Citation (APA)
Baumann, M. (2018). Fast Iterative Solution of the Time-Harmonic Elastic Wave Equation at Multiple Frequencies Delft University of Technology DOI: 10.4233/uuid:b1024bc5-46ad-450e-a3d3-090a166a67a7

Fast Iterative Solution of the Time-Harmonic Elastic Wave Equation at Multiple Frequencies

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
woensdag 10 januari 2018 om 15:00 uur

door

Manuel Matthias BAUMANN
Master of Science in Applied Mathematics, Technische Universiteit Delft,
en Kungliga Tekniska högskolan, Stockholm, Zweden,
geboren te Berlijn, Duitsland.

Dit proefschrift is goedgekeurd door de

promotor: Prof.dr.ir. Kees Vuik
copromotor: Dr.ir. Martin B. van Gijzen

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. Kees Vuik	Technische Universiteit Delft, promotor
Dr.ir. Martin B. van Gijzen	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. Dr. Andreas Frommer	Bergische Universität Wuppertal
Prof. Dr. Reinhard Nabben	Technische Universität Berlin
Prof.dr.ir. Cornelis W. Oosterlee	Technische Universiteit Delft
Dr.ir. Rob Remis	Technische Universiteit Delft
Prof.dr.ir. Arnold Heemink	Technische Universiteit Delft, reservelid
Andere leden:	
Dr. René-Édouard Plessix	Shell Global Solutions International B.V., Institut de Physique du Globe de Paris

Keywords: *Krylov subspace methods · preconditioning · shifted linear systems · time-harmonic elastic wave equation · MSSS matrix computations · spectral analysis*

AMS Subject Classification: *65F08 · 65F50 · 65M60 · 74B05*

Fast Iterative Solution of the Time-Harmonic Elastic Wave Equation at Multiple Frequencies.
Dissertation at Delft University of Technology.

Shell Global Solutions International B.V. is gratefully acknowledged for the financial support of this thesis work.

ISBN 978-94-6295-827-2

Printed by: ProefschriftMaken (`www.proefschriftmaken.nl`)
Cover design: Lucas Rozenboom (`www.lucasrozenboom.com`)

Contents

Summary

Fast Iterative Solution of the Time-Harmonic Elastic Wave Equation at Multiple Frequencies

Manuel M. Baumann

This work concerns the efficient numerical solution of the elastic wave equation. The elastic wave equation is a well-established partial differential equation (PDE) that models wave propagation through an elastic medium such as the earth subsurface, and is, therefore, of great importance in seismic applications. Geophysicists match simulation results of the elastic wave equation with measurements in a PDE-constrained optimization framework in order to gain information about the structure of the earth subsurface. After spatial discretization and Fourier transform in time, the time-harmonic elastic wave equation (forward problem) reads,

$$(K + i\omega_k C - \omega_k^2 M)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\omega, \tag{$\bigstar$}$$

where the primary challenge of this work is the efficient numerical solution of ($\bigstar$) when multiple (angular) wave frequencies are present, that is $N_\omega > 1$. This task becomes in particular challenging when the elastic wave equation in three spatial dimensions is considered because the matrices K, C and M in ($\bigstar$) become very large and ill-conditioned. In this situation, Krylov subspace methods are the common choice for the iterative numerical solution of ($\bigstar$). Without appropriate preconditioning, however, the Krylov iteration converges slowly to the solutions of the linear systems ($\bigstar$). The main contributions of this work are:

1. Development of an efficient shift-and-invert preconditioner designed for the simultaneous iterative solution when ($\bigstar$) is reformulated as a sequence of shifted linear systems. The shift-and-invert preconditioner is optimal with respect to a spectral convergence bound of multi-shift GMRES.

2. Implementation and development of an algorithmic framework that solves shifted linear systems in a nested inner-outer iteration loop. The new algorithm has been evaluated for different combinations of inner and outer multi-shift Krylov methods.

3. In a practical application, the preconditioner is usually applied inexactly. We, therefore, extend the recent theory of multilevel sequentially semiseperable (MSSS) matrix computations to the elastic operator in two and three spatial dimensions.

This work can be seen as a continuation of the extensive research of the last decade on the Complex Shifted Laplace preconditioner for the (discretized) acoustic wave equation that has been performed to a large extend at Delft University of Technology. The other way around, many of the contributions of this thesis also apply to the Helmholtz operator in a multi-frequency setting. We conclude this work with various numerical experiments in two ($d = 2$) and three ($d = 3$) spatial dimensions. In both cases, the size of the computational mesh in one spatial direction n is restricted by the highest wave frequency considered such that typically 20 points per wave length are guaranteed. If $\epsilon > 0$ denotes the viscous damping parameter, our algorithm has shown computational complexity

$$\mathcal{O}\left(n^{d+\mathbb{1}_{\{\epsilon=0\}}}\right), \quad \text{where } d \in \{2,3\} \text{ and } \mathbb{1}_{\{\epsilon=0\}} := \begin{cases} 0, & \text{if } \epsilon > 0, \\ 1, & \text{if } \epsilon = 0, \end{cases}$$

when the grid size and the frequency range are increased simultaneously, and multiple frequencies within this range are present.

Preconditioning techniques rely to a large extend on (numerical) approximations. The approximation can, in general, be motivated by physical insight of the dynamical behavior of the underlying PDE, or purely depend on the algebraic structure of the matrix that is obtained after discretization. In this work we have exploited both approaches to some extend: Inexact MSSS matrix computation techniques limit the growth of the off-diagonal sub-matrix rank and, hence, rely on the structure of the discretization matrix on a Cartesian grid. On the other hand, physical insight is used when the shift-and-invert preconditioner is solved at a damped frequency. Due to the damping, it is possible to efficiently slice a 3D problem into a sequence of 2D problems (block SSOR preconditioner) and additionally solve the problem on a smaller 3D grid (additive coarse grid correction).

Samenvatting

Snelle Iteratieve Oplossing van de Tijdsharmonische Elastische Golfvergelijking bij Verschillende Frequenties

Manuel M. Baumann

Dit werk beschrijft een efficiënte numerieke oplossing van de elastische golfvergelijking. De elastische golfvergelijking is een gevestigde partïele differentaalvergelijking (PDV) die golfvoortplanting modelleert door een elastisch medium zoals bijvoorbeeld het aardoppervlak, en is daarom van groot belang in seismische toepassingen. In de geofysica worden simulatieresultaten van de elastische golfvergelijking vergeleken met metingen in een PDV-begrensd optimalisatieframework om informatie te vergaren over de structuur van het aardoppervlak. Na ruimtelijke discretisatie en Fourier transformatie in de tijd, is de tijdsharmonische elastische golfvergelijking te schrijven als,

$$(K + i\omega_k C - \omega_k^2 M)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\omega, \tag{$\star$}$$

waarbij de eerste uitdaging van dit werk de efficiënte numerieke oplossing is van (★) waarbij meedere frequenties meegenomen worden, oftewel $N_\omega > 1$. Deze opgave wordt uitdagend als de elastische golfvergelijking in drie dimensies wordt beschouwd, want de matrices K, C and M in (★) worden erg groot en slecht geconditioneerd. In deze situatie worden Krylov deelruimtemethoden als de meest standaard keuze beschouwd for de iteratieve numerieke oplossing van (★). Zonder adequate preconditionering convergeren de Krylov iteraties echter zeer langzaam naar de oplossingen van de lineaire systemen (★). De algemene bijdrage van dit werk zijn:

1. Ontwikkeling van een efficiënte shift-and-invert preconditionering ontworpen voor de simultaan-iteratieve oplossing als (★) herschreven is als een rij van verschoven lineaire systemen. De shift-and-invert preconditionering is optimaal met betrekking tot de spectrale convergentiegrens van multi-shift GMRES.

2. Implementatie en ontwikkeling van een algoritmisch framework dat de verschoven lineaire systemen in een geneste binnen-buiten iteratie-loop oplost.

3. In een praktische toepassing wordt de preconditionering meestal niet exact toegepast. Wij breiden daarom de recente theorie van multilevel sequentiële semi-separabele (MSSS) matrixberekeningen uit naar de elastische operator in twee en drie dimensies.

Dit werk kan beschouwd worden als voortzetting van uitgebreid onderzoek van het laatste decennium naar de Complex Shifted Laplace preconditionering voor de gediscretiseerde akoestische golfvergelijking die voor het grootste deel is uitgevoerd door de Technische Universiteit Delft. Aan de andere kant zijn veel bijdragen van deze thesis ook van toepassing op de Helmholtz operator in een multi-frequentie setting. Wij concluderen dit werk met verscheidene numerieke experimenten in twee en drie ruimtelijke dimensies. In beide gevallen is de grootte van het grid in één richting n begrensd door de hoogste frequentie zodanig dat typisch 20 punten per golflengte gegarandeerd zijn. Als $\epsilon > 0$ de visceuze dempingparameter beschrijft, geeft ons algoritme een complexiteit van

$$\mathcal{O}\left(n^{d+\mathbb{1}_{\{\epsilon=0\}}}\right), \quad \text{met } d \in \{2,3\} \text{ en } \mathbb{1}_{\{\epsilon=0\}} := \begin{cases} 0, & \text{als } \epsilon > 0, \\ 1, & \text{als } \epsilon = 0, \end{cases}$$

als de gridgrootte en frequentieinterval simultaan worden opgehoogd en meerdere frequenties aanwezig zijn in dit interval.

Preconditioneringstechnieken zijn sterk afhankelijk van (numerieke) benaderingen. De benadering kan in het algemeen gemotiveerd worden door natuurkundig inzicht of dynamisch gedrag van de onderliggende PDV, of puur afhangen van de algebraïsche structuur van de matrix die na discretisatie geconstrueerd is. In dit werk worden beide benaderingen tot op zekere hoogte gebruikt: Niet-exacte MSSS matrixberekeningstechnieken beperken de groei van de niet-diagonale sub-matrix rang en hangen af van de structuur van de discretisatiematrix op een Cartesisch grid. Anderzijds, natuurkundig inzicht wordt gebruikt wanneer de shift-and-invert preconditionering wordt opgelost voor een gedempte frequentie. Door demping is het mogelijk om een 3D probleem efficiënt op te delen in een aantal 2D problemen (block SSOR preconditioner) en bovendien het probleem op een kleiner 3D grid op te lossen (additive coarse grid correction).

Chapter 1

Introduction

Understanding the structure of the earth subsurface is a major interest of seismological research. A better understanding has positive impact not only on oil exploration but also on earthquake warnings. In modern industry, research and development is often done in combination with computer simulations. In the concrete case of seismic Full-Waveform Inversion (FWI), simulation results are matched with measurements in order to improve the understanding of the earth subsurface layers, cf. Figure 1.1. Computer simulations on current hardwares are limited in computation speed as well as memory storage. Therefore, state-of-the-art Numerical Linear Algebra (NLA) is developing algorithms that are both fast and memory-efficient. In the overview paper [153] on seismic FWI, the authors state:

> *"The main advantage of the iterative approach is the low memory requirement, although the main drawback results from a difficulty to design an efficient preconditioner because the impedance matrix is indefinite. To our knowledge, the extension to elastic wave equations still needs to be investigated."* – J. Virieux and S. Operto (2009)

The above quote gives direct motivation for this thesis work. The main computational bottleneck in a FWI algorithm is the numerical solution of the elastic wave equation (forward problem). Especially for large wave frequencies and high-resolution simulations, the solution of the elastic wave equation requires to solve a large system of linear equations. Krylov subspace methods are a memory-efficient approach to solve such systems of linear equations, but their convergence can be slow and, therefore, preconditioning is subject to current research in NLA. The need for efficient preconditioning becomes even more challenging when the simultaneous solution of multiple linear systems that arise from multiple wave frequencies is required.

In the following introductory part, we derive the time-harmonic elastic wave equation and present several possibilities for its (spatial) discretization. Because our research is focused on iterative Krylov methods, we review preconditioning techniques that have recently been developed for the acoustic wave equation (Helmholtz equation). A main aspect of our work concerns the development of efficient Krylov algorithms for multiple wave frequencies. Therefore, we also review existing precondi-

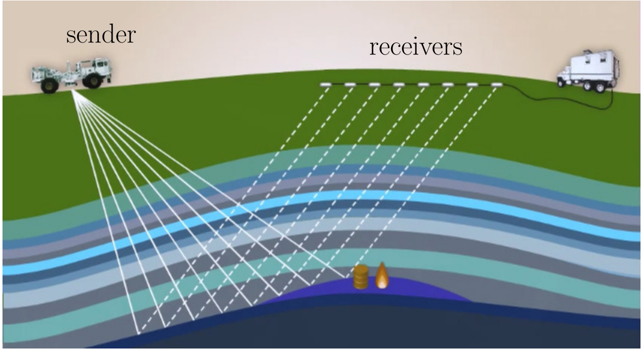


Figure 1.1: *A simplified scheme of a land seismic experiment with sender and receivers.* © *geophysicsRocks!*

tioning techniques for multi-shift Krylov methods. The introduction concludes with a closer look on seismic FWI. In the thesis outline, we emphasize the main contributions of this thesis work.

1.1 The Time-Harmonic Elastic Wave Equation

We derive the displacement formulation of the wave equation in an inhomogeneous elastic medium in two ($d = 2$) or three ($d = 3$) spatial dimensions. Our derivations are based on the standard textbooks [7, 28, 55, 71, 86]. We will denote the vector-valued displacement as,

$$\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} u_1(x, y, z, t) \\ u_2(x, y, z, t) \\ u_3(x, y, z, t) \end{pmatrix}, \quad \text{for } d = 3,$$

where $\mathbf{x} = (x, y, z) \in \Omega \subset \mathbb{R}^d$ denotes the spacial variables, and $t > 0$ denotes time. The equation of motion, which is an analog of Newton's second law *"mass × acceleration = forces"*, reads

$$\rho \ddot{\mathbf{u}} = \nabla \cdot \sigma + \mathbf{s}, \quad \text{for } \mathbf{x} \in \Omega,\ t > 0, \tag{1.1}$$

where the material density $\rho = \rho(\mathbf{x})$ in an inhomogeneous medium is space-dependent and $\mathbf{s}$ denotes external forces emitted by the sender in Figure 1.1. In one spatial dimension, the stress tensor σ is related linearly to the strain ϵ via Hooke's law, $\sigma = E \cdot \epsilon$, with Young's modulus E being determined for a specific material. For higher dimensions, the stress-strain relation reads,

$$\begin{aligned} \sigma = C : \epsilon &= \lambda(\mathbf{x}) \text{tr}(\epsilon) I_d + 2\mu(\mathbf{x})\epsilon & (1.2) \\ &= \lambda(\mathbf{x}) \left(\nabla \cdot \mathbf{u}\ I_d\right) + \mu(\mathbf{x}) \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathsf{T}}\right), & (1.3) \end{aligned}$$

with I_d being the identity matrix of dimension $d = \{2, 3\}$.

Here, the linearized strain tensor for isotropic† elastic‡ media is used,

$$\epsilon(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathsf{T}}\right), \tag{1.4}$$

and $tr(\cdot)$ in (1.2) denotes the trace of the strain tensor. The substitution of the strain tensor (1.4) in (1.2) yields a relation for the stress in terms of the displacement vector, i.e. $\sigma = \sigma(\mathbf{u})$ as in (1.3). Seismic waves travel in compressional (primary) P-waves and shear (secondary) S-waves.

Definition 1.1 (Tensor notation I, [86]) *The divergence and the gradient of a vector-valued function* $\mathbf{f} = (f_1, f_2, f_3)^{\mathsf{T}}$ *that depends on three spatial variables* x, y, z *are defined as:*

$$\nabla\cdot\mathbf{f} := \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z}, \qquad \nabla\mathbf{f} := \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial x} & \frac{\partial f_3}{\partial x} \\ \frac{\partial f_1}{\partial y} & \frac{\partial f_2}{\partial y} & \frac{\partial f_3}{\partial y} \\ \frac{\partial f_1}{\partial z} & \frac{\partial f_2}{\partial z} & \frac{\partial f_3}{\partial z} \end{pmatrix}.$$

The strain tensor ϵ *in* (1.4) *is known as the symmetric gradient. Note that with* ϵ *as in* (1.4) *it holds,* $\nabla\cdot\mathbf{f} = tr(\epsilon(\mathbf{f}))$.

Moreover, we introduce the Lamé parameters λ and μ in (1.5). The Lamé parameters λ and μ are directly related to the material density ρ and the speed of P-waves c_p and the speed of S-waves c_s via the well-know relations,

$$\mu = \frac{E\nu}{(1+\nu)(1-2\nu)} = c_s^2\rho, \quad \lambda = \frac{E}{2(1+\nu)} = \rho(c_p^2 - 2c_s^2), \tag{1.5}$$

with Poisson's ratio ν. Due to the relations (1.5), the set of three parameter $\{\rho, c_p, c_s\}$ is sufficient to fully describe problem (1.1) and the boundary conditions discussed in Section 1.1.1. The benchmark problems defined in Section 1.1.3 prescribe $\{\rho, c_p, c_s\}$ at all point in the computational domain $\mathbf{x} \in \Omega \subset \mathbb{R}^d$.

When the Fourier transform, cf. Definition 1.2, is applied to (1.1), the time-harmonic elastic wave equation yields,

$$-\omega^2\rho\,\hat{\mathbf{u}} - \nabla\cdot\sigma(\hat{\mathbf{u}}) = \hat{\mathbf{s}}, \qquad \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^d. \tag{1.6}$$

Here, the frequency-domain displacement vector $\hat{\mathbf{u}} = \hat{\mathbf{u}}(\mathbf{x}, \omega)$ is a function of space and the angular frequency $\omega = 2\pi f$. In a practical application, the solution of (1.6) at multiple frequencies $\{\omega_1, ..., \omega_{N_\omega}\}$ will be required, cf. Subsection 1.4.

Definition 1.2 (Fourier transform, [159]) *The continuous Fourier transform of a scalar function* $f \in L^1(\mathbb{R})$ *is defined as,*

$$\hat{f}(\omega) := (\mathcal{F}f)(\omega) = \int_{\mathbb{R}} f(t)e^{-i\omega t}dt \quad \forall\omega \in \mathbb{R}.$$

In particular, for the first and second (temporal) derivatives, it holds,

$$\mathcal{F}(\dot{f}) = i\omega\mathcal{F}(f) \quad \text{and} \quad \mathcal{F}(\ddot{f}) = -\omega^2\mathcal{F}(f),$$

and, if $f \in L^2(\mathbb{R})$, *Parseval's identity holds,*

$$\|f\|_{L^2(\mathbb{R})} = (2\pi)^{-1}\|\mathcal{F}f\|_{L^2(\mathbb{R})}.$$

---

† Those are materials with physical properties that are independent of direction in space.

‡ A medium that is able to resume its original shape after being stretched or compressed.

Remark 1.3 (The Helmholtz model) *The acoustic (scalar) wave equation reads,*

$$\ddot{p} - c^2 \Delta p = 0, \tag{1.7}$$

where p is the pressure and $c = c(\mathbf{x})$ the wave speed. A (finite difference) discretization with natural boundary conditions yields the well-known Helmholtz equation $(K + k^2 I)\hat{\mathbf{p}} = \mathbf{0}$, with wavenumber $k := 2\pi f/c = \omega/c$, for the time-harmonic pressure $\hat{\mathbf{p}}$. The acoustic wave equation is the scalar analogue of (1.6) when choosing $(c_p, c_s) \equiv (c, 0)$, and no source term in (1.7).

Remark 1.4 (Frequency versus time-domain simulations) *The elastic wave equation (1.1) can be solved numerically in a method-of-lines approach using an explicit time integration scheme. The computational complexity of both frequency and time domain has been compared multiple times [40, 103]. The comparison boils down to $n_t\mathcal{O}(n^d)$ complexity for the time-domain versus $n_{it}\mathcal{O}(n^d)$ complexity for an iterative Krylov method applied to a single frequency problem. We here assume that matrix-vector products are the dominating costs and can be performed in $\mathcal{O}(n^d)$. Since the number of time steps n_t (CFL condition) and the number of Krylov iterations n_{it} [103, 112] both are proportional to n, this simple comparison emphasizes the challenge to design an iterative algorithm simultaneous for multiple frequencies $n_f > 1$ in order for a frequency-domain approach to be competitive, i.e. $n_f \in \mathcal{O}(1)$.*

Remark 1.5 (Velocity-stress formulation) *Using the substitution $\mathbf{v} := \dot{\mathbf{u}}$, we can re-write the system (1.1) and (1.3) as the so-called velocity-stress formulation [151],*

$$\begin{aligned} \dot{\mathbf{v}} &= (\nabla\cdot\sigma + \mathbf{s})/\rho, \\ \dot{\sigma} &= \lambda\,(\nabla\cdot\mathbf{v}\ I_d) + \mu\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^{\mathsf{T}}\right), \end{aligned}$$

which is a coupled first-order system with parameter set $\{\rho, \lambda, \mu\}$ and source term $\mathbf{s}$.

1.1.1 Boundary Conditions and Damping Models

The time-harmonic elastic wave equation (1.6) is defined in the *inside* of the computational domain $\Omega \subset \mathbb{R}^d$. This section concerns the modeling aspect of suitable boundary conditions. We, therefore, split the boundary into $\partial\Omega = \partial\Omega_r \,\dot\cup\, \partial\Omega_a$, where we distinguish between absorbing (non-reflecting) boundary conditions on $\partial\Omega_a$ and reflecting boundary conditions on $\partial\Omega_r$.

Material-air Boundary Conditions

Typically, a free-surface boundary condition is prescribed in the North of the computational domain, cf. Figure 1.1. Due to the large difference in material density, no stresses are present normal to the boundary. The transformation of the free-surface boundary condition to frequency-domain is straightforward:

$$\sigma(\mathbf{u})\mathbf{n} = \mathbf{0} \quad \overset{\mathcal{F}}{\Rightarrow} \quad \sigma(\hat{\mathbf{u}})\mathbf{n} = \mathbf{0} \qquad \text{for } \mathbf{x} \in \partial\Omega_r. \tag{1.8}$$

Note that for a Cartesian domain, the normal vector is simply $\mathbf{n} = (0, 0, 1)^{\mathsf{T}}$ on the Northern boundary, $\partial\Omega_r = \Gamma_N$. This stress-free boundary condition can be included naturally in a finite element scheme, cf. Section 1.1.2.

Absorbing Boundary Conditions

Absorbing (non-reflecting) boundary conditions are necessary to best mimic an infinite dimensional domain. Outgoing waves are, therefore, enforced to have non-reflecting behavior when the computational domain is truncated, i.e. on $\partial\Omega_a = \partial\Omega_r \setminus \Gamma_N$. We differentiate between (primary, or compressional) P-waves and (secondary, or shear) S-waves. P-waves are parallel to the direction of wave propagation and S-waves are perpendicular to the direction of propagation. The waves travel with wave speeds c_p and c_s, respectively.

For simplicity, we consider first the south boundary and assume waves traveling in (negative) z-direction, i.e. $\partial\Omega_a = \Gamma_S$ and normal vector equals $\mathbf{n}_3 := (0, 0, -1)^\mathsf{T}$. A first order absorbing boundary condition [38, 65, 109] reads,

$$\left\{ \frac{\partial}{\partial t} - \begin{pmatrix} c_s & & \\ & c_s & \\ & & c_p \end{pmatrix} \frac{\partial}{\partial z} \right\} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{when } \mathbf{x} \in \partial\Omega_a = \Gamma_S. \tag{1.9}$$

Due to the sign in (1.9), waves can only traverse in negative z-direction, but not in the opposite direction. Hence, condition (1.9) imposes non-reflection for $\mathbf{x} \in \Gamma_S$. Assuming tangential derivatives to vanish and using relations (1.5) between the elastic parameters, the following equivalent boundary conditions hold,

$$\rho B \dot{\mathbf{u}} + \sigma(\mathbf{u})\mathbf{n}_3 = \mathbf{0}, \qquad \text{when } \mathbf{x} \in \partial\Omega_a = \Gamma_S, \text{ and } B := \operatorname{diag}(c_s, c_s, c_p).$$

After Fourier transform and going back to a general Cartesian boundary, the above relation reads,

$$i\omega\rho\, B(c_p, c_s)\hat{\mathbf{u}} + \sigma(\hat{\mathbf{u}})\mathbf{n} = \mathbf{0}, \quad \mathbf{x} \in \partial\Omega_a, \tag{1.10}$$

with $\mathbf{n}$ normal to the boundary, and with generalized expression for the $d \times d$ matrix B as given in [2, 92],

$$B(c_p, c_s) := c_p \mathbf{n}\mathbf{n}^\mathsf{T} + c_s \mathbf{t}\mathbf{t}^\mathsf{T} + c_s \mathbf{s}\mathbf{s}^\mathsf{T}, \qquad \text{with } \mathbf{t}, \mathbf{s} \perp \mathbf{n}. \tag{1.11}$$

Note the close relation of (1.10) to first order Sommerfeld radiation boundary conditions widely used for modeling absorption for the Helmholtz equation (1.7). In [37], a comparison of (1.9) with higher-order absorbing boundary conditions when the wave direction has an angle different from normal to the boundary is performed.

Perfectly Matched Layers (PML) Boundary Conditions

An artificial *sponge* layer called PML region can be added to the computational domain Ω at the absorbing boundary that damps and absorbs incoming waves. This idea has originally been developed for Maxwell's equation of electro-magnetic waves and recently been extended to elastic waves [62]. To realize the PML, a coordinate stretching function of the form

$$\tilde{x}_j := s_j(x_j, \omega)x_j, \quad \text{with } s_j(x_j, \omega) = \alpha_j(x_j)\left(1 + i\frac{\beta_j(x_j)}{\omega}\right), \ j = 1, ..., d, \tag{1.12}$$

for the spatial coordinates $\mathbf{x} = (x_1, x_2, x_3)$ is introduced. Inside the physical domain, the coordinates remain unchanged, i.e. $(\alpha_j, \beta_j) = (1, 0)$ for $\mathbf{x} \in \Omega$. Within the PML region the coefficient functions α_j, β_j realize non-reflective behavior [14, 141]. The elastic wave equation (1.6) is transformed accordingly. Due to the definition of the coordinate transformation (1.12), the elastic operator depends non-linearly on the (angular) frequencies which makes PML unpractical in a multi-frequency framework and for the focus of this thesis. The frequency-*in*dependent PML studied in [31] or a suitable linearization can be promising alternatives.

Models for Viscous Damping

Including damping boundary conditions, $\partial\Omega_a \neq \emptyset$, yields a discretized elastic operator that has spectral properties that make the problem generally easier to solve with an iterative method, cf. our spectral analysis in Section 3.3. Moreover, viscous damping can be added to the physical model by substitution of a complex-valued (angular) frequency. A common damping model for the Helmholtz equation is to substitute $k^2 \mapsto (1 - \hat{\epsilon} i)k^2$, i.e. $\omega_k^2 \mapsto (1 - \hat{\epsilon} i)\omega_k^2$, for $0 < \hat{\epsilon} \ll 1$; cf. [40, Remark 1.2.3]. This substitution does not include the Sommerfeld boundary conditions (1.10). In this work, we model damping via $\omega_k \mapsto (1 - \epsilon i)\omega_k$ in (1.13). For small damping parameters, both approaches coincide if $\hat{\epsilon} = \epsilon/2$ but note the consistent treatment within a multi-frequency framework of the second approach.

1.1.2 Spatial Discretizations of the Elastic Wave Equation

Concluding Section 1.1.1, we have derived the time-harmonic elastic wave equation (1.6) together with suitable boundary conditions (1.8) and (1.10) for reflecting and non-reflecting boundaries,

$$\left\{ \begin{aligned} -\omega_k^2 \rho\, \mathbf{u}_k - \nabla \cdot \sigma(\mathbf{u}_k) &= \mathbf{s}, && \text{for } \mathbf{x} \in \Omega \subset \mathbb{R}^d, \\ \sigma(\mathbf{u}_k)\mathbf{n} &= \mathbf{0}, && \text{for } \mathbf{x} \in \partial\Omega_r, \\ i\omega_k \rho\, B\mathbf{u}_k + \sigma(\mathbf{u}_k)\mathbf{n} &= \mathbf{0}, && \text{for } \mathbf{x} \in \partial\Omega_a, \end{aligned} \right\} \tag{1.13}$$

with $\mathbf{n}$ being normal to the respective boundary, σ and B as in (1.3) and (1.11), respectively, and $\mathbf{u}_k = \mathbf{u}(\mathbf{x}, \omega_k)$. In this thesis, we use a finite element discretization for the above boundary value problem. Alternative approaches used in literature are reviewed briefly.

Finite Difference Discretization

The most popular approach for the spatial discretization of the elastic wave equation might be a finite difference approach used for instance in [58, 97, 152]. A finite difference method can be slightly more memory-efficient than competing method but it is limited to structured meshes and the accurate implementation of, for instance, the stress-free boundary conditions (1.8) is non-trivial. A staggered-grid approach using a high order finite difference approximation is used in [58].

Boundary Integral Methods

The advantage of boundary integral methods is that a discretization is only required on the surface, i.e. on a lower dimensional domain [83]. The methods rely on solving a boundary integral equation based on Green's function for homogeneous media. Such fundamental (analytic) solution exist only in the case of homogeneous media which are not considered in this thesis. Semi-analytic extensions to domains with layered media exist [82].

Finite Element Discretization

The discretization approach followed in this thesis is a finite element method. The following discretization ansatz is made for the vector-valued displacement,

$$\mathbf{u}_k(\mathbf{x}) \approx \sum_{i=1}^{n_{\text{dofs}}} u_k^i \boldsymbol{\varphi}_i(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad u_k^i \in \mathbb{C}. \tag{1.14}$$

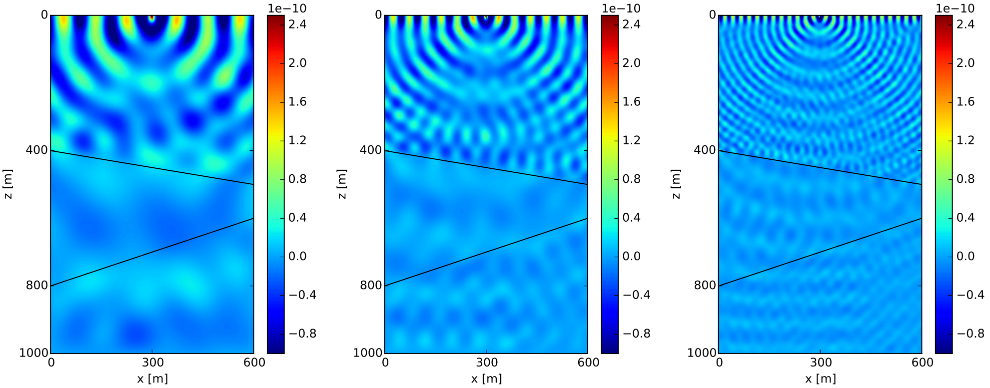
The *test functions* $\boldsymbol{\varphi}_i$ in (1.14) are typically polynomials on a compact support, see Appendix A for more details on the choice of basis functions used. The ansatz (1.14) can be substituted in (1.13). A Galerkin approach and integration by parts of the stress tensor term yields the following weak form: Find $\boldsymbol{\varphi}_i \in [H^1(\Omega)]^d$ such that,

$$\begin{aligned} -\omega_k^2 \sum_{i=1}^{n_{\text{dofs}}} u_k^i \int_\Omega \rho\, \boldsymbol{\varphi}_i \cdot \boldsymbol{\varphi}_j \; d\Omega - \sum_{i=1}^{n_{\text{dofs}}} u_k^i \int_\Omega \nabla \cdot \sigma(\boldsymbol{\varphi}_i) \cdot \boldsymbol{\varphi}_j \; d\Omega \\ = \int_\Omega \mathbf{s} \cdot \boldsymbol{\varphi}_j \; d\Omega, \quad \text{for all } \boldsymbol{\varphi}_j \in [H^1(\Omega)]^d, \end{aligned} \tag{1.15}$$

$j = 1, ..., n_{\text{dofs}}$, and for all source functions $\mathbf{s} \in [L^1(\Omega)]^d$. The boundary conditions (1.8) and (1.10) can be included naturally in (1.15) after applying the divergence theorem to the term $\int_\Omega \nabla \cdot \sigma(\boldsymbol{\varphi}_i) \cdot \boldsymbol{\varphi}_j d\Omega$. The numerical integration of the respective terms yield the linear systems,

$$(K + i\omega_k C - \omega_k^2 M)\mathbf{x}_k = \mathbf{b}, \quad \text{with } \omega_k := 2\pi f_k \text{ and } k = 1, ..., N_\omega, \tag{1.16}$$

with entries for $\{K, C, M\}$ derived in detail in (4.10)-(4.11) in Section 4.1.2.



(a) $\Re(u_2)$ at f=8 Hz (b) $\Re(u_2)$ at f=16 Hz (c) $\Re(u_2)$ at f=32 Hz

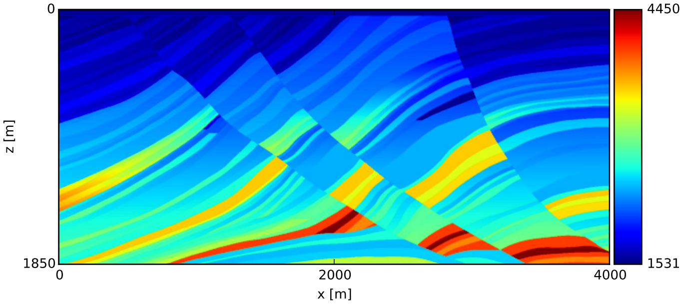
Figure 1.2: *Real part of the vertical displacement component for the 2D wedge problem at different frequencies.*

1.1.3 Benchmark Problems in 2D and 3D

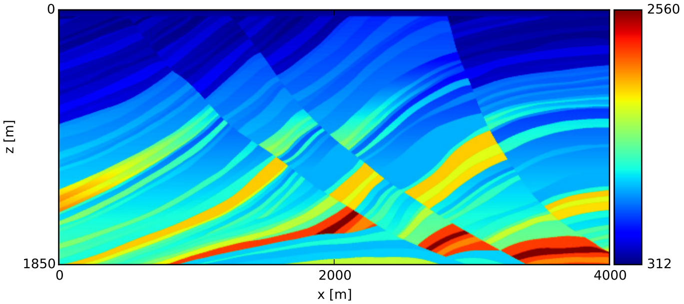
We present benchmark problems in 2D and 3D by defining the set of parameters $\{\rho, c_p, c_s\}$ for all $\mathbf{x} \in \Omega$. The Lamé parameters can be computed via (4.6). In the choice of test cases we restrict ourselves to Cartesian domains for Ω.

The marmousi-II Test Problem

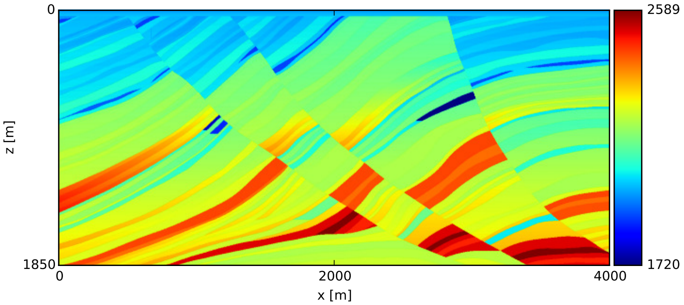
The marmousi-II problem is a standard two-dimensional benchmark problem for the elastic wave equation. In this thesis, we consider a subset of the original problem [85] on the domain $\Omega = [0, 4000] \times [0, 1850] \subset \mathbb{R}^2$, cf. Figure 1.3.



(a) Speed of P-waves c_p in m/s



(b) Speed of S-waves c_s in m/s



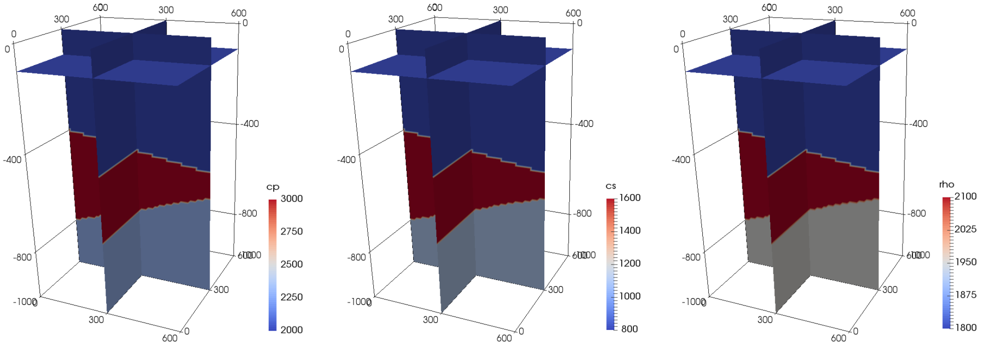
(c) Material density ρ in kg/m^3

Figure 1.3: *A subset of the well-known marmousi-II test problem [85] with removed top water layer as proposed in [113].*

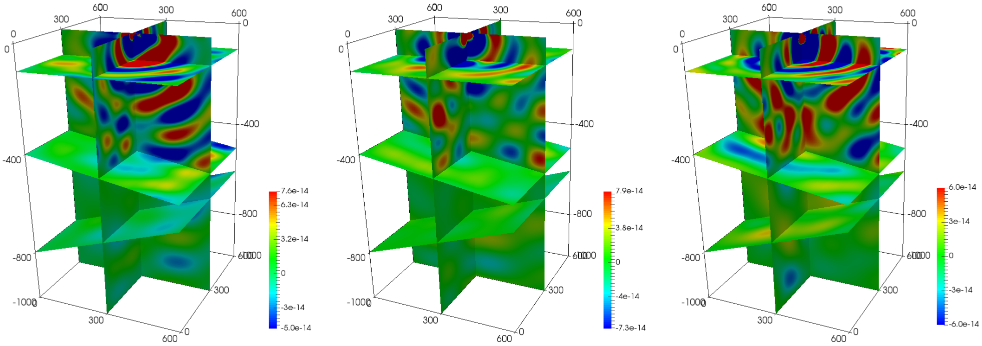
The data points in Figure 1.3 are given at an offset of $h_x = h_z = 1.25m$. This implies $n_{\text{dofs}} = 9,484,800$ degrees of freedom for the problem in Figure 1.3. For the wave velocities present in Figure 1.3 this implies a maximum frequency of approximately $f_{\max} = 12.5$ Hz when 20 points per wavelength are used.

An Elastic Wedge Test Case for 2D and 3D

A three-dimensional benchmark problem has been defined for the computational domain $\Omega = [0, 600] \times [0, 600] \times [0, 1000] \subset \mathbb{R}^3$ with elastic parameters $\{c_p, c_s, \rho\}$ that differ in three layers forming a wedge shape, cf. Figure 1.4a-1.4c. In some numerical experiments, we use a slice along the xz-plane as a 2D wedge test problem, see Figure 1.2. In Figure 1.4d-1.4f, we present the numerical solution of (1.13) at $f = 6$ Hz when a point source is placed at $(300, 300, 0)$ and a grid size of 10 m is used. This grid size yields a 3D elastic problem of roughly one million degrees of freedom, compare Table 1.1.



(a) Speed of P-waves in m/s (b) Speed of S-waves in m/s (c) Material density in kg/m^3



(d) $\Re(u_1)$ at f=6 Hz (e) $\Re(u_2)$ at f=6 Hz (f) $\Re(u_3)$ at f=6 Hz

Figure 1.4: *Parameter distribution 1.4a-1.4c and numerical solution 1.4d-1.4f at* $f = 6Hz$ *for the 3D wedge problem.*

Table 1.1: *Problem sizes of the 3D wedge benchmark problem for $\mathcal{Q}_1$ basis functions. The maximum frequency f_{max} is restricted by a minimum of* 20 *points per wavelength and the fastest wave speed* $\{c_p, c_s\}$ *in the test case.*

$h_x = h_y = h_z$	$d \times n_x \times n_y \times n_z$	n_{dofs}	$f_{\max}$
40m	$3 \times 15 \times 15 \times 25$	$16,875$	1 Hz
20m	$3 \times 30 \times 30 \times 50$	$135,000$	2 Hz
10m	$3 \times 60 \times 60 \times 100$	$1,080,000$	4 Hz
5m	$3 \times 120 \times 120 \times 200$	$8,640,000$	8 Hz

1.1.4 Reformulations of the Discrete Problem with an Emphasis on the Multi-Frequency Framework

The finite element discretization in Subsection 1.1.2 results in the discretized problem (1.16). For $N_\omega > 1$ (angular) frequencies, this yields a sequence of linear problems. The most straightforward approach is the successive numerical solution, i.e. looping over the frequency index k. In this thesis work, we consider two different approaches for the *simultaneous* solution of the multi-frequency problem (1.16).

Problem 1.6 (Shifted systems formulation) *Consider the sequence of linear problems,*

$$(K + i\omega_k C - \omega_k^2 M)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\omega.$$

A linearization [66] of the form,

$$(\mathcal{K} - \omega_k \mathcal{M})\underline{\mathbf{x}}_k = \underline{\mathbf{b}}, \quad k = 1, ..., N_\omega, \tag{1.17}$$

with $\underline{\mathbf{x}}_k := [\omega_k \mathbf{x}_k, \mathbf{x}_k]^\top$*, right-hand side vector* $\underline{\mathbf{b}} := [\mathbf{b}, \mathbf{0}]^\top$ *and* 2×2 *block matrices*

$$\mathcal{K} := \begin{bmatrix} iC & K \\ I & 0 \end{bmatrix}, \quad \textit{and} \quad \mathcal{M} := \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix},$$

yields a shifted problem with linear dependency on the frequencies.

Algorithms based on the re-formulation in Problem 1.6 are the focus of Chapter 2 and Chapter 3 of this thesis. An alternative approach to the shifted systems reformulation is considered in this thesis. In Problem 1.8, we state the re-formulation of (1.16) as a matrix equation. The matrix equation formulation is exploited in Chapter 4 and is subject to our comparison study in Chapter 5.

Remark 1.7 (Linearization (1.17) from a time-domain perspective) *In Problem 1.6, the auxiliary variable* $\hat{\mathbf{v}}_k := \omega_k \hat{\mathbf{u}}_k$ *is implicitly introduced in the unknown vector* $\underline{\mathbf{x}}_k$*. This is in close relation to the substitution* $\mathbf{v} := \dot{\mathbf{u}}$ *for the velocity-stress formulation in Remark 1.5.*

Problem 1.8 (Matrix equation formulation) *Consider, again, the sequence of linear problems,*

$$(K + i\omega_k C - \omega_k^2 M)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\omega.$$

The problem can be reformulated as the matrix equation,

$$K\mathbf{X} + iC\mathbf{X}\Omega - M\mathbf{X}\Omega^2 = \mathbf{B}, \quad \mathbf{B} := [\mathbf{b}, ..., \mathbf{b}],$$

where $\Omega := diag(\omega_1, ..., \omega_{N_\omega})$ *and the unknown* $\mathbf{X} := [\mathbf{x}_1, ..., \mathbf{x}_{N_\omega}]$ *consists of the stacked solution vectors. The block right-hand side* $\mathbf{B}$ *trivially allows multiple source terms.*

Remark 1.9 (Matrix properties in (1.16)**)** *If the matrices in Problem 1.6 and Problem 1.8 stem from a finite element discretization, we obtain* K, C *symmetric positive semidefinite, and* M *symmetric positive definite. For realistic seismic applications, the matrix* $(K + i\omega C - \omega^2 M)$ *is known to be highly ill-conditioned and indefinite [153].*

1.2 State-of-the-Art Preconditioning Techniques for Helmholtz Problems

Partial differential equations, such as the time-harmonic elastic wave equation described in detail in Section 1.1, yield after discretization a linear system of equations of the general form,

$$A\mathbf{x} = \mathbf{b}, \quad \text{with } \{\mathbf{x}, \mathbf{b}\} \in \mathbb{C}^N \text{ and } A \in \mathbb{C}^{N\times N} \text{ sparse.} \tag{1.18}$$

The sparsity of the matrix A naturally follows from the considered discretization scheme. If, for instance, a finite element approach is used then the ansatz (1.14) implies that $N = n_{\text{dofs}}$. Krylov subspace methods [81, 117] are a well-known tool to solve linear systems with large and sparse matrix A because matrix-vector products are comparably cheap. Generally speaking, an approximate solution $\mathbf{x}_m \in \mathcal{K}_m(A, \mathbf{b})$ to (1.18) is computed iteratively, and depending on the matrix properties different Krylov methods are favorable. In (1.19) we introduce the m-th Krylov subspace,

$$\mathcal{K}_m(A, \mathbf{b}) := \text{span}\left\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, ..., A^{m-1}\mathbf{b}\right\}. \tag{1.19}$$

Prominent Krylov subspace methods are GMRES [118], CG [64], Bi-CGSTAB [144] and, more recently developed, IDR(s) [130]. For the wave equation, convergence of Krylov methods is known to be very slow such that *preconditioning* [158] is required in order to speed up convergence and, eventually, to obtain $m \ll N$.

In this section, we review state-of-the-art preconditioning techniques for the acoustic wave equation (1.7) (so-called *Helmholtz problems*) based on the recent monographs [52, 78]. To some extend, our work can be seen as a continuation of the recent dissertations at Delft University of Technology [40, 76, 120].

Remark 1.10 (Direct methods for (1.18)**)** *Alternatively to Krylov subspace methods, direct methods compute an LU-factorization of A with lower (L) and upper (U) diagonal matrices,*

$$\mathbb{C}^{N\times N} \ni A = LU \quad \text{or} \quad A \approx LU.$$

On a 3D regular mesh the exact factorization requires $\mathcal{O}(N^{4/3})$ non-zeros in L and U, and $\mathcal{O}(N^2)$ flops [26], with $N = n^3$ if $n := n_x = n_y = n_z$. The references [3, 97] provide a feasibility study of (inexact) direct methods for the acoustic wave equation.

1.2.1 The Complex Shifted Laplace Preconditioner

Consider the single-frequency case in (1.16) and set $A := (K + i\omega C - \omega^2 M)$. A single preconditioner $\mathcal{P}$ can be applied to (1.18) in the following two ways,

$$\mathcal{P}^{-1}A\mathbf{x} = \mathcal{P}^{-1}\mathbf{b} \quad \text{or} \quad A\mathcal{P}^{-1}\mathbf{y} = \mathbf{b}, \text{ with } \mathbf{y} := \mathcal{P}\mathbf{x}.$$

For a *good* preconditioner, we typically require:

1. The spectrum of $\mathcal{P}^{-1}A$ (or of $A\mathcal{P}^{-1}$) is favorable to the spectrum of A which leads to a faster convergence of the considered Krylov method. This can, for instance, be achieved if $\mathcal{P} \approx A$ in some sense.

2. The preconditioner can be applied cheaply with respect to a matrix-vector product with A.

In [42, 44] the Complex Shifted Laplace (CSL) preconditioner is suggested as a right preconditioner for high-frequency Helmholtz problems. For parameters $\beta_1, \beta_2 \geq 0$, the CSL preconditioner yields,

$$\mathcal{P}_{\beta_1,\beta_2} := (K + i\omega C) - (\beta_1 - \beta_2 i)\omega^2 M = \tilde{K} - (\beta_1 - \beta_2 i)\omega^2 M. \tag{1.20}$$

Note that for $(\beta_1, \beta_2) = (1, 0)$ we get $\mathcal{P}_{\beta_1,\beta_2} = A$. Moreover, when $C \equiv 0$ in (1.20) and $\{K, M\}$ stem from a discretization of (1.7), the name Complex Shifted Laplace preconditioner is appropriate for $\beta_2 > 0$. In Figure 1.5 we show the effect of the CSL preconditioner on the spectrum. We point out that, for $C \equiv 0$, the unpreconditioned spectrum is indefinite and widely spread over the real axis. The CSL preconditioner yields a spectrum in the complex plane that is more clustered and enclosed by a circle. In [42], the authors also suggest to use a multigrid approach [140] in order to efficiently apply (1.20). The performance of the overall algorithm is demonstrated for different combinations of (β_1, β_2), and in particular for large β_2 multigrid works well because the imaginary part of the CSL preconditioner from a physics point of view corresponds to damping. The CSL preconditioner has lead to considerable attention and triggered further research. In particular the authors of [147] derive an optimal parameter $(\beta_1 - \beta_2 i) =: z_2$ for the CSL preconditioner with respect to a GMRES convergence bound. The work of [147] is generalized to the multi-frequency framework in Section 3. Extensions of the CSL preconditioner to the elastic case have been made in [2, 113].

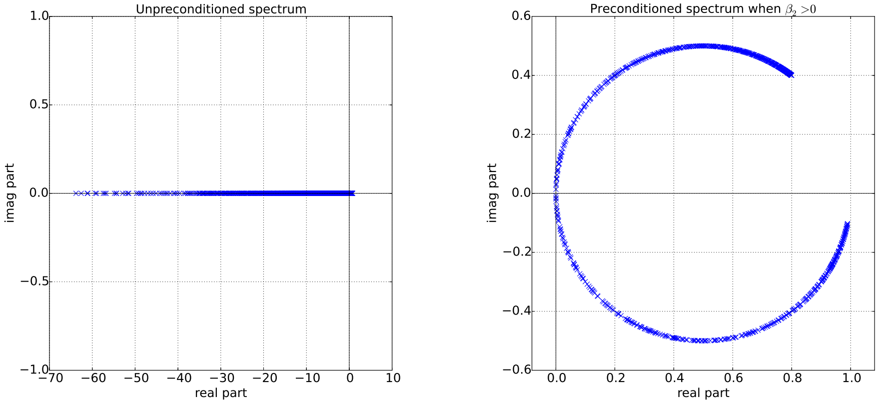


Figure 1.5: *Spectrum of a 2D Helmholtz surrogate problem for a single wave frequency, and for* $C = 0$ *in* (1.16)*. The CSL preconditioner (right) yields a much more clustered spectrum that is enclosed by a circle.*

1.2.2 Two-Level Preconditioning Techniques

We review two-level preconditioning techniques for the Helmholtz equation,

$$\mathcal{P}_2 \left(\mathcal{P}_1^{-1} A\right) \mathbf{x} = \tilde{\mathbf{b}}, \tag{1.21}$$

where $\tilde{\mathbf{b}} := \mathcal{P}_2 \mathcal{P}_1^{-1} \mathbf{b}$, and where typically $\mathcal{P}_1$ equals to the CSL preconditioner (1.20) solved with a multigrid method, cf. Subsection 1.2.1. The need for a second-level preconditioner is due to the fact that the spectrum solely preconditioned by the CSL yields near-zero eigenvalues and the circle indicated in Figure 1.5 (right) touches the origin.

Deflation as a second-level preconditioner has been studied in great detail and in a more general framework in [134, 135]. The authors of [41, 122] analyze and apply deflation on top of the CSL preconditioner, i.e,

$$\mathcal{P}_2 = I - AQ, \quad Q := ZE^{-1}Z^{\mathsf{T}}, \quad E := Z^{\mathsf{T}} AZ, \tag{1.22}$$

with prolongation and restriction operators as a choice for the deflation matrix Z. By this choice, the matrix E in (1.22) resembles a coarse grid approximation of the original matrix A. This second-level preconditioner is chosen such that the near-zero eigenvalues are deflated. An additional shift of the spectrum yields the so-called ADEF-1 methods [122, 134]. It is also possible to exchange the order of the two preconditioners in (1.21) as has been studied by [123]. When a Krylov method is used recursively for applying the inverse of E in (1.22), the overall algorithm is called a *multi-level* Krylov method [43]. In Subsection 3.4, we make use of a two-level preconditioning strategy. Spectral analysis for the single shift-and-invert preconditioner (first-level preconditioner) is exploited for an efficient second-level preconditioner, cf. subsections 3.4.1 and 3.4.2.

1.2.3 Domain Decomposition Methods

In Remark 1.10 we mention direct methods and point out their relatively large memory requirements. Many different approaches of incomplete LU factorizations limit the memory requirements, for instance by dropping non-zero elements. A block-LU factorization of a matrix A is given by,

$$A = \begin{bmatrix} S_1 & & & \\ L_1 & S_2 & & \\ & \ddots & \ddots & \\ & & L_{n-1} & S_n \end{bmatrix} \begin{bmatrix} I_1 & S_1^{-1}U_1 & & \\ & I_2 & S_2^{-1}U_2 & \\ & & \ddots & \ddots \\ & & & I_n \end{bmatrix} =: LU, \tag{1.23}$$

with so-called Schur complements,

$$S_1 = D_1, \quad S_j = D_j - L_{j-1}S_{j-1}^{-1}U_{j-1}, \text{if } j \geq 2. \tag{1.24}$$

An inexact block-LU factorization is given when the Schur complements (1.24) in (1.23) are approximated. We use an inexact block-LU factorization in Subsections 4.3.2 and 4.3.3 when inexact MSSS matrix computations are exploited on level-2 and level-3 of our scheme.

The block matrices S_j in (1.23) can be interpreted as a domain decomposition, for instance slices of lower dimensional blocks. In a (classical) domain decomposition algorithm, subdomains are defined $\Omega = \bigcup_j \Omega_j$ together with appropriate interface boundary conditions (transmission conditions) at $\Omega_i \bigcap \Omega_j, i \neq j$. The wave equation in then defined per subdomain, and an iteration scheme with overlapping or non-overlapping subdomains [34] can be defined.

For the elastic wave equation, similar ideas have been implemented in the sweeping preconditioner [141], overlapping Schwarz methods [99], and for multifrontal methods [3, 157]. The overview paper [52] unifies numerous domain decomposition approaches for the Helmholtz equation.

1.3 Krylov Methods for Shifted Linear Systems

The efficient numerical solution of shifted linear systems,

$$(A - \alpha_k I)\mathbf{x}_k = \mathbf{b}, \quad \text{where } k = 1, ..., N_\alpha \text{ and } \alpha_k \in \mathbb{C}, \tag{1.25}$$

with Krylov subspace methods heavily relies on the shift-invariance property stated in (1.26). Multi-shift variants exist for many Krylov methods, for example QMR [48], FOM(k) [117], BiCGstab(ℓ) [49], CG [142], MINRES [72] and, more recently developed, IDR(s) [33, 148] and QMRCGstab [90]. In Subsection 1.3.1 we exemplify how to exploit shift-invariance in a multi-shift Krylov algorithm for GMRES(k) based on the work by [50]. Recall the m-th Krylov subspace,

$$\mathcal{K}_m(A, \mathbf{b}) = \text{span}\left\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, ..., A^{m-1}\mathbf{b}\right\} \equiv \mathcal{K}_m(A - \alpha I, \beta\mathbf{b}), \tag{1.26}$$

and note the important property of *shift-invariance*, i.e. Krylov spaces are identical when the matrix A is shifted by a matrix with constant diagonal entries α as long as the starting vectors are collinear, that is β is a scalar in (1.26). In Subsection 1.3.2, we present an overview on recent preconditioning techniques for shifted systems that are designed such that the shift-invariance property (1.26) is preserved after multiplication with a preconditioner.

1.3.1 Shifted GMRES with Restarting

For the sake of numerical stability, an orthonormal basis of $\mathcal{K}_m(A, \mathbf{b})$ is computed by the Arnoldi method [117, Chapter 6.3]. A consequence of (1.26) is that this basis only needs to be computed *once* for all shifted systems, and the following shifted Hessenberg relation holds,

$$\begin{aligned} AV_m &= V_{m+1}\underline{\mathrm{H}}_m \\ \Rightarrow \quad (A - \alpha I)V_m &= V_{m+1}(\underline{\mathrm{H}}_m - \alpha \underline{\mathrm{I}}_m), \end{aligned}$$

Here, the m columns of $V_m \in \mathbb{C}^{N\times m}$ form an orthonormal basis of both spaces $\mathcal{K}_m(A, \mathbf{r}_0) = \mathcal{K}_m(A - \alpha I, \beta \mathbf{r}_0)$, and the matrix $\underline{\mathrm{H}}_m \in \mathbb{C}^{m+1\times m}$ is upper Hessenberg. Moreover, the matrix $\underline{\mathrm{I}}_m \in \mathbb{C}^{m+1\times m}$ denotes an $m \times m$ identity matrix with an extra zero row attached at the bottom.

GMRES is a *long-recurrence* Krylov methods and, in particular, storage of V_m can exceed memory availabilities. One way to overcome this drawback is restarting GMRES. In order to be able to restart shifted GMRES one needs to assure that residuals at restart are *collinear*. Because collinearity of the shifted residuals plays an important role in Section 2, we present the key result of [50]. Assuming $\mathbf{r}_0^{(k)} = \gamma_0^{(k)}\mathbf{r}_0$ (this can be guaranteed by choosing zero initial guess), we require after m iterations:

$$\begin{aligned} \mathbf{r}_m^{(k)} &= \gamma_m^{(k)}\mathbf{r}_m \\ \mathbf{b} - (A - \alpha_k I)(\mathbf{x}_0^{(k)} + V_m\mathbf{y}_m^{(k)}) &= \gamma_m^{(k)}V_{m+1}\mathbf{z}_{m+1} \\ \mathbf{r}_0^{(k)} - V_{m+1}(\underline{\mathrm{H}}_m - \alpha_k\underline{\mathrm{I}}_m)\mathbf{y}_m^{(k)} &= \gamma_m^{(k)}V_{m+1}\mathbf{z}_{m+1} \\ \gamma_0\mathbf{r}_0 - V_{m+1}(\underline{\mathrm{H}}_m - \alpha_k\underline{\mathrm{I}}_m)\mathbf{y}_m^{(k)} &= \gamma_m^{(k)}V_{m+1}\mathbf{z}_{m+1} \\ V_{m+1}\left((\underline{\mathrm{H}}_m - \alpha_k\underline{\mathrm{I}}_m)\mathbf{y}_m^{(k)} + \gamma_m^{(k)}\mathbf{z}_{m+1}\right) &= V_{m+1}(\gamma_0\|\mathbf{r}_0\|_2\mathbf{e}_1) \end{aligned}$$

Here, we used that the unshifted (base) system $A\mathbf{x}_m = \mathbf{b}$ is solved with GMRES implying for the base residual $\mathbf{r}_m = V_{m+1}\mathbf{z}_{m+1}$, with $\mathbf{z}_{m+1} = \|\mathbf{r}_0\|_2\mathbf{e}_1 - \underline{\mathrm{H}}_m\mathbf{y}_m$, cf. [50, 117]. Hence, the solution to the shifted system is $\mathbf{x}_m^{(k)} = V_m\mathbf{y}_m^{(k)}$, with $\mathbf{y}_m^{(k)}$ obtained from the solution of the $(m+1)\times(m+1)$ system,

$$\left[(\underline{\mathrm{H}}_m - \alpha_k\underline{\mathrm{I}}_m) \;\middle|\; \mathbf{z}_{m+1}\right]\begin{pmatrix}\mathbf{y}_m^{(k)} \\ \gamma_m^{(k)}\end{pmatrix} = \gamma_0\|\mathbf{r}_0\|_2\mathbf{e}_1. \tag{1.27}$$

When approximates $\mathbf{x}_m^{(k)}$ are computed via (1.27), corresponding residuals are collinear with collinearity factor $\gamma_m^{(k)}$ at restart. The matrix $\underline{\mathrm{H}}_m - \alpha_k\underline{\mathrm{I}}_m$ is often referred to as shifted Hessenberg matrix.

Remark 1.11 (Existence of base system) *For a given sequence of shifted matrices $(A-\alpha_k I), k = 1, ..., N_\alpha$, we can define $\hat{A} := A-\alpha_{\hat{k}}I$ for some index $\hat{k} \in \{1, ..., N_\alpha\}$*

such that the sequence of shifted matrices $(\hat{A} - (\alpha_k - \alpha_{\hat{k}})I)$ *yields an equivalent sequence of shifted systems that requires the solution of a base system* $\hat{A}\mathbf{x}_{\hat{k}} = \mathbf{b}$.

1.3.2 Shift-Invariant Preconditioning Techniques

Applying a preconditioner to (1.25) that preserves shift-invariance (1.26) requires,

$$\mathcal{K}_m(A\mathcal{P}^{-1}, \mathbf{b}) = \mathcal{K}_m((A - \alpha I)\mathcal{P}_\alpha^{-1}, \beta\mathbf{b}), \tag{1.28}$$

to hold. In this section we review recent preconditioning techniques that preserve shift-invariance and, hence, lead to a sequence of preconditioned shifted systems.

The Shift-And-Invert Preconditioner

Consider the shift-and-invert preconditioner $\mathcal{P}(\tau) := (A - \tau I)$ for some $\tau \in \mathbb{C}$. Then, we can find a scaled shift-and-invert preconditioner $\mathcal{P}_k$ such that,

$$\begin{aligned}(A - \alpha_k I)\mathcal{P}_k^{-1} &= A\mathcal{P}(\tau)^{-1} - \tilde{\alpha}_k I \\ &= A(A - \tau)^{-1} - \tilde{\alpha}_k I \\ &= [A - \tilde{\alpha}_k(A - \tau I)](A - \tau I)^{-1} \\ &= \left[A + \frac{\tilde{\alpha}_k \tau}{1 - \tilde{\alpha}_k} I\right](1 - \tilde{\alpha}_k)(A - \tau)^{-1}.\end{aligned}$$

If we now choose,

$$\frac{\tilde{\alpha}_k \tau}{1 - \tilde{\alpha}_k} = -\alpha_k \;\Leftrightarrow\; \tilde{\alpha}_k = \frac{\alpha_k}{\alpha_k - \tau}, \tag{1.29}$$

then the scaled shift-and-invert preconditioner,

$$\mathcal{P}_k^{-1} = (1 - \tilde{\alpha}_k)(A - \tau I)^{-1} = \frac{\tau}{\tau - \alpha_k}(A - \tau I)^{-1}, \tag{1.30}$$

preserves shift-invariance in the sense of (1.28), cf. also the work of [87]. Altogether, if we apply $\mathcal{P}_k$ as defined in (1.30) to a sequence of shifted systems,

$$(A - \alpha_k I)\frac{\tau}{\tau - \alpha_k}(A - \tau I)^{-1} = A(A - \tau I)^{-1} - \frac{\alpha_k}{\alpha_k - \tau} I, \tag{1.31}$$

the right hand side of the above equation yields a sequence of preconditioned shifted systems with newly defined shifts $\tilde{\alpha}$ according to (1.29). Because of the following equivalent formulation of (1.31),

$$(A - \alpha_k I)(A - \tau I)^{-1} = \frac{\tau - \alpha_k}{\tau} A(A - \tau I)^{-1} + \frac{\alpha_k}{\tau} I,$$

we note that, $\mathcal{K}_m\left((A - \alpha_k I)(A - \tau I)^{-1}, \mathbf{b}\right) \equiv \mathcal{K}_m\left(A(A - \tau I)^{-1}, \mathbf{b}\right)$, where the right Krylov space, again, does not depends on the shifts. The choice of τ in (1.31) yields some freedom that we exploit in detail in Section 3.

Multi-Preconditioning Techniques

It is also possible to apply several shift-and-invert preconditioners,

$$\mathcal{P}(\tau_j) := (A - \tau_j I), \quad j = 1, ..., J, \tag{1.32}$$

within a flexible variant of multi-shift GMRES [59, 119]. In [119], the authors point out that the number of preconditioners J can be significantly smaller than the number of shifted systems N_α. When a preconditioner (1.32) that differs per iteration is applied, additional storage for the new search space Z_m is required and the following shifted Arnoldi relation holds,

$$(A-\alpha_k I)Z_m =: (A-\alpha_k I)\left[\mathcal{P}(\tau_1)^{-1}\mathbf{v}_1, ..., \mathcal{P}(\tau_m)^{-1}\mathbf{v}_m\right] = V_{m+1}\left(\underline{\mathrm{H}}_m(T_m-\alpha_k I_m)+\underline{\mathrm{I}}_m\right),$$

where $T_m := \mathrm{diag}(\tau_1, ..., \tau_m)$ and $\tau_1, ..., \tau_m \in \{\tau_1, ..., \tau_J\}$ meaning that per GMRES iteration the preconditioner does not necessarily differ, i.e. $J \ll m$. In a more recent work, the authors suggest multipreconditioned multi-shift GMRES [6]. Here, preconditioners of the form (1.32) are applied *at once*, such that a block search space of the form,

$$\mathcal{Z}_m := \left[Z_1, ..., Z_m\right], \quad \text{with } Z_j = \left[\mathcal{P}(\tau_1)^{-1}V_j, ..., \mathcal{P}(\tau_J)^{-1}V_j\right],$$

it build. The block counterpart for the shifted Arnoldi relation then yields,

$$(A - \alpha_k I)\mathcal{Z}_m = \mathcal{V}_{m+1}\left(\underline{\mathcal{H}}_m(\mathcal{T}_m - \alpha_k I_m) + \underline{\mathcal{I}}_m\right),$$

where now $\mathcal{T}_m := \mathrm{blk\ diag}(T_1, ..., T_J)$, and $T_j = \mathrm{diag}(\tau_1, ..., \tau_1, ..., \tau_J, ..., \tau_J) \in \mathbb{C}^{J^j \times J^j}$, and $\underline{\mathcal{H}}_m$ being block-Hessenberg. However, growth of this flexible block search space $\mathcal{Z}_m$ is only linear in J as shown in [6]. Both methods fit the more general framework of rational Krylov methods [16]. Concerning the choice of $\{\tau_1, ..., \tau_J\}$ the authors give valuable experimental insight, cf. [6, 119].

Shifted Polynomial Preconditioners

Note that the (scaled) shift-and-invert matrix $\mathcal{P}_k$ (1.30) is a linear polynomial in A. In [69], the author notices that polynomials preserve shift-invariance and exploits this fact when constructing a linear polynomial preconditioner for shifted problems. For a general polynomial of degree n, $p_n(A) = \sum_{i=0}^n \gamma_i A^i$, it is shown in [1] that shift-invariance can be preserved in the following sense,

$$(A - \alpha_k I)p_{n,k}(A) = Ap_n(A) - \tilde{\alpha}_k I, \tag{1.33}$$

where $p_{n,k}(A) = \sum_{i=0}^n \gamma_{i,k} A^i$ is a polynomial preconditioner of degree n for the k-th shifted matrix $(A - \alpha_k I)$. In [1] formulas for the coefficients $\gamma_{i,k}$ and the new shifts $\tilde{\alpha}_k$ such that (1.33) holds are derived. Moreover, the authors of [1] suggest to use a Chebyshev polynomial for p_n.

A well-known theoretical result is the Neumann series $A^{-1} = \sum_{i=0}^\infty (I - A)^i$ which converges if the spectral radius of $(I - A)$ is less than 1. In practice, the Neumann preconditioner [117, Chapter 12.3] of degree n as an approximation to the inverse of A, i.e. $A^{-1} \approx \sum_{i=0}^n (I - \xi A)^i$, with free parameter ξ. In [160], the authors construct a truncated Neumann polynomial for each shifted matrix such that shift-invariance is preserved. In Subsection 3.4.1, we construct a Neumann polynomial for the (fixed) base matrix $A(A-\tau I)^{-1}$ in (1.31) with ξ chosen such that the spectral radius minimized. Based on the general framework derived in [1] shift-invariance can be preserved in this framework as well. The insight that a polynomial preconditioner can preserve shift-invariance in combination with the algorithm suggested by [119] triggered our research on inner-outer multi-shift Krylov methods presented in Section 2.

Recycling and Augmented Krylov Subspace Techniques

Sequences of shifted systems (1.25) can be viewed as a special case of the more general framework of sequences of linear systems, $A_k\mathbf{x}_k = \mathbf{b}$, for $k = 1, ..., N_\alpha$. The work of [29] and [93] addresses the fundamental question which subspace to reuse when GMRES is restarted. This question becomes more difficult when sequences of linear systems are considered and information from previous solves is reused, cf. [54, 98]. Suppose we want to recycle information from the subspace $\mathcal{U}$, the approximation of Recycled GMRES after m iterations is given by,

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{s}_m + \mathbf{t}_m, \qquad \text{with } \mathbf{s}_m \in \mathcal{U} \text{ and } \mathbf{t}_m \in \mathcal{K}_m(PA, \mathbf{r}_0),$$

where $\mathcal{U}$ contains information that is recycled (from previous solves) and P is the orthogonal projector onto $(A\mathcal{U})^\perp$. In [25], harmonic Ritz vectors are suggested for $\mathcal{U}$.

Recycling for sequences of shifted systems is addressed in [133]. A main result of [133] is the proof that projectors P preserve shift-invariance in the following situation,

$$\mathcal{K}_m(PA, \mathbf{v}) = \mathcal{K}_m(P(A-\alpha I), \mathbf{v}) \quad \text{iff. } \mathbf{v} \in (A\mathcal{U})^\perp. \tag{1.34}$$

Moreover, the author describes a procedure to compute a shifted augmented subspace $\mathcal{U}^{(k)}$ such that,

$$A\mathcal{U} = (A - \alpha_k I)\mathcal{U}^{(k)},$$

holds, and the same projector can be used in (1.34). This requires additional storage, and an approach with fixed memory is discussed in [133] as well. A block-version is discussed in [132], and, recently, an approach with general preconditioner applied to flexible multi-shifted GMRES [131] destroying shift-invariance (1.28) is presented.

1.4 Areas of Application: Frequency-Domain Full-Waveform Inversion

We want to briefly point the reader to a prominent field of application where the fast solution of the time-harmonic elastic wave equation at multiple frequencies is required. In seismic exploration, full-waveform inversion (FWI) is a computational approach to understand the structure of the earth subsurface. Therefore, reflected shock waves are measured (see Figure 1.1) and matched with simulation results in a least-squares sense, cf. the overview paper [153] and the references therein; and [94, 106] for frequency-domain FWI. From a mathematical point-of-view, let $N_\omega > 1$ and consider the PDE-constrained optimization problem,

$$\min_{\rho, c_p, c_s} \sum_{k=1}^{N_\omega} \|\hat{\mathbf{u}}_k(\rho, c_p, c_s) - \mathcal{F}(\mathbf{d}_k)\| \tag{1.35}$$

$$\text{subject to} \quad \mathcal{E}(\hat{\mathbf{u}}_k) = \mathbf{0} \quad \text{for all } k. \tag{1.36}$$

Here, $\mathcal{E}(\cdot)$ denotes the elastic wave equation in frequency-domain as described in Section 1.1 after spatial discretization and Fourier transform, i.e.,

$$\mathcal{E}(\hat{\mathbf{u}}_k) := (K + i\omega_k C - \omega_k^2 M)\hat{\mathbf{u}}_k - \hat{\mathbf{s}}.$$

The misfit function (objective function) (1.35) can differ for different FWI approaches and $\mathbf{d}_k$ denotes the measured reflection intensity for shock waves of frequency ω_k (in space-time). In particular, Parseval's identity allows to formulate (1.35) equivalently in time or frequency domain (see remark in Definition 1.2). Including multiple frequencies in the objective function (1.35) improves the problem statement, cf. [94]. The optimization problem (1.35)-(1.36) can be solved iteratively and typically requires the solution of the forward problem (1.36) at every iteration, cf. [91, 139].

1.5 Structure of this Thesis

This thesis concerns the efficient numerical solution of the time-harmonic elastic wave equation at multiple frequencies, hence, the forward problem in (1.35)-(1.36). In this work, the following main research questions are addressed:

1. Can we develop an efficient Krylov methods for the fast iterative solution of the elastic wave equation that works well for a (wide) range of frequencies?
2. Can we find a single preconditioner that can be applied efficiently and, at the same time, improves convergence of all linear systems that arise from different frequencies?
3. Can we implement an algorithm that exploits state-of-the-art matrix computations such that this preconditioner is efficient with respect to memory consumptions and computation costs?

The thesis is based to a large extend on the author's journal publications [8, 9, 10, 13]. The articles have been adapted in notation and partly extended. This thesis is structured as follows.

Chapter 2

In Chapter 2 we develop an efficient nested Krylov method for the simultaneous iterative solution of shifted linear systems. Therefore, the reformulation stated in Problem 1.6 is used. When solving shifted systems (1.25) with a nested method, it is particularly challenging to preserve the shift-invariance property (1.26). The framework that we have developed uses an inner multi-shift Krylov methods as a preconditioner for an outer, flexible multi-shift Krylov method. When the inner method yields collinear residuals, shift-invariance can be preserved after preconditioning in the sense of (1.28). In order to be able to use multi-shift IDR(s) as an inner method, we have developed a new IDR variant that guarantees collinear residuals. The inner-outer scheme avoids long recurrences by design which is demonstrated by numerical examples. This chapter is based on our publication [8].

Chapter 3

Chapter 3 is based on our work [9] and provides theoretical insight for the shift-and-invert preconditioner (1.30) when applied as a right preconditioner within multi-shift GMRES. We derive an analytic expression for the parameter τ in (1.31) that is optimal with respect to a convergence bound of multi-shift GMRES. Our spectral analysis is exploited as a second-level polynomial preconditioner for the multi-shift Problem 1.6 and as a spectral rotation in a matrix equation approach for Problem 1.8. Numerical experiments demonstrate the optimality of the newly derived optimal seed parameter in terms of GMRES iteration numbers.

Chapter 4

Chapter 4 addresses the fast implementation of the shift-and-invert preconditioner in a multi-frequency framework. Therefore, we use state-of-the-art multilevel sequentially semiseparable (MSSS) matrix computation that exploit the structure of the discretized elastic operator that stems from a Cartesian grid. We present an inexact block-LU factorization for $2D$ and $3D$ elastic problems based on our publication [13]. On the lowest level, inexact computations are performed that limit the grows of the off-diagonal rank. On higher levels, we exploit approximate Schur complements, and domain decomposition approaches, cf. Subsection 1.2.3. As a multi-frequency solver, our work [13] focuses on global IDR(s) for matrix equations, i.e. Problem 1.8.

Chapter 5

An extensive comparison study of the shifted systems approach (Problem 1.6) and the matrix equation reformulation (Problem 1.8) is presented in Chapter 5 and [10]. Our numerical experiments provide guidelines concerning the suitability of both approaches for multi-frequency wave propagation problems. In order to obtain an algorithm for the shift-and-invert preconditioner in 3D with linear computational complexity, we present recent improvements to [10] by means of a coarse grid correction in the newly added Section 5.3.

Chapter 2

Nested Krylov Methods for Shifted Linear Systems

Abstract. We consider Krylov subspace methods that are designed for sequences of shifted linear systems. For the efficient numerical solution of shifted problems, the shift-invariance property of the corresponding Krylov subspaces is used such that a Krylov basis is computed only once for all shifted systems. Preconditioners in general destroy this shift-invariance property. Known preconditioners that preserve the shift-invariance are the shift-and-invert preconditioner as well as polynomial preconditioners. In this work, we introduce a new approach to the preconditioning of multi-shift Krylov methods. In our new *nested* framework, we use an inner multi-shift Krylov method as a flexible preconditioner within an outer multi-shift Krylov method. In order to preserve the shift-invariance of the underlying Krylov subspaces, we require collinear residuals for the inner Krylov method. This new approach has been implemented for two possible combinations, namely FOM-FGMRES and IDR-FQMRIDR, and has been tested to various numerical examples arising from geophysical applications.

Introduction

We consider shifted linear systems with equal right-hand side of the form

$$(A - \sigma_k I)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\sigma, \tag{2.1}$$

This chapter is based on the journal article:

M. Baumann and M.B. van Gijzen (2015). *Nested Krylov methods for shifted linear systems.* SIAM Journal of Scientific Computing, **37**(5), S90–S112.

where the dimensions are $A \in \mathbb{C}^{N\times N}, \mathbf{x}_k \in \mathbb{C}^N, \mathbf{b} \in \mathbb{C}^N$ and N_σ denotes the number of distinct shifts $\sigma_k \in \mathbb{C}$. For simplicity, we will often write

$$(A - \sigma I)\mathbf{x}^{(\sigma)} = \mathbf{b}, \quad \sigma \in \mathbb{C}, \tag{2.2}$$

keeping in mind that we aim to solve (2.2) for a sequence of many shifts σ and that quantities with a superscript belong to the respective shifted system, i.e. $\mathbf{x}^{(\sigma)}$ is the solution of the linear system with system matrix $(A - \sigma I)$ and right-hand side $\mathbf{b}$.

For an early overview on the numerical solution of shifted linear systems using Krylov methods we refer to [69, 127]. Multi-shift variants exist for many Krylov methods, for example QMR [48], GMRES(k) [50], FOM(k) [124], BiCGstab(ℓ) [49], CG [142], MINRES [72] and, more recently developed, IDR(s) [33, 148] and QMR-CGstab [90].

A known preconditioning technique for shifted linear systems is the so-called shift-and-invert preconditioner of the form $(A - \tau I)$ where the *seed shift* $\tau \in \mathbb{C}$ has to be chosen carefully. This preconditioner has been applied to shifted Helmholtz problems for example in [44]. Since the shift-and-invert matrix has to be solved by a direct method, this approach can be computationally costly. This can be overcome by either a multigrid approach [45, 121] or by an approximation of the shift-and-invert preconditioner using a polynomial preconditioner as shown in [1]. Most recently, multiple shift-and-invert preconditioners have been combined in a flexible Krylov method in order to capture a wider range of shifts $\sigma_1, ..., \sigma_{N_\sigma}$, cf. [59, 119]. Our article is motivated by the question of whether we can use a Krylov method as a polynomial preconditioner within a flexible method. We will present two new algorithms that are both nested Krylov methods in the sense that an inner, collinear Krylov method is used as a (polynomial) preconditioner to solve shifted linear systems within a flexible outer Krylov iteration.

Our article is organized as follows: In Section 2.1, we review the multi-shift GMRES [50] and the multi-shift QMRIDR [148] algorithms without preconditioning. The flexible versions of both are used as an outer Krylov method for the new nested framework in Section 2.4 and 2.5, respectively. As for the inner, collinear Krylov method, we present the multi-shift FOM algorithm [124] that automatically leads to collinear residuals in Section 2.3.1. In order to use IDR(s) as an inner method, we present a new collinear variant in Section 2.3.2. The article concludes with various numerical tests in Section 2.6.

2.1 Review of Two Multi-Shift Krylov Algorithms

The main property that is used in Krylov subspace methods for shifted linear systems, is the shift-invariance of the Krylov subspaces that are generated by the matrix A and the shifted matrix $(A - \sigma I)$ when the same right-hand side $\mathbf{b}$ is used, i.e.

$$\mathcal{K}_m(A, \mathbf{b}) \equiv \operatorname{span}\{\mathbf{b}, A\mathbf{b}, ..., A^{m-1}\mathbf{b}\} = \mathcal{K}_m(A - \sigma I, \mathbf{b}) \quad \forall \sigma \in \mathbb{C}. \tag{2.3}$$

The immediate consequence of this invariance-property is that a basis for the underlying Krylov spaces only has to be built once for all shifted systems. Note that relation (2.3) also holds for collinear starting vectors but, in general, different right-hand sides in (2.1) destroy the shift-invariance.

Multi-Shift GMRES

The well-known GMRES method [118] can be adapted to shifted systems in a straightforward way. In [50], a restarted version of multi-shift GMRES has been developed that relies on collinear residuals at restart. In this section, we review the main ideas of [50] and point out how the shift-invariance property (2.3) is used in the algorithm in order to speed-up the computational performance when solving shifted systems numerically.

In the classical GMRES method for the unshifted system $A\mathbf{x} = \mathbf{b}$, an orthogonal basis of the m-th Krylov subspace is obtained by the Arnoldi method. This leads to the well-known Hessenberg relation [138, eqn. 33.3],

$$AV_m = V_{m+1}\underline{\mathrm{H}}_m, \tag{2.4}$$

where the columns of $V_m \in \mathbb{C}^{N\times m}$ span an orthonormal basis of $\mathcal{K}_m(A, \mathbf{b})$, and $\underline{\mathrm{H}}_m \in \mathbb{C}^{(m+1)\times m}$ is the respective Hessenberg matrix with entries h_{ij} that are uniquely determined by the Arnoldi iteration. Then, in classical GMRES, an approximation to the solution of $A\mathbf{x} = \mathbf{b}$ in the m-th iteration is given by

$$\mathbf{x}_m = V_m\mathbf{y}_m, \quad \text{where } \mathbf{y}_m = \underset{\mathbf{y}\in\mathbb{C}^m}{\operatorname{argmin}} \|\underline{\mathrm{H}}_m\mathbf{y} - \|\mathbf{b}\|\mathbf{e}_1\|, \tag{2.5}$$

with $\mathbf{e}_1$ being the first unit vector of $\mathbb{C}^{m+1}$, and $\mathbf{x}_0 = 0$. For simplicity, we will assume the initial guess to be equal to zero throughout the whole document. The optimization problem in (2.5) can be solved efficiently due to the Hessenberg structure of $\underline{\mathrm{H}}_m$ using for instance Givens rotations, cf. [57, Section 5.1.8]. Clearly, we see from (2.5) that $\mathbf{x}_m \in \mathcal{K}_m(A, \mathbf{b})$.

Because of the shift-invariance property (2.3), the matrix V_m which spans the m-th Krylov subspace can be re-used for any shift σ. Therefore, the Arnoldi iteration in multi-shift GMRES only needs to be performed once, and from the shifted Hessenberg relation,

$$(A - \sigma I)V_m = V_{m+1}(\underline{\mathrm{H}}_m - \sigma\underline{\mathrm{I}}_m),$$

we can derive an approximated solution to the shifted problem (2.2) via,

$$\mathbf{x}_m^{(\sigma)} = V_m\mathbf{y}_m^{(\sigma)}, \quad \text{where } \mathbf{y}_m^{(\sigma)} = \underset{\mathbf{y}\in\mathbb{C}^m}{\operatorname{argmin}} \|(\underline{\mathrm{H}}_m - \sigma\underline{\mathrm{I}}_m)\mathbf{y} - \|\mathbf{b}\|\mathbf{e}_1\|, \tag{2.6}$$

where $\underline{\mathrm{I}}_m$ is the identity matrix of size $m \times m$ with an extra zero row appended at the bottom. We note that the matrix $\underline{\mathrm{H}}_m(\sigma) \equiv \underline{\mathrm{H}}_m - \sigma\underline{\mathrm{I}}_m$ for the shifted system is of Hessenberg structure as well. Clearly, we get the original Hessenberg matrix of the unshifted problem back if $\sigma = 0$, i.e. $\underline{\mathrm{H}}_m(0) = \underline{\mathrm{H}}_m$.

By comparing (2.5) and (2.6), we note that the m-th iterate lies in both cases in the column space of the matrix V_m and, therefore, lies in the same Krylov subspace $\mathcal{K}_m(A, \mathbf{b})$. Moreover, we note from the derivation of the shifted Hessenberg matrix $\underline{\mathrm{H}}_m(\sigma)$ that the shift of the matrix A directly translates into a shift of the Hessenberg matrix.

In order to allow restarting for multi-shift GMRES, the authors of [50] require collinear residuals in order to preserve shift-invariance of the respective Krylov spaces after restart, cf. [50, Algorithm 2.4]. A numerically more robust implementation of restarted multi-shift GMRES has recently been proposed in [160].

Multi-Shift QMRIDR

The QMRIDR method presented in [148] is a variant of the Induced Dimension Reduction (IDR) method [130] that makes use of a so-called generalized Hessenberg decomposition and determines the m-th iterate via a quasi-minimization of the m-th residual. In [60, 148], the following relation is derived,

$$AG_mU_m = G_{m+1}\underline{\mathrm{H}}_m, \tag{2.7}$$

where $U_m \in \mathbb{C}^{m\times m}$ is upper triangular, $\underline{\mathrm{H}}_m \in \mathbb{C}^{(m+1)\times m}$ is of Hessenberg form, and $s+1$ consecutive vectors in G_m belong to the nested Sonneveld spaces $\mathcal{G}_0, ..., \mathcal{G}_j$. The entries of U_m, G_m and $\underline{\mathrm{H}}_m$ are uniquely determined by the specific IDR algorithm [130, 146], cf. [60] for a detailed derivation. Based on the generalized Hessenberg decomposition (2.7), a multi-shift IDR method, the QMRIDR(s) algorithm, has been derived [148].

The approach of QMRIDR(s) is very similar to the GMRES approach. Therefore, the m-th iterate is constructed as a linear combination of the columns of G_m by putting $\mathbf{x}_m = G_mU_m\mathbf{y}_m$, with a coefficient vector $\mathbf{y}_m \in \mathbb{C}^m$ that is determined via a least-squares problem that involves the Hessenberg matrix $\underline{\mathrm{H}}_m$ only. Altogether, the following minimization problem needs to be solved,

$$\mathbf{x}_m = G_mU_m\mathbf{y}_m, \quad \text{where } \mathbf{y}_m = \underset{\mathbf{y}\in\mathbb{C}^m}{\operatorname{argmin}} \|\underline{\mathrm{H}}_m\mathbf{y} - \|\mathbf{b}\|\mathbf{e}_1\|, \tag{2.8}$$

which is called 'quasi-minimization' of the m-th residual because the matrix G_m does not have orthogonal columns, cf. [148].

For shifted linear systems of the form (2.2), a very similar relation holds,

$$\mathbf{x}_m^{(\sigma)} = G_mU_m\mathbf{y}_m^{(\sigma)}, \quad \text{where } \mathbf{y}_m^{(\sigma)} = \underset{\mathbf{y}\in\mathbb{C}^m}{\operatorname{argmin}} \|(\underline{\mathrm{H}}_m - \sigma\underline{\mathrm{U}}_m)\mathbf{y} - \|\mathbf{b}\|\mathbf{e}_1\|, \tag{2.9}$$

and by comparing (2.8) with (2.9), we note that the approximate solutions to the respective systems lie in the same subspace and the matrix $\underline{\mathrm{H}}_m(\sigma) \equiv \underline{\mathrm{H}}_m - \sigma\underline{\mathrm{U}}_m$ is again of Hessenberg structure since $\underline{\mathrm{U}}_m$ consists of the upper triangular matrix U_m derived in [148] with an extra zero row appended.

From the derivations of multi-shift GMRES in Section 2.1 and of multi-shift QMRIDR in this section, we note that in both cases the efficient computation of the Hessenberg matrix of the shifted system $\underline{\mathrm{H}}_m(\sigma)$ is crucial for the design of the algorithm. Therefore, we will put emphasis on the computation of $\underline{\mathrm{H}}_m(\sigma)$ as a function of $\underline{\mathrm{H}}_m$ in the nested Krylov framework in Section 2.4 and Section 2.5 as well.

2.2 A Prototype Inner-Outer Krylov Method for Shifted Problems

This section is structured as follows: We will first point out the requirements of a single preconditioner for shifted linear systems that preserves the shift-invariance property of the Krylov subspaces in (2.3). In this work, we restrict ourselves to such types of preconditioners that preserve the shift-invariance, but we would like to mention the promising approaches of [15, 131, 133] which do not rely on shift-invariance after preconditioning. Based on the requirements of a single preconditioner that preserves shift-invariance, we design a flexible preconditioner in Subsection 2.2.2 that requires collinear residuals for the *inner* iteration. In Subsection 2.3, we present two Krylov methods that lead to collinear residuals; namely the FOM method that produces collinear residuals automatically and a new variant of the IDR(s) method where some modifications are necessary in order to obtain collinear residuals. Both methods are used as preconditioners in a flexible Krylov method in Subsection 2.4 and 2.5, respectively.

2.2.1 The Single Shift-And-Invert Preconditioner

In order to precondition a shifted linear system (2.2) without destroying the shift-invariance property of the respective Krylov spaces, we require the following equality after preconditioning,

$$\mathcal{K}_m((A-\sigma I)\mathcal{P}(\sigma)^{-1}, \mathbf{b}) = \mathcal{K}_m(A\mathcal{P}^{-1}, \mathbf{b}), \tag{2.10}$$

where $\mathcal{P}(\sigma)$ is a different preconditioner for every considered shifted system, and $\mathcal{P}$ is a preconditioner for the unshifted system $A\mathbf{x} = \mathbf{b}$, cf. [1, 69]. Relation (2.10) is satisfied if we find a parameter η that depends on the shift σ and a constant matrix $\mathcal{P}$ such that,

$$(A-\sigma I)\mathcal{P}(\sigma)^{-1} = A\mathcal{P}^{-1} - \eta(\sigma)I, \tag{2.11}$$

which in fact means that we can write the preconditioned shifted systems as shifted preconditioned systems with new shifts $\eta(\sigma)$. From [87, 88, 125], it is well-known that the so-called shift-and-invert preconditioner $\mathcal{P} \equiv (A - \tau I)$ applied to (2.11) leads to:

$$\begin{aligned}(A-\sigma I)\mathcal{P}(\sigma)^{-1} &= A(A-\tau I)^{-1} - \eta(\sigma)I \\ &= (1-\eta(\sigma))(A + \underbrace{\frac{\tau\eta(\sigma)}{1-\eta(\sigma)}}_{\equiv -\sigma} I)(A-\tau I)^{-1}.\end{aligned}$$

By choosing $\eta(\sigma)$ in an appropriate way, we can factor out the term $(A - \sigma I)$ on both sides which yields the following formulas:

$$\eta(\sigma) = \frac{\sigma}{\sigma-\tau}, \quad \mathcal{P}(\sigma) = \frac{1}{1-\eta(\sigma)}\mathcal{P} = \frac{\tau-\sigma}{\tau}\mathcal{P},$$

where the dependence of $\mathcal{P}(\sigma)$ on the shift becomes explicit. Therefore, only the seed shift τ has to be chosen and in order to invert $\mathcal{P}(\sigma)$ we only need to decompose $\mathcal{P}$. However, the one-time decomposition of $\mathcal{P}$ can numerically be very costly and the suitable choice of the seed shift τ is difficult for a large range of shifts $\sigma_1, ..., \sigma_{N_\sigma}$, as has been pointed out in [119]. In [1, 160], polynomial preconditioners are suggested that cheaply approximate $\mathcal{P}^{-1}$. Note that the right-hand side in (2.11) defines a new shifted problem for which reason the single shift-and-invert preconditioner can be applied as a *first layer* in a our later algorithm.

We remark that also in the more general case of shifted problems $(A - \sigma B)$ with a mass matrix $B \neq I$, the shift-and-invert preconditioner $\mathcal{P} = A - \tau B$ can be used. Applying this preconditioner yields a sequence of preconditioned shifted systems with a shifted identity in the same way as in the right-hand side of (2.11), cf. [87].

2.2.2 Flexible Preconditioning for Shifted Linear Systems

Flexible preconditioning of an iterative Krylov subspace method means that a different preconditioner can be used in every iteration j, see [115] where flexible GMRES (FGMRES) has been introduced for preconditioning systems of the form $A\mathbf{x} = \mathbf{b}$. For flexible preconditioning of shifted linear systems, we require a very similar relation to (2.11), namely,

$$(A - \sigma I)\mathcal{P}_j(\sigma)^{-1} = \alpha_j(\sigma)A\mathcal{P}_j^{-1} - \beta_j(\sigma)I, \tag{2.12}$$

where the parameters α_j and β_j will depend on the shift, and different preconditioners $\mathcal{P}_j$ and $\mathcal{P}_j(\sigma)$ are used in every iteration j. Note that the right-hand side in (2.12) is a shifted coefficient matrix and, thus, the shift-invariance is preserved by the flexible preconditioner. Since in a practical algorithm, the preconditioner is always directly applied to a vector $\mathbf{v}_j$, equation (2.12) translates into:

$$(A - \sigma I)\mathcal{P}_j(\sigma)^{-1}\mathbf{v}_j = \alpha_j(\sigma)A\mathcal{P}_j^{-1}\mathbf{v}_j - \beta_j(\sigma)\mathbf{v}_j. \tag{2.13}$$

We will next determine how α_j and β_j have to be chosen such that (2.12) and (2.13) hold. Therefore, we assume the preconditioning to be done by a multi-shift Krylov method itself (the *inner* method) which means that

$$\mathbf{z}_j = \mathcal{P}_j^{-1}\mathbf{v}_j, \quad \mathbf{z}_j^{(\sigma)} = \mathcal{P}_j(\sigma)^{-1}\mathbf{v}_j,$$

are computed via a truncated multi-shift Krylov method at step j. More precisely, the vectors $\mathbf{z}_j$ and $\mathbf{z}_j^{(\sigma)}$ denote the approximate (truncated) solution of the linear systems with system matrix A and $(A-\sigma I)$, respectively, and same right-hand side $\mathbf{v}_j$. Hence, the corresponding (inner) residuals are given by,

$$\mathbf{r}_j = \mathbf{v}_j - A\mathbf{z}_j = \mathbf{v}_j - A\mathcal{P}_j^{-1}\mathbf{v}_j, \tag{2.14}$$

$$\mathbf{r}_j^{(\sigma)} = \mathbf{v}_j - (A - \sigma I)\mathbf{z}_j^{(\sigma)} = \mathbf{v}_j - (A - \sigma I)\mathcal{P}_j(\sigma)^{-1}\mathbf{v}_j. \tag{2.15}$$

We require the residuals (2.14)-(2.15) of the inner method to be collinear, i.e.

$$\exists \gamma_j^{(\sigma)} \in \mathbb{C}: \quad \mathbf{r}_j^{(\sigma)} = \gamma_j^{(\sigma)}\mathbf{r}_j. \tag{2.16}$$

Note that the collinearity factor $\gamma_j^{(\sigma)}$ will be different at every iteration j and for every shift σ. Moreover, relation (2.16) is not a very strong assumption since for example multi-shift FOM [124], multi-shift BiCGstab [49], restarted multi-shift GMRES [50] and multi-shift IDR [33, 74] yield collinear residuals. With this assumption, α_j and β_j can be determined from (2.13) by using the collinearity relation (2.16) as the following calculation shows:

$$\begin{aligned}
(A - \sigma I)\mathbf{z}_j^{(\sigma)} &= \alpha_j A\mathbf{z}_j - \beta_j\mathbf{v}_j \\
\Leftrightarrow \quad \mathbf{v}_j - (A - \sigma I)\mathbf{z}_j^{(\sigma)} &= \alpha_j\mathbf{v}_j - \alpha_j A\mathbf{z}_j + (1 - \alpha_j + \beta_j)\mathbf{v}_j \\
\Leftrightarrow \quad \mathbf{r}_j^{(\sigma)} &= \underbrace{\alpha_j}_{\overset{!}{=}\gamma_j^{(\sigma)}}\mathbf{r}_j + \underbrace{(1 - \alpha_j + \beta_j)}_{\overset{!}{=}0}\mathbf{v}_j
\end{aligned}$$

Thus, with (2.13) the residuals are collinear if we choose

$$\alpha_j = \gamma_j^{(\sigma)}, \quad \beta_j = \alpha_j - 1 = \gamma_j^{(\sigma)} - 1$$

in every (outer) iteration $1 \leq j \leq m$. We show the following relation,

$$(A - \sigma I)\mathbf{z}_j^{(\sigma)} = \gamma_j^{(\sigma)} A\mathbf{z}_j - \left(\gamma_j^{(\sigma)} - 1\right) \mathbf{v}_j, \tag{2.17}$$

or, in terms of the flexible preconditioners $\mathcal{P}_j$ and $\mathcal{P}_j(\sigma)$, the following holds:

$$(A - \sigma I)\mathcal{P}_j(\sigma)^{-1}\mathbf{v}_j = \left(\gamma_j^{(\sigma)} A\mathcal{P}_j^{-1} - \left(\gamma_j^{(\sigma)} - 1\right) I\right) \mathbf{v}_j, \quad 1 \leq j \leq m.$$

Note that the factors α_j and β_j do depend on σ since the collinearity factors $\gamma_j^{(\sigma)}$ are different for every shift.

2.3 Shifted Krylov Methods with Collinear Residuals

In the previous section, we have derived the theoretical basis for a nested Krylov method for shifted linear systems. In order to be able to design a preconditioner that preserves the shift-invariance of the corresponding Krylov spaces, we assumed collinear residuals (2.16) for the inner multi-shift method. Next, we will present two multi-shift Krylov methods that lead to collinear residuals and, therefore, fulfill assumption (2.16). It is well-known from [124] that the multi-shift version of the Full Orthogonalization Method (FOM) leads to collinear residuals. We will review this result in Subsection 2.3.1. Moreover, we describe a new variant of the Induced Dimension Reduction (IDR) method that has collinear residuals in Subsection 2.3.2.

2.3.1 Collinear Residuals in Multi-Shift FOM

The multi-shift FOM method (`msFOM`) can be derived very similarly to the multi-shift GMRES method in Section 2.1, cf. [124]. In FOM, an orthogonal basis of $\mathcal{K}_m(A, \mathbf{b})$ is obtained via the Arnoldi iteration which yields,

$$V_m^{\mathsf{H}} A V_m = H_m,$$

and can be derived from (2.4) by left-multiplication with V_m^{H}, where H_m is a square matrix and the superscript H denotes Hermitian transpose. Assuming H_m is invertible, the m-th iterate is then obtained by

$$\mathbf{x}_m = V_m \mathbf{y}_m, \quad \text{where } \mathbf{y}_m = H_m^{-1}(\beta \mathbf{e}_1),$$

where, for simplicity, we assumed $\mathbf{x}_0 = 0$ as an initial guess, and $\beta \equiv \|\mathbf{r}_0\| = \|\mathbf{b}\|$. Similarly to multi-shift GMRES, the shifted Hessenberg matrix in `msFOM` can be derived as $H_m(\sigma) = H_m - \sigma I_m$, see [124]. The complete multi-shift FOM algorithm is repeated in Algorithm 2.1. It is well-known that the `msFOM` method as presented in Algorithm 2.1 leads to collinear residuals of the shifted system and the original ($\sigma = 0$) system. We repeat this result in the following Lemma.

Algorithm 2.1 `msFOM` for $(A-\sigma_k I)\mathbf{x}_k=\mathbf{b},\ k=1,...,N_\sigma$, [124]

```
 1: Initialize $\mathbf{r}_0=\mathbf{b},\ \beta=\|\mathbf{r}_0\|,\ \mathbf{v}_1=\mathbf{r}_0/\beta$      ▷ Initialization with $\mathbf{x}_0=\mathbf{0}$
 2: for $j=1$ to $m$ do
 3:     Compute $\mathbf{w}=A\mathbf{v}_j$
 4:     for $i=1$ to $j$ do                                             ▷ Arnoldi method
 5:         $h_{i,j}=\mathbf{w}^{\mathsf{H}}\mathbf{v}_i$
 6:         $\mathbf{w}=\mathbf{w}-h_{i,j}\mathbf{v}_i$
 7:     end for
 8:     Set $h_{j+1,j}=\|\mathbf{w}\|$ and $\mathbf{v}_{j+1}=\mathbf{w}/h_{j+1,j}$
 9: end for
10: Solve $\mathbf{y}_m=H_m^{-1}(\beta\mathbf{e}_1)$                              ▷ The unshifted case
11: Set $\mathbf{x}_m=V_m\mathbf{y}_m$
12: for $k=1$ to $N_\sigma$ do                                            ▷ Loop over shifts
13:     Construct $H_m(\sigma_k)=H_m-\sigma_k I_m$                ▷ Form shifted Hessenberg matrices
14:     Solve $\mathbf{y}_m^{(\sigma_k)}=H_m(\sigma_k)^{-1}(\beta\mathbf{e}_1)$
15:     Set $\mathbf{x}_m^{(\sigma_k)}=V_m\mathbf{y}_m^{(\sigma_k)}$        ▷ Note that $\mathbf{x}_m^{(\sigma_k)}\in\mathcal{K}_m(A,\mathbf{b})$ for all $\sigma_k$
16: end for
```

Lemma 2.1 (Collinearity of the residuals in Algorithm 2.1) *We use the notation of Algorithm 2.1. Let the respective residuals of the original and the shifted system after m steps be*

$$\mathbf{r}_m\equiv\mathbf{b}-A\mathbf{x}_m,$$
$$\mathbf{r}_m^{(\sigma)}\equiv\mathbf{b}-(A-\sigma I)\mathbf{x}_m^{(\sigma)}.$$

Then there exists a scalar $\gamma^{(\sigma)}$ that depends on the number of performed `msFOM` *iterations m, and the shift σ such that residuals are collinear, i.e.,*

$$\mathbf{r}_m^{(\sigma)}=\gamma^{(\sigma)}\mathbf{r}_m.$$

Proof 2.2 *This proof can be found in [117, Proposition 6.7] as well as in [124]. For the residual of the original system after m iterations of FOM (lines 1–11 in Algorithm 2.1), it holds,*

$$\begin{aligned}\mathbf{r}_m&=\mathbf{b}-A\mathbf{x}_m=\mathbf{b}-AV_m\mathbf{y}_m=\mathbf{r}_0-AV_m\mathbf{y}_m\\&=\underbrace{\beta\mathbf{v}_1-V_mH_m\mathbf{y}_m}_{=0}-h_{m+1,m}\mathbf{e}_m^{\mathsf{H}}\mathbf{y}_m\mathbf{v}_{m+1}=-h_{m+1,m}\mathbf{e}_m^{\mathsf{H}}\mathbf{y}_m\mathbf{v}_{m+1}.\end{aligned}$$

Here, the first two terms cancel due to the imposed Galerkin condition in FOM, i.e. $\mathbf{r}_m\perp\mathcal{K}_m(A,\mathbf{b})\Leftrightarrow V_m^{\mathsf{H}}\mathbf{r}_m=\mathbf{0}$. Repeating the same calculation for the shifted system yields,

$$\mathbf{r}_m^{(\sigma)}=-h_{m+1,m}^{(\sigma)}\mathbf{e}_m^{\mathsf{H}}\mathbf{y}_m^{(\sigma)}\mathbf{v}_{m+1},$$

with $\mathbf{e}_m\equiv[0,...,0,1]^{\mathsf{H}}\in\mathbb{C}^m$, and the Arnoldi vector $\mathbf{v}_{m+1}$ is identical to the unshifted case. Since the off-diagonal elements of the shifted Hessenberg matrix are identical to *the unshifted Hessenberg matrix in Algorithm 2.1, and the orthogonal basis vectors* $\mathbf{v}_i$ *obtained by the Arnoldi iteration are identical, we conclude,*

$$\gamma^{(\sigma)} = y_m^{(\sigma)}/y_m, \tag{2.18}$$

with $y_m, y_m^{(\sigma)}$ *being the last entry of the vectors* $\mathbf{y}_m, \mathbf{y}_m^{(\sigma)}$*, respectively. The residuals are collinear with the collinearity factor* $\gamma^{(\sigma)}$ *explicitly given by (2.18).* ☐

The above lemma shows that `msFOM` is suitable as a preconditioner for the nested framework derived in Section 2.2.2. The required collinearity factor $\gamma_j^{(\sigma)}$ of the j-th outer iteration in (2.17) is given by (2.18), where we assume that `msFOM` is stopped after m inner iterations

2.3.2 A New Variant of Multi-Shift IDR(s) with Collinear Residuals

The IDR(s) method as presented in [130] is a Krylov subspace method that is based on the idea that the residual is forced to lie in spaces $\mathcal{G}_j$ of shrinking dimensions as the number of iterations increases. In more detail, we require the residual $\mathbf{r}_{n+1}$ to fulfill,

$$\mathcal{G}_{j+1} \ni \mathbf{r}_{n+1} = (I - \omega_{j+1}A)\mathbf{v}_n, \quad \text{with } \mathbf{v}_n \in \mathcal{G}_j \cap \mathcal{S},\ \omega_{j+1} \in \mathbb{C} \setminus \{0\}, \tag{2.19}$$

with $\mathcal{G}_0 \equiv \mathcal{K}_N(A, \mathbf{b})$, and, without loss of generality, let $\mathcal{S} = \mathcal{N}(P^{\mathsf{H}})$ be the null space of some $N \times s$ matrix $P = [\mathbf{p}_1, ..., \mathbf{p}_s]$. It is known from the IDR theorem [130, Theorem 2.1] that the spaces $\mathcal{G}_j$ that are generated via the recursive definition (2.19) are of decreasing dimension and that, eventually, $\mathcal{G}_j = \{0\}$ for some $j \leq N$. This result guarantees that in exact arithmetic the residual will be equal to zero at some point. Moreover, note that the scalars ω_{j+1} in (2.19) can be chosen freely, which we will exploit in the following in order to derive a collinear variant of the IDR(s) method.

According to [130], the vectors $\mathbf{v}_n \in \mathcal{G}_j \cap \mathcal{S}$ can be computed in the following way,

$$(P^{\mathsf{H}} \Delta R_n)\mathbf{c} = P^{\mathsf{H}}\mathbf{r}_n, \tag{2.20}$$

$$\mathbf{v}_n = \mathbf{r}_n - \Delta R_n \mathbf{c}, \tag{2.21}$$

with $\Delta R_n \equiv [\Delta\mathbf{r}_{n-1}, ..., \Delta\mathbf{r}_{n-s}]$, and residual difference $\Delta\mathbf{r}_{n-1} \equiv \mathbf{r}_n - \mathbf{r}_{n-1}$. In a similar way, we denote the matrix of the last $s+1$ residuals by $R_n \equiv [\mathbf{r}_n, ..., \mathbf{r}_{n-s}]$. From the last two definitions, we note that the residual updates can be expressed as a product of the actual residuals and a difference matrix D in the following way,

$$[\mathbf{r}_n, -\Delta R_n] = R_n D, \tag{2.22}$$

where D is defined as the invertible matrix:

$$D \equiv \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & \ddots & -1 \\ & & & 1 \end{bmatrix} \in \mathbb{C}^{(s+1)\times(s+1)}.$$

Note that (2.22) holds in the same way for residuals of the shifted systems.

A first approach to adapt the IDR method to shifted linear systems that leads to collinear residuals has been done by [74]. The algorithm presented in [74] follows the classical approach of evaluating the shifted residual polynomial. For the collinear IDR-variant presented in this section, we derive a purely algebraic approach that exploits the IDR theorem and an implementation of the shifted algorithm that is very closely related to the first IDR version in [130]. After finishing this manuscript, we discovered that the same approach but a different implementation of it was independently followed by [33].

We first note that collinear residuals $\mathbf{r}_{n+1}$ and $\mathbf{r}_{n+1}^{(\sigma)}$ will lie in the same Sonneveld spaces. We therefore aim to construct the spaces $\mathcal{G}_j$ only once for all shifted systems. Moreover, we express the collinearity via a diagonal matrix $\tilde{\Gamma}_n^{(\sigma)} \equiv diag(\gamma_n^{(\sigma)}, ..., \gamma_{n-s}^{(\sigma)})$ that consists of the last $s+1$ consecutive collinearity factors such that it holds,

$$R_n^{(\sigma)} = R_n \tilde{\Gamma}_n^{(\sigma)}, \tag{2.23}$$

where $R_n^{(\sigma)} \equiv [\mathbf{r}_n^{(\sigma)}, ..., \mathbf{r}_{n-s}^{(\sigma)}]$ is constructed in the same way as R_n.

Our approach can be described in two phases: First, we note from (2.19), that in order to obtain collinear residuals, we need to produce collinear vectors $\mathbf{v}_n$ and $\mathbf{v}_n^{(\sigma)}$. Therefore, we want to calculate the intermediate vector $\mathbf{c}^{(\sigma)}$ in (2.20) of the shifted systems such that $\mathbf{v}_n^{(\sigma)} = \alpha^{(\sigma)} \mathbf{v}_n$, with $\alpha^{(\sigma)} \in \mathbb{C}$ to be determined. In the following calculation, we make use of (2.22) and (2.23) in order to manipulate relation (2.21) for the shifted system:

$$\begin{aligned} \alpha^{(\sigma)} \mathbf{v}_n = \mathbf{v}_n^{(\sigma)} = \mathbf{r}_n^{(\sigma)} - \Delta R_n^{(\sigma)} \mathbf{c}^{(\sigma)} &= [\mathbf{r}_n^{(\sigma)}, -\Delta R_n^{(\sigma)}] \begin{bmatrix} 1 \\ \mathbf{c}^{(\sigma)} \end{bmatrix} \\ &= R_n^{(\sigma)} D \begin{bmatrix} 1 \\ \mathbf{c}^{(\sigma)} \end{bmatrix} = R_n \tilde{\Gamma}_n^{(\sigma)} D \begin{bmatrix} 1 \\ \mathbf{c}^{(\sigma)} \end{bmatrix} \\ &= [\mathbf{r}_n, -\Delta R_n] D^{-1} \tilde{\Gamma}_n^{(\sigma)} D \begin{bmatrix} 1 \\ \mathbf{c}^{(\sigma)} \end{bmatrix}. \end{aligned}$$

By comparing with (2.21), we obtain the following $(s+1) \times (s+1)$ system of equations,

$$D^{-1} \tilde{\Gamma}_n^{(\sigma)} D \begin{bmatrix} 1 \\ \mathbf{c}^{(\sigma)} \end{bmatrix} = \alpha^{(\sigma)} \begin{bmatrix} 1 \\ \mathbf{c} \end{bmatrix}, \tag{2.24}$$

where the vector $\mathbf{c}$ is known, and $\mathbf{c}^{(\sigma)} \in \mathbb{C}^s$ and $\alpha^{(\sigma)} \in \mathbb{C}$ can be uniquely determined. Note that in contrast to (2.20)-(2.21), we have computed $\mathbf{c}^{(\sigma)}$ and $\mathbf{v}_n^{(\sigma)} = \alpha^{(\sigma)} \mathbf{v}_n$ of the shifted systems without storing additional residual differences. In the second step of our approach, we determine the free IDR parameter $\omega_{j+1}^{(\sigma)}$ and the factor $\gamma^{(\sigma)}$ such that the residuals are collinear, i.e.

$$\mathbf{r}_{n+1}^{(\sigma)} = \gamma^{(\sigma)} \cdot \mathbf{r}_{n+1}. \tag{2.25}$$

Algorithm 2.2 `coll_IDR(s)` for $(A - \sigma_k I)\mathbf{x}_k = \mathbf{b},\ k = 1, ..., N_\sigma$

```
 1: Set $\mathbf{x}_0 = \mathbf{x}_0^{(\sigma_k)} = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b}$, and $\gamma_0^{(\sigma_k)} = 1$
 2: for $n = 0, ..., s-1$ do                                        ▷ Initialization phase
 3:     $\mathbf{v} = A\mathbf{r}_n$; $\omega = (\mathbf{v}^\mathsf{H}\mathbf{r}_n)/(\mathbf{v}^\mathsf{H}\mathbf{v})$
 4:     $\Delta\mathbf{x}_n = \omega\mathbf{r}_n$; $\Delta\mathbf{r}_n = -\omega\mathbf{v}$
 5:     for $k = 1, ..., N_\sigma$ do
 6:         $\gamma_{n+1}^{(\sigma_k)} = \gamma_n^{(\sigma_k)}/(1-\omega\sigma_k)$            ▷ Collinearity factors, cf. (2.26)
 7:         $\omega^{(\sigma_k)} = \omega/(1-\omega\sigma_k)$            ▷ Choice for IDR parameter, cf. (2.26)
 8:         $\Delta\mathbf{x}_n^{(\sigma_k)} = \omega^{(\sigma_k)}\gamma_n^{(\sigma_k)}\mathbf{r}_n$
 9:         $\mathbf{x}_{n+1}^{(\sigma_k)} = \mathbf{x}_n^{(\sigma_k)} + \Delta\mathbf{x}_n^{(\sigma_k)}$
10:     end for
11:     Update: $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta\mathbf{x}_n$; $\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta\mathbf{r}_n$
12: end for
13: $\Delta R_{n+1} := (\Delta\mathbf{r}_n, ..., \Delta\mathbf{r}_0)$; $\Delta X_{n+1} := (\Delta\mathbf{x}_n, ..., \Delta\mathbf{x}_0)$
14: $\Delta X_{n+1}^{(\sigma_k)} := (\Delta\mathbf{x}_n^{(\sigma_k)}, ..., \Delta\mathbf{x}_0^{(\sigma_k)})$; $\boldsymbol{\gamma}^{(\sigma_k)} := (\gamma_{n+1}^{(\sigma_k)}, ..., \gamma_0^{(\sigma_k)})$
15: $n = s$
16: while $\|\mathbf{r}_n\| > TOL$ and $n < MAXIT$ do
17:     for $j = 0, ..., s$ do
18:         Solve $\mathbf{c}$ from $P^\mathsf{H}\Delta R_n\mathbf{c} = P^\mathsf{H}\mathbf{r}_n$
19:         $\mathbf{v} = \mathbf{r}_n - \Delta R_n\mathbf{c}$
20:         for $k = 1, ..., N_\sigma$ do                                ▷ Loop over shifts
21:             Set $\tilde{\Gamma}_n^{(\sigma_k)} := diag(\boldsymbol{\gamma}^{(\sigma_k)})$
22:             Solve $[1, \mathbf{c}^{(\sigma_k)}]$ from $D^{-1}\tilde{\Gamma}_n^{(\sigma_k)}D[1, \mathbf{c}^{(\sigma_k)}] = \alpha^{(\sigma_k)}[1, \mathbf{c}]$
                            ▷ The solve in line 22 implies $\mathbf{v}^{(\sigma_k)} = \alpha^{(\sigma_k)}\mathbf{v}$ acc. to (2.24)
23:         end for
24:         if j == 0 then
25:             $\mathbf{t} = A\mathbf{v}$
26:             $\omega = (\mathbf{t}^\mathsf{H}\mathbf{v})/(\mathbf{t}^\mathsf{H}\mathbf{t})$; $\omega^{(\sigma_k)} = \omega/(1-\omega\sigma_k)$       ▷ IDR parameter, cf. (2.26)
27:             $\Delta\mathbf{x}_n = -\Delta X_n\mathbf{c} + \omega\mathbf{v}$; $\Delta\mathbf{r}_n = -\Delta R_n\mathbf{c} - \omega\mathbf{t}$
28:         else
29:             $\Delta\mathbf{x}_n = -\Delta X_n\mathbf{c} + \omega\mathbf{v}$; $\Delta\mathbf{r}_n = -A\Delta\mathbf{x}_n$
30:         end if
31:         Update: $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta\mathbf{x}_n$; $\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta\mathbf{r}_n$
32:         for $k = 1, ..., N_\sigma$ do                                ▷ Loop over shifts
33:             $\gamma_{n+1}^{(\sigma_k)} = \alpha^{(\sigma_k)}/(1-\omega\sigma_k)$            ▷ Collinearity factors, cf. (2.26)
34:             $\Delta\mathbf{x}_n^{(\sigma_k)} = -\Delta X_n^{(\sigma_k)}\mathbf{c}^{(\sigma_k)} + \omega^{(\sigma_k)}\alpha^{(\sigma_k)}\mathbf{v}$
35:             $\mathbf{x}_{n+1}^{(\sigma_k)} = \mathbf{x}_n^{(\sigma_k)} + \Delta\mathbf{x}_n^{(\sigma_k)}$
36:         end for
37:         // The IDR-update:
38:         $n = n + 1$
39:         $\Delta R_n := (\Delta\mathbf{r}_{n-1}, ..., \Delta\mathbf{r}_{n-s})$; $\Delta X_n := (\Delta\mathbf{x}_{n-1}, ..., \Delta\mathbf{x}_{n-s})$
40:         $\Delta X_n^{(\sigma_k)} := (\Delta\mathbf{x}_{n-1}^{(\sigma_k)}, ..., \Delta\mathbf{x}_{n-s}^{(\sigma_k)})$; $\boldsymbol{\gamma}^{(\sigma_k)} := (\gamma_n^{(\sigma_k)}, ..., \gamma_{n-s}^{(\sigma_k)})$
41:     end for
42: end while
```

Therefore, we substitute the definition of the residuals from (2.19) and use the collinearity of the vectors $\mathbf{v}_n$ and $\mathbf{v}_n^{(\sigma)}$:

$$\begin{aligned} & & \mathbf{r}_{n+1}^{(\sigma)} &= \gamma^{(\sigma)}\mathbf{r}_{n+1} \\ \Leftrightarrow & & \left(I - \omega_{j+1}^{(\sigma)}(A-\sigma I)\right)\alpha^{(\sigma)}\mathbf{v}_n &= \gamma^{(\sigma)}\left((I-\omega_{j+1}A)\mathbf{v}_n\right) \\ \Leftrightarrow & & \left(1+\omega_{j+1}^{(\sigma)}\sigma\right)\alpha^{(\sigma)}\mathbf{v}_n - \omega_{j+1}^{(\sigma)}\alpha^{(\sigma)}A\mathbf{v}_n &= \gamma^{(\sigma)}\mathbf{v}_n - \gamma^{(\sigma)}\omega_{j+1}A\mathbf{v}_n. \end{aligned}$$

By matching the coefficients of the terms that belong to $\mathbf{v}_n$ and $A\mathbf{v}_n$, respectively, we obtain,

$$\alpha^{(\sigma)} + \omega_{j+1}^{(\sigma)}\sigma\alpha^{(\sigma)} = \gamma^{(\sigma)}, \quad \omega_{j+1}^{(\sigma)}\alpha^{(\sigma)} = \gamma^{(\sigma)}\omega_{j+1},$$

where $\gamma^{(\sigma)}$ and $\omega_{j+1}^{(\sigma)}$ can be calculated as:

$$\omega_{j+1}^{(\sigma)} = \frac{\omega_{j+1}}{1-\omega_{j+1}\sigma}, \quad \gamma^{(\sigma)} = \frac{\alpha^{(\sigma)}}{1-\omega_{j+1}\sigma}. \tag{2.26}$$

Thus, we have derived a formula for the collinearity factor $\gamma^{(\sigma)}$ which can be used in the nested framework in (2.17). In Algorithm 2.2, we present the collinear IDR variant (called `coll_IDR(s)`) using (2.26) for the choice of the free IDR parameter $\omega_{j+1}^{(\sigma)}$. Note that in (2.26) the explicit dependence of the collinearity factor on the shift becomes obvious.

Since in Algorithm 2.2 we loop over N_σ distinct shifts, all shifted quantities are labeled with a superscript that depends on this loop index k. Concerning the choice of the s-dimensional shadow space $\mathcal{S}$ and the parameter ω_{j+1} of the unshifted system, we refer to Section 4.1 and 4.2 of [130], respectively. As mentioned already, the implementation of Algorithm 2.2 does not require the storage of residual differences for the shifted systems. However, we need to additionally store s updates for the iterates.

2.4 Nested FOM-FGMRES for Shifted Linear Systems

We now present a special case of the nested Krylov framework of Section 2.2.2, namely a version where multi-shift FOM (`msFOM`) is used as inner preconditioner and flexible GMRES (`FGMRES`) is used as an outer Krylov iteration. Flexible GMRES has been introduced in [115] for unshifted systems $A\mathbf{x} = \mathbf{b}$ and allows a different preconditioner $\mathcal{P}_j$ in the j-th outer iteration. The Hessenberg relation (2.4) therein extends to,

$$AZ_m = V_{m+1}\underline{\mathrm{H}}_m, \quad (A-\sigma I)Z_m^{(\sigma)} = V_{m+1}\underline{\mathrm{H}}_m(\sigma), \tag{2.27}$$

where at step $1 \leq j \leq m$ the (flexible) preconditioning $\mathbf{z}_j \equiv \mathcal{P}_j^{-1}\mathbf{v}_j$, $\mathbf{z}_j^{(\sigma)} \equiv \mathcal{P}_j(\sigma)^{-1}\mathbf{v}_j$ is carried out, and $Z_m \equiv [\mathbf{z}_1, ..., \mathbf{z}_m]$ and $Z_m^{(\sigma)} \equiv [\mathbf{z}_1^{(\sigma)}, ..., \mathbf{z}_m^{(\sigma)}]$, respectively. Note that (2.27) is formulated for both, the shifted and unshifted case, and that one column of relation (2.27) yields,

$$A\mathbf{z}_j = V_{m+1}\underline{\mathbf{h}}_j, \quad (A-\sigma I)\mathbf{z}_j^{(\sigma)} = V_{m+1}\underline{\mathbf{h}}_j^{(\sigma)},$$

which we will use next in order to determine the shifted Hessenberg matrix $\underline{\mathrm{H}}_m(\sigma)$ *columnwise*. To this end, we assume that the preconditioner $\mathcal{P}_j(\sigma)$ is equivalent to a truncated `msFOM` inner iteration with collinear factors $\gamma_j^{(\sigma)}$ for the residuals according to (2.18). From the following calculation

$$\begin{aligned}
& & (A-\sigma I)\mathcal{P}_j(\sigma)^{-1}\mathbf{v}_j &= V_{m+1}\underline{\mathbf{h}}_j^{(\sigma)} \\
&\Leftrightarrow & \gamma_j^{(\sigma)}A\mathbf{z}_j - \left(\gamma_j^{(\sigma)}-1\right)\mathbf{v}_j &= V_{m+1}\underline{\mathbf{h}}_j^{(\sigma)} \\
&\Leftrightarrow & \gamma_j^{(\sigma)}V_{m+1}\underline{\mathbf{h}}_j - V_{m+1}\left(\gamma_j^{(\sigma)}-1\right)\underline{\mathbf{e}}_j &= V_{m+1}\underline{\mathbf{h}}_j^{(\sigma)} \\
&\Leftrightarrow & V_{m+1}\left(\gamma_j^{(\sigma)}\underline{\mathbf{h}}_j - \left(\gamma_j^{(\sigma)}-1\right)\underline{\mathbf{e}}_j\right) &= V_{m+1}\underline{\mathbf{h}}_j^{(\sigma)}
\end{aligned}$$

we can conclude the j-th column of the shifted Hessenberg matrix to be,

$$\underline{\mathbf{h}}_j^{(\sigma)} = \gamma_j^{(\sigma)}\underline{\mathbf{h}}_j - \left(\gamma_j^{(\sigma)}-1\right)\underline{\mathbf{e}}_j, \quad 1 \le j \le m, \tag{2.28}$$

with $\underline{\mathbf{e}}_j$ being the j-th unit vector of $\mathbb{C}^{m+1}$.

Aligning the columns of m outer iterations together, yields the following formula for the shifted Hessenberg matrix,

$$\underline{\mathrm{H}}_m(\sigma) = \left(\underline{\mathrm{H}}_m - \underline{\mathrm{I}}_m\right)\Gamma_m^{(\sigma)} + \underline{\mathrm{I}}_m, \tag{2.29}$$

where $\underline{\mathrm{I}}_m$ is the $m \times m$ identity matrix with an extra row of zeros attached and $\Gamma_m^{(\sigma)}$ is a diagonal matrix with the collinearity factors on the diagonal, i.e.

$$\Gamma_m^{(\sigma)} \equiv \begin{bmatrix} \gamma_1^{(\sigma)} & & \\ & \ddots & \\ & & \gamma_m^{(\sigma)} \end{bmatrix} \in \mathbb{C}^{m\times m}. \tag{2.30}$$

We use this notation in order to point out the similarities to the nested algorithm in Section 2.5. Note that for $\sigma = 0$, the shifted Hessenberg matrix (2.29) reduces to the original Hessenberg matrix, $\underline{\mathrm{H}}_m(0) = \underline{\mathrm{H}}_m$, because in this case the collinearity factors are all equal to one and, hence, $\Gamma_m^{(0)} = I$. The FOM-FGMRES nested Krylov method for shifted linear systems is summarized in Algorithm 2.3.

The least-squares problem in line 15 minimizes the residual of each shifted system as the following calculation proves,

$$\begin{aligned}
\mathbf{x}_j^{(\sigma)} &= \underset{\mathbf{x}\in\mathcal{Z}_j^{(\sigma)}}{\operatorname{argmin}} \left\|\mathbf{b} - (A-\sigma I)\mathbf{x}\right\| = \underset{\mathbf{y}\in\mathbb{C}^j}{\operatorname{argmin}} \left\|\mathbf{b} - (A-\sigma I)Z_j^{(\sigma)}\mathbf{y}\right\| \\
&= \underset{\mathbf{y}\in\mathbb{C}^j}{\operatorname{argmin}} \left\|\mathbf{b} - V_{j+1}\underline{\mathrm{H}}_j(\sigma)\mathbf{y}\right\| = \underset{\mathbf{y}\in\mathbb{C}^j}{\operatorname{argmin}} \left\|\beta\mathbf{e}_1 - \left(\left(\underline{\mathrm{H}}_j - \underline{\mathrm{I}}_j\right)\Gamma_j^{(\sigma)} + \underline{\mathrm{I}}_j\right)\mathbf{y}\right\|,
\end{aligned}$$

where we used the flexible shifted Arnoldi relation of (2.27) as well as (2.29).

Algorithm 2.3 Nested `FOM-FGMRES` for $(A - \sigma_k I)\mathbf{x}_k = \mathbf{b},\ k = 1, ..., N_\sigma$

```
Initialize $\mathbf{r}_0 = \mathbf{b},\ \beta = \|\mathbf{r}_0\|,\ \mathbf{v}_1 = \mathbf{r}_0/\beta$
for $j = 1$ to $m$ do
    Preconditioning: $\mathbf{z}_j^{(\sigma_k)} = \mathtt{msFOM}(A - \sigma_k I, \mathbf{v}_j)$    ▷ Call of Algorithm 2.1
    Compute $\gamma_j^{(\sigma_k)}$ according to (2.18)    ▷ Collinearity of msFOM
    Compute $\mathbf{w} = A\mathbf{z}_j^{(0)}$
    for $i = 1$ to $j$ do    ▷ (Outer) Arnoldi method
        $h_{i,j} = \mathbf{w}^{\mathsf{H}}\mathbf{v}_i$
        $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
    end for
    Set $h_{j+1,j} = \|\mathbf{w}\|$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
    for $k = 1$ to $N_\sigma$ do    ▷ Loop over shifts
        Define $Z_j^{(\sigma_k)} = [\mathbf{z}_1^{(\sigma_k)}, ..., \mathbf{z}_j^{(\sigma_k)}]$
        Construct $\underline{H}_j(\sigma_k)$ according to (2.29)
        Solve $\mathbf{y}_j^{(\sigma_k)} = \mathrm{argmin}_{\mathbf{y}} \left\|\beta\mathbf{e}_1 - \underline{H}_j(\sigma_k)\mathbf{y}\right\|$, with $\mathbf{e}_1 = [1, 0, ..., 0]^{\mathsf{H}} \in \mathbb{R}^{j+1}$
        Set $\mathbf{x}_j^{(\sigma_k)} = Z_j^{(\sigma_k)}\mathbf{y}_j^{(\sigma_k)}$
    end for
end for
```

We note that in the same way as in flexible GMRES for the unshifted case [115], extra storage is required because the matrices $Z_j^{(\sigma_k)}$ which span the solution space for every shifted problem need to be stored. This is in fact a major drawback of flexible GMRES that has already been pointed out by [115] and applies here for every shift. We, therefore, present in Section 2.5 a nested algorithm that uses flexible QMRIDR [148, Section 4] as an outer method and partly overcomes this storage requirement.

2.5 Nested IDR-FQMRIDR for Shifted Linear Systems

In a similar way to the nested FOM-FGMRES algorithm, we present a nested IDR-FQMRIDR method for shifted linear systems where `coll_IDR(s)` from Algorithm 2.2 is used as inner method. In contrast to the combination in Section 2.4, this is a combination of two short recurrence methods. For $V_m \equiv G_m U_m$, relation (2.7) in QMRIDR was given by,

$$AV_m = G_{m+1}\underline{H}_m.$$

In flexible QMRIDR (`FQMRIDR`) which has been introduced in [148, Section 4], this relation is replaced by

$$AZ_m = G_{m+1}\underline{H}_m, \quad (A - \sigma I)Z_m^{(\sigma)} = G_{m+1}\underline{H}_m(\sigma), \tag{2.31}$$

with $Z_m, Z_m^{(\sigma)}$ consisting of the respective columns $\mathbf{z}_j \equiv \mathcal{P}_j^{-1}\mathbf{v}_j,\ \mathbf{z}_j^{(\sigma)} \equiv \mathcal{P}(\sigma)_j^{-1}\mathbf{v}_j$ for $1 \le j \le m$, just as before. One column of (2.31) reads,

$$A\mathbf{z}_j = G_{m+1}\underline{\mathbf{h}}_j, \quad (A - \sigma I)\mathbf{z}_j^{(\sigma)} = G_{m+1}\underline{\mathbf{h}}_j^{(\sigma)},$$

which we will use in order to derive the shifted Hessenberg matrix of IDR-FQMRIDR.

We assume that the factors $\gamma_j^{(\sigma)}$ are given from (2.26) by an inner `coll_IDR(s)` iteration. Then, the flexible QMRIDR relation applied to a shifted problem yields:

$$\begin{aligned}
& & (A-\sigma I)\mathcal{P}_j(\sigma)^{-1}\mathbf{v}_j &= G_{m+1}\underline{\mathbf{h}}_j^{(\sigma)} \\
&\Leftrightarrow & \gamma_j^{(\sigma)}A\mathbf{z}_j - \left(\gamma_j^{(\sigma)}-1\right)\mathbf{v}_j &= G_{m+1}\underline{\mathbf{h}}_j^{(\sigma)} \\
&\Leftrightarrow & \gamma_j^{(\sigma)}G_{m+1}\underline{\mathbf{h}}_j - \left(\gamma_j^{(\sigma)}-1\right)G_m\mathbf{u}_j &= G_{m+1}\underline{\mathbf{h}}_j^{(\sigma)} \\
&\Leftrightarrow & G_{m+1}\left(\gamma_j^{(\sigma)}\underline{\mathbf{h}}_j - \left(\gamma_j^{(\sigma)}-1\right)\underline{\mathbf{u}}_j\right) &= G_{m+1}\underline{\mathbf{h}}_j^{(\sigma)}
\end{aligned}$$

and one column of the shifted Hessenberg matrix is given by,

$$\underline{\mathbf{h}}_j^{(\sigma)} = \gamma_j^{(\sigma)}\underline{\mathbf{h}}_j - \left(\gamma_j^{(\sigma)}-1\right)\underline{\mathbf{u}}_j, \quad 1\le j\le m, \tag{2.32}$$

where $\underline{\mathbf{u}}_j \equiv [\mathbf{u}_j, 0]^{\mathsf{H}}$ is the vector $\mathbf{u}_j$ from the j-th iteration of QMRIDR [148] with an extra zero.

Altogether, we have derived the shifted Hessenberg matrix,

$$\underline{\mathrm{H}}_m(\sigma) = \left(\underline{\mathrm{H}}_m - \underline{\mathrm{U}}_m\right)\Gamma_m^{(\sigma)} + \underline{\mathrm{U}}_m, \tag{2.33}$$

with $\Gamma_m^{(\sigma)}$ as defined in (2.30), and $\underline{\mathrm{U}}_m \equiv [\underline{\mathbf{u}}_1, ..., \underline{\mathbf{u}}_m]$. Here we see the close relation between the two nested methods: By comparing the expression for the shifted Hessenberg matrices in (2.29) and (2.33), we first note that in principle every inner Krylov method can be used that provides collinear residuals. Moreover, we use this factor within the (generalized) Hessenberg relation of the outer Krylov method in a very similar way which shows that in principle also every Krylov method as an outer iteration is suitable.

Note that Algorithm 2.4 is schematic. For a more detailed and memory-efficient implementation of the flexible `QMRIDR(s)` routine, see [148, Algorithm 1]. In fact, we only need to apply the `coll_IDR(s)` routine as a preconditioner in line 18 and change line 32 by the formula (2.32) in order to adapt [148, Algorithm 1] to our nested algorithm. In line 11 of Algorithm 2.4, the following *quasi-minimization* of the shifted residual is carried out,

$$\begin{aligned}
\mathbf{x}_j^{(\sigma)} &= \operatorname*{argmin}_{\mathbf{x}\in\mathcal{Z}_j^{(\sigma)}}\|\mathbf{b}-(A-\sigma I)\mathbf{x}\| = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{C}^j}\left\|\mathbf{b}-(A-\sigma I)Z_j^{(\sigma)}\mathbf{y}\right\| \\
&= \operatorname*{argmin}_{\mathbf{y}\in\mathbb{C}^j}\left\|\mathbf{b}-G_{j+1}\underline{\mathrm{H}}_j(\sigma)\mathbf{y}\right\| = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{C}^j}\left\|G_{j+1}\left(\beta\mathbf{e}_1-\underline{\mathrm{H}}_j(\sigma)\mathbf{y}\right)\right\| \\
&\le \operatorname*{argmin}_{\mathbf{y}\in\mathbb{C}^j}\left\|\beta\mathbf{e}_1-\underline{\mathrm{H}}_j(\sigma)\mathbf{y}\right\| = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{C}^j}\left\|\beta\mathbf{e}_1-\left(\left(\underline{\mathrm{H}}_j-\underline{\mathrm{U}}_j\right)\Gamma_j^{(\sigma)}+\underline{\mathrm{U}}_j\right)\mathbf{y}\right\|,
\end{aligned}$$

with an estimation for $\|G_{j+1}\|$ given in [148].

2.6 Numerical Experiments

The numerical examples we present are motivated from geophysical applications. We consider the numerical solution of the Helmholtz equation in Section 2.6 and of the

Algorithm 2.4 Nested `IDR-FQMRIDR` for $(A - \sigma_k I)\mathbf{x}_k = \mathbf{b},\ k = 1, ..., N_\sigma$

```
Initialize $\mathbf{r}_0 = \mathbf{b},\ \beta = \|\mathbf{r}_0\|,\ \mathbf{v}_1 = \mathbf{r}_0/\beta$
for $j = 1$ to $m$ do
    Preconditioning: $\mathbf{z}_j^{(\sigma_k)} = \texttt{coll\_IDR}(A - \sigma_k I, \mathbf{v}_j)$    ▷ Call of Algorithm 2.2
    Compute $\gamma_j^{(\sigma_k)}$ according to (2.26)    ▷ Collinearity in coll_IDR
    Compute $\underline{\mathbf{h}}_j, \underline{\mathbf{u}}_j$ as in QMRIDR    ▷ See [148, Algorithm 1]
    Compute $\underline{\mathbf{h}}_j^{(\sigma_k)}$ from (2.32)
    for $k = 1$ to $N_\sigma$ do    ▷ Loop over shifts
        Define $Z_j^{(\sigma_k)} = [\mathbf{z}_1^{(\sigma_k)}, ..., \mathbf{z}_j^{(\sigma_k)}]$
        Construct $\underline{H}_j(\sigma_k)$ according to (2.33)
        Solve $\mathbf{y}^{(\sigma_k)} = \mathrm{argmin}_{\mathbf{y}} \left\|\beta\mathbf{e}_1 - \underline{H}_j(\sigma_k)\mathbf{y}\right\|$, with $\mathbf{e}_1 = [1, 0, ..., 0]^{\mathsf{H}} \in \mathbb{R}^{j+1}$
        Set $\mathbf{x}_j^{(\sigma_k)} = Z_j^{(\sigma_k)}\mathbf{y}^{(\sigma_k)}$
    end for
end for
```

time-harmonic elastic wave equation (Navier equation) in Section 2.6. In both cases, we consider the numerical solution at multiple frequencies that arise from a frequency-domain model of acoustic and elastic waves, respectively. We will point out that there exists a one-to-one relation between the considered shifts in (2.1) and the frequencies of the waves.

All examples have been implemented in MATLAB R2011B on a `Intel(R) Xeon(R) CPU E3-1240 V2 @ 3.40GHz` CPU. For a more detailed description of the numerical tests, we refer to our extended technical report[†].

For the numerical solution of shifted linear systems of the form

$$(A - \sigma_k I)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\sigma, \tag{2.34}$$

it is of practical use to re-formulate the problem (2.34) by the following substitutions:

$$\begin{aligned} \bar{\sigma}_k &\equiv \sigma_k - \sigma^*, \\ \bar{A} &\equiv A - \sigma^* I, \end{aligned}$$

for some $\sigma^* \in \{\sigma_1, ..., \sigma_{N_\sigma}\}$. This way, it is equivalent to solve the shifted linear systems

$$(\bar{A} - \bar{\sigma}_k I)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\sigma, \tag{2.35}$$

with the only difference that in the formulation (2.35) the unshifted ($\bar{\sigma}_k = 0$) solution corresponds to one of the N_σ solutions we are interested in, compare Remark 1.11.

[†] M. Baumann and M.B. van Gijzen. *Nested Krylov methods for shifted linear systems.* Technical report **14-01**, Delft University of Technology, 2014.

An Inhomogeneous Helmholtz Problem

As a first example, we consider acoustic wave propagation which can be modeled by the Helmholtz equation,

$$-\Delta p - \left(\frac{2\pi f_k}{c(\mathbf{x})}\right)^2 p = s, \quad \text{in } \Omega \subset \mathbb{R}^2, \tag{2.36}$$

where p stands for the pressure and f_k is the k-th wave frequency. We consider the *wedge problem* which was introduced in [104]. Therein, the computational domain is $\Omega = [0, 600] \times [0, 1000]$ and the underlying sound velocity $c(\mathbf{x})$ is heterogeneous and represents three different layers, cf. [104] for more details. Moreover, the sound source is given by a point source at the top of the computational domain, $s = \delta(x - 300, z)$. We distinguish between reflecting boundary conditions (homogeneous Neumann boundary conditions),

$$\frac{\partial p}{\partial n} = 0, \quad \text{on } \partial\Omega, \tag{2.37}$$

and absorbing boundary conditions (so-called Sommerfeld radiation boundary conditions) of the form,

$$\frac{\partial p}{\partial n} - i\left(\frac{2\pi f_k}{c(\mathbf{x})}\right) p = 0, \quad \text{on } \partial\Omega, \tag{2.38}$$

where i is the imaginary unit. Preconditioning techniques for the Helmholtz problem in the single-shift case ($N_\sigma = 1$) are for instance discussed in [42, 44, 45, 121, 147]. When non-homogeneous Neumann boundary conditions (2.38) are included in (2.36), we end up with a discretization of the form,

$$(K + i(2\pi f_k)C - (2\pi f_k)^2 M)\mathbf{p} = \mathbf{s}, \quad \sigma_k = 2\pi f_k, \tag{2.39}$$

where C represents the boundary conditions (2.38), K is the (negative) discrete Laplacian and M is a mass matrix. The unknown vector $\mathbf{p}$ consists of discrete pressure values. Note that (2.39) is quadratic in σ_k and can be linearized in the following way,

$$\left\{\begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} - \sigma_k \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}\right\} \begin{bmatrix} \sigma_k \mathbf{p} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{s} \\ 0 \end{bmatrix}, \quad k = 1, ..., N_\sigma, \tag{2.40}$$

as suggested in [117, 125] which yields a shifted problem. For reflecting boundary conditions (2.37), we get $C = 0$ and a shifted linear system can be obtained by simply multiplying (2.39) with M^{-1}, which we avoid in the following. The shift-and-invert preconditioner (as introduced in Section 2.2.1) for (2.40) is given by,

$$\mathcal{P} = \begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} - \tau \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} iC - \tau M & K \\ I & -\tau I \end{bmatrix}, \tag{2.41}$$

with damping parameter $\tau \in \mathbb{C}$ and can be seen as the analogue of the shifted Laplace preconditioner of [42, 44, 45] applied to the block system (2.40). We remark that (2.41) can be decomposed and inverted in the following way,

$$\mathcal{P}^{-1} = \begin{bmatrix} I & \tau I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & (K + i\tau C - \tau^2 M)^{-1} \end{bmatrix} \begin{bmatrix} 0 & I \\ I & -iC + \tau M \end{bmatrix}. \tag{2.42}$$

Therefore, from a computational point of view, applying the preconditioner (2.41) to (2.40) reduces to efficiently solving systems with the shifted Laplacian $(K + i\tau C - \tau^2 M)$ in (2.42) which is of the same dimension as (2.39).

For large-scale applications, a multigrid approach as in [42, 45, 121] can be used to approximate the shifted Laplace preconditioner. This gives rise to the question to which accuracy a multigrid method needs to be applied. Using the nested method of Section 2.4 or Section 2.5 for the preconditioned shifted problem yields a setting similar to [143] where the authors analyze inexact nested Krylov methods for the unshifted case. The insight of [143] on the accuracy that is needed for matrix-vector multiplications may give guidelines for an extension to the shifted framework.

In the present test case, we aim to solve (2.40) for a range of six frequencies $f_k = \{1, 2, 4, 8, 16, 32\} Hz$ and restrict ourselves to a direct method for the shift-and-invert preconditioner. For the resolution of high frequencies and due to the doubling of unknowns from (2.39) to (2.40), the system size becomes more than $30,000$ equations. We present the convergence behavior for multi-shift GMRES and multi-shift QMRIDR(8) using only the single shift-and-invert preconditioner (2.42) in Figure 2.1, respecively. The convergence curves show that the residual norms first stagnate or even increase in the case of QMRIDR(8). Moreover, the convergence rates are nearly linear as soon as the residual norms start decreasing. Therefore, it is intuitive to truncate the inner iterations in this region. The convergence plots of nested FOM-FGMRES and nested IDR-FQMRIDR(8) are presented in Figure 2.2 and Figure 2.3, respectively.

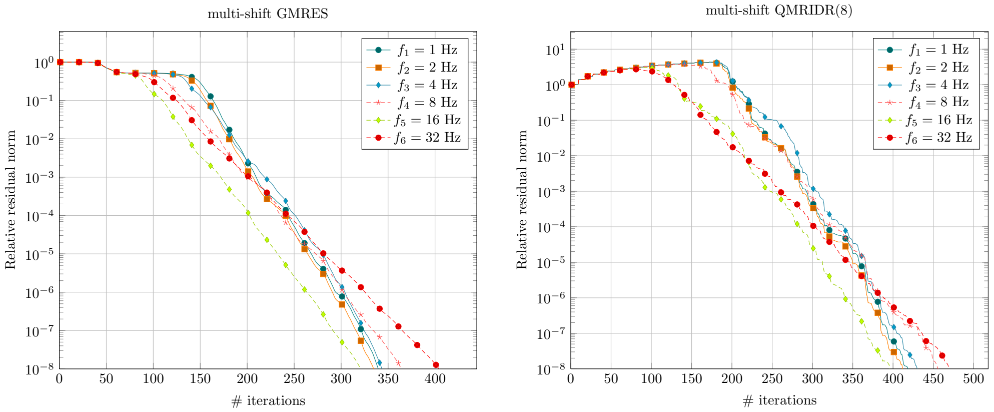


Figure 2.1: *Convergence behavior of multi-shift GMRES (left) and multi-shift QMRIDR(8) (right) for* (2.36)-(2.38).

In both nested algorithms, the number of *inner* iterations is chosen in such a way that the relative residual norms are of the size 0.1 or smaller which seems to be a good choice for truncation of the inner algorithm. The convergence rate of the outer Krylov method is in both cases very fast.

In Table 2.1, we want to point out the CPU time that is required in order to solve all six shifted systems up to a relative tolerance of 10^{-8}. Comparing multi-shift GMRES and multi-shift GMRES preconditioned by a nested FOM method (FOM-FGMRES), we observe that the nested method is more than five times faster. Since

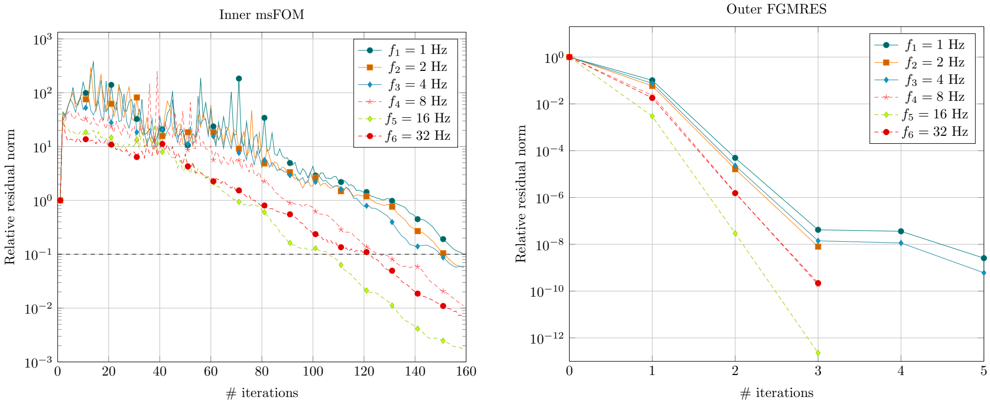


Figure 2.2: *Convergence behavior of `FOM-FGMRES` for (2.36)-(2.38): Typical inner convergence (left) and outer convergence (right).*

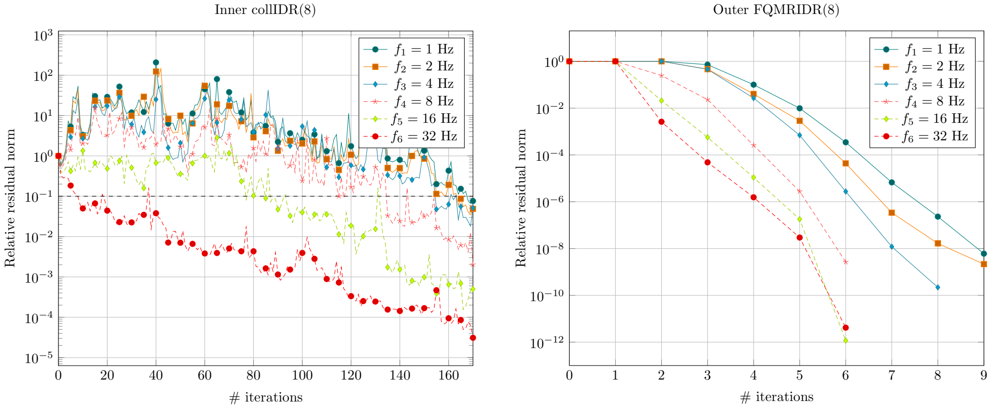


Figure 2.3: *Convergence behavior of `IDR-FQMRIDR(8)` for (2.36)-(2.38): Typical inner convergence (left) and outer convergence (right).*

the total number of iterations, and therefore, the number of matrix-vector multiplications is larger in the nested method, we conclude that the observed speed-up is due to shorter recurrence of the Arnoldi orthogonalization process. This also explains why we observe no speed-up for multi-shift QMRIDR which is a short recurrence method by design.

Moreover, we included other possible combinations of the nested algorithm which show that a combination of inner `msFOM` with outer `FQMRIDR(8)` performs best. Table 2.1 also shows the performance of `coll_IDR(8)` as a stand-alone multi-shift algorithm as well as restarted multi-shift GMRES [50] (`rest_msGMRES`) which can be seen as a nested combination of multi-shift GMRES with an outer Richardson iteration. For all considered cases, the seed shift τ of the shift-and-invert preconditioner (2.41) has been tuned such that optimal convergence was obtained.

Table 2.1: *Comparison of multi-shift and nested multi-shift algorithms for the wedge problem* (2.36) *with absorbing boundary conditions* (2.38). *Here, we set* $\hat{\sigma} \equiv \max\{\sigma_k, k = 1, ..., N_\sigma\}$ *when choosing the seed parameter in* (2.41).

	multi-shift Krylov methods			
	`msGMRES`	`rest_msGMRES`	`QMRIDR(8)`	`coll_IDR(8)`
# inner iterations	-	200	-	-
# outer iterations	404	4	471	584
seed shift τ	$(0.7-0.7i)\hat{\sigma}$	$(0.7-0.7i)\hat{\sigma}$	$(0.7-0.7i)\hat{\sigma}$	$(0.7-0.7i)\hat{\sigma}$
CPU time	157.33s	39.4s	62.51s	79.13s
	nested multi-shift Krylov methods			
	`FOM-FGMRES`	`IDR(8)-FGMRES`	`FOM-FQMRIDR(8)`	`IDR(8)-FQMRIDR(8)`
# inner iterations	160	250	100	170
# outer iterations	5	13	8	9
seed shift τ	$(0.7-0.7i)\hat{\sigma}$	$(0.7-0.7i)\hat{\sigma}$	$(0.7-0.7i)\hat{\sigma}$	$(0.7-0.7i)\hat{\sigma}$
CPU time	28.56s	216.5s	24.4s	75.41s

The Homogeneous Time-Harmonic Elastic Wave Equation

Our second example considers the wave propagation of sound waves through an elastic medium. We are interested in the numerical solution of time-harmonic waves at multiple (angular) frequencies $\sigma_k = 2\pi f_k, k = 1, ..., N_\sigma$. The scattering of time-harmonic waves is described in [2] by a Navier equation,

$$-\sigma_k^2 \rho(\mathbf{x})\mathbf{u} - \nabla \cdot \tau(\mathbf{u}) = \mathbf{s}, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \tag{2.43}$$

where $\mathbf{u} : \Omega \to \mathbb{R}^2$ is the unknown displacement vector, $\mathbf{s}$ is typically a point source, and $\rho(\mathbf{x})$ is the density of the material which is assumed to be space-dependent. The strain and stress tensor are derived from Hooke's law and are given by,

$$\tau(\mathbf{u}) \equiv \lambda(\mathbf{x}) \left(\nabla \cdot \mathbf{u}\right) + 2\mu(\mathbf{x})\epsilon(\mathbf{u}), \quad \epsilon(\mathbf{u}) \equiv \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathsf{T}}\right).$$

Note that the underlying density $\rho(\mathbf{x})$ as well as the Lamé parameters $\lambda(\mathbf{x})$ and $\mu(\mathbf{x})$ have to be prescribed in the considered model, see Table 2.2.

In contrast to the example in Section 2.6, we will consider more realistic boundary conditions. Therefore, the following impedance boundary condition is prescribed,

$$i\gamma(\mathbf{x})\sigma_k \rho(\mathbf{x}) B\mathbf{u} + \tau(\mathbf{u})\mathbf{n}(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial\Omega, \tag{2.44}$$

where γ is the absorption coefficient, $i \equiv \sqrt{-1}$, and $B_{i,j}(\mathbf{x}) \equiv c_p(\mathbf{x}) n_i n_j + c_s(\mathbf{x}) t_i t_j$. Here, n_i and t_i are the components of the (outer) normal vector $\mathbf{n}$ and the tangential vector $\mathbf{t}$, respectively. For $\Omega \subset \mathbb{R}^2$ we consequently get a 2×2 matrix B at every boundary point $\mathbf{x} \in \partial\Omega$. The quantities c_p and c_s are the speed of pressure wave and shear wave, respectively (see Table 2.2). In the following, we prescribed absorbing boundary conditions on whole $\partial\Omega$ by setting $\gamma \equiv 1$.

Table 2.2: *Value of constant parameters taken from [2].*

ρ [kg/m^3]	c_p [m/s]	c_s [m/s]	λ [Pa]	μ [Pa]
$2.7 \cdot 10^3$	$6.8983 \cdot 10^3$	$4.3497 \cdot 10^3$	$2.6316 \cdot 10^{10}$	$5.1084 \cdot 10^{10}$

From a discretization of (2.43)-(2.44) using linear finite elements, we obtain the linear systems

$$(K + i\sigma_k C - \sigma_k^2 M)\underline{\mathbf{u}} = \underline{\mathbf{s}}, \quad k = 1, ..., N_\sigma, \tag{2.45}$$

with K, C, M being symmetric and sparse block matrices, and $\underline{\mathbf{u}}, \underline{\mathbf{s}}$ being the discretized counterpart of $\mathbf{u}, \mathbf{s}$ in lexicographical order. Here, C contains the boundary conditions (2.44), and K and M are a stiffness and mass matrix, respectively. Reformulation of (2.45) in the same way as (2.40) yields a block system of the form

$$\left\{ \begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} - \sigma_k \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \right\} \begin{bmatrix} \sigma_k \underline{\mathbf{u}} \\ \underline{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{s}} \\ 0 \end{bmatrix}, \quad k = 1, ..., N_\sigma, \tag{2.46}$$

which is again a shifted linear system with shifts $\sigma_1, ..., \sigma_{N_\sigma}$.

The considered numerical setting is taken from [2, 87]. Therein, the parameters are set to the values presented in Table 2.2, and the unit square is considered as computational domain Ω. The angular frequencies σ_k are given by $\sigma_k = 2\pi f_k$ and range from $f_1 = 5{,}000Hz$ until $f_6 = 30{,}000Hz$ in uniform steps. For more details on the discretization and the numerical results, we refer to our corresponding technical report.

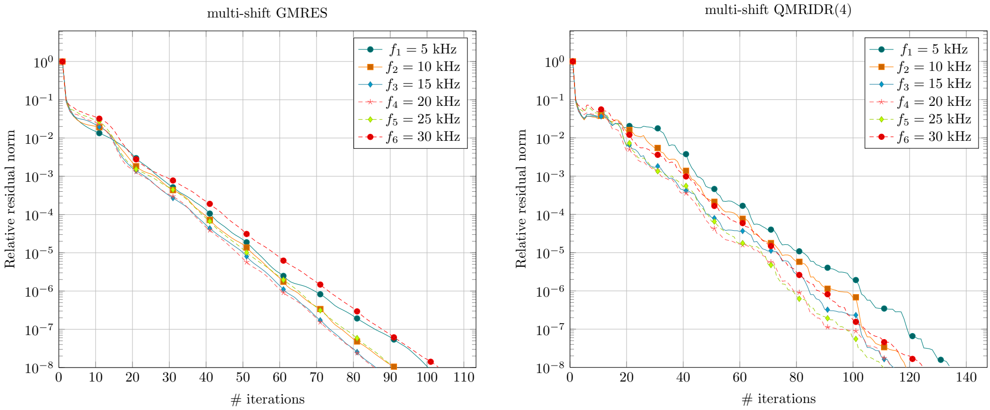


Figure 2.4: *Convergence behavior of multi-shift GMRES (left) and multi-shift QMRIDR(4) (right) for* (2.46).

We were again running our numerical experiments using an additional shift-and-invert preconditioner (2.42) with seed shift τ as shown in Table 2.3. In Figure 2.4, we present the convergence curves of multi-shift GMRES and multi-shift QMRIDR(4) without nested preconditioning. In this experiments, we observe a flat convergence behavior that gives rise to an early truncation in the nested framework.

For nested FOM-FGMRES, we chose the number of inner `msFOM` iterations such that the relative residual norms are below a threshold tolerance of 0.1, cf. Figure 2.5. Our numerical experiments have proven that this leads to a rapid convergence in only 7 iterations in the outer FGMRES loop. When measuring the actual CPU time that is required to solve all $N_\sigma = 6$ shifted systems with multi-shift GMRES and nested FOM-FGMRES, we observe a speed-up of almost two, cf. Table 2.3.

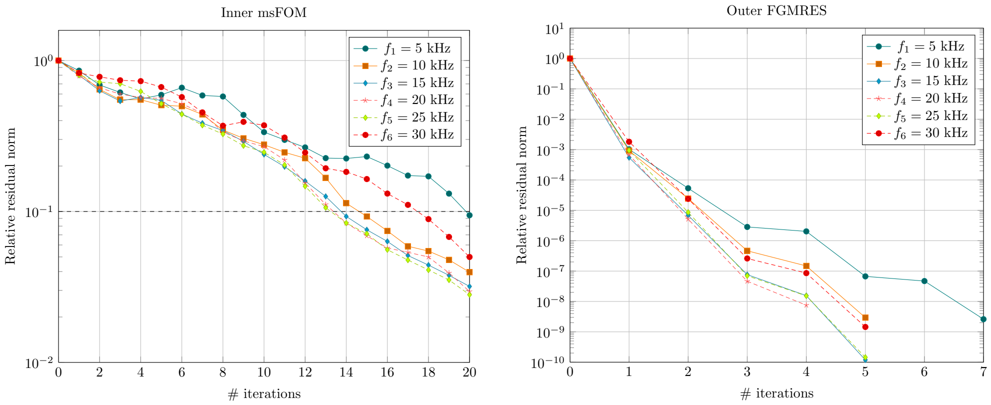


Figure 2.5: *Convergence behavior of `FOM-FGMRES` for (2.46): Typical inner convergence (left) and outer convergence (right).*

Table 2.3: *Comparison of multi-shift and nested multi-shift algorithms for the linear elastic wave equation (2.43)-(2.44). Again,* $\hat{\sigma} \equiv \max\{\sigma_k, k = 1, ..., N_\sigma\}$.

	multi-shift Krylov methods			
	`msGMRES`	`rest_msGMRES`	`QMRIDR(4)`	`coll_IDR(4)`
# inner iterations	-	20	-	-
# outer iterations	103	7	136	134
seed shift τ	$(0.7 - 0.7i)\hat{\sigma}$	$(0.7 - 0.7i)\hat{\sigma}$	$(0.7 - 0.7i)\hat{\sigma}$	$(0.7 - 0.7i)\hat{\sigma}$
CPU time	17.71s	6.13s	22.35s	22.58s
	nested multi-shift Krylov methods			
	`FOM-FGMRES`	`IDR(4)-FGMRES`	`FOM-FQMRIDR(4)`	`IDR(4)-FQMRIDR(4)`
# inner iterations	20	25	30	30
# outer iterations	7	9	5	15
seed shift τ	$(0.7 - 0.7i)\hat{\sigma}$	$(0.7 - 0.7i)\hat{\sigma}$	$(0.7 - 0.7i)\hat{\sigma}$	$(0.7 - 0.7i)\hat{\sigma}$
CPU time	9.12s	32.99s	8.14s	58.36s

Moreover, we applied nested IDR-FQMRIDR(4) to (2.46) in Figure 2.6. From the convergence behavior of the inner `coll_IDR` iteration (Algorithm 2.2), we note that the convergence curves show irregular jumps which makes a proper truncation of the inner preconditioner rather difficult. As in the previous tests, we do not observe a speed-up in CPU time for the nested algorithm which is mostly due to the fact of short recurrence of QMRIDR(s).

In Table 2.3 we present as well numerical results for an implementation of restarted multi-shift GMRES (`rest_msGMRES`) and our IDR-variant that exploits collinear residuals (`coll_IDR(s)`) which can be seen as a multi-shift Krylov method when being applied as a stand-alone algorithm as presented in Algorithm 2.2. Moreover, we compare performance of the nested Krylov methods with different inner-outer methods combined. For the specific setting considered in Table 2.3, we first note that `QMRIDR(4)` and `coll_IDR(4)` require similar CPU times. Moreover, we observe that a combination of inner `msFOM` and outer `FQMRIDR(4)` perform best among the nested algorithms. The restarted version of multi-shift GMRES performs best in this setting but did not converge in some examples described in our technical report.

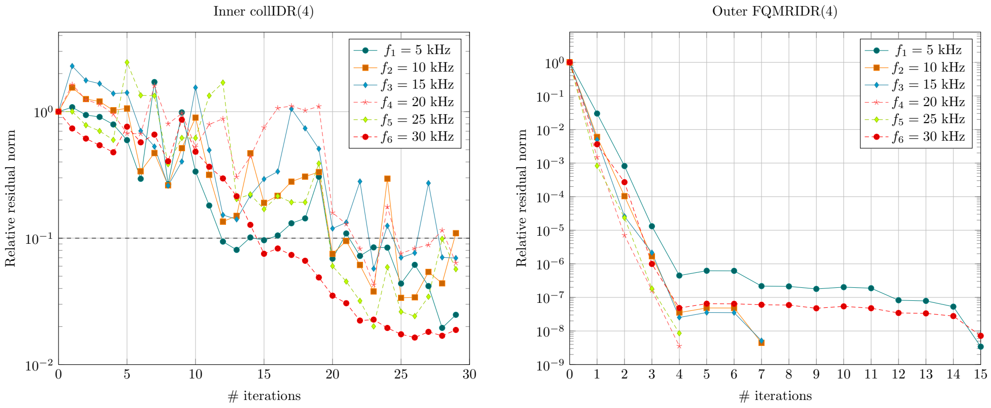


Figure 2.6: *Convergence behavior of* `IDR-FQMRIDR(4)` *for* (2.46)*: Typical inner convergence (left) and outer convergence (right).*

Conclusion

This work presents an algorithmic framework for the numerical solution of shifted linear systems (2.1) with inner-outer Krylov methods that allow flexible preconditioning. In this sense, it can be seen as a generalization of the work of [115, 126, 156] to sequences of shifted problems. The most general algorithm of this paper can be summarized in the following way,

1. the flexible preconditioner $\mathcal{P}_j(\sigma)$ is itself an *inner* multi-shift Krylov method which produces collinear residuals in the sense of (2.16),
2. the collinearity factor is used in the j-th iteration of an *outer* Krylov method in order to derive the Hessenberg matrix of the shifted systems, cf. (2.28), (2.32).

We call this new framework a *nested* Krylov method for shifted linear systems since the inner Krylov iteration is considered as a flexible preconditioner for the outer Krylov method. Moreover, the fact that we can use a Krylov method as flexible preconditioner shows that a Krylov polynomial can be used as polynomial preconditioner in the sense of [1].

This general framework has been illustrated and tested for two possible combinations of inner-outer Krylov methods. We present a combination of inner FOM and outer FGMRES in Algorithm 2.3. Therefore, the collinearity factor for the inner Krylov method (multi-shift FOM) is given by (2.18) without any further manipulations. When combining multi-shift IDR(s) and FQMRIDR(s) as presented in Algorithm 2.4, a new variant of IDR(s) has been developed which leads to collinear residuals with collinearity factor given by (2.26), cf. Algorithm 2.2. In both cases, a shifted Hessenberg matrix has been derived using the respective collinearity factors. This has been done for flexible GMRES (2.29) and flexible QMRIDR(s) (2.33), respectively.

Various numerical tests have been performed that showed an optimal performance of the nested algorithm when the inner Krylov method was truncated as the relative residuals satisfy $\|\mathbf{r}_j^{(\sigma_k)}\|/\|\mathbf{r}_0^{(\sigma_k)}\| < 10^{-1}$ at every (outer) iteration j and for all shifts

$\sigma_1, ..., \sigma_{N_\sigma}$. This way, we were able to obtain a computational speed-up up to a factor of five when comparing multi-shift GMRES with nested FOM-FGMRES in Section 2.6.

2.7 Extension: Nested Block Krylov Methods for Shifted Linear Systems with Multiple Right-Hand Sides

We briefly describe the extension of the inner-outer multi-shift Krylov algorithm to the situation when multiple right-hand sides are present. This is in particular of importance for seismic problems used in Section 2.6 because different source term locations result in multiple right-hand sides. We exemplify this extension for the combination of nested block versions of (inner) FOM with (outer) FGMRES[†]. Therefore, consider a sequence of shifted linear systems as in (2.1), with multiple right-hand sides,

$$(A - \sigma_k I)\mathbf{x}_k^\ell = \mathbf{b}_\ell, \quad k = 1, ..., N_\sigma, \quad \ell = 1, ..., N_b.$$

This sequence can be re-written as a block version of (2.1), i.e.,

$$(A - \sigma_k I)\mathbf{X}_k = \mathbf{B}, \quad \mathbf{B} := [\mathbf{b}_1, ..., \mathbf{b}_{N_b}], \quad k = 1, ..., N_\sigma. \tag{2.47}$$

Block Krylov methods are discussed for instance in [117, Chapter 6.12]. Shift-invariance (2.3) for block Krylov spaces reads,

$$\mathcal{K}_m(A, \mathbf{R}_0) \equiv \mathcal{K}_m(A - \sigma I, \mathbf{R}_0), \quad \forall \sigma \in \mathbb{C}, \tag{2.48}$$

where we note that $\mathcal{K}_m(A, \mathbf{R}_0)$ has dimension $m \cdot N_b$, and $\mathbf{R}_0 = \mathbf{B}$ if zero initial guess is used. An orthonormal basis for (2.48) is obtained by the block Arnoldi method: Initially a QR factorization of the first residual $\mathbf{R}_0 = V_1 R$ yields the first N_b block Arnoldi vectors, then $V_j, 1 < j \leq m$, is computed sequentially using Arnoldi orthogonalization (Algorithm 6.22 in [117]). The block Hessenberg relation for the shifted and the unshifted case reads,

$$\begin{aligned} A\mathbf{V}_m &= \mathbf{V}_{m+1}\underline{\mathbf{H}}_{m+1} = \mathbf{V}_m\mathbf{H}_m + V_{m+1}H_{m+1,m}E_m^{\mathsf{T}}, \\ (A - \sigma I)\mathbf{V}_m &= \mathbf{V}_{m+1}\underline{\mathbf{H}}_{m+1}^{(\sigma)} \equiv \mathbf{V}_{m+1}(\underline{\mathbf{H}}_{m+1} - \sigma\underline{\mathbf{I}}_m), \end{aligned}$$

where $\mathbf{V}_m := [V_1, ..., V_m]$, E_m is the matrix of the last N_b columns of the identity matrix $I_{m \cdot N_b}$, $\mathbf{H}_m$ is block-Hessenberg, and $\underline{\mathbf{I}}_m$ is an $mN_b \times mN_b$ identity matrix with N_b zero columns attached at the bottom, i.e. $\underline{\mathbf{I}}_m = I_{(m+1)N_b} - E_{m+1}$; cf. [117, Chapter 6.12] for details.

The extension of the nested multi-shift methods to block Krylov methods is based on the fact that in analogy to multi-shift FOM yielding collinear residuals, the matrix of blocks of residuals in multi-shift block FOM [61] yields linearly dependent columns,

---

[†] This work is based on a discussion with Kirk M. Soudhalter (at this time: Johann Radon Institute for Computational and Applied Mathematics, Linz) in 2015; and is not included in [8].

i.e., there exists a matrix $\Gamma_j^{(\sigma)} \in \mathbb{C}^{N_b \times N_b}$ that relates block residuals after j block multi-shift FOM iterations such that,

$$\mathbf{R}_j^{(\sigma)} = \mathbf{R}_j \Gamma_j^{(\sigma)}, \quad 1 \leq j \leq m, \tag{2.49}$$

where Γ_j can be seen as a generalization of the collinearity relation (2.25). This fact is exploited for restarting shifted block FOM in [61].

Proof 2.3 *Let $\mathbf{X}_m^{(\sigma)}$ be an approximate solution to (2.47) with shift σ after m shifted block FOM iterations. Consider, in analogy to the proof of Lemma 2.1, the corresponding block residual,*

$$\begin{aligned}
\mathbf{R}_m^{(\sigma)} &= \mathbf{B} - (A - \sigma I)\mathbf{X}_m^{(\sigma)} = \mathbf{R}_0 - (A - \sigma I)\mathbf{V}_m \mathbf{Y}_m^{(\sigma)} \\
&= \mathbf{R}_0 - \mathbf{V}_m \mathbf{H}_m^{(\sigma)} \mathbf{Y}_m^{(\sigma)} - V_{m+1} H_{m+1,m} E_m^{\mathsf{T}} \mathbf{Y}_m^{(\sigma)} \\
&= -V_{m+1} H_{m+1,m} E_m^{\mathsf{T}} \mathbf{Y}_m^{(\sigma)},
\end{aligned}$$

and, in the same way,

$$\mathbf{R}_m = -V_{m+1} H_{m+1,m} E_m^{\mathsf{T}} \mathbf{Y}_m.$$

The block residuals are, therefore, linearly dependent $\mathbf{R}_m^{(\sigma)} = \mathbf{R}_m \Gamma_m^{(\sigma)}$, with matrix $\Gamma_m^{(\sigma)}$ in (2.49) given as,

$$\Gamma_m^{(\sigma)} = \underbrace{(H_{m+1,m} E_m^{\mathsf{T}} \mathbf{Y}_m)}_{N_b \times N_b}{}^{-1} \underbrace{(H_{m+1,m} E_m^{\mathsf{T}} \mathbf{Y}_m^{(\sigma)})}_{N_b \times N_b} = Y_m^{-1} Y_m^{(\sigma)} \in \mathbb{C}^{N_b \times N_b},$$

with $Y_m, Y_m^{(\sigma)}$ being the last $N_b \times N_b$ blocks of $\mathbf{Y}_m, \mathbf{Y}_m^{(\sigma)}$, respectively. ☐

When using block multi-shift FOM as a preconditioner within an (outer) flexible block multi-shift GMRES loop, relation (2.27) can be generalized to a block variant using,

$$\underline{\mathbf{H}}_m(\sigma) = (\underline{\mathbf{H}}_m - \underline{\mathbf{I}}_m)\, \mathbf{\Gamma}_m^{(\sigma)} + \underline{\mathbf{I}}_m,$$

i.e., a block version of (2.29) in Section 2.4. Note that the matrix that relates inner and outer method,

$$\mathbf{\Gamma}_m^{(\sigma)} := \begin{bmatrix} \Gamma_1^{(\sigma)} & & \\ & \ddots & \\ & & \Gamma_m^{(\sigma)} \end{bmatrix} \in \mathbb{C}^{mN_b \times mN_b},$$

is now block-diagonal compared to (2.30), and m denotes here the number of (outer) flexible block multi-shift GMRES iterations.

Chapter 3

An Optimized Shift-And-Invert Preconditioner for Multi-Frequency Wave Propagation Problems

Abstract. We consider wave propagation problems that are modeled in the frequency-domain, and that need to be solved simultaneously for multiple frequencies within a fixed range. For this, a single shift-and-invert preconditioner at a so-called *seed* frequency is applied. The choice of the seed is crucial for the performance of preconditioned multi-shift GMRES and is closely related to the parameter choice for the Complex Shifted Laplace preconditioner. Based on a classical GMRES convergence bound, we present an optimal seed parameter that purely depends on the original frequency range. The new insight is exploited in a two-level preconditioning strategy: A shifted Neumann preconditioner with minimized spectral radius is additionally applied to multi-shift GMRES. Moreover, we present a reformulation of the multi-shift problem to a matrix equation solved with, for instance, global GMRES. Here, our analysis allows for rotation of the spectrum of the linear operator. Numerical experiments for the time-harmonic visco-elastic wave equation demonstrate the performance of the new preconditioners.

This chapter is based on the technical report:

M. Baumann and M.B. van Gijzen (2017). *An Efficient Two-Level Preconditioner for Multi-Frequency Wave Propagation Problems.* DIAM Technical Report **17–03**, Delft University of Technology.

Introduction

We consider the efficient iterative solution of a sequence of $n_s > 1$ shifted systems of the form,

$$(\mathcal{K} - s_k\mathcal{M})\mathbf{x}_k = \mathbf{b}, \quad \text{for } k = 1, ..., n_s, \tag{3.1}$$

where the matrices $\mathcal{K}$ and $\mathcal{M}$ depend on the specific problem discretization, and $\{s_k\}_{k=1}^{n_s}$ is a sequence of (possibly complex) shifts. Problems of the form (3.1) arise, for instance, in oscillatory hydraulic tomography [119] and lattice quantum chromodynamic [50]. Moreover, the extension of acoustic Helmholtz problems [22, 42, 123, 147] to a multi-frequency setting results in the framework (3.1). The focus of the present work, however, lies on situations where the discretization matrices $\mathcal{K}, \mathcal{M}$ stem from a discretization of the time-harmonic elastic wave equation [28]. Depending on the specific choice of boundary conditions the structure of the matrices varies, and the shifts s_k are either equal to the (angular) wave frequencies [8, 147] or to the squared (angular) wave frequencies [113]. For both situations, we will consider viscous damping by substituting $s_k \mapsto (1 - \epsilon i)s_k$, where $\epsilon > 0$ is the damping parameter and $i \equiv \sqrt{-1}$, cf. [2, 8, 113, 147].

Throughout this document we put emphasis on the case when the set of shifts $\{s_1, ..., s_{n_s}\}$ in (3.1) is distinct, and multiple frequencies are considered, i.e. $n_s > 1$. Without loss of generality, we assume the frequencies to be ordered, and, in particular, $s_1 = \min_k\{s_k\} =: s_{\min}$ and $s_{n_s} = \max_k\{s_k\} =: s_{\max}$ are the extreme frequencies. A large area of application where the fast solution of (3.1) at multiple frequencies is required is the so-called Full-Waveform Inversion modeled in frequency-domain; cf. [94, 105, 106, 153].

If the matrices in (3.1) are large and sparse, Krylov subspace methods are the common choice for the iterative numerical solution of (3.1). When re-formulating problem (3.1) to problems with shifted identity, Krylov methods can be particularly efficient, and variants of almost all popular Krylov methods have been derived for this type of problems (such as GMRES(k) [50], FOM(k) [117], BiCGstab(ℓ) [49] and IDR(s) [8, 148] among others). It is, however, difficult to apply a preconditioner and, at the same time, preserve the shifted structure: Most recently, polynomial preconditioners [1], flexible preconditioners [119], nested methods [8], and multi-preconditioned methods [6] have successfully been developed. Alternative approaches to solve sequences of linear systems such as (3.1) are the reformulation as a matrix equation [13], and the usage of information of previous solves called *recycling* [132, 133].

In most cases, a single preconditioner of the form,

$$\mathcal{P}(\tau) := (\mathcal{K} - \tau\mathcal{M}), \quad \text{with seed shift } \tau \in \mathbb{C}, \tag{3.2}$$

is applied where the choice of $\tau \in \mathbb{C}$ for a given set $\{s_1, ..., s_{n_s}\}$ is crucial for the convergence behavior of the overall algorithm, as has been pointed out in [6, 13, 119]. The present paper addresses the following:

1. We present an *optimal* choice for the seed parameter τ in (3.2) when a single shift-and-invert preconditioner is applied to (3.1). Our proposed choice is based on spectral analysis and the minimization of a classical GMRES convergence bound that also holds in the multi-shift framework.

2. Once a preconditioner of the form (3.2) is applied, the spectra are known to be bounded by circles which gives rise to the efficient application of a shifted Neumann preconditioner [1] as a second-level preconditioner. Our choice for τ minimizes its spectral radius.

3. The spectral analysis of the multi-shift framework is exploited for an equivalent matrix equation formulation of (3.1) studied in [13]. A simple post-rotation of the block spectrum yields a second-level preconditioner for the matrix equation and significantly speeds up convergence of global GMRES [68].

We point out that the analysis of an efficient seed parameter τ is fundamentally different from the single-frequency case studied in [147] since there is no trivial solution that needs to be excluded from the optimization. Moreover, note that we can *not* include directly frequency-dependent Sommerfeld boundary conditions in the preconditioner (3.2), as recommended in [84]. When (3.2) is applied inexactly using a multi-grid algorithm [112] or deflation [77], the choice of τ is usually combined with damping such that the multi-grid solver works well, cf. [22, 53] for an analysis in the Helmholtz case. We do not consider this aspect in this paper but note that the MSSS preconditioner developed in [13] allows to apply the inverse of (3.2) fast, even for large frequencies. We conclude with numerical examples obtained from a finite element discretization of the time-harmonic visco-elastic wave equation at multiple wave frequencies.

3.1 Treatment of The Time-Harmonic Elastic Wave Equation at Multiple Frequencies

The aim of this work is the efficient iterative solution of the elastic wave equation in a frequency-domain formulation. The displacement vector $\mathbf{u}(t, x)$ at time t and with spatial component x satisfies the elastic wave equation,

$$\rho \ddot{\mathbf{u}} = \nabla \cdot \sigma(\mathbf{u}) + \mathbf{s}, \quad \text{in } (0, T] \times \Omega, \quad \Omega \subset \mathbb{R}^d, \quad d \in \{2, 3\}, \tag{3.3}$$

with inhomogeneous material density $\rho = \rho(x)$, stress tensor σ, and source term $\mathbf{s}$, cf. [28]. We consider the following set of boundary conditions,

$$\rho \dot{\mathbf{u}} = -\sigma(\mathbf{u})\mathbf{n} \;\text{ for } x \in \partial\Omega_a \quad \text{and} \quad \sigma(\mathbf{u})\mathbf{n} = 0 \;\text{ for } x \in \partial\Omega_r, \tag{3.4}$$

where the condition on $\partial\Omega_r$ models reflection of waves, and the Sommerfeld radiation condition on $\partial\Omega_a$ is one way to model absorption. For the time-harmonic ansatz $\mathbf{u}(t, x) = \hat{\mathbf{u}}(x)e^{-i\omega t}$ substituted into (3.3)-(3.4) we obtain:

$$-\omega^2 \rho \hat{\mathbf{u}} - \nabla \cdot \sigma(\hat{\mathbf{u}}) = \hat{\mathbf{s}}, \quad \text{in } \Omega, \tag{3.5a}$$

$$i\omega\rho \; B(c_p, c_s) \; \hat{\mathbf{u}} + \sigma(\hat{\mathbf{u}})\mathbf{n} = 0, \quad \text{on } \partial\Omega_a, \tag{3.5b}$$

$$\sigma(\hat{\mathbf{u}})\mathbf{n} = 0, \quad \text{on } \partial\Omega_r. \tag{3.5c}$$

Note that, in the frequency-domain formulation (3.5a)-(3.5c), the Sommerfeld condition yields a term that is proportional to the frequency. For the definition of $B(c_p, c_s)$

in (3.5b) we refer to [2, 13]. In [13] a detailed derivation of a finite element discretization of (3.5a)-(3.5c) is presented that yields linear systems of the form,

$$(K + i\omega_k C - \omega_k^2 M)\hat{\mathbf{u}}_k = \hat{\mathbf{s}}, \quad k = 1, ..., N_\omega, \tag{3.6}$$

with K being a stiffness matrix, M being a mass matrix and C includes Sommerfeld boundary conditions. The angular frequencies ω_k appear quadratic in (3.6). Therefore, we apply a linearization [125] that results in block-systems of doubled dimensions,

$$\left(\begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} - \omega_k \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}\right) \begin{bmatrix} \omega_k \hat{\mathbf{u}}_k \\ \hat{\mathbf{u}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{s}} \\ 0 \end{bmatrix}, \quad k = 1, ..., N_\omega. \tag{3.7}$$

Let $\epsilon > 0$. We formally add *viscous damping* to (3.6) by introducing the set of complex frequencies $\hat{\omega}_k := (1 - \epsilon i)\omega_k$. The damped problem reads, $(K + i\hat{\omega}_k C - \hat{\omega}_k^2 M)\hat{\mathbf{u}}_k = \mathbf{b}$, for $k = 1, ..., N_\omega$, where the ansatz $\mathbf{u}(t, x) = \hat{\mathbf{u}}(x)e^{-i\hat{\omega}t} = \hat{\mathbf{u}}(x)e^{-i\omega t}e^{-\epsilon\omega t}$ now includes a damping term. When damping is added to the problem, spectral properties change and, in particular, the bounding circles that we describe in Section 3.3 do no longer touch the origin. The systems (3.7) are of the form (3.1), see Problem 3.1.

We also consider different types of absorbing boundary conditions on $\partial\Omega_a$: The case of purely non-mixed boundary conditions in (3.4) trivially yields $C \equiv 0$ in (3.6) and we, again, obtain a problem of the form (3.1). Absorption can also be modeled by introducing a *sponge* layer [113] or using perfectly matched layers (PML) [14]. The frequency-independent PML derived in [31] yields a term $C(\omega_0)$. In general, however, including PML boundary conditions yields a nonlinear term $C(\omega)$ that is not considered here. We summarize the above derivations by the following two problem statements.

Problem 3.1 *Consider the discretized time-harmonic elastic wave equation* (3.5a)-(3.5c) *with Sommerfeld radiation boundary conditions [2, 13] on $\partial\Omega_a \neq \emptyset$, and multiple angular frequencies $\omega_k \equiv 2\pi f_k$,*

$$(K + i\omega_k C - \omega_k^2 M)\hat{\mathbf{u}}_k = \hat{\mathbf{s}}, \quad k = 1, ..., N_\omega, \quad \{K, C, M\} \in \mathbb{C}^{N\times N},$$

where the matrices K and C are symmetric positive semi-definite, and M is symmetric positive definite. The linearization (3.7) *yields a shifted problem of the form* (3.1) *with, in particular, block matrices,*

$$\mathcal{K} := \begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} \in \mathbb{C}^{2N\times 2N}, \quad \mathcal{M} := \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \in \mathbb{C}^{2N\times 2N}, \quad \textit{shifts } s_k := \omega_k, \tag{3.8}$$

and right-hand side vector $\mathbf{b} := [\hat{\mathbf{s}}, 0]^{\mathsf{H}}$.

Remark 3.2 (Block decomposition) *In reformulation* (3.7)*, dimensions are doubled compared to the original problem size in* (3.6)*. For the preconditioner* (3.2)*, however, the following decomposition holds,*

$$\mathcal{P}(\tau)^{-1} = (\mathcal{K} - \tau\mathcal{M})^{-1} = \left(\begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} - \tau \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}\right)^{-1} \tag{3.9}$$

$$= \begin{bmatrix} I & \tau I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & (K + i\tau C - \tau^2 M)^{-1} \end{bmatrix} \begin{bmatrix} 0 & I \\ I & -iC + \tau M \end{bmatrix}, \tag{3.10}$$

where the (inexact) inversion is required only at the original problem size.

Problem 3.3 *Consider the discretized time-harmonic elastic wave equation* (3.5a)-(3.5c) *with frequency-independent PML [31] or 'sponge layer' [113] boundary conditions replacing* (3.5b)*, and angular frequencies* ω_k*,*

$$(K - \omega_k^2 M)\hat{\mathbf{u}}_k = \hat{\mathbf{s}}, \quad k = 1, ..., N_\omega,$$

where K *is symmetric positive semi-definite, and* M *is symmetric positive definite. The problem is, trivially, of the form* (3.1) *for* $\mathcal{K} := K, \mathcal{M} := M$ *and shifts* $s_k := \omega_k^2$ *equal to the squared angular frequencies.*

Remark 3.4 *For* $\epsilon > 0$*, the damped problem corresponding to Problem 3.1 and Problem 3.3, respectively, is given by the substitution,*

$$(\mathcal{K} - \hat{s}_k \mathcal{M})\hat{\mathbf{x}}_k = \mathbf{b}, \quad \text{where } \hat{s}_k := (1 - \epsilon i)s_k \ \text{ for } k = 1, ..., n_s. \tag{3.11}$$

In our following notation, we indicate quantities related to the damped problem with a hat.

3.2 The Seed Parameter of the Shift-And-Invert Preconditioner for Multi-Shift GMRES

In this section we briefly review the multi-shift GMRES method introduced in [50]. Throughout this paper we always consider the case when multi-shift GMRES is right-preconditioned by a preconditioner of the form (3.2). When applying a (scaled) shift-and-invert preconditioner as a right preconditioner to systems (3.1) the resulting preconditioned systems are shifted linear systems. Moreover, the shift parameter (sometimes called *seed frequency*) gives some freedom. Recall that we consider a sequence of problems of the form (3.1),

$$(\mathcal{K} - s_k \mathcal{M})\mathbf{x}_k = \mathbf{b}, \quad \text{for } k = 1, ..., n_s,$$

where the matrices $\mathcal{K}, \mathcal{M}$ are defined in Problem 3.1 or Problem 3.3, respectively. For $\tau \in \mathbb{C} \setminus \{0\}$, we define the shift-and-invert preconditioner $\mathcal{P}(\tau) = (\mathcal{K} - \tau\mathcal{M})$ as in (3.2). Applying right preconditioning to (3.1) using the scaled preconditioner $\mathcal{P}_k := 1/(1 - \eta_k)(\mathcal{K} - \tau\mathcal{M}) = 1/(1 - \eta_k)\mathcal{P}(\tau)$ yields,

$$(\mathcal{K} - s_k\mathcal{M})\mathcal{P}_k^{-1}\mathbf{y}_k = \mathbf{b} \quad \Leftrightarrow \quad (\mathcal{K}(\mathcal{K} - \tau\mathcal{M})^{-1} - \eta_k I)\mathbf{y}_k = \mathbf{b}, \tag{3.12}$$

where $\eta_k := s_k/(s_k - \tau)$. Note that the latter is a (preconditioned) shifted linear system with (possibly complex) shifts η_k and system matrix $\mathcal{K}\mathcal{P}(\tau)^{-1}$. Note further that the back-substitution $\mathbf{x}_k = \mathcal{P}_k^{-1}\mathbf{y}_k = (1 - \eta_k)(\mathcal{K} - \tau\mathcal{M})^{-1}\mathbf{y}_k$ can be computed efficiently for $k = 1, ..., n_s$. A similar equivalence as (3.12) that yields shifted systems with base matrix $\mathcal{M}(\mathcal{K} - \tau\mathcal{M})^{-1}$ is used in [119].

Remark 3.5 *It is well-known that Krylov subspaces are shift-invariant, i.e. for the matrix* $\mathcal{A} := \mathcal{K}(\mathcal{K} - \tau\mathcal{M})^{-1}$ *it holds,*

$$\mathcal{K}_m(\mathcal{A}, \mathbf{b}) \equiv \mathit{span}\{\mathbf{b}, \mathcal{A}\mathbf{b}, ..., \mathcal{A}^{m-1}\mathbf{b}\} = \mathcal{K}_m(\mathcal{A} - \eta I, \mathbf{b}) \quad \forall \eta \in \mathbb{C}, \forall m \in \mathbb{N}.$$

As a consequence, the shifted Arnoldi relation holds,

$$(\mathcal{A} - \eta I)V_m = V_{m+1}(\underline{H}_m - \eta \underline{I}),$$

where the columns of V_m *are an orthonormal basis of* $\mathcal{K}_m(\mathcal{A}, \mathbf{b})$ *that are computed by the Arnoldi method only once for all shifted systems, cf. Algorithm 3.1.*

Algorithm 3.1 Multi-shift GMRES with right preconditioning for (3.12), cf. [50]

```
Set $\mathbf{r}^{(0)} = \mathbf{b}$, $\beta = \|\mathbf{r}^{(0)}\|$, $\mathbf{v}_1 = \mathbf{r}^{(0)}/\beta$    ▷ Initialization with zero initial guess
for $j = 1$ to $m$ do
    Apply $\mathbf{w} = (\mathcal{K} - \tau\mathcal{M})^{-1}\mathbf{v}_j$    ▷ Apply preconditioner (3.2) (cf. Section 3.3)
    Compute $\mathbf{w} = \mathcal{K}\mathbf{w}$
    for $i = 1$ to $j$ do    ▷ Arnoldi method
        $h_{i,j} = \mathbf{w}^{\mathsf{H}}\mathbf{v}_i$
        $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
    end for
    Set $h_{j+1,j} = \|\mathbf{w}\|$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
    Set $\underline{\mathrm{H}}_m = [h_{i,j}]_{i=1,...,m+1}^{j=1,...,m}$ and $V_m = [\mathbf{v}_1, ..., \mathbf{v}_m]$    ▷ Orthogonal basis of $\mathcal{K}_m(\mathcal{A}, \mathbf{b})$, $\mathcal{A} := \mathcal{K}(\mathcal{K} - \tau\mathcal{M})^{-1}$
end for
for $k = 1$ to $n_s$ do    ▷ Loop over shifts
    Solve $\mathbf{y}_k = \mathrm{argmin}_{\mathbf{y}} \| \beta\mathbf{e}_1 - (\underline{\mathrm{H}}_m - \eta_k\underline{\mathrm{I}})\mathbf{y} \|$    ▷ Solve shifted Hessenberg systems
    Compute $\mathbf{x}_k = (1 - \eta_k)(\mathcal{K} - \tau\mathcal{M})^{-1}V_m\mathbf{y}_k$    ▷ Back-substitution
end for
```

In [147] the authors analyze spectral properties of the shifted Laplace preconditioner in the single-frequency case, $n_s = 1$, and exploit their analysis within preconditioned GMRES [118]. One of the results of [147] is the fact that the preconditioned spectrum (3.12) lies within circles of radius R and center c. Both are, in the single frequency case, a function of s_1 and τ (denoted by z_1 and z_2 in [147], respectively). Moreover, the authors of [147] show that in the absence of viscous damping the circles touch the origin, i.e. $R = |c|$. We state the following convergence bound for the GMRES residual norm that is in the absence of damping (compare Remark 3.4) of little practical use.

Theorem 3.6 (Classical convergence bound for GMRES, [117]) *Let the eigenvalues of the preconditioned matrix be enclosed by a circle with radius* R *and center* c. *Then the GMRES-residual norm after* j *iterations* $\|\mathbf{r}^{(j)}\|$ *satisfies,*

$$\frac{\|\mathbf{r}^{(j)}\|}{\|\mathbf{r}^{(0)}\|} \leq c_2(X) \left(\frac{R}{|c|}\right)^j, \quad \textit{where } \mathbf{r}^{(0)} = \mathbf{b} \;\; \textit{if } \mathbf{x}_0 \equiv \mathbf{0},$$

and where X *is the matrix of eigenvectors, and* $c_2(X)$ *denotes its condition number in the 2-norm.*

The convergence bound for GMRES described in Theorem 3.6 can be extended to the preconditioned multi-shift variant presented in Algorithm 3.1 in a straight-forward way. That is because in multi-shift GMRES optimality for the shifted residuals holds for all individual systems.

Corollary 3.7 (Convergence bound for multi-shift GMRES, [50, 117]) *An extension of the bound described in Theorem 3.6 to the (preconditioned) multi-shift GMRES-residual norms computed by Algorithm 3.1 is given by,*

$$\frac{\left\|\mathbf{r}_k^{(j)}\right\|}{\left\|\mathbf{r}^{(0)}\right\|} \leq c_2(X)\left(\frac{R_k}{|c_k|}\right)^j, \quad k=1,...,n_s, \quad j \leq m, \tag{3.13}$$

where the spectrum of the k-th shifted system after preconditioning is assumed to be enclosed by a circle of radius R_k and center point c_k, respectively.

The following section gives detailed explanations on the suitable choice of the bounding circles (c_k, R_k) in terms of the seed frequency τ. In particular, we will derive explicit formulas for these quantities, and make use of the fact that $|c_k| > R_k$ when viscous damping is added such that the bound in Corollary 3.7 can be exploited.

3.3 Spectral Analysis and Optimal Seed Shift Parameter τ^*

We describe the main result of this paper: The efficiency of the preconditioner in (3.12) highly depends on the choice of the seed parameter $\tau \in \mathbb{C}$. The following lemma provides insight how to choose this parameter such that the bound of Corollary 3.7 is minimized. The result yields an explicit formula for τ in terms of the considered frequency range $[s_{\min}, s_{\max}]$, and the damping parameter $\epsilon \geq 0$ as introduced in Remark 3.4.

Lemma 3.8 (Optimal seed frequency for preconditioned shifted GMRES) *Let $\{s_k\}_{k=1}^{n_s} \subseteq [s_{min}, s_{max}] \subset \mathbb{R}$. Consider the sequence of problems $(\mathcal{K} - s_k\mathcal{M})\boldsymbol{x}_k = \boldsymbol{b}$ with a right preconditioner $\mathcal{P}_k := 1/(1-\eta_k)(\mathcal{K} - \tau\mathcal{M})$,*

$$(\mathcal{K} - s_k\mathcal{M})\mathcal{P}_k^{-1}\mathbf{y}_k = \mathbf{b}, \quad \mathbf{x}_k = \mathcal{P}_k^{-1}\mathbf{y}_k, \quad \textit{for} \;\; k=1,...,n_s. \tag{3.14}$$

For $\eta_k = s_k/(s_k - \tau)$, problem (3.14) is equivalent to,

$$(\mathcal{K}(\mathcal{K} - \tau\mathcal{M})^{-1} - \eta_k I)\mathbf{y}_k = \mathbf{b}, \quad k=1,...,n_s, \tag{3.15}$$

where $\mathcal{P}(\tau) = (\mathcal{K} - \tau\mathcal{M})$ is the shift-and-invert preconditioner at seed frequency τ, and the matrices $\mathcal{K}$ and $\mathcal{M}$ and shifts $\{s_k\}_{k=1}^{n_s}$ are defined in Problem 3.1 and Problem 3.3, respectively. The following statements give guidance on choosing $\tau \in \mathbb{C}$ in an optimal sense:

(i) For λ from the spectrum of $\mathcal{K}\mathcal{M}^{-1}$, $\lambda \in \mathbf{\Lambda}[\mathcal{K}\mathcal{M}^{-1}]$, it holds $\Im(\lambda) \geq 0$.

(ii) *Let $\tau = \Re(\tau) + i\Im(\tau)$. The preconditioned spectra of (3.15) are enclosed by circles of radii R_k and center points c_k,*

$$c_k = \left(\frac{1}{2} - \frac{s_k(s_k - \Re(\tau))}{(s_k - \Re(\tau))^2 + \Im(\tau)^2}, \frac{\Re(\tau)}{2\Im(\tau)} - \frac{s_k\Im(\tau)}{(s_k - \Re(\tau))^2 + \Im(\tau)^2}\right) \in \mathbb{C},$$

$$R_k = \frac{1}{2}\sqrt{1 + \left(\frac{\Re(\tau)}{\Im(\tau)}\right)^2} =: R(\tau).$$

The preconditioned spectra of (3.15) with viscous damping $\hat{s}_k := (1 - \epsilon i)s_k$ are enclosed by circles of radii $\hat{R}_k = R(\tau)$ and center points $\hat{c}_k$,

$$\hat{c}_k = \left(\frac{1}{2} - \frac{(1+\epsilon^2)s_k^2 + (\epsilon\Im(\tau) - \Re(\tau))s_k}{(s_k - \Re(\tau))^2 + (\epsilon s_k + \Im(\tau))^2}, \frac{\Re(\tau)}{2\Im(\tau)} - \frac{(\Im(\tau) + \epsilon\Re(\tau))s_k}{(s_k - \Re(\tau))^2 + (\epsilon s_k + \Im(\tau))^2}\right).$$

(iii) *The set of points $\{\hat{c}_k\}_{k=1}^{n_s} \subset \mathbb{C}$ described in statement (ii) lie on a circle with center $\underline{c}$ and radius $\underline{R}$ given by,*

$$\underline{c} = \left(0, \frac{\epsilon|\tau|^2}{2\Im(\tau)(\Im(\tau) + \epsilon\Re(\tau))}\right), \quad \underline{R} = \sqrt{\frac{|\tau|^2(\epsilon^2+1)}{4(\Im(\tau) + \epsilon\Re(\tau))^2}}.$$

In the undamped case, $\epsilon = 0$, this center is equal to the origin and, therefore, $\underline{R} = R(\tau)$.

(iv) *Consider the preconditioner $\mathcal{P}(\tau^*) = (\mathcal{K} - \tau^*\mathcal{M})$ in (3.15). An optimal seed frequency τ^* for Algorithm 3.1 that minimizes the GMRES-bound in Corollary 3.7 is given by,*

$$\begin{aligned}\tau^*(\epsilon) &= \underset{\tau\in\mathbb{C}}{\operatorname{argmin}} \max_{k=1,..,n_s} \left(\frac{R(\tau)}{|\hat{c}_k|}\right)\\ &= \frac{2s_{min}s_{max}}{s_{min}+s_{max}} - i\frac{\sqrt{[\epsilon^2(s_{min}+s_{max})^2 + (s_{max}-s_{min})^2]\, s_{min}s_{max}}}{s_{min}+s_{max}},\end{aligned} \tag{3.16}$$

where $s_{min} := \min_k\{s_k\}$ and $s_{max} := \max_k\{s_k\}$.

Proof 3.9 *We prove this lemma 'step-by-step'.*

(i) *For $\mathcal{K}, \mathcal{M}$ as in Problem 3.3 the statement trivially holds since $\lambda \in \mathbb{R}$. With the assumption on K, C and M in Problem 3.1 we know that all eigenvalues μ_k of the quadratic eigenvalue problem,*

$$(K + \mu_k C + \mu_k^2 M)v_k = 0, \quad k = 1, ..., 2N,$$

are stable, i.e. $\Re(\mu_k) \le 0$, cf. [Table 1.1][137]. Eigenvalues of $\mathcal{K}\mathcal{M}^{-1}$ are then given by $\lambda_k := -i\mu_k$, and the corresponding eigenvectors are $\mathbf{v}_k := \mathcal{M}[-i\mu_k v_k, v_k]^\mathsf{T}$:

$$\begin{aligned} &\begin{bmatrix} iC & K \\ I & 0 \end{bmatrix}\begin{bmatrix} M^{-1} & 0 \\ 0 & I \end{bmatrix}\mathbf{v}_k = \lambda_k\mathbf{v}_k \\ \Leftrightarrow\quad &\begin{bmatrix} iC & K \\ I & 0 \end{bmatrix}\begin{bmatrix} -i\mu_k v_k \\ v_k \end{bmatrix} = -i\mu_k\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} -i\mu_k v_k \\ v_k \end{bmatrix} \\ \Leftrightarrow\quad &\begin{bmatrix} \mu_k C v_k + K v_k \\ -i\mu_k v_k \end{bmatrix} = \begin{bmatrix} -\mu_k^2 M v_k \\ -i\mu_k v_k \end{bmatrix}. \end{aligned}$$

The definition of λ_k *together with* $\Re(\mu_k) \leq 0$ *imply that* $\Im(\lambda_k) \geq 0$ *for Problem 3.1.*

(ii) *Consider the system matrix of* (3.15),

$$(\mathcal{K}(\mathcal{K}-\tau\mathcal{M})^{-1}-\eta_k I) = \mathcal{K}\mathcal{M}^{-1}(\mathcal{K}\mathcal{M}^{-1}-\tau I)^{-1}-\eta_k I, \quad \text{with } \eta_k = \frac{s_k}{s_k-\tau}.$$

The latter is a Möbius transformation with complex shift, hence the spectrum satisfies the mapping,

$$\mathbf{\Lambda}[\mathcal{K}\mathcal{M}^{-1}] \ni \lambda \mapsto \frac{\lambda}{\lambda-\tau} - \frac{s_k}{s_k-\tau}.$$

Since $\Im(\lambda) \geq 0$, *it is well-known [147] that for* $s_k \equiv 0$ *the Möbius transformation maps the spectrum within a circle of radius* $R = \left|\frac{\tau}{\tau-\bar{\tau}}\right|$ *and center* $c_0 = \frac{-\bar{\tau}}{\tau-\bar{\tau}}$. *In the shifted case, it holds:*

$$\hat{R}_k \equiv R(\tau) = \left|\frac{\tau}{\tau-\bar{\tau}}\right| = \frac{1}{2}\sqrt{1+\left(\frac{\Re(\tau)}{\Im(\tau)}\right)^2}, \tag{3.17}$$

$$\begin{aligned}\hat{c}_k &= \frac{-\bar{\tau}}{\tau-\bar{\tau}} - \frac{\hat{s}_k}{\hat{s}_k-\tau} = \frac{-\Re(\tau)+i\Im(\tau)}{2i\Im(\tau)} - \frac{s_k - i\epsilon s_k}{(s_k-\Re(\tau))-i(\epsilon s_k+\Im(\tau))} \\ &= \left(\frac{1}{2} - \frac{(1+\epsilon^2)s_k^2+(\epsilon\Im(\tau)-\Re(\tau))s_k}{(s_k-\Re(\tau))^2+(\epsilon s_k+\Im(\tau))^2}\right) + \\ &\qquad i\left(\frac{\Re(\tau)}{2\Im(\tau)} - \frac{(\Im(\tau)+\epsilon\Re(\tau))s_k}{(s_k-\Re(\tau))^2+(\epsilon s_k+\Im(\tau))^2}\right),\end{aligned} \tag{3.18}$$

where we again write $\tau = \Re(\tau)+i\Im(\tau)$. *The case* $\epsilon = 0$ *yields the corresponding result for* c_k *in the absence of viscous damping. Note that the radii* (3.17) *are independent of* k.

(iii) *We prove this fact by first constructing a center point* $\underline{c}$. *Therefore, we use two points* $\hat{c}_k$ *that are opposite to each other with real part zero,*

$$\Re(\hat{c}_k) = 0 \quad \Leftrightarrow \quad s_k^{1/2} = \pm|\tau|(\epsilon^2+1)^{-1},$$

where we note that negative frequencies are not considered. Substituting $s_k^{1/2}$ *into the imaginary part of* (3.18) *and computing the middle point yields,*

$$\underline{c} = \left(0, \frac{\epsilon|\tau|^2}{2\Im(\tau)(\Im(\tau)+\epsilon\Re(\tau))}\right). \tag{3.19}$$

We use Maple *to show that every point* $\hat{c}_k$ *has a constant distance from* $\underline{c}$. *This distance is the squared radius* $\underline{R}$,

$$\underline{R}^2 = \|\hat{c}_k - \underline{c}\|_2^2 = \frac{|\tau|^2(\epsilon^2+1)}{4(\Im(\tau)+\epsilon\Re(\tau))^2}, \quad \text{independent of } s_k. \tag{3.20}$$

(iv) *In part (iii) of this proof, we have shown that the center points* $\hat{c}_k$ (3.18) *of the preconditioned spectra lie on a circle with center* $\underline{c}$ (3.19) *and radius* $\underline{R}$ (3.20). *Therefore, an alternative parametrization of the distance to the origin* $|\hat{c}_k|$ *is given by,*

$$|\hat{c}_k|^2 = \underline{R}^2 + \Im(\underline{c})^2 + 2\Im(\underline{c})\underline{R}\sin(\varphi_k) = \underline{R}^2 - \Im(\underline{c})^2 + 2\Im(\underline{c})\Im(\hat{c}_k),$$

where the imaginary part of $\hat{c}_k$ *is given explicitly by* (3.18), *and* φ_k *is the corresponding phase angle, cf. Figure 3.1 (right). The expression for the GMRES bound in Corollary 3.7 can, hence, be simplified,*

$$\begin{aligned} \tau^*(\epsilon) &= \underset{\tau\in\mathbb{C}}{\operatorname{argmin}} \max_{k=1,..,n_s} \left(\frac{R(\tau)}{|\hat{c}_k|}\right) = \underset{\tau\in\mathbb{C}}{\operatorname{argmin}} \max_{k=1,..,n_s} \left(\frac{R(\tau)^2}{|\hat{c}_k|^2}\right) && (3.21)\\ &= \underset{\tau\in\mathbb{C}}{\operatorname{argmin}} \max_{k=1,..,n_s} \left(\frac{R^2}{\underline{R}^2 + \Im(\underline{c})^2 + 2\Im(\underline{c})\underline{R}\sin(\varphi_k)}\right) \\ &\overset{(*)}{=} \underset{\tau\in\mathbb{C}}{\operatorname{argmin}} \max_{k\in\{1,n_s\}} \left(\frac{R^2}{\underline{R}^2 - \Im(\underline{c})^2 + 2\Im(\underline{c})\Im(\hat{c}_k)}\right) && (3.22)\\ &\overset{(**)}{=} \{\tau\in\mathbb{C}\,|\,\Im(\hat{c}_1) = \Im(\hat{c}_{n_s})\}\,, && (3.23)\end{aligned}$$

where in step $(*)$ *we use that* $\sin(\varphi_k)$ *obtains its minimum at the boundary (the case* $\sin(\varphi_k) = -1$ *has been excluded in part (iii)), and in step* $(**)$ *we use that the minimum of the maximum of two functions occurs when the two functions are equal. Setting the imaginary parts equal* (3.21) *yields:*

$$\begin{aligned} &\frac{s_{min}}{(s_{min} - \Re(\tau^*))^2 + (\epsilon s_{min} + \Im(\tau^*))^2} = \frac{s_{max}}{(s_{max} - \Re(\tau^*))^2 + (\epsilon s_{max} + \Im(\tau^*))^2} \\ \Rightarrow \quad &\Re(\tau^*)^2 + \Im(\tau^*)^2 = s_{min}s_{max}(1+\epsilon^2).\end{aligned}$$

We next express $\tau^* = s^* e^{i\varphi^*}$ *in polar coordinates, with length given by* $s^* = \sqrt{s_{min}s_{max}(1+\epsilon^2)}$. *Expressing the objective function* $\mathcal{J} := R^2[\underline{R}^2 - \Im(\underline{c})^2 + 2\Im(\underline{c})\Im(\hat{c}_k)]^{-1}$ *in terms of* (s,φ), *we can use* Maple[†] *to solve* $\frac{\partial \mathcal{J}(s^*,\varphi)}{\partial\varphi} = 0$ *for the unique minimum,*

$$\varphi^* = \varphi^*(s_{min}, s_{max}, \epsilon) = \arctan\left(-\sqrt{\frac{\epsilon^2(s_{min}+s_{max})^2 + (s_{max}-s_{min})^2}{4s_{min}s_{max}}}\right). \tag{3.24}$$

The conversion to Cartesian coordinates completes the proof. ☐

Remark 3.10 *Note that for* $\epsilon = 0$, *the ratio* (3.16) *equals to* 1, *cf.* [147]. *In this limit case, the optimal seed frequency yields,*

$$\begin{aligned}\tau^*(0) &= \frac{2s_{min}s_{max}}{s_{min}+s_{max}} - i\frac{|s_{max}-s_{min}|\sqrt{s_{min}s_{max}}}{s_{min}+s_{max}} \\ &= \sqrt{s_{min}s_{max}}\exp\left(i\arctan\left(-\frac{|s_{max}-s_{min}|}{2\sqrt{s_{min}s_{max}}}\right)\right), && (3.25)\end{aligned}$$

[†] We added corresponding `Maple [v 18.02]` scripts to our public repository (`https://github.com/ManuelMBaumann/opt_tau`).

which is a function of the geometric mean of the extreme frequencies, and their distance.

Remark 3.11 *For the mono-frequency case, $s_{min} = s_{max} \equiv s$, we obtain the expected (trivial) results $\tau^*(\epsilon) = s\sqrt{1+\epsilon^2}e^{i\arctan(-\epsilon)} = (1-\epsilon i)s = \hat{s}$, and $\tau^*(0) = s$.*

We illustrate the results of Lemma 3.8 in Figure 3.1: The left figure demonstrates the optimality of τ^* as stated in (3.16). Moreover, we plot the angle along which we have optimized in (3.24) as a dashed line. In the right figure, we show the corresponding preconditioned spectrum with bounding circles for a surrogate problem. This distribution of the circles corresponds to the case when τ minimizes the bound (3.16). Since the radii of all preconditioned spectra have the same magnitude, we see that the two extreme frequencies are expected to converge slowest because the respective distances to the origin is smallest which yields a worst case for the bound in Corollary 3.7.

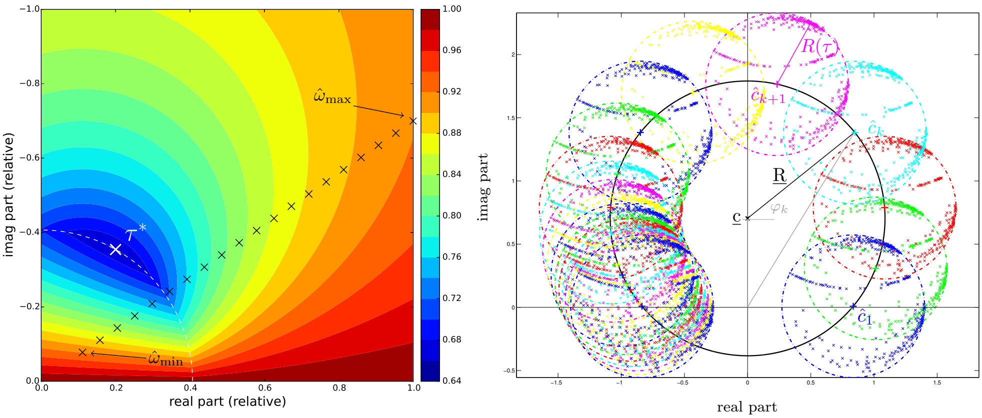


Figure 3.1: *Left: Convergence bound and optimal seed frequency τ^* according to* (3.16)*. Along the dashed line the function is differentiable. Right: Preconditioned spectra and surrounding circles for positive damping $\epsilon = 0.7$ added to Problem 3.1. Here, we use $N_\omega = 20$ frequencies equally spaced within the interval $[1, 9]$Hz. The imaginary parts that belong to the extreme frequencies are equal as imposed in* (3.21)*, but not equal to zero.*

As a result of Lemma 3.8, we see that the GMRES-bound in Corollary 3.7 and the location of the optimal seed frequency (3.16) are explicit functions of the damping parameter ϵ and the extreme frequencies $[s_{\min}, s_{\max}]$. This is both illustrated in Figure 3.2. The optimization of the seed parameter is obtained based on the damped problem. Because of continuity in Figure 3.2 (left), we note that τ^* smoothly depends on the damping factor ϵ which motivates the choice for the seed parameter in Remark 3.10 in the limit case $\epsilon \to 0$.

Lemma 3.8 does not give information about the actual value of the multi-shift GMRES bound (3.13) other than $R_k/|c_k| \equiv 1$ when $\epsilon = 0$. In the following corollary we show that when τ^* is chosen according to (3.16), the bound $R_k/|c_k|$ evaluated at τ^* is a function of the damping parameter $\epsilon > 0$ and the ratio $\rho := s_{\max}/s_{\min}$ only.

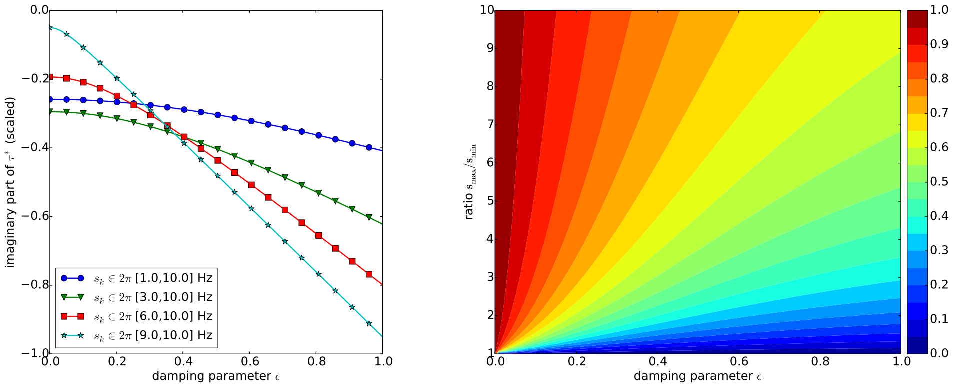


Figure 3.2: *Left: Imaginary part of τ^* scaled by s_{max} for different intervals $[s_{\min}, s_{\max}]$ and varying damping parameter ϵ. Right: GMRES-bound (3.13) at τ^* as a function of the ratio s_{max}/s_{min} and ϵ. Note that in (3.16) the real part of τ^* is independent of ϵ and is, therefore, not plotted here.*

Corollary 3.12 *Let $\epsilon > 0$, and $\tau^* = \tau^*(\epsilon, s_{min}, s_{max})$ as in (3.16) for a frequency interval $[s_{min}, s_{max}]$. Then there exists a function $f(\epsilon, s_{max}/s_{min})$ such that,*

$$\frac{R_k(\tau^*)}{|c_k(\epsilon,\tau^*)|} = f(\epsilon,\rho), \quad \text{where } \rho := s_{max}/s_{min}, \tag{3.26}$$

i.e. the bound in (3.13) depends only on the damping parameter ϵ and the ratio of the interval boundaries ρ. The quantities R_k and c_k are,

$$R_k(\tau^*) = R(\tau^*) = \frac{1}{2}\sqrt{1+\left(\frac{\Re(\tau^*)}{\Im(\tau^*)}\right)^2},$$

$$c_k(\epsilon,\tau^*) \overset{\epsilon\geq 0}{=} \hat{c}_k(\tau^*) = \begin{pmatrix} \frac{1}{2} - \frac{(1+\epsilon^2)s_k^2 + (\epsilon\Im(\tau^*)-\Re(\tau^*))s_k}{(s_k-\Re(\tau^*))^2+(\epsilon s_k+\Im(\tau^*))^2} \\ \frac{\Re(\tau^*)}{2\Im(\tau^*)} - \frac{(\Im(\tau^*)+\epsilon\Re(\tau^*))s_k}{(s_k-\Re(\tau^*))^2+(\epsilon s_k+\Im(\tau))^2} \end{pmatrix} \in \mathbb{C},$$

according to Lemma 3.8(ii).

Proof 3.13 *We show that*

$$\frac{R(\tau^*(\epsilon,s_{min},s_{max}))}{|c_k(\epsilon,\tau^*(\epsilon,s_{min},s_{max}))|} = \frac{R(\tau^*(\epsilon,d\cdot s_{min},d\cdot s_{max}))}{|c_k(\epsilon,\tau^*(\epsilon,d\cdot s_{min},d\cdot s_{max}))|}, \quad k\in\{1,n_s\}$$

for any scalar d. First, note that

$$\tau^*(\epsilon,s,t) = \frac{2st}{s+t} - i\frac{\sqrt{[\epsilon^2(s+t)^2+(t-s)^2]\,st}}{s+t},$$

is a homogeneous function of degree 1 with respect to the second and third argument, i.e., $\tau^*(\epsilon, ds_{min}, ds_{max}) = d\cdot\tau^*(\epsilon, s_{min}, s_{max})$. Therefore, the real and imaginary part *of* τ^* *scale with* d *in the same way. This implies,*

$$R(d\ \tau^*) = \frac{1}{2}\sqrt{1 + \left(\frac{d\ \Re(\tau^*)}{d\ \Im(\tau^*)}\right)^2} = R(\tau^*),$$

and, moreover,

$$\Re(c_1(\epsilon, d\tau^*)) = \frac{1}{2} - \frac{d^2(1+\epsilon^2)s_1^2 + d^2(\epsilon\Im(\tau^*) - \Re(\tau^*))s_1}{d^2(s_1 - \Re(\tau^*))^2 + d^2(\epsilon s_1 + \Im(\tau^*))^2} = \Re(c_1(\epsilon, \tau^*)),$$
$$\Im(c_1(\epsilon, d\tau^*)) = \frac{d\Re(\tau^*)}{2d\Im(\tau^*)} - \frac{d^2(\Im(\tau^*) + \epsilon\Re(\tau^*))s_1}{d^2(s_1 - \Re(\tau^*))^2 + d^2(\epsilon s_1 + \Im(\tau^*))^2} = \Im(c_1(\epsilon, \tau^*)),$$

and, in the same way, $c_{n_s}(\epsilon, d\tau^*) = c_{n_s}(\epsilon, \tau^*)$*, where we associate* s_{min} *with* s_1 *(i.e.* $k = 1$*), and* s_{max} *with* s_{n_s}*, cf.* (3.22) *in Lemma 3.8(iv). Thus,* s_1 *and* s_{n_s} *scale with* d *as the interval boundaries do.* ☐

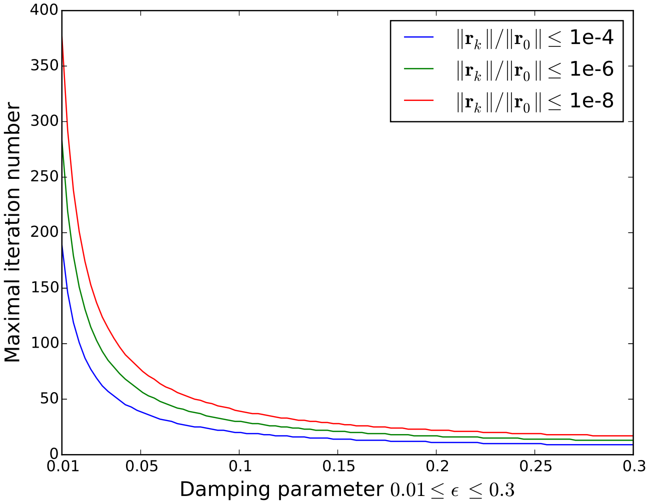


Figure 3.3: *Number of iterations* j *such that the relative residual norm is bounded by* $\{$*1e-4, 1e-6, 1e-8*$\}$*. The residual* $\mathbf{r}_k$ *is associated with the (angular) wave frequency* $s_k \in [s_{min}, s_{max}]$ *with fixed ratio* $\rho = s_{max}/s_{min} = 2$*, and* $c_2(X)$ *set to* 1 *in* (3.13)*.*

The bound (3.26) is plotted as a function of the damping parameter ϵ and the ration $\rho = s_{\max}/s_{\min}$ in Figure 3.2 (right). If ρ is kept constant, the GMRES bound implies an a priori known maximum iteration number for a fixed relative residual tolerance, cf. Figure 3.3. The numerical experiment 3.19 exploits the insight of Corollary 3.12 when splitting a frequency interval $[s_{\min}, s_{\max}]$ into subintervals in a balanced way.

3.4 Areas of Application in a Two-Level Preconditioning Framework

We present two examples in which the insight of the previous section is exploited for the design of efficient two-level preconditioners: In Section 3.4.1 we make use of the spectral bounds of Lemma 3.8 in order to choose the parameter of the shifted Neumann polynomial preconditioner [1] such that its spectral radius is minimal. In Section 3.4.2 we present a reformulation of the multi-frequency problem (3.1) as a matrix equation, cf. [13, 10]. Here, the convergence behavior depends on the union of the spectra of all considered frequencies and, hence, a suitable rotation of the bounding circles yields an efficient second-level preconditioner for global GMRES [68].

3.4.1 Shifted Neumann Preconditioning Techniques

For a set of frequencies $\{s_k\}_{k=1}^{n_s}$ and viscous damping parameter $\epsilon > 0$, the preconditioned shifted problems (3.15) at an optimal seed frequency τ^* given explicitly by (3.16) read,

$$(\mathcal{A} - \eta_k I)\mathbf{y}_k = \mathbf{b}, \quad \text{with } \mathcal{A} := \mathcal{K}(\mathcal{K} - \tau^* \mathcal{M})^{-1} \text{ and } \eta_k := \frac{\hat{s}_k}{\hat{s}_k - \tau^*} = \frac{(1 - \epsilon i)s_k}{(1 - \epsilon i)s_k - \tau^*}.$$

For $s_k = 0$, the spectrum of the matrix $\mathcal{A}$ is bounded by a circle of radius R and center c_0 as stated in part (ii) of Lemma 3.8,

$$R = \frac{1}{2}\sqrt{1 + \left(\frac{\Re(\tau^*)}{\Im(\tau^*)}\right)^2} \quad \text{and} \quad c_0 = \frac{\bar{\tau}^*}{\bar{\tau}^* - \tau^*} = \left(\frac{1}{2}, \frac{\Re(\tau^*)}{2\Im(\tau^*)}\right).$$

We consider a Neumann preconditioner [117, Chapter 12.3] of degree n as an approximation to the inverse of $\mathcal{A}$,

$$\mathcal{A}^{-1} \approx \sum_{i=0}^{n} (I - \xi \mathcal{A})^i =: p_n(\mathcal{A}), \quad \text{with parameter } \xi = \frac{1}{c_0} = \frac{\bar{\tau}^* - \tau^*}{\bar{\tau}^*}. \tag{3.27}$$

and its monic representation $p_n(\mathcal{A}) = \sum_{i=0}^{n} \gamma_i \mathcal{A}^i$. Shift-invariance, cf. Remark 3.5, can be preserved by the Neumann preconditioner if the following holds,

$$(\mathcal{A} - \eta_k I) p_{n,k}(\mathcal{A}) = \mathcal{A} p_n(\mathcal{A}) - \tilde{\eta}_k I, \tag{3.28}$$

where $p_{n,k}(\mathcal{A}) = \sum_{i=0}^{n} \gamma_{i,k} \mathcal{A}^i$ is a polynomial preconditioner of degree n for the k-th shifted matrix $(\mathcal{A} - \eta_k I)$. Substitution yields,

$$\sum_{i=0}^{n} \gamma_{i,k} \mathcal{A}^{i+1} - \sum_{i=0}^{n} \eta_k \gamma_{i,k} \mathcal{A}^i - \sum_{i=0}^{n} \gamma_i \mathcal{A}^{i+1} + \tilde{\eta}_k I = 0. \tag{3.29}$$

The latter (3.29) is a difference equation and can be solved backwards [1]:

$$\begin{aligned} \gamma_{n,k} &= \gamma_n, \\ \gamma_{i-1,k} &= \gamma_{i-1} + \eta_k \gamma_{i,k}, \quad \text{for } i = n, ..., 1, \\ \tilde{\eta}_k &= \eta_k \gamma_{0,k}. \end{aligned}$$

As a result, we solve the shifted systems on the right-hand side in (3.28) with Algorithm 3.1 using the Neumann preconditioner (3.27) of degree n. An alternative polynomial preconditioner for shifted systems is derived in [160].

Remark 3.14 *Note that* $p_n(\mathcal{A}) = c_0\mathcal{A}^{-1}$*, as* $n \to \infty$*, converges to the scaled inverse of* $\mathcal{A}$*. From* (3.28) *we conclude that* $p_{n,k}(\mathcal{A}) = (c_0 - \tilde{\eta}_k)(A - \eta_k I)^{-1}, n \to \infty$*, and, hence,* $p_{n,k}$ *is a polynomial preconditioner of degree* n *for the* k*-th shifted problem* $(A - \eta_k I)$*.*

3.4.2 A Spectral Scaling Strategy for the Matrix Equation Formulation

An alternative approach to efficiently solve multi-frequency wave problems is to rewrite the discretized problem as a matrix equation $A(\mathbf{X}) = B$, where the block unknown $\mathbf{X}$ is the stacked numerical solution at different frequencies [13] and A is a linear operator of the form, $A(\mathbf{X}) = \sum_{j=1}^{J} A_j \mathbf{X} B_j^{\mathsf{H}} = C,$ where $J = 2$ in (3.30). Consider, for instance, Problem 3.3 and the reformulation,

$$A(\mathbf{X}) := K\mathbf{X} - M\mathbf{X}\Omega^2 = B, \quad \text{where } \Omega := \operatorname{diag}(\omega_1, ..., \omega_{N_\omega}),\ B := \hat{\mathbf{s}}\mathbf{1}^{\mathsf{T}}, \tag{3.30}$$

and where the unknown is $\mathbf{X} := [\hat{\mathbf{u}}_1, ..., \hat{\mathbf{u}}_{N_\omega}]$. The spectrum of the operator A is equal to the spectrum of the block diagonal matrix of the corresponding vectorized problem,

$$\begin{bmatrix} (K - \omega_1^2 M) & & \\ & \ddots & \\ & & (K - \omega_{N_\omega}^2 M) \end{bmatrix} \begin{pmatrix} \hat{\mathbf{u}}_1 \\ \vdots \\ \hat{\mathbf{u}}_{N_\omega} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{s}} \\ \vdots \\ \hat{\mathbf{s}} \end{pmatrix},$$

and, hence, the spectrum of A equals the union of the spectra of the shifted systems in Problem 3.3. Consider the iterative solution of (3.30) with a global Krylov method such as global GMRES [68]. The analysis of Section 3.3 can be used in order to improve the convergence of global GMRES for the matrix equation (3.30). Therefore, we define the preconditioners,

$$P_1^{-1}(\mathbf{Y}) := (K - \tau^2 M)^{-1}\mathbf{Y}\Gamma, \quad \text{where } \Gamma := \operatorname{diag}(1 - \eta_1, ..., 1 - \eta_{N_\omega}), \tag{3.31a}$$

$$P_2^{-1}(\mathbf{Y}) := \mathbf{Y}R, \quad \text{where } R := \operatorname{diag}\left(1, e^{-i(\phi_2 - \phi_1)}, ..., e^{-i(\phi_{N_\omega} - \phi_{N_\omega - 1})}\right), \tag{3.31b}$$

where ϕ_k is the angular component of $\hat{c}_k$ in Lemma 3.8(ii), and $\eta_k := \omega_k^2/(\omega_k^2 - \tau)$ as in (3.12). Because of the correspondence of the spectrum of the matrix operator with the shifted systems, the above preconditioners have the following interpretation that are illustrated in Figure 3.4: The application of P_1^{-1} as a right preconditioner in global GMRES is equivalent to (3.14) and the left-hand side in (3.12) and yields a spectrum that is equal to the union of the circles described in Lemma 3.8. For the fast convergence this spectrum is not favorable. Therefore, we apply P_2^{-1} as a second-level preconditioner that yields a rotation of the spectrum to the right half plane.

Note that the preconditioners (3.31a) and (3.31b) commute, i.e. $(A \circ P_1^{-1}) \circ P_2^{-1} = (\mathcal{A} \circ P_2^{-1}) \circ P_1^{-1}$. For global GMRES, this allows to apply P_1 inexactly and, trivially, apply P_2 exactly.

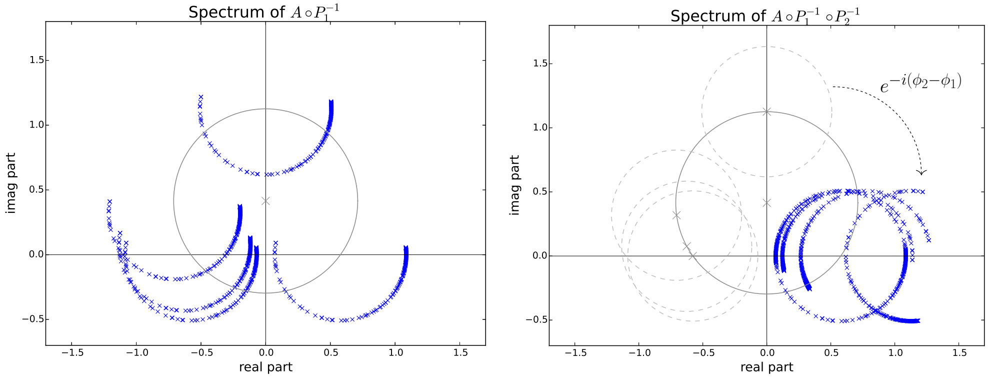


Figure 3.4: *Left: Block spectrum of the linear operator* (3.30) *with shift-and-invert preconditioner* (3.31a) *that is equivalent to the multi-shift approach. Right: Block spectrum after additional rotation with* (3.31b)*. We use a surrogate problem with* $N_\omega = 5$ *frequencies equally spaced within* $f_k \in [1, 9]$ *Hz.*

Remark 3.15 *In the next section we also consider the case when Problem 3.1 is reformulated as the matrix equation* $A(\mathbf{X}) := K\mathbf{X} + iC\mathbf{X}\Omega - K\mathbf{X}\Omega^2 = B$*, i.e. the situation when absorbing boundary conditions are present in a matrix equation reformulation, cf. Experiment 3.18.*

3.5 Numerical Experiments

We present numerical examples for a finite element discretization[†] of the time-harmonic elastic wave equation (3.5a)-(3.5c) in 2D. The problem setting shown in Figure 3.5 (left) is an inhomogeneous *wedge* problem inspired by the acoustic analogue proposed in [104] which has been used for the demonstration of spectral analysis of Helmholtz problems in [147]. The computational domain is $\Omega = [0, 600] \times [0, 1000]$ meter. Whenever Problem 3.1 is solved, we place a point source at $(300, 0)$ meter and prescribe absorbing Sommerfeld conditions on the upper boundary, cf. Figure 3.5 (middle). Generally speaking, Problem 3.3 is easier to solve numerically. In our numerical examples we consider Problem 3.3 with a point source at the center of Ω and consider reflecting walls on the entire boundary, cf. Figure 3.5 (right).

Numerical Experiment 3.16 (Proof-of-concept) *We numerically demonstrate the key findings of Lemma 3.8. In particular, we show that* $\tau^*(\epsilon, \omega_{min}, \omega_{max})$ *is independent of the number of frequencies within* $[\omega_{min}, \omega_{max}]$*. Moreover, we show mesh-independency of* τ^* *and demonstrate the direct connection between the bounding circles described in Lemma 3.8 (ii) and the convergence behavior of multi-shift GMRES applied to Problem 3.1.*

[†] For the finite element discretization we use the Python package `nutils` [149].

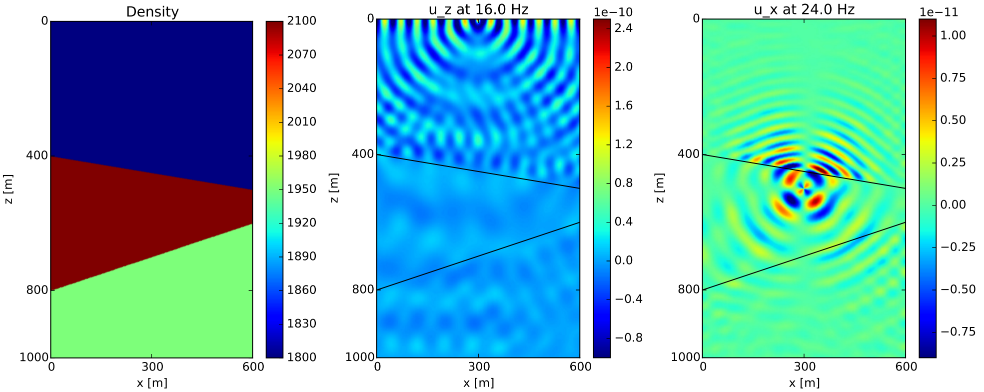


Figure 3.5: *Numerical set-up: Material density ρ in $[kg/m^2]$ (left), and numerical solution of Problem 3.1 at $\epsilon = 0$ (middle) and Problem 3.3 with damping $\epsilon = 0.05$ (right) at different frequencies. See [13] for a detailed description of the test problem.*

The experiments reported in Table 3.1 are performed at fixed frequency range and for a fixed damping parameter. The shift-and-invert preconditioner is applied to multi-shift GMRES at the optimal seed frequency corresponding to (3.16). We consider N_ω equidistantly-spaced frequencies. The results show the expected result that more frequencies within the same interval can be solved at no extra iterations, and at low extra computational costs. When repeating some of the experiments on a finer mesh we conclude mesh-independence of the optimal seed parameter, cf. Table 3.2.

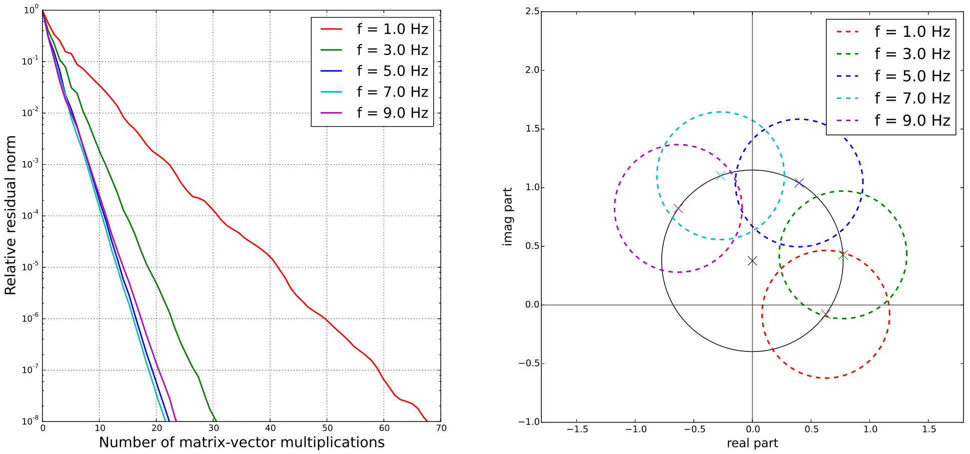
Table 3.1: *Multi-shift GMRES using the optimal seed parameter τ^* according to Lemma 3.8, and fixed damping $\epsilon = 0.05$. Discretization size of $h_x = h_z = 5m$ implies $N = 48,642$ dofs.*

$\omega_{\min}/2\pi$ [Hz]	$\omega_{\max}/2\pi$ [Hz]	N_ω	# iterations	`CPU time [s]`
1	5	5	106	45.6
		10	106	48.7
		20	106	47.3
1	10	5	251	205.1
		10	252	223.7
		20	252	243.5

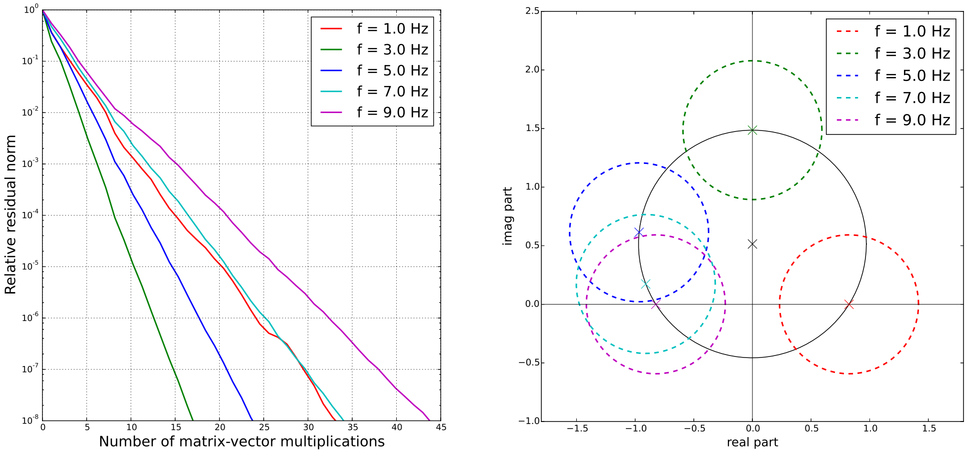
Table 3.2: *Setting as in Table 3.1 and discretization size half compared to Table 3.1.*

$\omega_{\min}/2\pi$ [Hz]	$\omega_{\max}/2\pi$ [Hz]	N_ω	# iterations	`CPU time [s]`
1	5	10	103	189.4
1	10	10	246	770.10

We next demonstrate the close relation between the spectral bounds derived in Lemma 3.8 and the convergence behavior of multi-shift GMRES preconditioned with a shift-and-invert preconditioner at optimal seed frequency τ^*. We, therefore, consider the same multi-frequency setting at two different seed frequencies. Figure 3.6 shows the respective convergence curves next to the bounding circles described in Lemma 3.8. When comparing the two choices for τ, we note that the circles corresponding to the optimal τ are further away from the origin which yields a smaller bound in Corollary 3.7. Moreover, the *outlier* in Figure 3.6a motivates the min-max criterion chosen for the optimization in (3.16).



(a) Convergence curves for $\tau = (0.3 - 0.7i)\omega_{\max}$ (b) Bounding circles when $\tau = (0.3 - 0.7i)\omega_{\max}$



(c) Convergence curves exploiting Lemma 3.8, $\tau = \tau^* = (0.2 - 0.35i)\omega_{\max}$ (d) Bounding circles exploiting Lemma 3.8, $\tau = \tau^* = (0.2 - 0.35i)\omega_{\max}$

Figure 3.6: *Relation between convergence of multi-shift GMRES and spectral bounds of Lemma 3.8. Here, we chose* $\epsilon = 0.7$ *which yields a value for the multi-shift GMRES bound of* `0.812` *at* $\tau = (0.3 - 0.7i)\omega_{max}$ *which is significantly larger than* `0.659` *obtained at the optimum.*

Numerical Experiment 3.17 (Shifted Neumann preconditioner) *In this numerical experiment we study the effect of the shifted Neumann preconditioner* (3.27) *on the convergence behavior of multi-shift GMRES within the two-level preconditioning technique described in Section 3.4.1.*

A major drawback of GMRES (and its multi-shift variant) is the increasing computational work and memory requirement when the number of iterations grows. This can be overcome by restarting Algorithm 3.1, cf. [50]. If the matrix-vector operation is relatively cheap, polynomial preconditioners [145] are an important alternative. If a Neumann polynomial (3.27) of degree n is applied, the number of matrix-vector products per iteration is $n + 1$. The experiment in Table 3.3 shows that GMRES iteration numbers and computation times can be reduced by approximately a factor of 4 compared to the case without shifted Neumann preconditioner ($n = 0$).

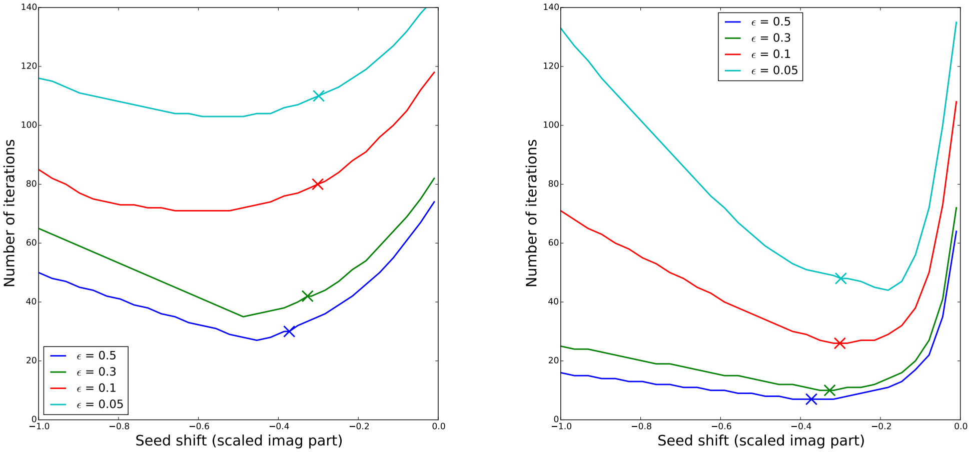
In Figure 3.7 we compare convergence for different values of the seed frequency, and Neumann preconditioners of degree $n = 0$ (only shift-and-invert) and $n = 5$. We observe that the seed parameter τ^* that minimizes the bound in Corollary 3.7 yields an iteration number close to the optimum. Moreover, we note in Figure 3.7a that in the case of large damping the bound in Corollary 3.7 is more descriptive and the choice of τ has a larger influence on the convergence of multi-shift GMRES. For small damping ($\epsilon = 0.05$) on the other hand, the parameter choice in the Neumann preconditioner at degree $n = 5$ gains importance and iteration numbers can be reduced up to a factor of 2, cf. Figure 3.7b.

Table 3.3: *Multi-shift GMRES without restarting using optimal seed parameter* τ^* *according to Lemma 3.8,* $\epsilon = 0.05$*, and a Neumann polynomial preconditioner* (3.27) *of degree n. We consider* $N_\omega = 10$ *frequencies in Problem 3.1 equally spaced within the interval* $\omega_k/2\pi = [1, 10]$*Hz. The problem size is* $N = 48,642$ *dofs.*

$n =$	10	5	4	3	2	1	0
# iterations	45	64	94	80	121	150	252
`CPU time [s]`	`52.10`	`46.88`	`64.69`	`47.39`	`73.21`	`89.74`	`213.18`

Numerical Experiment 3.18 (Matrix equation (3.30) with spectral rotation) *In this experiment we use global GMRES [68] to solve the matrix equation* (3.30) *preconditioned by* (3.31a). *The experiment demonstrates the benefit of spectral rotation* (3.31b) *as a second-level preconditioner for the matrix equation approach described in Section 3.4.2.*

We solve Problem 3.3 reformulated as a matrix equation (3.30) using global GMRES [68]. As explained in Section 3.4.2 the block preconditioner P_1 (3.31a) yields the spectral situation of Lemma 3.8 in a matrix equation framework. When $C = 0$ the eigenvalues of the preconditioned linear operator lie *on* the bounding circles described in Lemma 3.8(ii), see Figure 3.4. In Table 3.4 we evaluate the effect of the rotation P_2 (3.31b) for two different frequency ranges and for different total number of frequencies $N_\omega = \{5, 15\}$. The comparison shows clearly the benefit of rotating the spectrum. This becomes more evident when the number of frequencies is increased from 5 to 15. Since all circles are rotated on top of each other, the clustering of the



(a) Without Neumann preconditioner ($n = 0$) (b) Neumann preconditioner of degree $n = 5$

Figure 3.7: *Optimality of the shifted Neumann preconditioner. Left: Without preconditioner. Right: Neumann polynomial preconditioner at* $n = 5$. *The optimal seed parameter* τ^* *is marked with a cross, and compared against values of* τ *with varying imaginary part. All values are scaled by* $\omega_{\max}$. *In the present experiment, the frequency range is fixed at* $f_k \in [1, 5]$ *Hz with* $N_\omega = 5$ *and* $12,322$ *dofs.*

spectrum is the same in both cases and we observe an almost equal iteration number for global GMRES. Moreover, we consider the case where Sommerfeld boundary conditions are present and the matrix equation in Remark 3.15 is solved. The effects with respect to P_2 are similar.

Numerical Experiment 3.19 (An interval splitting strategy) *We consider an interval* $\mathcal{I} = [\omega_{\min}, \omega_{\max}]$ *of equidistantly spaced frequencies, and assume* $n_p > 1$ *available parallel processors. This experiment investigates a strategy for splitting* $\mathcal{I}$ *into* n_p *subintervals based on the choice for* τ^* *according to* (3.16) *for each subinterval such that a balanced load in agreement with Corollary 3.12 is achieved.*

For fixed damping parameter $\epsilon > 0$, the optimal seed frequency in (3.16) is a function of the frequency range of the original problem only, i.e. $\tau^* = \tau(\epsilon, \omega_{\min}, \omega_{\max})$. Consider first the case where $n_p = 2$ CPUs are present and the interval of frequencies can be split into two parts, $\mathcal{I} = [\omega_{\min}, \omega_{\max}] = [\omega_{\min}, \omega_{\text{mid}}] \cup [\omega_{\text{mid}}, \omega_{\max}]$ with seed parameter chosen optimally according to (3.16) for both subinterval. In Figure 3.8a this splitting point ω_{mid} is varied, and the largest iteration number (marked by crosses) and the larger bound (3.13) at the respective optimum for the two subintervals is reported. We conclude that the best splitting point is when the boundary ratios are equal, i.e. $\omega_{\text{mid}}/\omega_{\min} = \omega_{\max}/\omega_{\text{mid}}$, which is obtained at the geometric mean at $\omega_{\text{mid}} = 3$ Hz in Figure 3.8a. In the subsequent experiments in Figure 3.8b and 3.8c, we report the upper interval boundary of the first subinterval and apply the previously derived splitting strategy *inductively* to the $n_p - 1$ remaining subintervals. In conclusion, a splitting equidistantly on a logarithmic scale yields best results, and in

Table 3.4: *Solution of* (3.30) *with global GMRES: The preconditioners* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *are as defined in* (3.31a) *and* (3.31b)*, respectively. Damping is introduced via the substitution* $\omega_k^2 \mapsto (1 - i\epsilon)\omega_k^2$*, with* $\epsilon = 0.1$ *in this table. The considered problem has* 48,642 *dofs, and global GMRES is restarted after* 200 *iterations.*

	frequency range	N_ω	$A(P_1^{-1}(\mathbf{X})) = B$	$A(P_1^{-1}(P_2^{-1}(\mathbf{X}))) = B$
$C = 0$	$f_k \in [1,3]$Hz	5	220.2 (301 iter.)	93.8 (164 iter.)
	$f_k \in [1,3]$Hz	15	2296.9 (702 iter.)	203.2 (171 iter.)
	$f_k \in [6,9]$Hz	5	1356.5 (983 iter.)	22.2 (66 iter.)
	$f_k \in [6,9]$Hz	15	no convergence	53.8 (65 iter.)
$C \neq 0$	$f_k \in [1,3]$Hz	5	94.9 (203 iter.)	24.8 (72 iter.)
	$f_k \in [1,3]$Hz	15	502.1 (300 iter.)	66.0 (76 iter.)
	$f_k \in [6,9]$Hz	5	499.4 (566 iter.)	18.5 (54 iter.)
	$f_k \in [6,9]$Hz	15	1827.2 (627 iter.)	42.8 (53 iter.)

the present test case the number of iterations can be reduced by this strategy from 43 (at $n_p = 1$) to 14 (at $n_p = 4$).

Numerical Experiment 3.20 (The undamped ($\epsilon \equiv 0$) case) *In this experiment we study the quality of* $\tau^*(0)$ *as in* (3.25) *for the case when no viscous damping is present, cf. Remark* 3.10*. This choice is compared to choices found in different literature.*

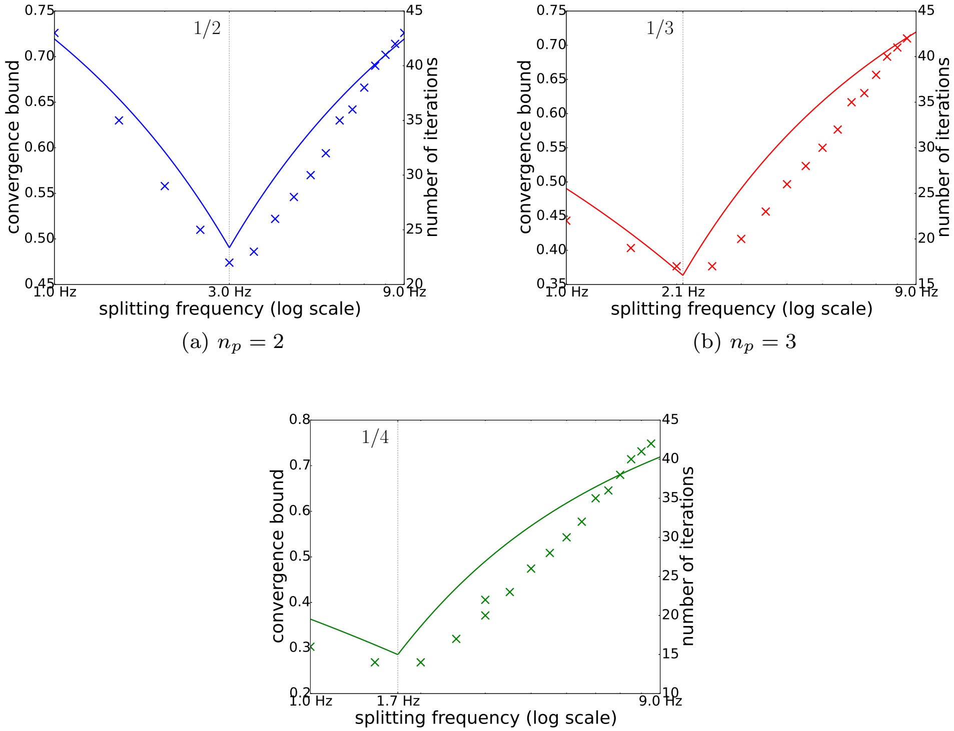
The optimality of τ^* in (3.16) is derived for positive damping parameter $\epsilon > 0$ because only then the circles that bound the preconditioned spectra do not touch the origin, i.e. $|c_k| > R_k$ in (3.13). The graph in Figure 3.2 (left), however, shows a smooth dependence of $\tau^*(\epsilon, s_{\min}, s_{\max})$ on ϵ and, in particular, yield an optimal value in the case of $\epsilon = 0$ stated in Remark 3.10. In Table 3.5 we compare the seed parameter $\tau^*(0)$ with two choices found in the literature: $\tau = (1 - 0.5i)\omega_{\max}$ in [147] and the hand optimized value $\tau = (0.7 - 0.3i)\omega_{\max}$ used in [10, 13]. Moreover, we use an alternative optimization criteria for minimizing the bound in Corollary 3.7 as a third comparison value. The results in Table 3.5 show that optimality in terms of GMRES iteration numbers is lost but, on the other hand, comparable results to the established choices in literature are obtained.

Table 3.5: *Multi-shift GMRES without restarting using different seed parameters* τ *and no damping* ($\epsilon = 0$)*. We consider* $N_\omega = 10$ *frequencies equally spaced within the interval* $\omega_k/2\pi = [5, 10]$*Hz. The problem size is* 48,642 *dofs.*

seed $\tau/\omega_{\max}$	$\tau^* = 0.66 - 0.26i$	$\tau = 0.7 - 0.3i$	$\tau = 1 - 0.5i$	$\min_\tau \mathrm{mean}_k(R/\lvert c_k\rvert)$
# iterations	226	201	295	300
CPU time [s]	139.5	109.3	247.6	257.6

Conclusions

We have derived an optimal seed parameter τ^* for the shift-and-invert preconditioner $\mathcal{P}(\tau^*) = (\mathcal{K} - \tau^* \mathcal{M})$ applied to a sequence of shifted systems. For a given set of



(a) $n_p = 2$

(b) $n_p = 3$

(c) $n_p = 4$

Figure 3.8: *Effect of interval splitting for* $n_p = \{2, 3, 4\}$ *when* $\mathcal{I} = 2\pi\ [1, 9]$ *Hz and* $\epsilon = 0.5$ *fixed. We investigate the splitting point for the first out of* n_p *subintervals. From left to right, the splitting strategy is applied 'recursively' to the* $n_p - 1$ *remaining subintervals.*

frequencies $s_k \in [s_{\min}, s_{\max}]$ the optimal seed is an explicit function of the extreme frequencies, and the viscous damping parameter ϵ, cf. (3.16) in Lemma 3.8. The optimality of the parameter is derived with respect to a well-known GMRES convergence bound that has been extended to the multi-shift setting, and in the presence of viscous damping, i.e. $\epsilon > 0$. Our numerical experiments, however, prove the usefulness even for the case without damping (Experiment 3.20). Comparisons with shift-and-invert preconditioners with parameter different from τ^* show a slower convergence behavior of multi-shift GMRES and, therefore, numerically prove optimality of τ^* (see Experiment 3.16 and Experiment 3.17).

The spectral analysis that has been carried out for the derivation of τ^* gives valuable insight that we exploit within two applications: In Section 3.4.1, a shifted Neumann preconditioner is derived that has minimum spectral radius. The numerical examples in Experiment 3.17 show that an increase of the degree of the Neumann polynomial leads to a significant reduction of GMRES iteration numbers and, hence, of memory requirements. Moreover, numerical tests have shown that the Neumann preconditioner based on τ^* yields fast convergence especially in the case of a small damping parameter (cf. Experiment 3.17). In Section 3.4.2, we apply global GMRES to a matrix equation reformulation of the shifted problem. Then, the bounding circles of the shifted spectra can be rotated which yields a more favorable spectrum for the matrix equation approach, cf. Experiment 3.18.

We have also considered the situation when more than one CPU is present, and a sequence of shifted problems within a fixed interval $[s_{\min}, s_{\max}]$ can be split into subintervals that are solved simultaneously on each available CPU. In Experiment 3.19 we give strong numerical evidence that an optimal interval splitting strategy is to split equidistantly on a logarithmic scale. With respect to future work, we would like to point out that multi-shift GMRES with a shift-and-invert preconditioner (as in Algorithm 3.1) fits the more general framework of rational Krylov methods [114]. The fact that we apply a single shift-and-invert preconditioner corresponds to the situation where a rational Krylov space with denominator degree equals to one is chosen. The recent RKFIT algorithm [17] provides a strategy for pole selection in rational Krylov methods and can, thus, be used for comparison. Moreover, the presented interval splitting strategy yields (optimal) seed parameters for each subinterval and can be exploited in the framework of multi-preconditioned GMRES for shifted systems [6].

Chapter 4

A Preconditioner for the Elastic Wave Equation Based on Multi-Level Sequentially Semiseparable Matrix Computations

Abstract. In this work we present a new numerical framework for the efficient solution of the time-harmonic elastic wave equation at multiple frequencies. We show that multiple frequencies (and multiple right-hand sides) can be incorporated when the discretized problem is written as a matrix equation. This matrix equation can be solved efficiently using the preconditioned IDR(s) method. We present an efficient and robust way to apply a single preconditioner using MSSS matrix computations. For 3D problems, we present a memory-efficient implementation that exploits the solution of a sequence of 2D problems. Realistic examples in two and three spatial dimensions demonstrate the performance of the new algorithm.

Introduction

The understanding of the earth subsurface is a key task in geophysics, and Full-Wave-

This chapter is based on the journal article:

M. Baumann, R. Astudillo, Y. Qiu, E.Y.M. Ang, M.B. van Gijzen, and R.-É. Plessix (2017). *An MSSS-Preconditioned Matrix Equation Approach for the Time-Harmonic Elastic Wave Equation at Multiple Frequencies.* Computational Geosciences, Springer, DOI: 10.1007/s10596-017-9667-7.

form Inversion (FWI) is a computational approach that matches the intensity of reflected shock waves (measurements) with simulation results in a least squares sense; cf. [153] and the references therein for an overview of state-of-the-art FWI algorithms. From a mathematical point of view, the problem of matching measurements with simulation results leads to a PDE-constrained optimization problem where the objective function is defined by the respective FWI approach, and the constraining partial differential equation (PDE) is the wave equation. Since the earth is an elastic medium, the elastic wave equation needs to be considered. In order to design an efficient optimization algorithm, a fast numerical solution of the elastic wave equation is required at every iteration of the optimization loop. This so-called *forward problem* is the focus of this work.

More recently, FWI has been considered for an equivalent problem formulated in the frequency-domain [94, 106]. The frequency-domain formulation of wave propagation has shown specific modeling advantages for both acoustic and elastic media. For the efficient FWI, notably the waveform tomography [105, 153], a fast numerical solution of the respective time-harmonic forward problem is required. More precisely, the forward problem requires the fast numerical solution of the discretized time-harmonic elastic wave equation at multiple wave frequencies and for multiple source terms. In this context, many efficient numerical solution methods have been proposed mostly for the (acoustic) Helmholtz equation [100, 102, 104, 112]. In this work, we present an efficient solver of the time-harmonic *elastic* wave equation that results from a finite element discretization, cf. [27, 46].

Especially for large 3D problems, the efficient numerical solution with respect to computation time and memory requirements is subject to current research. When an iterative Krylov method is considered, the design of efficient preconditioners for the elastic wave equation is required. In [2] a damped preconditioner for the elastic wave equation is presented. The authors of [113] analyze a multi-grid approach for the damped problem. Both works are extensions of the work of Erlangga et al. [112] for the acoustic case. The recent low-rank approach of the MUMPS solver [3] makes use of the hierarchical structure of the discrete problem and can be used as a preconditioner, cf. [4, 157]. When domain decomposition is considered, the sweeping preconditioner [141] is an attractive alternative.

In this work we propose a *hybrid* method that combines the iterative Induced Dimension Reduction (IDR) method with an efficient preconditioner that exploits the multilevel sequentially semiseparable (MSSS) matrix structure of the discretized elastic wave equation on a Cartesian grid. Moreover, we derive a matrix equation formulation that includes multiple frequencies and multiple right-hand sides, and present a version of IDR that solves linear matrix equations at a low memory requirement. The paper is structured as follows: In Section 4.1, we derive a finite element discretization for the time-harmonic elastic wave equation with a special emphasis on the case when multiple frequencies are present. Section 4.2 presents the IDR(s) method for the iterative solution of the resulting matrix equation. We discuss an efficient preconditioner in Section 4.3 based on the MSSS structure of the discrete problem. We present different versions of the MSSS preconditioner for 2D and 3D problems in Section 4.3.2 and 4.3.3, respectively. The paper concludes with extensive numerical tests in Section 4.4.

4.1 Finite Element Discretization of The Time-Harmonic Elastic Wave Equation at Multiple Frequencies

In this section we present a finite element discretization of the time-harmonic elastic wave equation with a special emphasis on the mathematical and numerical treatment when multiple frequencies (and possibly multiple right-hand sides) are present.

4.1.1 Problem Description

The time-harmonic elastic wave equation describes the displacement vector $\mathbf{u} : \Omega \to \mathbb{C}^d$ in a computational domain $\Omega \subset \mathbb{R}^d, d \in \{2, 3\}$, governed by the following partial differential equation (PDE),

$$-\omega_k^2 \rho(\mathbf{x})\mathbf{u}_k - \nabla \cdot \sigma(\mathbf{u}_k) = \mathbf{s}, \;\; \mathbf{x} \in \Omega \subset \mathbb{R}^d, \;\; k = 1, ..., N_\omega. \tag{4.1}$$

Here, $\rho(\mathbf{x})$ is the density of an elastic material in the considered domain Ω that can differ with $\mathbf{x} \in \Omega$ (inhomogeneity), $\mathbf{s}$ is a source term, and $\{\omega_1, ..., \omega_{N_\omega}\}$ are multiple angular frequencies that define N_ω problems in (4.1). The stress and strain tensor follow from Hooke's law for isotropic elastic media,

$$\sigma(\mathbf{u}_k) := \lambda(\mathbf{x}) \left(\nabla \cdot \mathbf{u}_k \; I_d\right) + 2\mu(\mathbf{x})\epsilon(\mathbf{u}_k), \tag{4.2}$$

$$\epsilon(\mathbf{u}_k) := \frac{1}{2}\left(\nabla \mathbf{u}_k + (\nabla \mathbf{u}_k)^{\mathsf{T}}\right), \tag{4.3}$$

with λ, μ being the Lamé parameters (4.6). On the boundary $\partial\Omega$ of the domain Ω, we consider the following boundary conditions,

$$i\omega_k \rho(\mathbf{x}) B \mathbf{u}_k + \sigma(\mathbf{u}_k)\hat{\mathbf{n}} = 0, \quad \mathbf{x} \in \partial\Omega_a, \tag{4.4}$$

$$\sigma(\mathbf{u}_k)\hat{\mathbf{n}} = 0, \quad \mathbf{x} \in \partial\Omega_r, \tag{4.5}$$

where Sommerfeld radiation boundary conditions at $\partial\Omega_a$ model absorption, and we typically prescribe a free-surface boundary condition in the north of the computational domain $\partial\Omega_r$, with $\partial\Omega_a \cup\!\!\!\cdot\; \partial\Omega_r = \partial\Omega$. In (4.4), B is a $d \times d$ matrix that depends on c_p and c_s, $B \equiv B(\mathbf{x}) := c_p(\mathbf{x})\hat{\mathbf{n}}\hat{\mathbf{n}}^{\mathsf{T}} + c_s(\mathbf{x})\hat{\mathbf{t}}\hat{\mathbf{t}}^{\mathsf{T}} + c_s(\mathbf{x})\hat{\mathbf{s}}\hat{\mathbf{s}}^{\mathsf{T}}$, with vectors $\{\hat{\mathbf{n}}, \hat{\mathbf{t}}, \hat{\mathbf{s}}\}$ being normal or tangential to the boundary, respectively; cf. [2] for more details. Note that the boundary conditions (4.4)-(4.5) can naturally be included in a finite element approach as explained in Section 4.1.2.

We assume the set of five parameters $\{\rho, c_p, c_s, \lambda, \mu\}$ in (4.1)-(4.5) to be space-dependent. The Lamé parameters λ and μ are directly related to the material density ρ and the speed of P-waves c_p and speed of S-waves c_s via,

$$\mu = c_s^2 \rho, \quad \lambda = \rho(c_p^2 - 2c_s^2). \tag{4.6}$$

All parameter functions are assumed in $L^1(\Omega)$. More specifically, we interpolate data points using Q_1 basis functions.

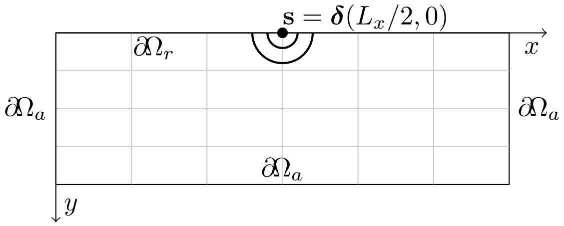


Figure 4.1: *Boundary conditions and source term for $d = 2$. For $d = 3$, the source is for instance located at $(L_x/2, L_y/2, 0)^\mathsf{T}$.*

4.1.2 Spline-Based Finite Element Discretization

For the discretization of (4.1)-(4.5) we follow a classical finite element approach using the following ansatz,

$$\mathbf{u}_k(\mathbf{x}) \approx \sum_{i=1}^{N} u_k^i \boldsymbol{\varphi}_i(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad u_k^i \in \mathbb{C}. \tag{4.7}$$

In the numerical examples presented in Section 4.4 we restrict ourselves to Cartesian grids and basis functions $\boldsymbol{\varphi}_i$ that are B-splines of degree p as described for instance in [24, Chapter 2]. The number of degrees of freedom is, hence, given by

$$N = d \prod_{i \in \{x,y,z\}} (n_i - 1 + p), \quad d \in \{2, 3\}, p \in \mathbb{N}^+, \tag{4.8}$$

with n_i grid points in the respective spatial direction (in Figure 4.1 we illustrate the case where $d = 2$ and $n_x = 7, n_y = 5$).

Definition 4.1 (Tensor notation II, [36]) *The dot product between two vector-valued quantities $\mathbf{u} = (u_x, u_y), \mathbf{v} = (v_x, v_y)$ is denoted as, $\mathbf{u} \cdot \mathbf{v} := u_x v_x + u_y v_y$. Similarly, we define the componentwise multiplication of two matrices $U = [u_{ij}], V = [v_{ij}]$ as, $U : V := \sum_{i,j} u_{ij} v_{ij}$.*

A Galerkin finite element approach applied to (4.1) yields the following weak form: Find $\boldsymbol{\varphi}_i \in [H^1(\Omega)]^d$ such that,

$$\begin{aligned} -\omega_k^2 \sum_{i=1}^{N} u_k^i \int_\Omega \rho(\mathbf{x}) \boldsymbol{\varphi}_i \cdot \boldsymbol{\varphi}_j \; d\Omega - \sum_{i=1}^{N} u_k^i \int_\Omega \nabla \cdot \sigma(\boldsymbol{\varphi}_i) \cdot \boldsymbol{\varphi}_j \; d\Omega \\ = \int_\Omega \mathbf{s} \cdot \boldsymbol{\varphi}_j \; d\Omega, \quad \text{for all } \boldsymbol{\varphi}_j \in [H^1(\Omega)]^d, \end{aligned}$$

$j = 1, ..., N$, and for all source functions $\mathbf{s} \in [L^1(\Omega)]^d$. We exploit the boundary conditions (4.4)-(4.5) in the following way,

$$\begin{aligned} &\int_\Omega \nabla \cdot \sigma(\boldsymbol{\varphi}_i) \cdot \boldsymbol{\varphi}_j \; d\Omega \\ &= \int_{\partial\Omega} \sigma(\boldsymbol{\varphi}_i) \boldsymbol{\varphi}_j \cdot \hat{\mathbf{n}} \; d\Gamma - \int_\Omega \sigma(\boldsymbol{\varphi}_i) : \nabla \boldsymbol{\varphi}_j \; d\Omega \\ &= -i\omega_k \int_{\partial\Omega_a} \rho(\mathbf{x}) B \boldsymbol{\varphi}_i \cdot \boldsymbol{\varphi}_j \; d\Gamma - \int_\Omega \sigma(\boldsymbol{\varphi}_i) : \nabla \boldsymbol{\varphi}_j \; d\Omega \end{aligned}$$

Note that the stress-free boundary condition (4.5) can be included naturally in a finite element discretization by excluding $\partial\Omega_r$ from the above boundary integral.

We summarize the finite element discretization of the time-harmonic, inhomogeneous elastic wave equation at multiple frequencies ω_k by,

$$(K + i\omega_k C - \omega_k^2 M)\mathbf{x}_k = \mathbf{b}, \quad k = 1, ..., N_\omega, \tag{4.9}$$

with unknown vectors $\mathbf{x}_k := [u_k^1, ..., u_k^N]^\mathsf{T} \in \mathbb{C}^N$ consisting of the coefficients in (4.7), and mass matrix M, stiffness matrix K and boundary matrix C given by,

$$[K]_{ij} = \int_\Omega \sigma(\boldsymbol{\varphi}_i) : \nabla\boldsymbol{\varphi}_j \; d\Omega, \quad [M]_{ij} = \int_\Omega \rho\boldsymbol{\varphi}_i \cdot \boldsymbol{\varphi}_j \; d\Omega, \tag{4.10}$$

$$[C]_{ij} = \int_{\partial\Omega_a} \rho B\boldsymbol{\varphi}_i \cdot \boldsymbol{\varphi}_j \; d\Gamma, \qquad [\mathbf{b}]_j = \int_\Omega \mathbf{s} \cdot \boldsymbol{\varphi}_j \; d\Omega. \tag{4.11}$$

In a 2D problem (see Figure 4.1), the unknown $\mathbf{x}_k$ contains the x-components and the y-components of the displacement vector. When lexicographic numbering is used, the matrices in (4.9) have the block structure

$$K = \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix}, \quad C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix}, \quad M = \begin{bmatrix} M_{xx} & M_{xy} \\ M_{yx} & M_{yy} \end{bmatrix},$$

as shown in Figure 4.3 (left) for $d = 2$, and Figure 4.2 (top left) for $d = 3$. When solving (4.9) with an iterative Krylov method, it is necessary to apply a preconditioner. Throughout this document, we consider a preconditioner of the form

$$\mathcal{P}(\tau) = (K + i\tau C - \tau^2 M), \tag{4.12}$$

where τ is a single *seed frequency* that needs to be chosen with care for the range of frequencies $\{\omega_1, ...\omega_{N_\omega}\}$, cf. the considerations in [8, 119]. The efficient application of the preconditioner (4.12) for problems of dimension $d = 2$ and $d = 3$ on a structured domain is presented in Section 4, and the choice of τ is discussed in Section 4.4.2.

4.1.3 Reformulation as a Matrix Equation

We next describe a new approach to solve (4.9) at multiple frequencies. Therefore, we define the block matrix $\mathbf{X}$ consisting of all unknown vectors, $\mathbf{X} := [\mathbf{x}_1, ..., \mathbf{x}_{N_\omega}] \in \mathbb{C}^{N \times N_\omega}$, and note that (4.9) can be rewritten as,

$$\mathcal{A}(\mathbf{X}) := K\mathbf{X} + iC\mathbf{X}\Sigma - M\mathbf{X}\Sigma^2 = B, \tag{4.13}$$

where $\Sigma := \text{diag}(\omega_1, ..., \omega_{N_\omega})$, and with block right-hand side $B := [\mathbf{b}, ..., \mathbf{b}]$. In (4.13), we also define the linear operator $\mathcal{A}(\cdot)$ which defines the matrix equation (4.13) in short-hand notation as $\mathcal{A}(\mathbf{X}) = B$. This reformulation gives rise to use an extension of the IDR(s) method to solve linear matrix equations [5].

Note that an alternative approach to efficiently solve (4.9) at multiple frequencies ($N_\omega > 1$) leads to the solution of shifted linear systems as presented in [8, Section 4.2] and the references therein. The memory-efficient approach followed by [8] relies on the shift-invariance property of the Krylov spaces belonging to different frequencies.

Some restrictions of this approach like collinear right-hand sides in (4.9) and the difficulty of preconditioner design are, however, not present in the matrix equation setting (4.13).

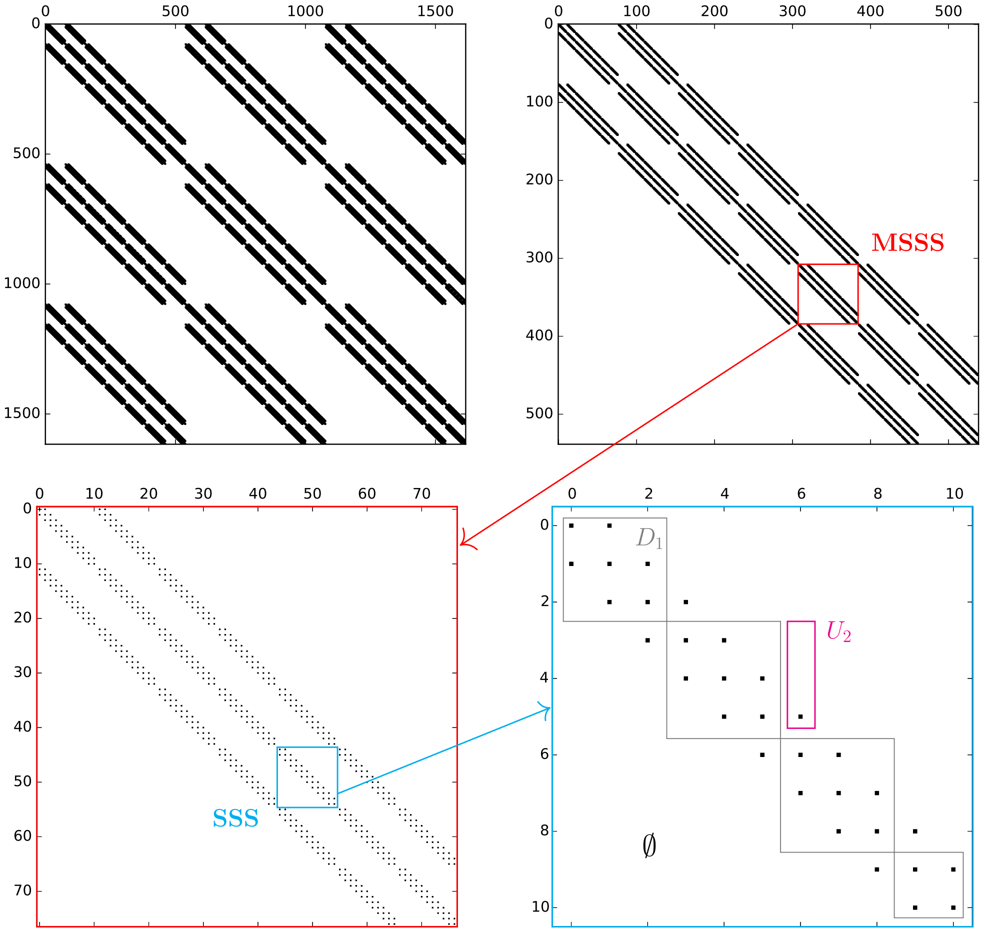


Figure 4.2: *A* `spy` *plot of* (4.12) *for a 3D elastic problem when linear basis functions* ($p = 1$) *are used: In the top row we show the discretized problem for lexicographic (top left) and nodal-based ordering (top right). Appropriate zooming demonstrates the hierarchically repeating structure of the matrix on level 2 (bottom left) and level 1 (bottom right). For level 1, we indicate the SSS data structure used in Section 4.3.1. Here, the rank of* U_2 *equals one.*

4.2 The Induced Dimension Reduction (IDR) Method

Krylov subspace methods are an efficient tool for the iterative numerical solution of large-scale linear systems of equations [81]. In particular, the matrices K, C, M that typically are obtained from a discretization of the time-harmonic elastic wave equation (4.9) are ill-conditioned and have very large dimensions, especially when high frequencies are considered. For these reasons, the numerical solution is computationally challenging, and factors like memory consumption and computational efficiency have to be taken into account when selecting a suitable Krylov method.

The Generalized Minimum Residual (GMRES) method [118] is one of the most widely-used Krylov method because of its rather simple implementation and optimal convergence property. Nevertheless, GMRES is a *long-recurrence* Krylov method, i.e., its requirements for memory and computation grow in each iteration which is unfeasible when solving linear systems arising from the elastic wave equation. On the other hand, short-recurrence Krylov methods keep the computational cost constant per iteration; one of the most used method of this class is the Bi-conjugate gradient stabilized (Bi-CGSTAB) method [144].

In this work we propose to apply an alternative short-recurrence Krylov method: the Induced Dimension Reduction (IDR) method [130, 146]. IDR(s) uses recursions of depth $s+1$, with $s \in \mathbb{N}^+$ being typically small, to solve linear systems of equations of the form,

$$A\mathbf{x} = \mathbf{b}, \quad A \in \mathbb{C}^{N\times N}, \quad \{\mathbf{x}, \mathbf{b}\} \in \mathbb{C}^N, \tag{4.14}$$

where the coefficient matrix A is a large, sparse, and in general non-Hermitian. We mention some important numerical properties of the IDR(s) method: First, finite termination of the algorithm is ensured with IDR(s) computing the exact solution in $N + \frac{N}{s}$ iterations in exact arithmetics. Second, Bi-CGSTAB and IDR(1) are mathematically equivalent [129]. Third, IDR(s) with $s > 1$ often outperforms Bi-CGSTAB for numerically difficult problems, for example, for convection-diffusion-reaction problems where the convection term is dominating, or problems with a large negative reaction term, cf. [130] and [146], respectively.

4.2.1 IDR(s) for Linear Systems of Equations

We present a brief introduction of the IDR(s) method that closely follows [130]. In Section 4.2.2, we explain how to use IDR(s) for solving (4.13) for multiple frequencies in a matrix equation setting. We introduce the basic concepts of the IDR(s) method. The IDR(s) algorithm is based on the following theorem.

Theorem 4.2 (The IDR(s) theorem) *Let A be a matrix in $\mathbb{C}^{N\times N}$, let $\mathbf{v}_0$ be any non-zero vector in $\mathbb{C}^N$, and let $\mathcal{G}_0$ be the full Krylov subspace, $\mathcal{G}_0 := \mathcal{K}_N(A, \mathbf{v}_0)$. Let $\mathcal{S}$ be a (proper) subspace of $\mathbb{C}^N$ such that $\mathcal{S}$ and $\mathcal{G}_0$ do not share a nontrivial invariant subspace of A, and define the sequence:*

$$\mathcal{G}_j := (I - \xi_j A)(\mathcal{G}_{j-1} \cap \mathcal{S}), \quad j = 1, 2, \ldots, \tag{4.15}$$

where ξ_j are nonzero scalars. Then it holds:

1. $\mathcal{G}_{j+1} \subset \mathcal{G}_j$, *for* $j \geq 0$, *and,*
2. *dim*$(\mathcal{G}_{j+1}) <$ *dim*$(\mathcal{G}_j)$, *unless* $\mathcal{G}_j \equiv \{0\}$.

Proof 4.3 *Can be found in [130].* □

Exploiting the fact that the subspaces $\mathcal{G}_j$ are shrinking and $\mathcal{G}_j = \{0\}$ for some j, IDR(s) solves the problem (4.14) by constructing residuals $\mathbf{r}_{k+1}$ in the subspaces $\mathcal{G}_{j+1}$, while in parallel, it extracts the approximate solutions $\mathbf{x}_{k+1}$. In order to illustrate how to create a residual vector in the space $\mathcal{G}_{j+1}$, let us assume that the space $\mathcal{S}$ is the left null space of a full rank matrix $P := [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_s]$, $\{\mathbf{x}_i\}_{i=k-(s+1)}^{k}$ are $s+1$ approximations to (4.14) and their corresponding residual vectors $\{\mathbf{r}_i\}_{i=k-(s+1)}^{k}$ are in $\mathcal{G}_j$. IDR(s) creates a residual vector $\mathbf{r}_{k+1}$ in $\mathcal{G}_{j+1}$ and obtains the approximation $\mathbf{x}_{k+1}$ using the following $(s+1)$-term recursions,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \xi_{j+1}\mathbf{v}_k + \sum_{j=1}^{s} \gamma_j \Delta\mathbf{x}_{k-j},$$

$$\mathbf{r}_{k+1} = (I - \xi_{j+1}A)\mathbf{v}_k, \quad \mathbf{v}_k = \mathbf{r}_k - \sum_{j=1}^{s} \gamma_j \Delta\mathbf{r}_{k-j},$$

where $\Delta\mathbf{y}_k$ is the forward difference operator $\Delta\mathbf{y}_k := \mathbf{y}_{k+1} - \mathbf{y}_k$. The vector $\mathbf{c} = (\gamma_1, \gamma_2, \ldots, \gamma_s)^{\mathsf{T}}$ can be obtained imposing the condition $\mathbf{r}_{k+1} \in \mathcal{G}_{j+1}$ by solving the $s \times s$ linear system,

$$P^{\mathsf{H}}[\Delta\mathbf{r}_{k-1}, \Delta\mathbf{r}_{k-2}, \ldots, \Delta\mathbf{r}_{k-s}]\mathbf{c} = P^{\mathsf{H}}\mathbf{r}_k.$$

At this point, IDR(s) has created a new residual vector $\mathbf{r}_{k+1}$ in $\mathcal{G}_{j+1}$. However, using the fact that $\mathcal{G}_{j+1} \subset \mathcal{G}_j$, $\mathbf{r}_{k+1}$ is also in $\mathcal{G}_j$, IDR(s) repeats the above computation in order to create $\{\mathbf{r}_{k+1}, \mathbf{r}_{k+2}, \ldots, \mathbf{r}_{k+s+1}\}$ in $\mathcal{G}_{j+1}$. Once $s+1$ residuals are in $\mathcal{G}_{j+1}$, IDR(s) is able to sequentially create new residuals in $\mathcal{G}_{j+2}$.

4.2.2 Preconditioned IDR(s) for Linear Matrix Equations

The IDR(s) theorem 4.2 can be generalized to solve linear problems in any finite-dimensional vector space. In particular, IDR(s) has recently been adapted to solve linear matrix equations [5]. In this work, we use this generalization of the IDR(s) method to solve the time-harmonic elastic wave equation at multiple frequencies. Using the definition of the linear operator $\mathcal{A}(\cdot)$ in (4.13) yields a matrix equation in short-hand notation, $\mathcal{A}(\mathbf{X}) = B$, which is close to (4.14). Here, the block right-hand side B equals

$$B := \mathbf{b}[1, 1, \ldots, 1]_{N_\omega} \quad \text{or} \quad B := [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{N_\omega}]$$

depending whether we consider a constant source term for each frequency as in (4.1) or allow variations.

IDR(s) for solving (4.13) uses the same recursions described in Section 4.2.1 acting on block matrices. The main differences with the original IDR(s) algorithm of [130] are the substitution of the matrix-vector product $A\mathbf{x}$ by the application of the linear operator $\mathcal{A}(\mathbf{X})$, and the use of Frobenius inner products, see Definition 4.4. Note that two prominent long-recurrence Krylov methods have been generalized to the solution of linear matrix equations in [68] using a similar approach. In Algorithm 4.1, we present IDR(s) for solving the matrix equation (4.13) with biorthogonal residuals (see details in [5, 146]). The preconditioner used in Algorithm 4.1 is described in the following Section.

Definition 4.4 (Frobenius inner product, [68]) *The Frobenius inner product of two real matrices* A, B *of the same size is defined as* $\langle A, B\rangle_F := tr(A^{\mathsf{H}}B)$, *where* $tr(\cdot)$ *is the trace of the matrix* $A^{\mathsf{H}}B$. *The matrix Frobenius norm is,* $\|A\|_F^2 := \langle A, A\rangle_F$.

Algorithm 4.1 Preconditioned IDR(s) for matrix equations [5]

```
Input: $\mathcal{A}$ as defined in (4.13), $B \in \mathbb{C}^{N\times N_\omega}$, $tol \in (0,1)$, $s \in \mathbb{N}^+$, $P \in \mathbb{C}^{N\times(s\times N_\omega)}$,
    $\mathbf{X}_0 \in \mathbb{C}^{N\times N_\omega}$, preconditioner $\mathcal{P}$
Output: $\mathbf{X}$ such that $\|B - \mathcal{A}(\mathbf{X})\|_F/\|B\|_F \leq tol$
$G = 0 \in \mathbb{C}^{N\times s\times N_\omega}$, $U = 0 \in \mathbb{C}^{N\times s\times N_\omega}$
$M = I_s \in \mathbb{C}^{s\times s}$, $\xi = 1$
$R = B - \mathcal{A}(\mathbf{X}_0)$                                ▷ Apply operator (4.13)
while $\|R\|_F \leq tol \cdot \|B\|_F$ do
    Compute $[\mathbf{f}]_i = \langle P_i, R\rangle_F$ for $i = 1, \ldots, s$
    for $k = 1$ to $s$ do
        Solve $\mathbf{c}$ from $M\mathbf{c} = \mathbf{f}$, $(\gamma_1, \ldots, \gamma_s)^{\mathsf{T}} = \mathbf{c}$
        $V = R - \sum_{i=k}^{s} \gamma_i G_i$
        $V = \mathcal{P}^{-1}(V)$                    ▷ Apply preconditioner, see Section 4.3
        $U_k = U\mathbf{c} + \xi V$
        $G_k = \mathcal{A}(U_k)$                            ▷ Apply operator (4.13)
        for $i = 1$ to $k-1$ do
            $\alpha = \langle P_i, G_k\rangle_F/[M]_{i,i}$
            $G_k = G_k - \alpha G_i$
            $U_k = U_k - \alpha U_i$
        end for
        $[M]_{ik} = \langle P_i, G_k\rangle_F$
        $\beta = [\mathbf{f}]_k/[M]_{k,k}$
        $R = R - \beta G_k$
        $\mathbf{X} = \mathbf{X} + \beta U_k$
        if $k + 1 \leq s$ then
            $[\mathbf{f}]_i = 0$ for $i = 1, \ldots, k$
            $[\mathbf{f}]_i = [\mathbf{f}]_i - \beta[M]_{i,k}$ for $i = k+1, \ldots, s$
        end if
        Overwrite $k$-th block of $G$ and $U$ by $G_k$ and $U_k$
    end for
    $V = \mathcal{P}^{-1}(R)$                        ▷ Apply preconditioner, see Section 4.3
    $T = \mathcal{A}(V)$                                    ▷ Apply operator (4.13)
    $\xi = \langle T, R\rangle_F/\langle T, T\rangle_F$
    $\rho = \langle T, R\rangle_F/(\|T\|_F\|R\|_F)$
    if $|\rho| < \rho_0$ then                       ▷ $\rho_0 = 0.7$ is recommended in [128]
        $\xi = \rho_0 \times \xi/|\rho|$
    end if
    $R = R - \xi T$
    $\mathbf{X} = \mathbf{X} + \xi V$
end while
return $\mathbf{X} \in \mathbb{C}^{N\times N_\omega}$
```

4.3 An MSSS-Based Preconditioner for the Elastic Wave Equation

Semiseparable matrices [150] and the more general concept of sequentially semiseparable (SSS) matrices [19, 20] are structured matrices represented by a set of generators. Matrices that arise from the discretization of 1D partial differential equations typically have an SSS structure [111], and submatrices taken from the strictly lower/upper-triangular part yield generators of low rank. Multiple applications from different areas can be found [30, 70, 110] that exploit this structure. Multilevel sequentially semiseparable (MSSS) matrices generalize SSS matrices to the case when $d > 1$. Again, discretizations of higher-dimensional PDEs give rise to matrices that have an MSSS structure [107], and the multilevel paradigm yields a hierarchical matrix structure with MSSS generators that are themselves MSSS of a lower hierarchical level. This way, at the lowest level, generators are SSS matrices. The advantages of Cartesian grids in higher dimensions and the resulting structure of the corresponding discretization matrices depicted in Figure 4.2 is directly exploited in MSSS matrix computations. For unstructured meshes we refer to [161] where hierarchically semiseparable (HSS) matrices are used. MSSS preconditioning techniques were first studied for PDE-constrained optimization problems in [107] and later extended to computational fluid dynamics problems [108]. In this work, we apply MSSS matrix computations to precondition the time-harmonic elastic wave equation. Appropriate splitting of the 3D elastic operator leads to a sequence of 2D problems in level-2 MSSS structure. An efficient preconditioner for 2D problems is based on model order reduction of level-1 SSS matrices.

4.3.1 Definitions and Basic SSS Operations

We present the formal definition of an SSS matrix used on 1D level in Definition 4.5.

Definition 4.5 (SSS matrix structure, [20]) *Let A be an $n \times n$ block matrix in SSS structure such that A can be written in the following block-partitioned form,*

$$A_{ij} = \begin{cases} U_i W_{i+1} \cdots W_{j-1} V_j^{\mathsf{H}}, & \text{if} \quad i < j, \\ D_i, & \text{if} \quad i = j, \\ P_i R_{i-1} \cdots R_{j+1} Q_j^{\mathsf{H}}, & \text{if} \quad i > j. \end{cases} \tag{4.16}$$

Here, the superscript 'H' denotes the conjugate transpose of a matrix. The matrices $\{U_s, W_s, V_s, D_s, P_s, R_s, Q_s\}_{s=1}^{n}$ are called generators of the SSS matrix A, with their respective dimensions given in Table 4.1. As a short-hand notation for (4.16), we use $A = SSS(P_s, R_s, Q_s, D_s, U_s, W_s, V_s)$.

The special case of an SSS matrix when $n = 4$ is presented in the appendix.

Table 4.1: *Generators sizes for the SSS matrix A in Definition 4.5. Note that, for instance, $m_1 + ... + m_n$ equals the dimension of A.*

U_i	W_i	V_i	D_i	P_i	R_i	Q_i
$m_i \times k_i$	$k_{i-1} \times k_i$	$m_i \times k_{i-1}$	$m_i \times m_i$	$m_i \times l_i$	$l_{i-1} \times l_i$	$m_i \times l_{i+1}$

In general, every matrix can be represented in SSS format. In Figure 4.2 (bottom right) we show that the 1D level of the elastic operator is tridiagonal if $p = 1$. Therefore, diagonal blocks D_i are copies of the 1D operator, and off-diagonal blocks can, for instance, be represented by the product of rank-p matrices, $U_2 V_3^{\mathsf{H}}$, where the last element of U_2 is identical to the respective entry of the 1D operator and V_3 is the first unit vector. Basic operations such as addition, multiplication and inversion are closed under SSS structure and can be performed in linear computational complexity if k_i and l_i in Table 4.1 are bounded by a constant. The rank of the off-diagonal blocks, formally defined as the *semiseparable order* in Definition 4.6, plays an important role in the computational complexity analysis of SSS matrix computations.

Definition 4.6 (Semiseparable order, [35]) *Let A be an $n \times n$ block matrix in SSS structure satisfying Definition 4.5. We use a colon-style notation: $A(i\!:\!j, k\!:\!\ell)$ selects rows of blocks from i to j and columns of blocks from k to ℓ of the SSS matrix A, i.e. $A(2\!:\!2, 3\!:\!3) = U_2 V_3^{\mathsf{H}}$. Let*

$$\text{rank } A(s+1\!:\!n, 1\!:\!s) =: l_s, \;\; s = 1, 2, \ldots, n-1, \;\; \text{and let further,}$$

$$\text{rank } A(1\!:\!s, s+1\!:\!n) =: u_s, \;\; s = 1, 2, \ldots, n-1.$$

Setting $r^l := \max\{l_s\}$ and $r^u := \max\{u_s\}$, we call r^l the lower semiseparable order and r^u the upper semiseparable order of A, respectively.

If the upper and lower semiseparable order are bounded by say r^*, i.e., $\{r^l,\ r^u\} \leq r^*$, then the computational cost for the SSS matrix computations is of $\mathcal{O}((r^*)^3 n)$ complexity [20], where n is the number of blocks of the SSS matrix as introduced in Definition 4.5. We will refer to r^* as the maximum off-diagonal rank. Matrix-matrix operations are closed under SSS structure, but performing SSS matrix computations will increase the semiseparable order, cf. [20]. We use model order reduction in the sense of Definition 4.8 in order to bound the semiseparable order.

Using the aforementioned definition of semiseparable order, we next introduce the following lemma to compute the (exact) LU factorization of an SSS matrix.

Lemma 4.7 (*LU* factorization of an SSS matrix) *Let $A = SSS(P_s, R_s, Q_s, D_s, U_s, W_s, V_s)$ be given in generator form with semiseparable order $(r^l,\ r^u)$. Then the factors of an LU factorization of A are given by the following generators representation,*

$$L = SSS(P_s, R_s, \hat{Q}_s, D_s^L,\ 0,\ 0,\ 0),$$
$$U = SSS(0,\ 0,\ 0,\ D_s^U, \hat{U}_s, W_s, V_s).$$

The generators of L and U are computed by Algorithm 4.2. Moreover, L has semiseparable order $(r^l, 0)$, and U has semiseparable order $(0, r^u)$.

Definition 4.8 (Model order reduction of an SSS matrix) *Let $A = SSS(P_s, R_s, Q_s, D_s, U_s, W_s, V_s)$ be an SSS matrix with lower order numbers l_s and upper order numbers u_s. The SSS matrix $\tilde{A} = SSS(\tilde{P}_s, \tilde{R}_s, \tilde{Q}_s, D_s, \tilde{U}_s, \tilde{W}_s, \tilde{V}_s)$ is called a reduced order approximation of A, if $\|A - \tilde{A}\|_2$ is small, and for the lower and upper order numbers it holds, $\tilde{l_s} < l_s, \tilde{u_s} < u_s$ for all $1 \leq s \leq n-1$.*

Algorithm 4.2 LU factorization and inversion of an SSS matrix A [19, 150]

```
Input: $A = SSS(P_s, R_s, Q_s, D_s, U_s, W_s, V_s)$ in generator form
// Perform LU factorization
$D_1 =: D_1^L D_1^U$                                   ▷ $LU$ factorization on generator level
Let $\hat{U}_1 := (D_1^L)^{-1} U_1$, and $\hat{Q}_1 := (D_1^L)^{-\mathsf{H}} Q_1$
for $i = 2 : n-1$ do
    if $i = 2$ then
        $M_i := \hat{Q}_{i-1}^{\mathsf{H}} \hat{U}_{i-1}$
    else
        $M_i := \hat{Q}_{i-1}^{\mathsf{H}} \hat{U}_{i-1} + R_{i-1} M_{i-1} W_{i-1}$
    end if
    $\left(D_i - P_i M_i V_i^{\mathsf{H}}\right) =: D_i^L D_i^U$          ▷ $LU$ factorization of generators
    Let $\hat{U}_i := (D_i^L)^{-1} (U_i - P_i M_i W_i)$, and
    let $\hat{Q}_i := (D_i^U)^{-\mathsf{H}} (Q_i - V_i M_i^{\mathsf{H}} R_i^{\mathsf{H}})$
end for
$M_n := \hat{Q}_{n-1}^{\mathsf{H}} \hat{U}_{n-1} + R_{n-1} M_{n-1} W_{n-1}$
$\left(D_n - P_n M_n V_n^{\mathsf{H}}\right) =: D_n^L D_n^U$              ▷ $LU$ factorization of generators
// Perform inversion
$L := SSS(P_s, R_s, \hat{Q}_s, D_s^L, 0, 0, 0)$
$U := SSS(0, 0, 0, D_s^U, \hat{U}_s, W_s, V_s)$
$A^{-1} = U^{-1} L^{-1}$        ▷ SSS inversion & matrix-matrix multiplication (see App. B)
```

4.3.2 Approximate Block-*LU* Decomposition Using MSSS Computations for 2D Problems

Similar to Definition 4.5 for SSS matrices, the generators representation for MSSS matrices (level-k SSS matrices) is given in Definition 4.9.

Definition 4.9 (MSSS matrix structure, [107]) *The matrix A is said to be a level-k SSS matrix if it has a form like (4.16) and all its generators are level-$(k-1)$ SSS matrices. The level-1 SSS matrix is the SSS matrix that satisfies Definition 4.5 We call A to be in MSSS matrix structure if $k > 1$.*

Most operations for SSS matrices can directly be extended to MSSS matrix computations. In order to perform a matrix-matrix multiplication of two MSSS matrices in linear computational complexity, model order reduction which is studied in [20, 107, 108] is necessary to keep the computational complexity low. The preconditioner (4.12) for a 2D elastic problem is of level-2 MSSS structure. We present a block-*LU* factorization of a level-2 MSSS matrix in this Section. Therefore, model order reduction is necessary which results in an *approximate* block-*LU* factorization. This approximate factorization can be used as a preconditioner for IDR(s) in Algorithm 4.1. On a two-dimensional Cartesian grid, the preconditioner (4.12) has a 2×2 block structure as presented in Figure 4.3 (left).

Definition 4.10 (Permutation of an MSSS matrix, [107]) *Let $\mathcal{P}(\tau)$ be a 2×2 level-2 MSSS block matrix arising from the FEM discretization of* (4.12) *using linear*

B-splines $(p = 1)$*,*

$$\mathcal{P}(\tau) = \begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} \\ \mathcal{P}_{21} & \mathcal{P}_{22} \end{bmatrix} \in \mathbb{C}^{2n_xn_y \times 2n_xn_y}, \tag{4.17}$$

with block entries being level-2 MSSS matrices in generator form,

$$\mathcal{P}_{11} = MSSS(P_s^{11}, R_s^{11}, Q_s^{11}, D_s^{11}, U_s^{11}, W_s^{11}, V_s^{11}), \tag{4.18a}$$
$$\mathcal{P}_{12} = MSSS(P_s^{12}, R_s^{12}, Q_s^{12}, D_s^{12}, U_s^{12}, W_s^{12}, V_s^{12}), \tag{4.18b}$$
$$\mathcal{P}_{21} = MSSS(P_s^{21}, R_s^{21}, Q_s^{21}, D_s^{21}, U_s^{21}, W_s^{21}, V_s^{21}), \tag{4.18c}$$
$$\mathcal{P}_{22} = MSSS(P_s^{22}, R_s^{22}, Q_s^{22}, D_s^{22}, U_s^{22}, W_s^{22}, V_s^{22}), \tag{4.18d}$$

where $1 \le s \le n_x$*. Note that all generators in* (4.18a)-(4.18d) *are SSS matrices of (fixed) dimension* n_y*. Let* $\{m_s\}_{s=1}^{n}$ *be the dimensions of the diagonal generators of such an SSS matrix, cf. Table 4.1, with* $\sum_{s=1}^{n} m_s = n_y$*. Then there exists a permutation matrix* Ψ*,* $\Psi\Psi^{\mathsf{T}} = \Psi^{\mathsf{T}}\Psi = I$*, given by*

$$\Psi = \begin{bmatrix} I_{n_x} \otimes \begin{bmatrix} \Psi_{1D} \\ 0 \end{bmatrix} & I_{n_x} \otimes \begin{bmatrix} 0 \\ \Psi_{1D} \end{bmatrix} \end{bmatrix}, \tag{4.19}$$

where

$$\Psi_{1D} := \begin{bmatrix} blkdiag\left(\begin{bmatrix} I_{m_s} \\ 0 \end{bmatrix}\right)_{s=1}^{n} & blkdiag\left(\begin{bmatrix} 0 \\ I_{m_s} \end{bmatrix}\right)_{s=1}^{n} \end{bmatrix},$$

such that $\mathcal{P}_{2D}(\tau) = \Psi^{\mathsf{T}}\mathcal{P}(\tau)\Psi$ *is of global MSSS level-2 structure.*

We illustrate the effect of the permutation matrix Ψ in Figure 4.3. For a matrix (4.12) that results from a discretization of the 2D time-harmonic elastic wave equation, P_{2D} is of block tri-diagonal MSSS structure.

Corollary 4.11 (Block tri-diagonal permutation) *Consider in Definition 4.10 the special case that the block entries in* (4.17) *are given as,*

$$\mathcal{P}_{11} = MSSS(P_s^{11}, 0, \underline{I}, D_s^{11}, U_s^{11}, 0, \underline{I}), \tag{14.20a}$$
$$\mathcal{P}_{12} = MSSS(P_s^{12}, 0, \underline{I}, D_s^{12}, U_s^{12}, 0, \underline{I}), \tag{14.20b}$$
$$\mathcal{P}_{21} = MSSS(P_s^{21}, 0, \underline{I}, D_s^{21}, U_s^{21}, 0, \underline{I}), \tag{14.20c}$$
$$\mathcal{P}_{22} = MSSS(P_s^{22}, 0, \underline{I}, D_s^{22}, U_s^{22}, 0, \underline{I}), \tag{14.20d}$$

with rectangular matrix $\underline{I} = [I, 0]$*. Then the matrix* $\Psi^{\mathsf{T}}\mathcal{P}(\tau)\Psi$ *is of block tri-diagonal MSSS structure.*

Proof 4.12 *This result follows from formula (2.13) of Lemma 2.4 in the original proof [107] when generators* $R_s^{ij} = W_s^{ij} \equiv 0$ *for* $i, j \in \{1, 2\}$*.* □

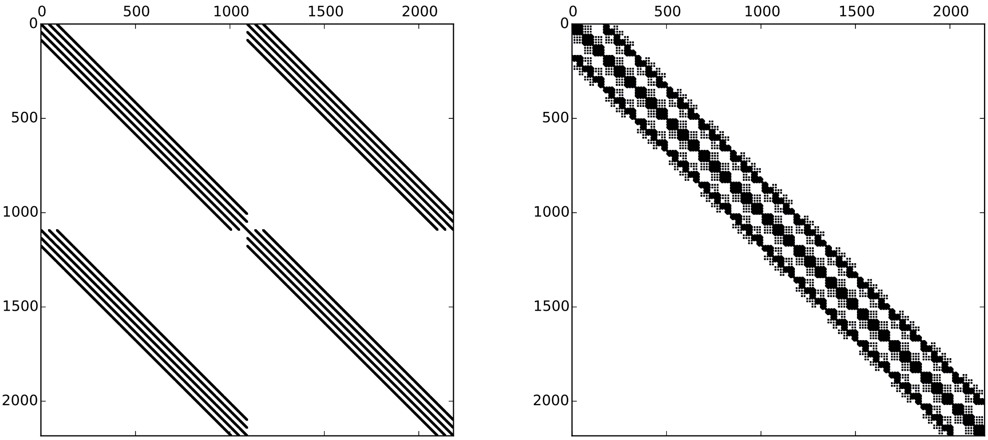


Figure 4.3: *A* `spy` *plot of* $\mathcal{P}(\tau)$ *for the* `wedge` *problem (left) and* $\Psi^{\mathsf{T}}\mathcal{P}(\tau)\Psi$ *(right) for* $d = p = 2$*, and* `nnz=100,587` *in both cases. Clearly, the permutation leads to a reduction in bandwidth, and the permuted matrix is block tri-diagonal.*

If the matrix (4.17) is sparse, it is advisable to use a sparse data structure on generator-level for (14.20a)-(14.20d) as well. Because of Corollary 4.11, the permuted 2D preconditioner can be written as,

$$\mathcal{P}_{2D} = \Psi^{\mathsf{T}}\mathcal{P}(\tau)\Psi = \begin{bmatrix} P_{1,1} & P_{1,2} & & \\ P_{2,1} & P_{2,2} & P_{2,3} & \\ & \ddots & \ddots & \ddots \\ & & \ddots & P_{n_x,n_x} \end{bmatrix} \tag{4.21}$$

with block entries $P_{i,j}$ in SSS format according to Definition 4.5, compare Figure 4.3 (right). We perform a block-LU factorization of the form $\mathcal{P}_{2D} = LSU$, with

$$L_{i,j} = \begin{cases} I & \text{if } i = j \\ P_{i,j}S_j^{-1} & \text{if } i = j+1 \end{cases}, \quad U_{i,j} = \begin{cases} I & \text{if } j = i \\ S_i^{-1}P_{i,j} & \text{if } j = i+1 \end{cases}, \tag{4.22}$$

and Schur complements given by

$$S_i = \begin{cases} P_{i,i} & \text{if } i = 1 \\ P_{i,i} - P_{i,i-1}S_{i-1}^{-1}P_{i-1,i} & \text{if } 2 \leq i \leq n_x. \end{cases} \tag{4.23}$$

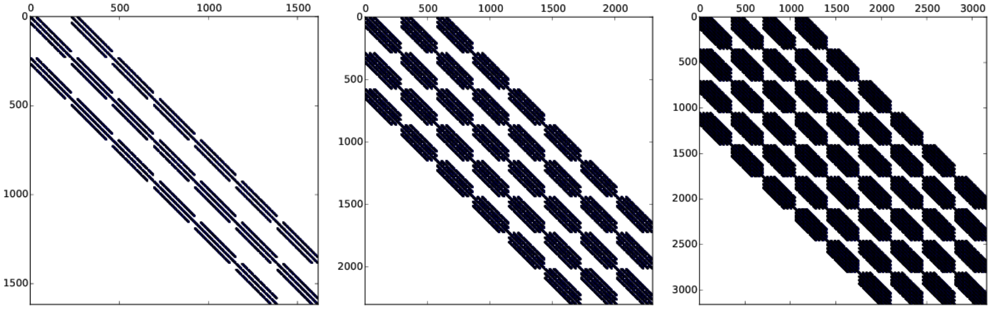
The Schur complements in (4.22)-(4.23) are SSS matrices and inverses can be computed with Algorithm 4.2. From Lemma 4.7, we conclude that this does not increase the respective off-diagonal ranks. However, in (4.22)-(4.23), we also need to perform matrix-matrix multiplications and additions of SSS matrices which lead to an increase in rank, cf. [20] and Appendix B. Therefore, we apply model order reduction in the sense of Definition 4.8 at each step i of the recursion (4.23) in order to limit the off-diagonal rank. An algorithm that limits the off-diagonal ranks to a constant, say r^*, can be found in [107]. This leads to approximate Schur complements and, hence, an inexact LU factorization. In Experiment 4.15, we show that for small off-diagonal ranks this approach results in a very good preconditioner for 2D elastic problems.

4.3.3 Block SSOR Splitting Using MSSS Computations for 3D Problems

For 3D problems, we consider a nodal-based FEM discretization of (4.12) with n_z being the outermost dimension, as shown in Figure 4.4 for different order of B-splines. In order to derive a memory-efficient algorithm for 3D problems, we consider the matrix splitting,

$$\mathcal{P}_{3D}(\tau) = \underline{L} + \hat{S} + \overline{U}, \quad \hat{S} = \text{blkdiag}(\hat{S}_1, ..., \hat{S}_{n_z}), \tag{4.24}$$

where $\underline{L}$ and $\overline{U}$ are the (sparse) strictly lower and strictly upper parts of $\mathcal{P}_{3D}(\tau)$, and $\hat{S}$ is a block-diagonal matrix with blocks $\hat{S}_i$ being in level-2 MSSS structure. This data structure is illustrated in Figure 4.5a.



(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

Figure 4.4: *Nodal-based discretization of $\mathcal{P}_{3D}(\tau)$ in 3D for different degrees p of FEM basis function.*

According to [117, Section 4.1.2], the SSOR preconditioner based on the splitting (4.24) is given by,

$$\mathcal{P}_{3D}(\tau) = \frac{1}{\eta(2-\eta)}(\eta\,\underline{L} + \hat{S})\hat{S}^{-1}(\eta\,\overline{U} + \hat{S})$$

which for $\eta = 1$ equals,

$$\mathcal{P}_{3D}(\tau) = (\underline{L}\hat{S}^{-1} + I)\hat{S}(\hat{S}^{-1}\overline{U} + I). \tag{4.25}$$

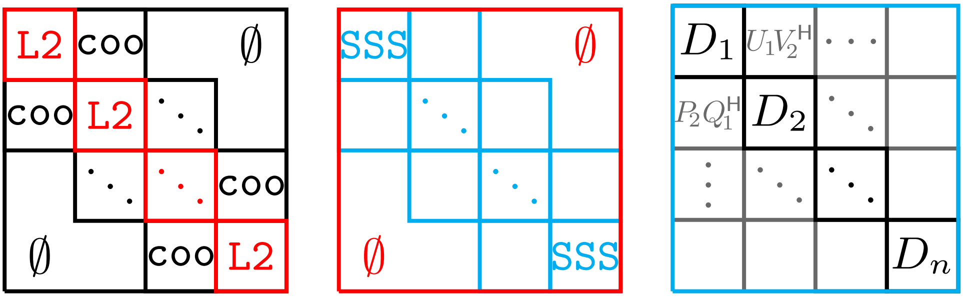
In (4.25) we note that this decomposition coincides with the 2D approach (4.22)-(4.23) when the term "$P_{i,i-1}S_{i-1}^{-1}P_{i-1,i}$" in the Schur complements (4.23) is neglected. This choice avoids a rank increase due to multiplication and addition, but yields a *worse* preconditioner than in 2D. The block entries $\hat{S}_i$,$i = 1,..,n_z$, are in level-2 MSSS structure and, hence, formula (4.22)-(4.23) can be applied sequentially for the inverses that appear in (4.25). In order to invert level-1 SSS matrices that recursively

Table 4.2: *Overview of algorithms applied at different levels for the (approximate) inversion of the preconditioner* (4.25).

Level	**Algorithm for** $(\cdot)^{-1}$	**Datatype**
3D MSSS	SSOR decomposition (4.25)	`sparse` + `L2_SSS`
2D MSSS	Schur (4.22)-(4.23) & MOR	`tridiag. L2_SSS`
1D SSS	Algorithm 4.2	`L1_SSS` (4.16)
generator	`LAPACK` routines	set of sparse matrices

appear in (4.23), we use Algorithm 4.2. On the generator level, we use suitable `LAPACK` routines; cf. Table 4.2 for an overview of the different algorithms used at each level.

We illustrate the data structure of the preconditioner (4.25) in 3D for the case of linear B-splines ($p = 1$) in Figure 4.5. On level-3, we use a mixed data format that is most memory-efficient for the splitting (4.24). Since only diagonal blocks need to be inverted, we convert those to level-2 MSSS format, and keep the off-diagonal blocks of $\underline{L}$ and $\overline{U}$ in sparse format.



(a) `L3_SSS` (b) `L2_SSS`, definition 4.9 (c) `SSS`, definition 4.5

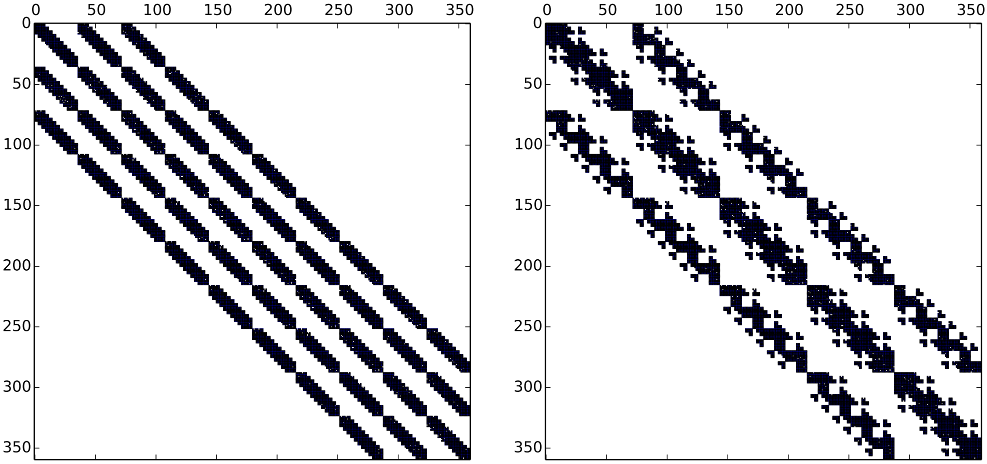
Figure 4.5: *Nested data structure for the preconditioner* (4.21) *after permutation for* $d = 3$ *and* $p = 1$*. With* `'coo'` *we abbreviate the coordinate-based sparse data structure as used, for instance, in [116].*

For $p > 1$, we apply the permutation of Definition 4.10 on each diagonal block of $\hat{S}$, cf. Figure 4.6. This way, the Schur decomposition described in Section 4.3.2 can be applied for inverting block tri-diagonal level-2 MSSS matrices.

4.3.4 Memory Analysis for the 2D and 3D MSSS Preconditioner

We finish our description of MSSS preconditioners with a memory analysis of the suggested algorithms described for 2D problems in Section 4.3.2, and for 3D problems in Section 4.3.3, respectively. The following Corollary 4.13 shows that in both cases we obtain linear memory requirements in terms of the problem size (4.8).

Corollary 4.13 (Linear memory requirement) *Consider* $p = 1$ *and a three-dimensional problem of size* $n_x \times n_y \times n_z$*. For simplicity, we assume on the generator-level* $m_i \equiv m$*, and the off-diagonal ranks of the inverse Schur complements* S_i *in* (4.23)



(a) $\hat{S}_i,\ 1 \leq i \leq n_x$

(b) $\Psi^{\mathsf{T}} \hat{S}_i \Psi$

Figure 4.6: *Permutation on level-2 leads to a block tri-diagonal level-2 MSSS matrix for* $p > 1$.

being limited by $k_i = l_i \equiv r^*$. *The grid size in y-direction on level-1 implies* n *generators via* $n = d n_y m^{-1}$, *with* m *being a constant and* $d \in \{2, 3\}$. *The memory requirement of the preconditioners* $\mathcal{P}_{2D}$ *and* $\mathcal{P}_{3D}$ *presented in Section 4.3.2 and Section 4.3.3, respectively, is linear in the respective problem dimension* (4.8).

Proof 4.14 *Consider the preconditioner* $\mathcal{P}_{2D} = LSU$ *given by* (4.22)-(4.23). *Besides blocks of the original operator, an additional storage of* n_x *inverse Schur complements* S_i^{-1} *in SSS format is required,*

$$\mathit{mem}(\mathcal{P}_{2D}^{-1}, r^*) = \mathit{mem}(\mathcal{P}_{2D}) + \sum_{i=1}^{n_x} \mathit{mem}(S_i^{-1}, r^*) \in \mathcal{O}(n_x n_y).$$

The approximate Schur decomposition described in Section 4.3.2 allows dense, full rank diagonal generators $D_i, 1 \leq i \leq n$, *of size* $m \times m$, *and limits the rank of all off-diagonal generators by* r^* *using model order reduction techniques:*

$$\mathit{mem}(S_i^{-1}, r^*) = \underbrace{n \cdot m^2}_{\sim D_i} + \underbrace{4(n-1)mr^*}_{\sim \{U_i, V_i, P_i, Q_i\}} + \underbrace{2(n-2)r^* r^*}_{\sim \{W_i, R_i\}} \in \mathcal{O}(n_y).$$

Concerning the memory requirement for storing $\mathcal{P}_{2D}$ *in MSSS format, we first note that the permutation described in Corollary 4.11 does not affect the memory consumption. Since we use sparse generators in* (14.20a)-(14.20d), *the memory requirement is of the same order as the original, sparse matrix* (4.12) *obtained from the FEM discretization.*

For 3D problems, we suggest the usage of $\mathcal{P}_{3D}$ *as in* (4.25) *based on the splitting* (4.24). *For the data structure, we keep the strictly lower and upper diagonal*

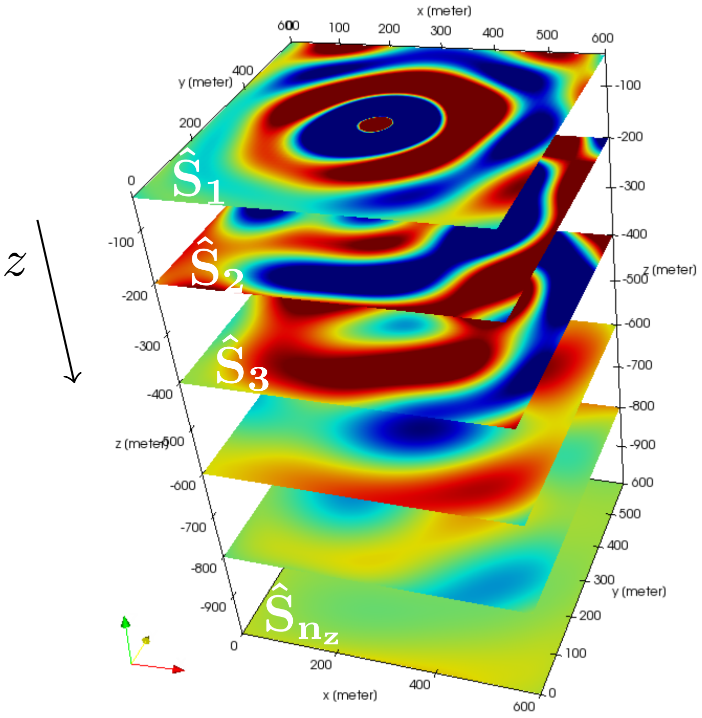


Figure 4.7: *Schematic illustration: The diagonal blocks of $\hat{S}$ in (4.24) correspond to a sequence of n_z 2D problems in the xy−plane.*

parts in sparse format and convert the diagonal blocks to level-2 MSSS format, cf. Figure 4.7,

$$\begin{aligned}\mathit{mem}(\mathcal{P}_{3D}^{-1}, r^*) &= n_z \cdot \mathit{mem}(\mathcal{P}_{2D}^{-1}, r^*) + \mathit{nnz}(\underline{L}) + \mathit{nnz}(\overline{U}) \\ &\in \mathcal{O}(n_x n_y n_z).\end{aligned}$$

□

Note that the case $p > 1$ also yields a linear memory requirement but is, for simplicity, not addressed here.

4.4 Numerical Experiments

We present numerical example for the two-dimensional, elastic `Marmousi-II` model [85] as well as for a three-dimensional elastic `wedge` problem which has been inspired by the well-known acoustic test case introduced in [75, 104] for 2D and 3D, respectively. In the examples, we restrict ourselves to Cartesian grids with fixed discretization size $h \equiv h_x = h_y = h_z$. Depending on the specific problem parameters, the maximum frequency we allow is restricted by,

$$f_{\max} < \frac{\min_{\mathbf{x}\in\Omega}\{c_p, c_s\}}{ppw \cdot h}, \quad ppw = 20, \tag{4.26}$$

where in the following experiments a minimum of 20 points per wavelength (ppw) is guaranteed, and $\omega_k = 2\pi f_k$.

All numerical examples presented in this section have been implemented in FORTRAN 90 using the GNU/gfortran compiler running over GNU/Debian Linux, and executed on a computer with 4 CPUs Intel I5 with 32 GB of RAM.

4.4.1 Proof of Concepts

We begin our numerical tests with a sequence of experiments performed on an *academic* two-dimensional `wedge` problem described in Figure 4.8. The aim of these first experiments is to prove the following concepts for the 2D algorithm introduced in Section 4.3.2:

- Demonstrate the dependency of the iterative solution method on the maximum off-diagonal rank, $r^* = \max\{r^l, r^u\}$. In Experiment 4.15 we show that a small value of r^* leads to a very good preconditioner in terms of number of Krylov iterations.
- Show that the 2D algorithm yields linear computational complexity when all problem parameters are unchanged and the grid size doubles (Experiment 4.16).
- In Experiments 4.17 and 4.18, we evaluate the frequency dependency of the MSSS-preconditioner (4.12) when $\tau \neq \omega$. This is in particular important when multiple frequencies in a matrix equation framework are considered in Section 4.4.2.

We perform parameter studies on a two-dimensional slice (xz-plane) of the `wedge` problem described in Figure 4.14. The values of ρ, c_p and c_s in the respective layers are given in Table 4.3, and the considered computational domain $\Omega = [0, 600] \times [0, 1000]$ meters is shown in Figure 4.8.

Table 4.3: *Parameter configuration of the elastic* `wedge` *problem. The Lamé parameters can be computed via* (4.6).

Parameter	Layer #1	Layer #2	Layer #3
$\rho[kg/m^3]$	1800	2100	1950
$c_p[m/s]$	2000	3000	2300
$c_s[m/s]$	800	1600	1100

In the first set of experiments, we restrict ourselves to the single-frequency case, $N_\omega = 1$. The discrete problem is, thus, given by,

$$(K + i\omega C - \omega^2 M)\mathbf{x} = \mathbf{b},$$

with a preconditioner that approximates the original operator, $\mathcal{P}(\tau) \approx (K + i\tau C - \tau^2 M), \tau = \omega$, by taking low-rank approximations in the block-LU factorization.

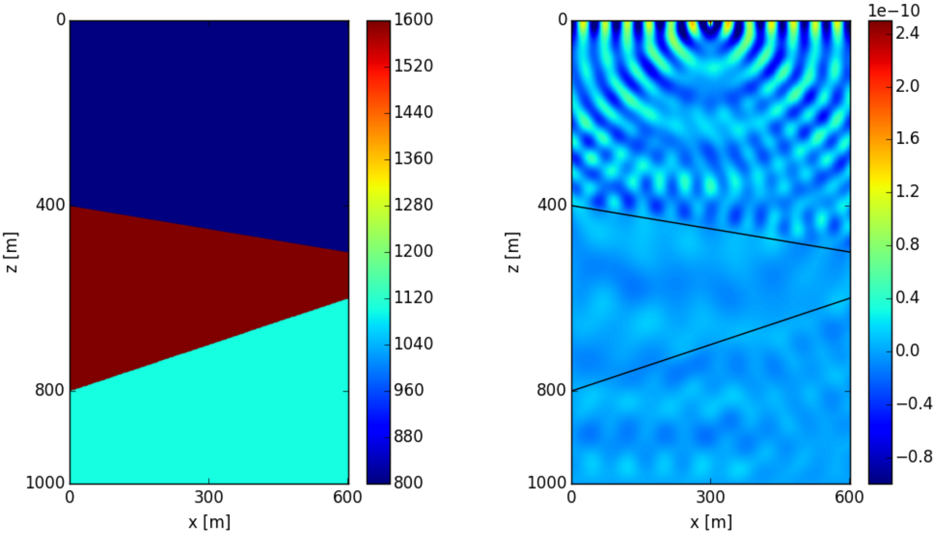


Figure 4.8: *2D elastic* `wedge` *problem used for parameter study: Speed of S-waves in m/s (left) and real part of z-component of displacement vector at* $f = 16$ *Hz (right).*

Numerical Experiment 4.15 (Off-diagonal rank) *This experiment evaluates the performance of the MSSS-preconditioner* (4.21) *for 2D problems when the maximal off-diagonal rank* r^* *is increased.*

In Experiment 4.15, we apply the approximate block-LU decomposition (4.22)-(4.23) as described in Section 4.3.2 to the 2D `wedge` problem at the frequencies $f = 8$ Hz and $f = 16$ Hz. The maximum off-diagonal rank $r^* = \max\{r^l, r^u\}$ of the Schur complements (4.23) is restricted using model order reduction techniques, cf. [107]. The dimension of the diagonal constructors has been chosen to be $m_i = 40$, cf. Table 4.1. Figure 4.9 shows the convergence behavior of preconditioned IDR(4) (Algorithm 4.1 with $N_\omega = 1$) and preconditioned BiCGStab [144]. We note that even in the high-frequency case, an off-diagonal rank of $r^* = 10$ leads to a very efficient preconditioner, and an (outer) Krylov method that converges within at most 40 iterations to a residual tolerance `tol=10e-8`. Moreover, we observe that IDR(s) outperforms BiCGStab in the considered example when the same preconditioner is applied. For a rank $r^* > 15$, we observe convergence within very few iterations.

Numerical Experiment 4.16 (Computational complexity in 2D) *The inexact block-LU factorization yields linear computational complexity when applied as a preconditioner within MSSS-preconditioned IDR(s), demonstrated for the 2D wedge problem.*

In our second numerical experiment, the maximum off-diagonal rank is fixed to $r^* = 15$ such that very few IDR iterations are required, and the computational costs in Figure 4.10 are dominated by the MSSS preconditioner. We solve the 2D `wedge` problem at frequency 8 Hz for different mesh sizes and a finite element discretization with B-splines of degree $p = \{1, 2\}$.

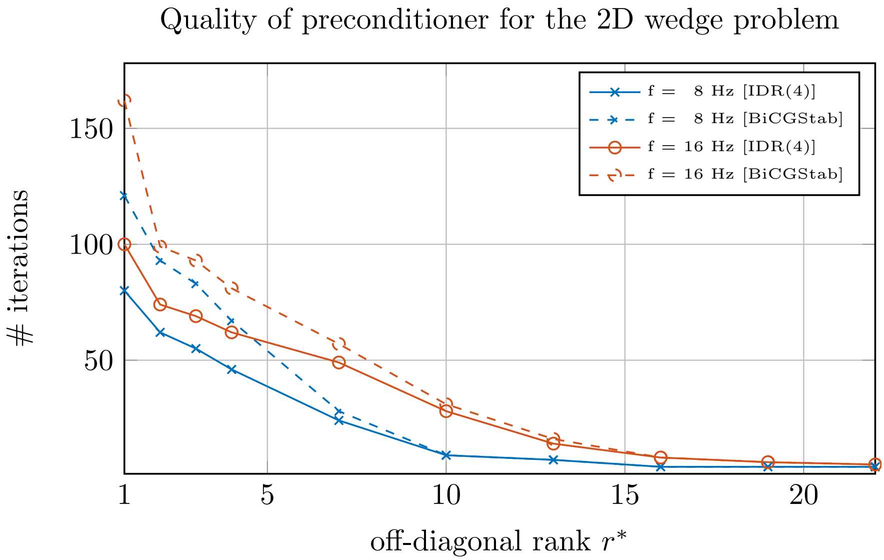


Figure 4.9: *Number of Krylov iterations when the maximum off-diagonal rank of the inverse Schur complements is restricted to* r^*.

In Figure 4.10, the CPU time is recorded for different problem sizes: The mesh size h is doubled in both spatial directions such that the number of unknowns quadruples according to (4.8). From our numerical experiments we see that the CPU time increases by a factor of ~ 4 for both, linear and quadratic, splines. This gives strong numerical evidence that the 2D MSSS computations are performed in linear computational complexity.

Numerical Experiment 4.17 (Constant points per wavelength) *Convergence behavior of MSSS-preconditioned IRD(s) when the problem size and wave frequency are increased simultaneously.*

In the previous example, the wave frequency is kept constant while the problem size is increased which is of little practical use due to oversampling. We next increase the wave frequency *and* the mesh size simultaneously such that a constant number of points per wavelength, $ppw = 20$, is guaranteed. In Table 4.4, we use the freedom in choosing the maximum off-diagonal rank parameter r^* such that the overall preconditioned IDR(s) algorithm converges within a total number of iterations that grows linearly with the frequency. This particular choice of r^* shows that the MSSS preconditioner has comparable performance to the multi-grid approaches in [101, 113] where the authors numerically prove $\mathcal{O}(n^3)$ complexity for 2D problems of size $n_x = n_y \equiv n$.

The off-diagonal rank parameter r^* can on the other hand be used to tune the preconditioner in such a way that the number of IDR iterations is kept constant for various problem sizes. In Table 4.5, we show that a constant number of ~ 30 IDR

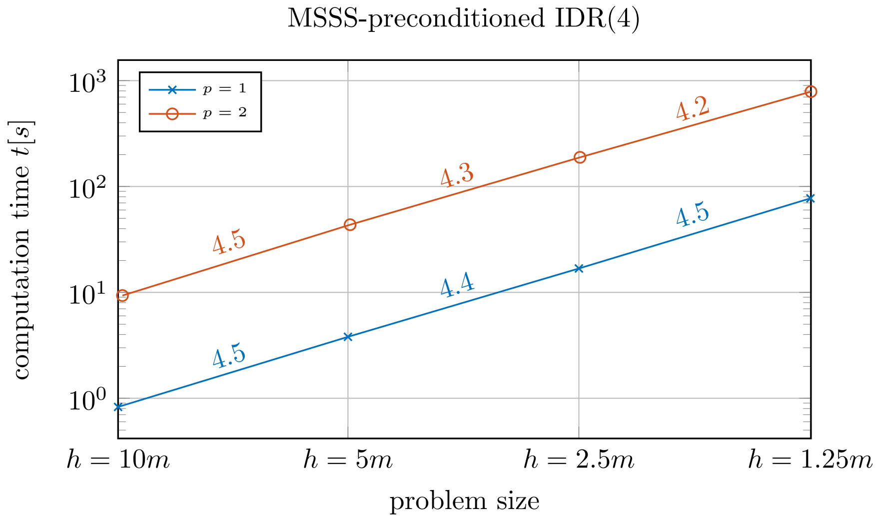


Figure 4.10: *Linear computational complexity of preconditioned IDR(4) for the 2D* `wedge` *problem at* $f = 8$ *Hz.*

Table 4.4: *Performance of the MSSS preconditioner when problem size and frequency are increased simultaneously such that* $ppw = 20$ *and* `tol=10e-8`*:* $\mathcal{O}(n^3)$ *complexity.*

f	$h[m]$	r^*	MSSS	IDR(4)	total CPU time
4 Hz	10.0	5	0.55 sec	16 iter.	0.71 sec
8 Hz	5.0	7	2.91 sec	33 iter.	4.2 sec
16 Hz	2.5	10	15.3 sec	62 iter.	31.8 sec
32 Hz	1.25	16	95.4 sec	101 iter.	242.5 sec

iterations can be achieved by a moderate increase of r^* which yields an algorithm that is nearly linear.

Numerical Experiment 4.18 (Quality of $\mathcal{P}_{2D}(\tau)$ when $\tau \neq \omega$) *Single-frequency experiments when seed frequency differs from the original problem.*

This experiments bridges to the multi-frequency case. We consider single-frequency problems at $f \in \{2, 3, 4, 5\}$ Hz, and vary the parameter τ of the preconditioner (4.21). The off-diagonal rank r^* is chosen sufficiently large such that fast convergence is obtained when $\tau = \omega$. From Figure 4.11 we conclude that the quality of the preconditioner heavily relies on the seed frequency, and a fast convergence of preconditioned IDR(4) is only guaranteed when τ is close to the original frequency.

4.4.2 The Elastic `Marmousi-II` Model

We now consider the case when $N_\omega > 1$, and the matrix equation,

$$K\mathbf{X} + iC\mathbf{X}\Sigma - M\mathbf{X}\Sigma^2 = B, \quad \mathbf{X} \in \mathbb{C}^{N \times N_\omega}, \tag{4.27}$$

Table 4.5: *Performance of the MSSS preconditioner when problem size and frequency are increased simultaneously such that ppw* $= 20$ *and* `tol=10e-8`*: Constant number of iterations.*

f	$h[m]$	r^*	MSSS	IDR(4)	total CPU time
4 Hz	10.0	3	0.50 sec	29 iter.	0.83 sec
8 Hz	5.0	7	2.91 sec	33 iter.	4.2 sec
16 Hz	2.5	11	16.9 sec	27 iter.	24.5 sec
32 Hz	1.25	18	107.1 sec	33 iter.	163.2 sec

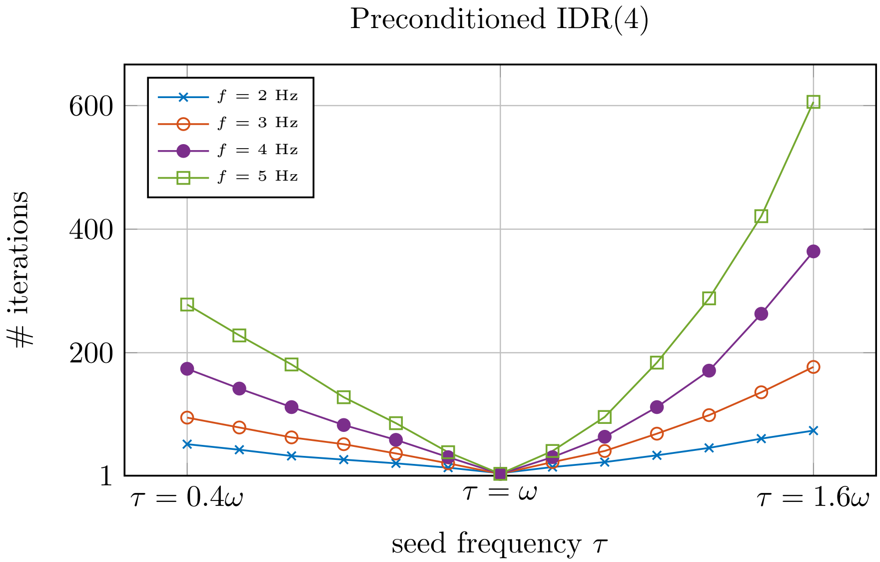


Figure 4.11: *Number of iterations of preconditioned IDR(s) when* $\tau \neq \omega$ *in* (4.21)*. We perform the experiment for different frequencies, and keep a constant grid size* $h = 5m$ *and residual tolerance* `tol = 10e-8`*.*

is solved. Note that this way we can incorporate multiple wave frequencies in the diagonal matrix $\Sigma = \text{diag}(\omega_1, ..., \omega_{N_\omega})$, and different source terms lead to a block right-hand side of the form $B = [\mathbf{b}_1, ..., \mathbf{b}_{N_\omega}]$. When multiple frequencies are present, the choice of seed frequency τ is crucial as we demonstrate for the `Marmousi-II` problem in Experiment 4.20. We solve the matrix equation (4.27) arising from the realistic `Marmousi-II` problem [85]. We consider a subset of the original computational domain, $\Omega = [0, 4000] \times [0, 1850]m$, as suggested in [113], cf. Figure 4.13.

Numerical Experiment 4.19 (`Marmousi-II` at multiple right-hand sides) *Performance of the MSSS-preconditioned IDR(s) method for the two-dimensional, elastic* `Marmousi-II` *problem when multiple source locations are present.*

We consider the `Marmousi-II` problem depicted in Figure 4.13 at $h = 5$m and frequency $f = 2$ Hz. We present the performance of MSSS-preconditioned IDR(4) for N_ω

equally-spaced source locations (right-hand sides) in Table 4.6. The CPU time required for the preconditioner as well as the iteration count is constant when $N_\omega > 1$ because we consider a single frequency. The overall wall clock time, however, scales better than N_ω due to the efficient implementation of block matrix-vector products in the IDR algorithm. The experiment for $N_\omega = 20$ shows that there is an optimal number of right-hand sides for a single-core algorithm.

Table 4.6: *Numerical experiments for the* `Marmousi-II` *problem at $f = 2$ Hz using a maximum off-diagonal rank of $r^* = 15$.*

# RHSs	MSSS fact. [sec]	IDR(4) [sec]
1	60.2	8.18 (8 iter.)
5	60.2	25.0 (8 iter.)
10	60.1	43.5 (8 iter.)
20	60.3	108.3 (8 iter.)

Numerical Experiment 4.20 (`Marmousi-II` at multiple frequencies) *Performance of MSSS-preconditioned IDR(s) for the two-dimensional* `Marmousi-II` *problem at multiple frequencies.*

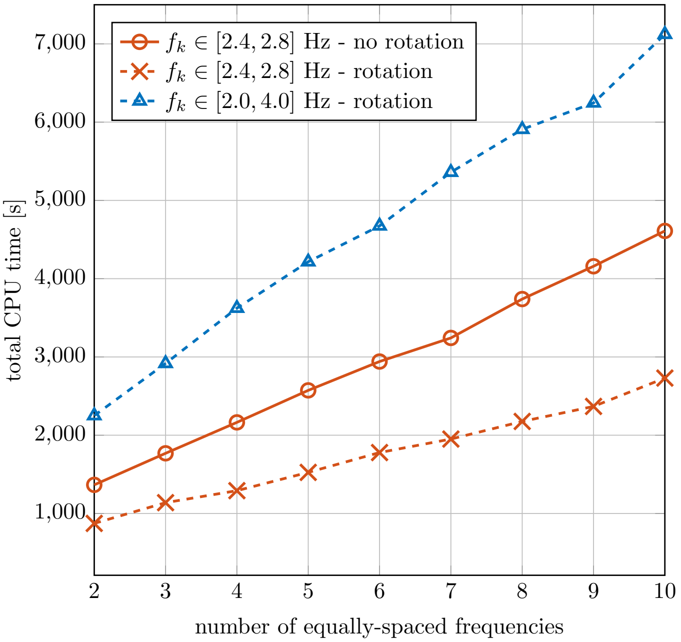


Figure 4.12: *Total CPU times of MSSS-IDR(4) for $N_\omega > 1$ frequencies equally-spaced within a fixed range. Additional scaling (dashed lines) following [9] improves convergence, and allows for larger frequency ranges.*

In Experiment 4.20, we consider a single source term located at $(L_x/2, 0)^\mathsf{T}$ and N_ω frequencies equally-spaced in the intervals $f_k \in [2.4, 2.8]$ Hz and $f_k \in [2.0, 4.0]$ Hz.

The seed frequency is chosen at $\tau = (1 - 0.5i)\omega_{max}$ for which we recorded the best convergence behavior. When the number of frequencies is increased, we observe an improved performance compared to an extrapolation of the $N_\omega = 2$ case. We also observed that the size of interval in which the different frequencies range is crucial for the convergence behavior. In [9], we describe how the convergence of global GMRES [68] can be improved by scaling the k-th column of the block unknown $\mathbf{X}$ by the factor $e^{-i\varphi_k}$. Spectral analysis shows that the angles φ_k can be chosen such that

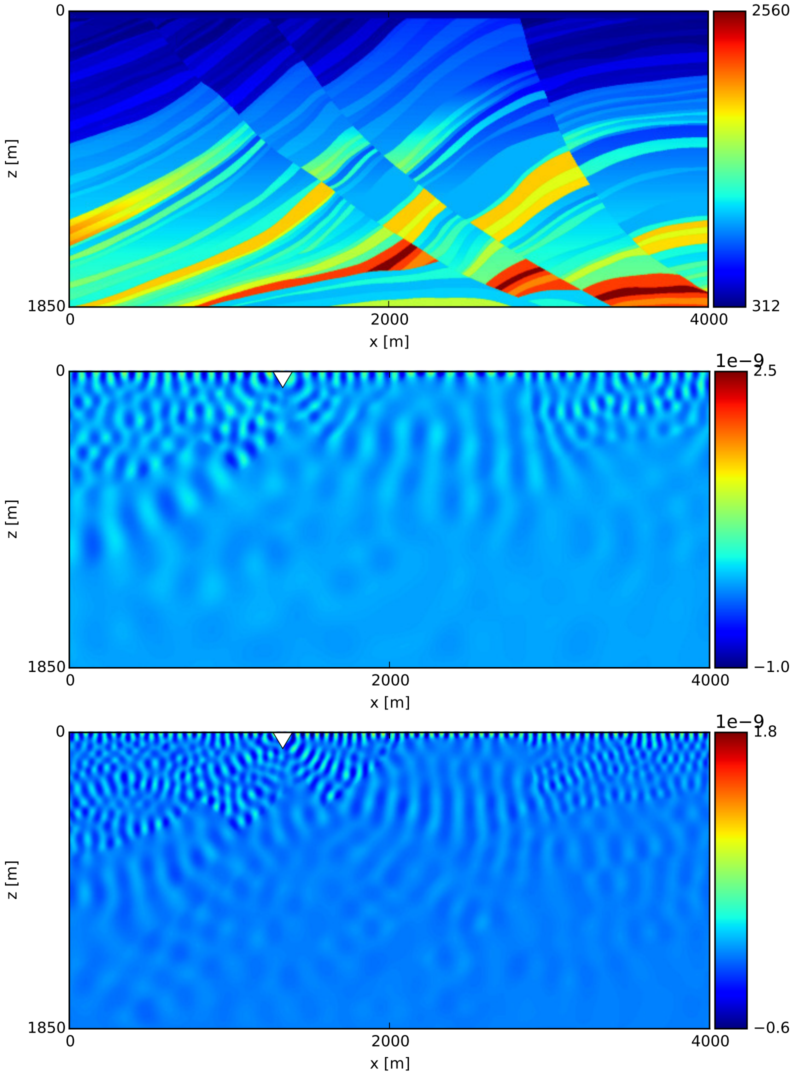


Figure 4.13: *Speed of S-waves in m/s (top), and real part of the z-component of the displacement vector in frequency-domain at $f = 4$ Hz (middle) and $f = 6$ Hz (bottom) for the* `Marmousi-II` *model, cf. [85] for a complete parameter set. The source location is indicated by the symbol '∇'.*

the spectrum of the preconditioned operator is rotated and convergence is improved, cf. [9]. In the present case of global IDR(s) (Algorithm 4.1) combined with an inexact MSSS preconditioner (4.21) we record a reduction to 60% of the CPU time when spectral rotation is applied to the $N_\omega = 10$ case, cf. Figure 4.12.

4.4.3 A Three-Dimensional Elastic `Wedge` Problem

The `wedge` problem with parameters presented in Table 4.3 is extended to a third spatial dimension, resulting in $\Omega = [0, 600] \times [0, 600] \times [0, 1000] \subset \mathbb{R}^3$.

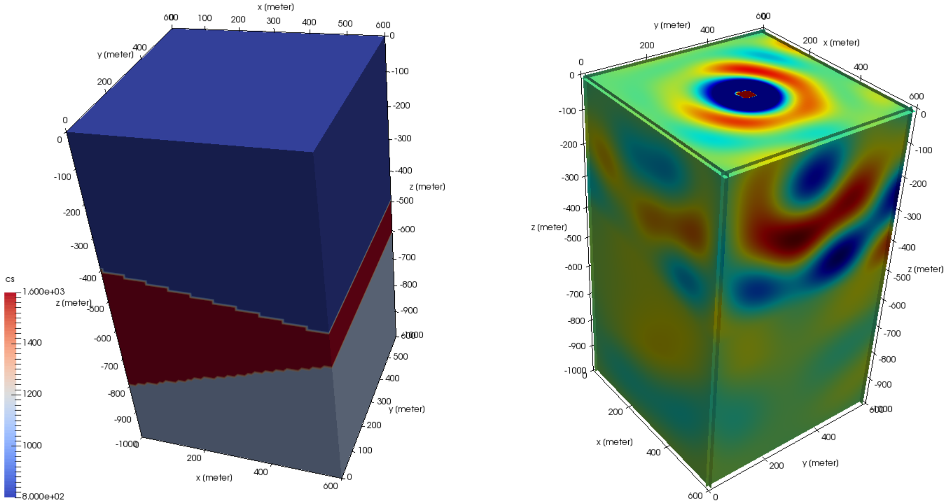


Figure 4.14: *Left: Parameter configuration of the elastic* `wedge` *problem for* $d = 3$ *according to Table 4.3. Right: Numerical solution of* $\Re(\mathbf{u}_z)$ *at* $f = 4Hz$*.*

Numerical Experiment 4.21 (A 3D elastic wedge problem) *A three-dim- ensional, inhomogeneous elastic wedge problem with physical parameters specified in Table 4.3 is solved using the SSOR-MSSS preconditioner described in Section 4.3.3.*

Similar to Experiment 4.17, we consider a constant number of 20 points per wavelength, and increase the wave frequency from 2Hz to 4Hz while doubling the number of grid points in each spatial direction. In Figure 4.15 we observe a factor of ~ 4 which numerically indicates a complexity of $\mathcal{O}(n^5)$ for 3D problems. Moreover, we note that IDR outperforms BiCGStab in terms of number of iterations. The corresponding CPU times are presented in Table 4.7: From the previous analysis, a factor of ~ 32 for the overall CPU times is expected since the number of unknowns in three spatial directions is doubled (linear complexity yields a factor of 8), and Figure 4.15 motivates an additional factor of 4 in iteration numbers.

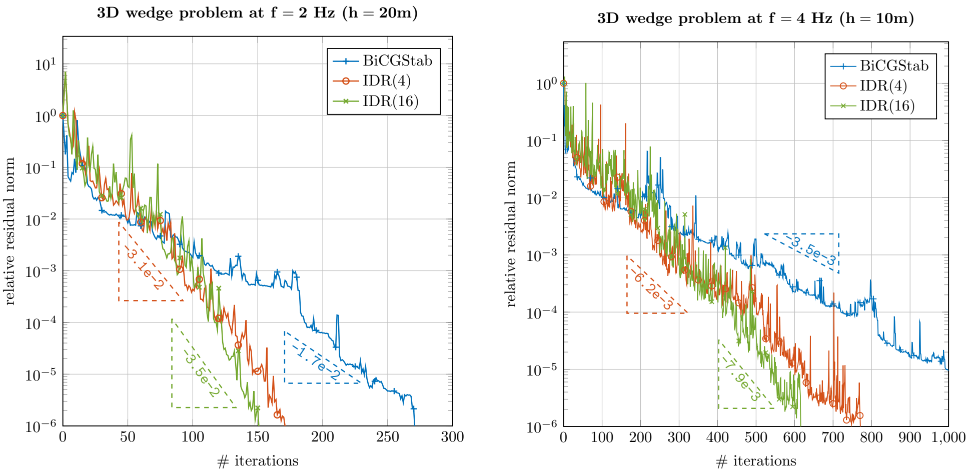


Figure 4.15: *Convergence history of different Krylov methods preconditioned with the SSOR-MSSS preconditioner* (4.25) *for the 3D* `wedge` *problem of Figure 4.14. We indicate (approximate) slopes based on a linear fit of the convergence curves.*

Table 4.7: *Total CPU times in seconds corresponding to the convergence plots in Figure 4.15. Note that BiCGStab at $f = 4$ Hz is stopped after* 1,000 *iterations, cf. Figure 4.15.*

frequency	BiCGStab	IDR(4)	IDR(16)
$f = 2$ Hz	144.5	95.5	91.3
$f = 4$ Hz	4430.7	3536.4	3100.5

Conclusions

We present an efficient *hybrid* method for the numerical solution of the inhomogeneous time-harmonic elastic wave equation. We use an incomplete block-LU factorization based on MSSS matrix computations as a preconditioner for IDR(s). The presented framework further allows to incorporate multiple wave frequencies and multiple source locations in a matrix equation setting (4.13). The suggested MSSS preconditioner is conceptional different for 2D and 3D problems:

- We derive an MSSS permutation matrix (4.19) that transforms the 2D elastic operator into block tridiagonal level-2 MSSS matrix structure. This allows the application of an approximate Schur factorization (4.22)-(4.23). In order to achieve linear computational complexity, the involved SSS operations (level-1) are approximated using model order reduction techniques that limit the off-diagonal rank.

- A generalization to 3D problems is not straight-forward because no model order reduction algorithms for level-2 MSSS matrices are currently available [107]. We therefore suggest the SSOR splitting (4.25) where off-diagonal blocks are treated as sparse matrices and diagonal blocks resemble a sequence of 2D problems in level-2 MSSS structure.

We present a series of numerical experiments on a 2D elastic `wedge` problem (Figure 4.8) that prove theoretical concepts. In particular, we have numerically shown that a small off-diagonal rank $r^* \sim 10$ yields a preconditioner such that IDR(s) converges within very few iterations (Experiment 4.15).

Further numerical experiments for 2D elastic problems are performed on the realistic `Marmousi-II` data set. The newly derived matrix equation approach shows computational advantages when multiple right-hand sides (Experiment 4.19) and multiple frequencies (Experiment 4.20) are solved simultaneously.

In Corollary 4.13, we prove that the MSSS preconditioner has linear memory requirements for 2D and 3D problems. The overall computational complexity is investigated for the case of a constant number of wavelength, i.e. the number of grid points n in one spatial direction in linearly increased with the wave frequency. Numerical experiments show $\mathcal{O}(n^3)$ complexity for 2D (Experiment 4.17) and $\mathcal{O}(n^5)$ complexity for 3D (Experiment 4.21) problems. The 3D preconditioner solves a sequence of 2D problems and can be parallelized in a straight forward way.

Chapter 5

An Algorithmic Comparison Study

Abstract. In this paper, we present a comparison study for three different iterative Krylov methods that we have recently developed for the simultaneous numerical solution of frequency-domain wave propagation problems when multiple wave frequencies are present. The three approaches have in common that they require the application of a single shift-and-invert preconditioner at a suitable *seed* frequency. The focus of the present work lies on the performance of the respective iterative method. In particular for three-dimensional problems, the efficient application of the shift-and-invert preconditioner is discussed. We conclude with numerical examples that provide guidance concerning the suitability of the three methods.

Introduction

After spatial discretization using, for instance, the finite element method with N degrees of freedom [13, Section 2], the time-harmonic wave equation has the form,

$$(K + i\omega_k C - \omega_k^2 M)x_k = b, \quad \omega_k := 2\pi f_k, \quad k = 1, ..., N_\omega, \tag{5.1}$$

with stiffness matrix K, mass matrix M, and C consisting of Sommerfeld boundary conditions [2] modeling absorption. Note that (5.1) yields a sequence of N_ω linear

This chapter is based on the journal article:

M. Baumann and M.B. van Gijzen (2017). *Efficient iterative methods for multi-frequency wave propagation problems: A comparison study.* Procedia Computer Science Vol. **108**, pp. 645–654.

The extended paper version is invited for publication in the ICCS 2017 Special Issue of the Journal of Computational Science.

systems of equations. One way to solve the systems (5.1) simultaneously is to define the block matrix of unknowns, $\mathbf{X} := [x_1, ..., x_{N_\omega}] \in \mathbb{C}^{N \times N_\omega}$, and to note that (5.1) can be rewritten as a linear matrix equation,

$$\mathcal{A}(\mathbf{X}) := K\mathbf{X} + iC\mathbf{X}\Omega - M\mathbf{X}\Omega^2 = B, \quad \text{with } \Omega := \operatorname{diag}(\omega_1, ..., \omega_{N_\omega}) \text{ and } B := b1^\mathsf{T}. \tag{5.2}$$

The matrix equation (5.2) can then be solved using a global Krylov method, cf. [68]. A second approach is to consider a linearization [125] of the form,

$$\left(\begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} - \omega_k \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \right) \begin{bmatrix} \omega_k x_k \\ x_k \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad k = 1, ..., N_\omega, \tag{5.3}$$

where the angular frequencies $\omega_1, ..., \omega_{N_\omega}$ appear as a (linear) shift. For short-hand notation, we define the block matrices,

$$\mathcal{K} := \begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} \in \mathbb{C}^{2N \times 2N} \quad \text{and} \quad \mathcal{M} := \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \in \mathbb{C}^{2N \times 2N}, \tag{5.4}$$

and write (5.3) as $(\mathcal{K} - \omega_k \mathcal{M})\mathbf{x}_k = \mathbf{b}$, for $\mathbf{x}_k := [\omega_k x_k, x_k]^\mathsf{T}$ and $\mathbf{b} := [b, 0]^\mathsf{T}$. We will consider the case $C \equiv 0$ independently. The matrix equation (5.2) then reduces to two terms, and we can identify $\mathcal{K} = K$ as well as $\mathcal{M} = M$ and avoid doubling of dimensions in (5.3). In this paper, we review and compare the following recently developed algorithms:

- Global GMRES [68] for the matrix equation approach (5.2) (cf. Algorithm 5.1 and [13]),
- Polynomial preconditioners [1, 9] for multi-shift GMRES (cf. Algorithm 5.2),
- Nested multi-shift FOM-FGMRES as presented in [8] (cf. Algorithm 5.3-5.4).

Note that this list does not consider a comparison with the algorithms suggested by [6, 119] and by [132]. Moreover, we restrict ourselves to GMRES-variants of the respective algorithms, and refer to [5] for global IDR(s) and to [8] for the more memory-efficient combination nested IDR-QMRIDR(s). In [1] a shifted polynomial preconditioner is used within multi-shift BiCG. The derivations in Section 5.1 emphasize that the costs-per-iteration of each proposed algorithm are comparable, and are dominated by the shift-and-invert preconditioner. In Section 5.2, we evaluate the three approaches for a benchmark problem of the discretized time-harmonic elastic wave equation. Special emphasis is put on the practical application of the shift-and-invert preconditioner in three spatial dimensions in Section 5.3.

5.1 Review of the Developed Algorithms

The review of the subsequent algorithms is based on our works [8, 9, 13].

5.1.1 The Preconditioned Matrix Equation Approach

The matrix equation (5.2) with right preconditioning reads,

$$\mathcal{A}(P(\tau)^{-1}\mathbf{Y}) = B, \quad \mathbf{X} = P(\tau)^{-1}\mathbf{Y}, \quad \text{where } P(\tau) := (K + i\tau C - \tau^2 M)^{-1}, \tag{5.5}$$

and $\mathcal{A}(\cdot)$ as in (5.2). A similar reformulation has been suggested in [132]. We note that the preconditioner $P(\tau)$ can be applied inexactly using, for instance, an incomplete LU factorization. The (possibly complex) parameter τ is called the *seed* frequency. In Algorithm 5.1, we state the global GMRES method [68]. Note that in the block Arnoldi method the trace inner product is used, and norms are replaced by the Frobenius norm $\|\cdot\|_\mathsf{F}$ for block matrices. After m iterations, an approximate solution to (5.2) in the block Krylov subspace $\mathcal{K}_m(\mathcal{A}P(\tau)^{-1}, B)$ is obtained. An efficient preconditioner will yield convergence for $m \ll N$.

Algorithm 5.1 Right-preconditioned global GMRES for the matrix equation (5.5) with linear operator as defined in (5.2), cf. [68]

```
Set $R_0 = B,\ V_1 = R_0/\|R_0\|_\mathsf{F}$                          ▷ Initialization ($\mathbf{X}_0 = 0$)
for $j = 1$ to $m$ do
    Apply $W = \mathcal{A}(P(\tau)^{-1}V_j)$                        ▷ Preconditioner (might be inexact)
    for $i = 1$ to $j$ do                                            ▷ Block-Arnoldi method
        $h_{i,j} = \mathrm{tr}(W^\mathsf{H} V_i)$
        $W = W - h_{i,j} V_i$
    end for
    Set $h_{j+1,j} = \|W\|_\mathsf{F}$ and $V_{j+1} = W/h_{j+1,j}$
end for
Set $\underline{\mathrm{H}}_m = [h_{i,j}]_{i=1,\dots,m}^{j=1,\dots,m+1}$, $\mathbf{V}_m = [V_1, \dots, V_m]$      ▷ $\mathbf{V}_m$ spans block Krylov space
Solve $\mathbf{y}_m = \mathrm{argmin}_\mathbf{y} \|\underline{\mathrm{H}}_m \mathbf{y} - \|B\|_\mathsf{F}\mathbf{e}_1\|_2$      ▷ $\mathbf{e}_1$ is first unit vector in $\mathbb{C}^{m+1}$
Compute $\mathbf{X}_m = P(\tau)^{-1}(\mathbf{V}_m * \mathbf{y}_m)$      ▷ '$*$' denotes the star product, cf. [68]
```

5.1.2 Preconditioners for Shifted Linear Systems

The methods presented in this section are both two-level preconditioning approaches. As a first-level preconditioner, a shift-and-invert preconditioner of the form,

$$\begin{aligned}\mathcal{P}(\tau)^{-1} = (\mathcal{K} - \tau\mathcal{M})^{-1} &\overset{(5.4)}{=} \left(\begin{bmatrix} iC & K \\ I & 0 \end{bmatrix} - \tau \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} I & \tau I \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & (K + i\tau C - \tau^2 M)^{-1} \end{bmatrix} \begin{bmatrix} 0 & I \\ I & -iC + \tau M \end{bmatrix},\end{aligned} \tag{5.6}$$

is applied. Based on the decomposition (5.6) we note that $P(\tau)^{-1} = (K + i\tau C - \tau^2 M)^{-1}$ as defined in (5.5) is the main computational work and, hence, the work-per-iteration is comparable to Algorithm 5.1. When applying a scaled shift-and-invert preconditioner to the block systems (5.3), the following equivalence holds,

$$(\mathcal{K} - \omega_k\mathcal{M})\mathcal{P}_k^{-1}\mathbf{y}_k = \mathbf{b} \quad \Leftrightarrow \quad (\mathcal{K}\mathcal{P}(\tau)^{-1} - \eta_k I)\mathbf{y}_k = \mathbf{b}, \tag{5.7}$$

if we introduce $\eta_k := \omega_k/(\omega_k - \tau)$, and the scaled preconditioners in (5.7) yield $\mathcal{P}_k^{-1} := (1-\eta_k)\mathcal{P}(\tau)^{-1} = (1-\eta_k)(\mathcal{K}-\tau\mathcal{M})^{-1}$. Note that the latter is a preconditioned shifted linear system with (complex) shifts η_k and system matrix $\mathcal{C} := \mathcal{K}\mathcal{P}(\tau)^{-1} = \mathcal{K}(\mathcal{K} - \tau\mathcal{M})^{-1}$. Due to the equivalence in (5.7), the preconditioner (5.6) needs to be applied exactly. Moreover, right-preconditioning requires the back-substitution $\mathbf{x}_k = \mathcal{P}_k^{-1}\mathbf{y}_k$.

Shifted Neumann Preconditioners

After applying the shift-and-invert preconditioner (5.6) to (5.3), we remain with solving,

$$(\mathcal{C} - \eta_k I)\mathbf{y}_k = \mathbf{b}, \quad \mathbf{x}_k = \mathcal{P}_k^{-1}\mathbf{y}_k, \quad k = 1, ..., N_\omega, \tag{5.8}$$

where $\mathcal{C} = \mathcal{K}\mathcal{P}(\tau)^{-1}$, and with (complex) shifts $\eta_k = \omega_k/(\omega_k - \tau)$. Efficient algorithms for shifted linear systems (5.8) rely on the shift-invariance property, i.e. the identity, $\mathcal{K}_m(\mathcal{C}, \mathbf{b}) \equiv \mathcal{K}_m(\mathcal{C} - \eta I, \mathbf{b})$, for any shift $\eta \in \mathbb{C}$; cf. [50, 124]. The (preconditioned) spectrum of the matrix $\mathcal{C}$ is known to be enclosed by a circle of radius R and center c [9, 147]. Therefore, the Neumann preconditioner p_n [117, Chapter 12.3] of degree n,

$$\mathcal{C}^{-1} \approx \sum_{i=0}^{n} (I - \xi\mathcal{C})^i =: p_n(\mathcal{C}), \quad \text{with } \xi = \frac{1}{c} = -\frac{\tau - \bar{\tau}}{\bar{\tau}}, \tag{5.9}$$

has optimal spectral radius [9]. The polynomial preconditioner (5.9) can also be represented in a different basis as, $p_n(\mathcal{C}) = \sum_{i=0}^{n} \alpha_i \mathcal{C}^i$. Shift-invariance can then be preserved if the following holds,

$$(\mathcal{C} - \eta_k I)p_{n,k}(\mathcal{C}) = \mathcal{C}p_n(\mathcal{C}) - \tilde{\eta}_k I, \tag{5.10}$$

where $p_{n,k}(\mathcal{C}) = \sum_{i=0}^{n} \alpha_{i,k}\mathcal{C}^i$ is a polynomial preconditioner for $(\mathcal{C} - \eta_k I)$. Substitution yields,

$$\sum_{i=0}^{n} \alpha_{i,k}\mathcal{C}^{i+1} - \sum_{i=0}^{n} \eta_k \alpha_{i,k}\mathcal{C}^i - \sum_{i=0}^{n} \alpha_i \mathcal{C}^{i+1} + \tilde{\eta}_k I = 0. \tag{5.11}$$

The latter (5.11) is a difference equation and can be solved in closed form [1]:

$$\alpha_{n,k} = \alpha_n, \tag{12a}$$
$$\alpha_{i-1,k} = \alpha_{i-1} + \eta_k \alpha_{i,k}, \quad \text{for } i = n, ..., 1, \tag{12b}$$
$$\tilde{\eta}_k = \eta_k \alpha_{0,k}. \tag{12c}$$

Inner-Outer Krylov Methods

In our approach [8], we modify (5.8) by the substitutions, $\bar{\mathcal{K}} := \mathcal{K} - \omega_1\mathcal{M}$, $\bar{\mathcal{C}} := \bar{\mathcal{K}}\mathcal{P}(\tau)^{-1}$, and solve the equivalent shifted systems,

$$(\bar{\mathcal{C}} - \bar{\eta}_k I)\mathbf{y}_k = \mathbf{b}, \quad \bar{\eta}_k := \frac{\omega_k - \omega_1}{\omega_k - \tau}, \quad k = 1, ..., N_\omega, \tag{5.13}$$

Algorithm 5.2 Multi-shift GMRES with polynomial preconditioner (5.9) for systems (5.8), cf. [1, 9]

```
Set $\mathbf{r}_0 = \mathbf{b},\ \mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|$    ▷ Initialization ($\mathbf{x}_0 = \mathbf{0}$)
for $j = 1$ to $m$ do
    Apply $\mathbf{w} = \mathcal{C}p_n(\mathcal{C})\mathbf{v}_j$    ▷ Polynomial preconditioner (5.9) of degree $n$
    for $i = 1$ to $j$ do    ▷ Arnoldi method
        $h_{i,j} = \mathbf{w}^{\mathsf{H}}\mathbf{v}_i$
        $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
    end for
    Set $h_{j+1,j} = \|\mathbf{w}\|$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
end for
Set $\underline{\mathrm{H}}_m = [h_{i,j}]_{i=1,\dots,m}^{j=1,\dots,m+1}$ and $V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$
for $k = 1$ to $N_\omega$ do
    Solve $\mathbb{C}^m \ni \mathbf{z}_k = \mathrm{argmin}_{\mathbf{z}} \|(\underline{\mathrm{H}}_m - \tilde{\eta}_k \underline{\mathrm{I}}_m)\mathbf{z} - \|\mathbf{r}_0\|\mathbf{e}_1\|$    ▷ Shifts $\tilde{\eta}_k$ acc. to (12c)
    Resubstitute $\mathbf{y}_k = p_{n,k}(\mathcal{C})V_m\mathbf{z}_k$    ▷ Coefficients of $p_{n,k}$ acc. to (12a)-(12b)
end for
```

with the advantage that for $k = 1$ we solve the *base* system $\bar{\mathcal{C}}\mathbf{y}_1 = \mathbf{b}$ (unshifted case). A nested multi-shift Krylov algorithm consists in general of m_i inner iterations and m_o outer iterations. The nested FOM-FGMRES algorithm [8] is a combination of inner multi-shift FOM (Algorithm 5.3) with outer flexible multi-shift GMRES (Algorithm 5.4).

Algorithm 5.3 Inner multi-shift FOM for (5.13), cf. [124]

```
Set $\mathbf{r}_0 = \mathbf{b},\ \mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|$    ▷ Initialization ($\mathbf{x}_0 = \mathbf{0}$)
for $j = 1$ to $m_i$ do
    Apply $\mathbf{w} = \bar{\mathcal{K}}(\mathcal{K} - \tau\mathcal{M})^{-1}\mathbf{v}_j$    ▷ Apply matrix $\bar{\mathcal{C}}$, cf. definition in (5.13)
    for $i = 1$ to $j$ do    ▷ Arnoldi method
        $h_{i,j} = \mathbf{w}^{\mathsf{H}}\mathbf{v}_i$
        $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
    end for
    Set $h_{j+1,j} = \|\mathbf{w}\|$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
end for
Set $H_{m_i} = [h_{i,j}]_{i=1,\dots,m_i}^{j=1,\dots,m_i}$ and $V_{m_i} = [\mathbf{v}_1, \dots, \mathbf{v}_{m_i}]$
for $k = 1$ to $N_\omega$ do
    Solve $\mathbb{C}^{m_i} \ni \mathbf{y}_k = (H_{m_i} - \bar{\eta}_k I_{m_i})^{-1}(\|\mathbf{r}_0\|\mathbf{e}_1)$    ▷ Shifted Hessenberg systems
    Compute $\gamma_k = \mathbf{y}_k(m_i)/\mathbf{y}_1(m_i)$    ▷ Collinearity factors, cf. [8]
    Compute $\mathbf{x}_k = V_{m_i}\mathbf{y}_k$
end for
```

In [8], we derive that if the inner method yields collinear residuals in the sense,

$$\mathbf{r}_j^{(k)} = \gamma_j^{(k)}\mathbf{r}_j, \quad \gamma_j^{(k)} \in \mathbb{C} \text{ for } k = 1, \dots, N_\omega, \tag{5.14}$$

for $\mathbf{r}_j$ being the residual of the base system after m_i inner iterations, we can preserve shift-invariance in the outer method. The consecutive collinearity factors of the inner method then appear on a diagonal matrix Γ_k of a modified Hessenberg matrix in the outer loop (see line 13 in Algorithm 5.4 and [8], respectively). More precisely, after m_o outer iterations, the solution to,

$$\bar{\mathbf{z}}_k = \operatorname*{argmin}_{\mathbf{z}\in\mathbb{C}^{m_o}} \left\| \left((\underline{\mathrm{H}}_{m_o} - \underline{\mathrm{I}}_{m_o})\Gamma_k + \underline{\mathrm{I}}_{m_o} \right) \mathbf{z} - \|\mathbf{r}_0\| \mathbf{e}_1 \right\|_2, \quad \mathbf{y}_k = Z_{m_o}^{(k)} \bar{\mathbf{z}}_k, \tag{5.15}$$

yields approximate solutions to (5.13) in the search spaces $Z_{m_o}^{(k)} \in \mathbb{C}^{2N\times m_o}$ that minimize the 2-norm of the residual of the k-th shifted system, cf. [8]. In (5.15), the Hessenberg matrix $\underline{\mathrm{H}}_{m_o}$ corresponds to the base system, and $\Gamma_k := \mathrm{diag}(\gamma_1^{(k)}, ..., \gamma_{m_o}^{(k)})$ is constructed from the collinearity factors in (5.14). Note that multi-shift FOM (Algorithm 5.3) yields collinear residuals by default [8, 124].

Algorithm 5.4 Outer multi-shift FGMRES for (5.13), cf. [8, 50]

```
 1: Set $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|$          ▷ Initialization $(\mathbf{x}) = \mathbf{0}$)
 2: for $j = 1$ to $m_o$ do
 3:     $[\mathbf{z}_j^{(k)}, \{\gamma_j^{(k)}\}_{k=1}^{N_\omega}]$ = msFOM($\bar{\mathcal{C}}, \{\bar{\eta}_k\}_{k=1}^{N_\omega}, \mathbf{v}_j$, maxit=$m_i$)      ▷ Inner method
                                                                  (call of Algorithm 5.3)
 4:     Apply $\mathbf{w} = \bar{\mathcal{K}}(\mathcal{K} - \tau\mathcal{M})^{-1}\mathbf{z}_j^{(k=1)}$     ▷ Apply matrix $\bar{\mathcal{C}}$ to base system
 5:     for $i = 1$ to $j$ do                                       ▷ Arnoldi method
 6:         $h_{i,j} = \mathbf{w}^{\mathsf{H}}\mathbf{v}_i$
 7:         $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
 8:     end for
 9:     Set $h_{j+1,j} = \|\mathbf{w}\|$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
10: end for
11: Set $\underline{\mathrm{H}}_{m_o} = [h_{i,j}]_{i=1,...,m_o}^{j=1,...,m_o+1}$ and $Z_{m_o}^{(k)} = [\mathbf{z}_1^{(k)}, ..., \mathbf{z}_{m_o}^{(k)}]$     ▷ Collect search spaces
12: for $k = 1$ to $N_\omega$ do
13:     Set $\underline{\mathrm{H}}_{m_o}^{(k)} = \left(\underline{\mathrm{H}}_{m_o} - \underline{\mathrm{I}}_{m_o}\right)\Gamma_k + \underline{\mathrm{I}}_{m_o}$, where $\Gamma_k := \mathrm{diag}(\gamma_1^{(k)}, ..., \gamma_{m_o}^{(k)})$
14:     Solve $\mathbb{C}^{m_o} \ni \bar{\mathbf{z}}_k = \mathrm{argmin}_{\mathbf{z}} \left\|\underline{\mathrm{H}}_{m_o}^{(k)}\mathbf{z} - \|\mathbf{r}_0\|\mathbf{e}_1\right\|$     ▷ Hessenberg systems (5.15)
15:     Compute $\mathbf{y}_k = Z_{m_o}^{(k)}\bar{\mathbf{z}}_k$
16: end for
```

5.2 Comparison Study and Convergence Behavior

We focus our numerical experiments on linear systems (5.1) that stem from a finite element discretization [2, 13] of the time-harmonic elastic wave equation [28]:

$$-\omega_k^2 \rho \mathbf{u}_k - \nabla\cdot\sigma(\mathbf{u}_k) = \mathbf{s}, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^{d=\{2,3\}}, \tag{5.16a}$$

$$i\omega_k\rho\ B(c_p, c_s)\mathbf{u}_k + \sigma(\mathbf{u}_k)\hat{\mathbf{n}} = 0, \quad \mathbf{x} \in \partial\Omega_a, \tag{5.16b}$$

$$\sigma(\mathbf{u}_k)\hat{\mathbf{n}} = 0, \quad \mathbf{x} \in \partial\Omega_r. \tag{5.16c}$$

The Stress tensor in (5.16a) fulfills Hooke's law, $\sigma(\mathbf{u}_k) = \lambda(\mathbf{x})\,(\nabla\cdot\mathbf{u}_k\ I_d) + \mu(\mathbf{x})\big(\nabla\mathbf{u}_k + (\nabla\mathbf{u}_k)^{\mathsf{T}}\big)$, and we consider Sommerfeld radiation boundary conditions on $\partial\Omega_a$ that model absorption, and a free-surface boundary condition on $\partial\Omega_r$ (reflection). A finite element discretization[†] with basis functions that are B-splines [24, Chapter 2] of degree $p \in \mathbb{N}_{>0}$ yields,

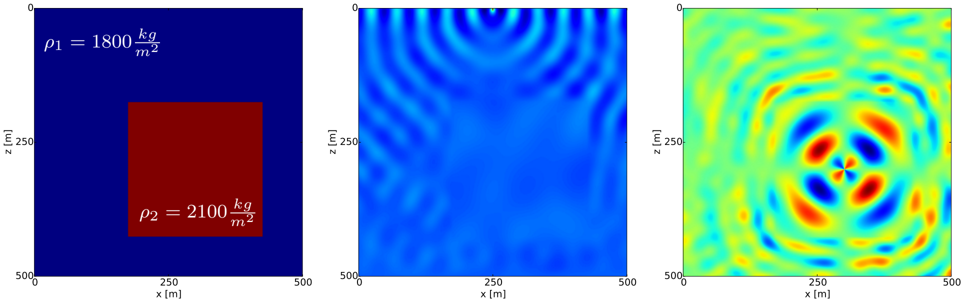
$$(K + i\omega_k C - \omega_k^2 M)\mathbf{u}_k = \mathbf{s}, \quad k = 1, ..., N_\omega, \tag{5.17}$$

where $\mathbf{u}_k$ contains FEM coefficients of the k-th displacement vector, and $\mathbf{s}$ models a time-harmonic source term. In the case of purely reflecting boundary conditions, $\partial\Omega_a = \emptyset$, we obtain $C = 0$; cf. [13].

The inhomogeneous set of parameters $\{\rho, c_p, c_s\}$ is described in Figure 5.1a. In Figure 5.1b, we prescribe material-air boundary conditions at the upper boundary only, and a point source at $(L_x/2, 0)^\mathsf{T}$. When comparing convergence behavior of the matrix equation approach (5.2) with the shifted system re-formulation (5.3), we make use of the identity,

$$\|\mathbf{R}_m\|_\mathsf{F} = \sqrt{\sum_{k=1}^{N_\omega} \left\|\mathbf{r}_m^{(k)}\right\|_2^2}, \qquad \text{for } \mathbf{R}_m := [\mathbf{r}_m^{(1)}, ..., \mathbf{r}_m^{(N_\omega)}] \in \mathbb{C}^{N\times N_\omega},$$

where $\left\{\mathbf{r}_m^{(k)}\right\}_{k=1}^{N_\omega}$ are the columns of $\mathbf{R}_m$ and *not* the residuals of the shifted systems. Since this way the block residual in Frobenius norm naturally is larger than an individual residual norm in 2-norm, we use the maximum 2-norm of the residuals of (5.3) as a *fair* stopping criteria. All numerical examples presented in Section 5.2 have been implemented in `Python-3`, and executed on a computer with 4 CPUs `Intel I5` with 32 GB of RAM.



(a) Density distribution. (b) $\Re(u_3)$ at $f = 16$Hz, $C \neq 0$. (c) $\Re(u_1)$ at $f = 20$Hz, $C \equiv 0$.

Figure 5.1: *Set-up of the 2D numerical experiments: Density distribution (left), and real part of z-component of the displacement at* $f = 16$*Hz (middle) as well as of x-component at* $f = 20$*Hz (right). The speed of pressure waves and shear waves are* $c_p = \{2000, 3000\}\frac{m}{s}$ *and* $c_s = \{800, 1600\}\frac{m}{s}$*, respectively, and the Lamé parameters* $\{\lambda, \mu\}$ *in Hooke's law are calculated accordingly. The case* $C \equiv 0$ *implies purely reflecting boundary conditions, and we apply viscous damping of* 5%*.*

[†] For the finite element discretization of (5.16a)-(5.16c) we use the Python package `nutils` [149].

5.2.1 Convergence Study in the Presence of Viscous Damping

As a first numerical experiment we consider the case when viscous damping is added to (5.17) via the substitution $\omega_k \mapsto (1-\epsilon i)\omega_k$ for $\epsilon > 0$. As we explain in Section 5.1.2, the spectral radius of the polynomial preconditioner (5.9) can be minimized as a result of the optimal seed frequency $\tau^*(\epsilon)$ derived in [9]. Table 5.1 demonstrates that an increase of the polynomial degree n reduces the number of iterations of Algorithm 5.2, cf. [47, 160]. The best CPU time is obtained for $n = 3$ in (5.9).

Table 5.1: *Performance of Algorithm 5.2 for the case $C = 0$ and viscous damping parameter $\epsilon = 0.05$. We consider a fixed frequency range of $N_\omega = 5$ equally-spaced frequencies in $f_k \in [8, 16]$Hz, and $2 \times 200 \times 200$ `dofs`. The seed parameter τ is chosen according to [9].*

$n =$	10	5	4	3	2	1	0
# iterations	12	20	25	29	39	57	106
`CPU time [s]`	`24.20`	`20.77`	`20.66`	`19.84`	`20.27`	`22.51`	`36.87`

Table 5.2 compares the performance of the three algorithms when viscous damping is present, i.e. $\epsilon > 0$; cf. Figure 5.1c. Clearly, the shifted systems approaches outperform the matrix equation approach.

Table 5.2: *Comparison of the three algorithms for the setup described in Table 5.1. The degree of the polynomial preconditioner is fixed at $n = 3$. We report CPU time in seconds and in parenthesis the number of iterations until `tol=1e-8` is reached.*

problem size	frequency range	N_ω	`Gl-GMRES`	`poly-msGMRES`	`FOM-FGMRES`
$2\times200\times200$	$\omega_k \in 2\pi[12, 16]$Hz	5	`29.3` (48)	`12.65` (12)	`12.63` $(7 \cdot 8)$
$2\times200\times200$	$\omega_k \in 2\pi[10, 16]$Hz	5	`46.6` (75)	`15.31` (19)	`16.04` $(12 \cdot 8)$
$2\times200\times200$	$\omega_k \in 2\pi[8, 16]$Hz	5	`79.9` (112)	`19.80` (29)	`19.90` $(17 \cdot 8)$
$2\times200\times200$	$\omega_k \in 2\pi[12, 16]$Hz	15	`64.8` (47)	`15.71` (12)	`13.41` $(7 \cdot 8)$
$2\times200\times200$	$\omega_k \in 2\pi[10, 16]$Hz	15	`115.9` (73)	`18.37` (19)	`16.86` $(12 \cdot 8)$
$2\times200\times200$	$\omega_k \in 2\pi[8, 16]$Hz	15	`198.9` (109)	`22.49` (29)	`20.71` $(17 \cdot 8)$

5.2.2 Suitability for Wide Frequency Ranges

We next consider the undamped problem ($\epsilon = 0$) with Sommerfeld boundary conditions (see Figure 5.1b) which is numerically more challenging. Here, we use $n = 0$ in Algorithm 5.2 because the spectral radius of the polynomial preconditioner is $R/|c| \equiv 1$, cf. [9, 147]. The experiments in Table 5.3 and 5.4 show that the matrix equation approach requires a large number of iterations, especially when the number of frequencies is increased. This is due to the fact that the union of the preconditioned spectra needs to be well approximated by the global GMRES method.

The equivalent, vectorized reformulation of the matrix equation (5.2),

$$\begin{bmatrix} (K + i\omega_1 C - \omega_1^2 M) & & \\ & \ddots & \\ & & (K + i\omega_{N_\omega} C - \omega_{N_\omega}^2 M) \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{N_\omega} \end{pmatrix} = \begin{pmatrix} b \\ \vdots \\ b \end{pmatrix},$$

Table 5.3: *Comparison for undamped case and increased frequency range at a fixed seed parameter* $\tau = (0.7 - 0.3i)\omega_{\max}$*, with* $\omega_{\max} = 2\pi \cdot 8$ *Hz in this table.*

problem size	frequency range	N_ω	`Gl-GMRES`	`poly-msGMRES`	`FOM-FGMRES`
$2\times 100\times 100$	$\omega_k \in 2\pi[7,8]$Hz	5	`14.2` (111)	`9.98` (96)	`5.40` $(20\cdot 8)$
$2\times 100\times 100$	$\omega_k \in 2\pi[4,8]$Hz	5	`16.3` (124)	`10.81` (96)	`5.55` $(20\cdot 8)$
$2\times 100\times 100$	$\omega_k \in 2\pi[1,8]$Hz	5	`29.5` (193)	`12.40` (106)	`8.40` $(20\cdot 11)$
$2\times 100\times 100$	$\omega_k \in 2\pi[7,8]$Hz	15	`42.6` (116)	`11.42` (96)	`5.86` $(20\cdot 8)$
$2\times 100\times 100$	$\omega_k \in 2\pi[4,8]$Hz	15	`50.5` (127)	`11.69` (96)	`6.02` $(20\cdot 8)$
$2\times 100\times 100$	$\omega_k \in 2\pi[1,8]$Hz	15	`148.9` (324)	`13.68` (106)	`8.97` $(20\cdot 11)$

Table 5.4: *Setting as in Table 5.3 using quadratic B-splines* ($p = 2$)*.*

problem size	frequency range	N_ω	`Gl-GMRES`	`poly-msGMRES`	`FOM-FGMRES`
$2\times 100\times 100$	$\omega_k \in 2\pi[7,8]$Hz	15	`86.9` (117)	`18.74` (97)	`13.86` $(20\cdot 8)$
$2\times 100\times 100$	$\omega_k \in 2\pi[4,8]$Hz	15	`98.6` (130)	`19.59` (97)	`13.96` $(20\cdot 8)$
$2\times 100\times 100$	$\omega_k \in 2\pi[1,8]$Hz	15	`267.4` (332)	`28.87` (107)	`18.95` $(20\cdot 11)$

shows that the preconditioner (5.5) acts on the block diagonals which demonstrates that the block Krylov subspace in Algorithm 5.1 needs to approximate the union of the spectra whereas in the shifted systems approach only one space is built due to shift-invariance. This drawback is partly overcome by applying appropriate rotations to the spectrum as we show in detail in [9].

5.2.3 Inexact Solves for the Shift-And-Invert Preconditioner

In Table 5.5 we exploit the use of an inexact LU factorization[†] for the shift-and-invert preconditioner in Algorithm 5.1. Therefore, we extend the test case in Figure 5.1a to 3D by an expansion in y-direction, see Figure 5.2a. The measured CPU times indicate the trade-off between decomposition time and overall number of iterations. In practice, more advanced inexact preconditioners such as multigrid [112, 113] or hierarchical matrix decompositions [3, 13] are used for seismic applications.

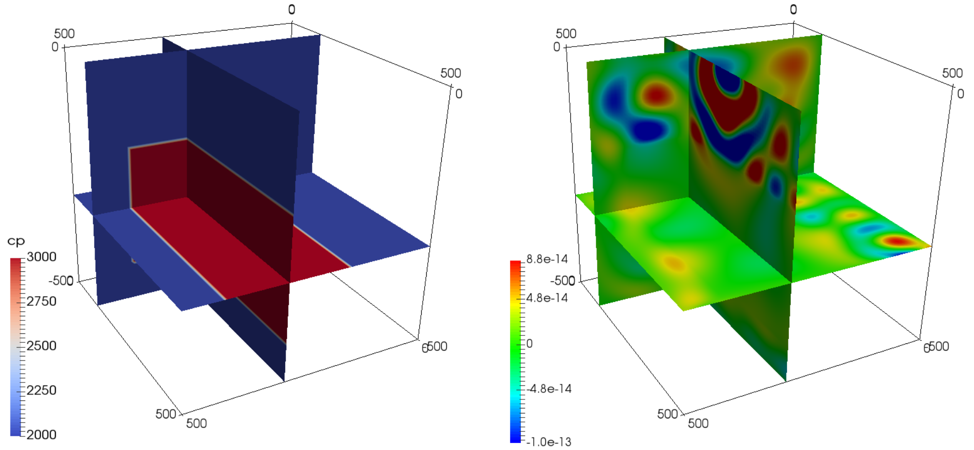
Table 5.5: *Inexact solves for the shift-and-invert preconditioner in Algorithm 5.1. We consider* $N_\omega = 10$ *equally-spaced frequencies with seed parameter* $\tau = (0.7 - 0.3i)\omega_{\max}$*. We use* $\|\mathbf{R}_m\|_{\mathsf{F}} <$ `1e-8` *as stopping criteria.*

problem size	frequency range	preconditioner	`setup time`	`CPU time`	# iter.
$3\times 35\times 35\times 35$	$\omega_k \in 2\pi[1,3]$Hz	exact inverse	`4533.9`	`5396.2`	53
$3\times 35\times 35\times 35$	$\omega_k \in 2\pi[1,3]$Hz	iLU(10.0)	`332.9`	`2852.3`	482
$3\times 35\times 35\times 35$	$\omega_k \in 2\pi[1,3]$Hz	iLU(20.0)	`559.2`	`2179.0`	367
$3\times 35\times 35\times 35$	$\omega_k \in 2\pi[1,3]$Hz	iLU(30.0)	`1061.4`	`2129.8`	197

[†] We use Python's built-in incomplete LU factorization `scipy.sparse.linalg.spilu`.

5.3 Additive Coarse Grid Correction for the 3D Elastic Preconditioner

We turn our attention to the shifted systems approach, i.e. Algorithm 5.2 or Algorithm 5.3-5.4, and consider numerical problems that stem from elasticity problems in three spatial dimensions, cf. Figure 5.2. It has been noted multiple times in this thesis that the shift-and-invert preconditioner (5.6), other than in Section 5.2.3, needs to be applied to full accuracy due to the equivalence used in (5.7). In Section 4.4.3, we have proposed a 3D block-SSOR preconditioner that efficiently solves a sequence of 2D problems. The numerical results of Figure 4.15, however, yield unsatisfying convergence behavior even if the block-SSOR preconditioner is applied at the original frequency. We note that in Algorithm 5.2 and Algorithm 5.3-5.4, the shift-and-invert preconditioner needs to be applied at a seed frequency τ that is chosen based on the spectral analysis performed in Section 3, i.e. according to (3.16), and that contains damping.



(a) Speed of P-waves in m/s.

(b) $\Re(u_1)$ at $f = 8$Hz, $C \neq 0$.

Figure 5.2: *Three-dimensional test case. Left: The underlying parameters* $\{\rho, c_p, c_s\}$ *are chosen according to Figure 5.1a, and have been duplicated in* y*-direction. Right:* x*-component of the displacement vector at* $f = 8$ *Hz.*

Here, we present an extension of the block-SSOR preconditioner by an additive coarse grid correction (CGC). The application of an additive CGC is a standard tool used for Schwarz methods [34], i.e. in the situation where the numerical solution is computed separately on sub-domains and where the CGC yields an additional connectivity between the domains. In [34], an analysis of the CGC is presented based on the ratio h/H between the fine grid associated with grid size h and the coarse grid associated with grid size H. In [117, Section 13.4.2], coarse grid correction is interpreted as a two-grid multigrid method with no smoothing. The observation of [95, 135, 136] that a suitable choice for the deflation matrices Z and $Z^\top$ in (1.22) yields an additive CGC leads immediately to the applicability of additive CGC as a preconditioner. The comparison with multiplicative CGC in [96] concludes similar results for both variants.

We recall the splitting for the damped elastic operator $P(\tau^*)$ (5.5) introduced in Section 4.3.3,

$$K + i\tau^* C - (\tau^*)^2 M =: \underline{\mathrm{L}} + \hat{S} + \overline{U}, \quad \text{with } \hat{S} = \mathrm{blkdiag}(\hat{S}_1, ..., \hat{S}_{n_z}), \tag{5.18}$$

and with τ^* chosen according to (3.16). With appropriate splitting, the block entries $\{\hat{S}_i\}_{i=1}^{n_z}$ yield a sequence of 2D problems that can be solves efficiently using level-2 MSSS techniques as in Section 4.3.2. Based on the splitting (5.18), a block-SSOR preconditioner P_{SSOR} can be defined. If an additive coarse grid correction P_{CGC} is applied, the (damped) preconditioner reads,

$$P_h(\tau^*)^{-1} = P_{\mathrm{SSOR}}^{-1} + P_{\mathrm{CGC}} = \left[(\underline{\mathrm{L}}\hat{S}^{-1} + I)\hat{S}(\hat{S}^{-1}\overline{U} + I)\right]^{-1} + PP_H(\tau^*)^{-1}R, \tag{5.19}$$

with prolongation and restriction matrices, P and R, that realize a mapping between the fine and the coarse grid, respectively. If the discretization size H for the coarse grid is much larger than h for the fine grid, the inverse $P_H(\tau^*)^{-1}$ in (5.19) can be computed efficiently; for instance using a direct method. The coarse operator P_H here stems from a *direct* (finite element) discretization of the damped elastic wave equation on a coarse grid and, thus, replaces the so-called Galerkin operator $(RP_hP)^{-1}$ used in the multigrid literature [18, 140]. A block Jacobi preconditioner has been combined with additive coarse grid correction in [155].

Prolongation and Restriction Operators

The prolongation matrix P and restriction matrix R in (5.19) realize a mapping between the fine and coarse grid. Here, we assume a three-dimensional grid ($d = 3$) and first-order B-splines for the FEM discretization ($p = 1$). Let the fine grid have $N = 3n_xn_yn_z$ grid points and the coarse grid $N^c = 3n_x^cn_y^cn_z^c$, with $N^c \ll N$ and, in general a grid spacing of $H \neq 2h$.

In one spatial direction, the prolongation and restriction operators are rectangular matrices of the dimensions,

$$P_x \in \mathbb{R}^{n_x \times n_x^c} \quad \text{and} \quad R_x \in \mathbb{R}^{n_x^c \times n_x}.$$

In Figure 5.3, we illustrate the definition of both operators in our implementation: Since H_x is not necessarily a multiple of h_x, we suggest for interior grid points to use linear interpolation from nearest neighbors, and so-called *injection* at the boundary. Since the solution of the damped elastic wave equation is smooth, we expect linear interpolation to be sufficient. Using the numbering introduced in Appendix A, the tensor product,

$$P = I_3 \otimes P_x \otimes P_y \otimes P_z \in \mathbb{R}^{N^c \times N},$$
$$R = I_3 \otimes R_x \otimes R_y \otimes R_z \in \mathbb{R}^{N \times N^c},$$

yields prolongation and restriction operators in 3D. This *grid-based* definition of restriction and prolongation is valid when first-order finite elements are considered. We refer to [67] for a more general framework of prolongation and restriction operators mapping between function spaces of higher-order B-splines.

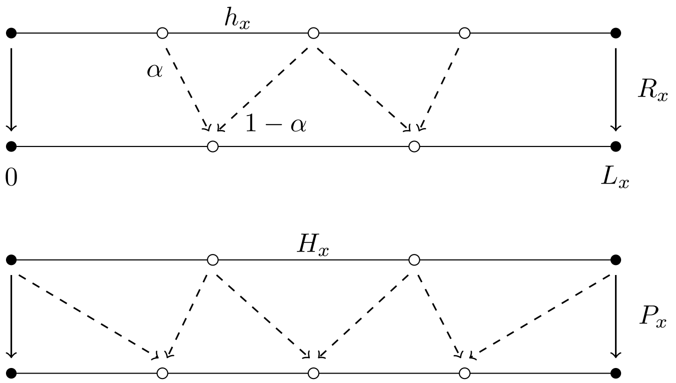


Figure 5.3: *Restriction and prolongation operators for a one-dimensional grid discretizing the interval* $[0, L_x]$*, exemplified for a fine grid of* $n_x = 5$ *and a coarse grid of* $n_x^c = 4$ *grid points. Dashed arrows indicate linear interpolation, and straight arrows visualize injection on the boundary. Note that here* H_x *is not a multiple of* h_x*.*

5.3.1 An SSOR-MSSS Preconditioner with Additive Coarse Grid Correction for Damped 3D Problems

The efficiency of an additive CGC preconditioner is demonstrated by means of numerical experiments performed for the damped three-dimensional elastic problem described in Figure 5.2. In order to have an efficient algorithm for the shifted linear systems approach (using Algorithm 5.2 or Algorithm 5.3-5.4), we require the preconditioner (5.6) to have optimal computational complexity, that is $\mathcal{O}(n^3)$ if $n := n_x = n_y = n_z$. In order to achieve the required accuracy, we apply preconditioner (5.19) to PGMRES. In Figure 5.4, we demonstrate that the additive CGC (5.19) leads to a constant number of PGMRES iterations if the (damped) seed frequency τ^* is chosen according to (3.16). The convergence results in Figure 5.4, moreover, demonstrate grid-independence of the CGC since we double the maximum wave frequency in each experiment which in 3D implies an increase of the (fine) computational grid by a factor of 8.

Table 5.6 contains the CPU times corresponding to Figure 5.4. The additive CGC has been implemented upon our Fortran 90 implementation used in Section 4.4, and a precomputed *LU* decomposition using SuperLU [79] is used for the inversion at the coarse grid level. The experiments in Table 5.6 and Figure 5.4 show a constant number of PGMRES iterations only when an additive CGC is applied. Since we double the number of grid points in every spatial direction, we observe a factor of ~ 8 in CPU time per iteration. A small increase of this optimal complexity is due to the fact that the *LU* factorization of the coarse grid operator needs to be applied. For the largest test case, the storage requirement of the SuperLU implementation exceeds the RAM memory. This can be overcome with a recursive strategy, i.e. a 3-grid cycle.

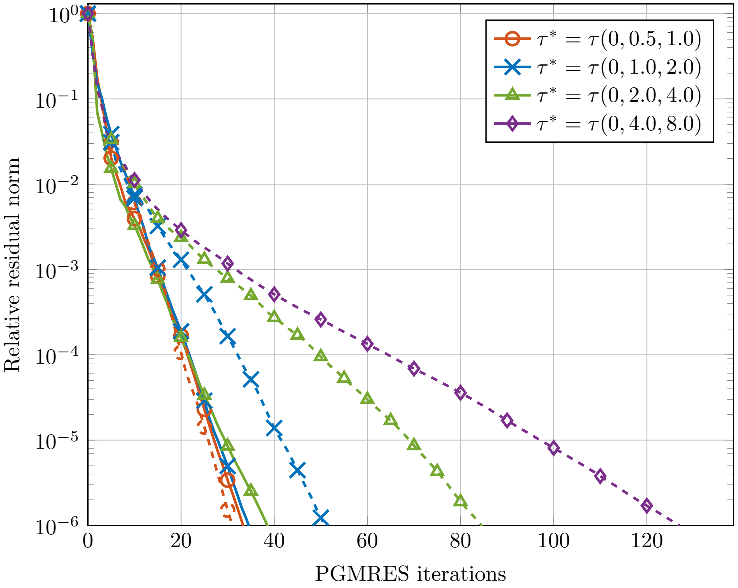


Figure 5.4: *Convergence of preconditioned GMRES applied to the damped 3D problem* $(K + i\tau^* C - (\tau^*)^2 M)$ *with preconditioner* (5.19). *The grid size is increased proportional to the largest frequency* $f_{\max}$*, cf. Table* 5.6. *The 'seed' is chosen as* $\tau^* = \tau(\epsilon, 2\pi f_{\min}, 2\pi f_{\max})$ *according to* (3.16). *We compare convergence with (straight lines) and without (dashed lines) additive coarse grid correction in* (5.19).

Table 5.6: *Iteration numbers and CPU times for PGMRES when the preconditioner* (5.19) *is applied to the damped 3D problem. We report the effect of an additive CGC at the absence of viscous damping, i.e.* $\epsilon = 0$.

$h_x = h_y = h_z$ freq. range	40m $[0.5, 1]$Hz	20m $[1, 2]$Hz	10m $[2, 4]$Hz	5m $[4, 8]$Hz
$n_{\text{dofs}} =$	5,184	46,875	375,000	3,000,000
$\mathcal{P}_H(\tau^*)^{-1}$ at	$H = 100$m	$H = 50$m	$H = 25$m	$H = 12.5$m
P_{SSOR}^{-1}	31 (0.43)	51 (6.2)	85 (87.0)	127 (1145.4)
$P_{\text{SSOR}}^{-1} + P_{\text{CGC}}$	34 (0.50)	35 (4.7)	39 (47.3)	-

Table 5.7: *Performance of PGMRES with SSOR and additive CGC preconditioner* (5.19) *applied to the damped 3D problem with viscous damping* $0 \leq \epsilon \ll 1$.

$h_x = h_y = h_z$ freq. range	40m $[0.5, 1]$Hz	20m $[1, 2]$Hz	10m $[2, 4]$Hz
$\epsilon = 0.5$	31 (0.43)	30 (4.1)	33 (42.5)
$\epsilon = 0.1$	34 (0.47)	35 (4.8)	38 (46.8)
$\epsilon = 0.05$	34 (0.46)	35 (4.7)	38 (46.1)
$\epsilon = 0.0$	34 (0.48)	35 (4.8)	39 (47.3)

We repeat the experiment in the presence of viscous damping, $\epsilon > 0$, for the CGC. Again, the constant iteration numbers in Table 5.7 imply optimal computational complexity for applying the preconditioner in 3D.

5.3.2 Computational Complexity Study of the Overall Shifted Algorithm

We next study the overall computational complexity of the shifted systems (5.3) approach for two-dimensional ($d = 2$) and three-dimensional ($d = 3$) elasticity problems at multiple frequencies. In order to develop an efficient multi-shift Krylov method, the computational costs per iteration are required to be of the order $\mathcal{O}(n^d)$ which is the order of a matrix-vector product. In Section 5.3.1, it has been shown that the shift-and-invert preconditioner (5.6) can be applied at optimal complexity in the 3D case. In order to analyze the performance of the overall multi-shift algorithms, we first focus on the two-dimensional problem described in Figure 5.1. For the shift-and-invert preconditioner, we use an approximate block-LU decomposition based on MSSS matrix computations described in detail in Section 4.3.2. Table 5.8 shows that if the MSSS-preconditioner is applied to a damped problem at seed frequency $\tau^*(\epsilon = 0)$ (3.16), a constant off-diagonal rank (semiseparable order) of $r^* = 5$ is sufficient such that PGMRES convergences within few iterations. In particular, we measure a factor of ~ 4 when the problem size is doubled in both spatial directions which is optimal.

Table 5.8: *Performance of MSSS-preconditioned GMRES as developed in Section 4.3.2 applied to the 2D damped elastic wave equation. The semiseparable order (Definition 4.6) is bounded by $r^* = 5$ in all experiments.*

$h_x = h_z$ freq. range	10m $[2, 4]$Hz	5m $[4, 8]$Hz	2.5m $[8, 16]$Hz	1.25m $[16, 32]$Hz
$\epsilon = 0.5$	4 (0.02)	4 (0.09)	4 (0.37)	4 (1.5)
$\epsilon = 0.1$	5 (0.03)	5 (0.11)	4 (0.36)	6 (2.1)
$\epsilon = 0.05$	5 (0.03)	5 (0.11)	5 (0.45)	6 (2.1)
$\epsilon = 0.0$	5 (0.03)	5 (0.11)	5 (0.45)	6 (2.1)

We next apply preconditioned multi-shift GMRES (Algorithm 5.2 at $n = 0$) to multi-frequency problems in 2D up to a frequency of 32 Hz, cf. Table 5.9. As in the previous experiment, the preconditioner is applied at the (optimal) seed frequency τ^* (3.16). As a consequence of Corollary 3.12, we compare convergence at frequency intervals such that the ratio $\omega_{\max}/\omega_{\min}$ is constant. If the viscous damping parameter ϵ is large, the convergence bound visualized in Figure 3.3 (green line for the tolerances used in Table 5.8) is descriptive, and a constant number of iterations for the different experiments is observed. In Table 5.9, we show that the iteration numbers stay almost constant if $\epsilon > 0$. According to Figure 3.3, an upper bound for the multi-shift GMRES iteration number exists. In CPU time, we observe an increase of a factor ~ 4 due to the increase in mesh size. In the absence of viscous damping, $\epsilon = 0$, the iteration number is proportional to the largest wave frequency [112, 113] and we observe a factor of ~ 8 in CPU time in Table 5.9.

Table 5.9: *Preconditioned multi-shift GMRES (Algorithm 5.2 at* $n = 0$*) applied to* $N_\omega = 10$ *equidistantly spaced frequencies. The shift-and-invert preconditioner is applied as studied in Table 5.8.*

$h_x = h_z$ freq. range	10m $[2, 4]$Hz	5m $[4, 8]$Hz	2.5m $[8, 16]$Hz	1.25m $[16, 32]$Hz
$\epsilon = 0.5$	10 (0.62)	11 (2.8)	12 (12.8)	12 (49.4)
$\epsilon = 0.1$	20 (0.90)	28 (5.8)	33 (31.8)	37 (160.7)
$\epsilon = 0.05$	24 (1.02)	40 (8.0)	53 (47.8)	64 (264.9)
$\epsilon = 0.0$	31 (1.22)	77 (14.0)	163 (134.6)	358 (1391.2)

(Multi-shift) GMRES is a long-recurrence Krylov method, and the costs per iteration grow due to Arnoldi's orthogonalization method. Moreover, the storage requirement is proportional to the iteration number. A more memory-efficient multi-shift Krylov method is, therefore, the nested approach [8] presentend in Algorithm 5.3-5.4. In Table 5.10, we show that in the case of $\epsilon = 0$ the same complexity $\mathcal{O}(n^3)$ can be obtained using short recurrences due to the inner-outer scheme.

Table 5.10: *Comparison of multi-shift GMRES (Algorithm 5.2) with nested multi-shift FOM-FGMRES (Algorithm 5.3-5.4) for the test case without viscous damping,* $\epsilon = 0$*, in Table 5.9. The iteration numbers for the nested algorithm are reported as* $m_o \times m_i$*, where the inner method (Algorithm 5.3) is stopped if either the residual norm drops below* `0.1`*, or the maximum iteration number* m_i *is reached.*

$h_x = h_z$ freq. range	10m $[2, 4]$Hz	5m $[4, 8]$Hz	2.5m $[8, 16]$Hz	1.25m $[16, 32]$Hz
msGMRES	31 (1.22)	77 (14.0)	163 (134.6)	358 (1391.2)
FOM-FGMRES	3×15 (2.1)	6×15 (19.0)	7×30 (179.7)	16×30 (1835.1)

Consider the extension of the previous test case to three-dimensions in Figure 5.2. We have demonstrated in Section 5.3.1 that the shift-and-invert preconditioner can be applied at optimal computational complexity if an additive coarse grid correction is applied. In Table 5.11, we repeat the numerical experiment for multi-shift GMRES for a three-dimensional test problem, and observe a similar behavior: In the presence of viscous damping the iteration number is (almost) constant and we obtain an algorithm in $\mathcal{O}(n^3)$. If $\epsilon = 0$, the iteration number doubles and we obtain $\mathcal{O}(n^4)$ complexity. In particular, this complexity is favourable to what has been observed in Section 4.4.3 when a block-SSOR preconditioner has been applied *directly* to the wave frequency of the original single-shift problem. In Appendix C, the complexity study is repeated for the 3D wedge problem described in Figure 1.4 .

Conclusions

We have compared three GMRES-based algorithms for the simultaneous iterative solution of frequency-domain wave propagation problems at multiple frequencies that have the discretized form (5.1). The three approaches share that they require the

Table 5.11: *Multi-shift GMRES with a block-SSOR and additive CGC preconditioner* (5.19) *applied to the 3D problem shown in Figure 5.2 at* $N_\omega = 10$ *equally spaced frequencies.*

$h_x = h_y = h_z$ freq. range	40m $[0.5, 1]$Hz	20m $[1, 2]$Hz	10m $[2, 4]$Hz
$\epsilon = 0.5$	6 (9.2)	8 (103.7)	9 (1201.2)
$\epsilon = 0.1$	7 (10.5)	11 (140.6)	16 (2020.8)
$\epsilon = 0.05$	8 (11.2)	12 (148.0)	20 (2376.3)
$\epsilon = 0.0$	8 (11.2)	14 (163.4)	26 (2889.4)

application of a single shift-and-invert preconditioner at a so-called *seed* frequency. From our numerical experiments we draw the following conclusions:

- In the presence of viscous damping (experiments in Subsection 5.2.1) the optimal seed parameter derived in [9] implies a polynomial preconditioner which, depending on the degree n of the polynomial, leads to a significant reduction of the number of multi-shift GMRES iterations (Algorithm 5.2). Without viscous damping, however, the spectral radius of the polynomial preconditioner equals one and no improvement has been observed.

- The matrix equation approach (Algorithm 5.1) builds up a block Krylov subspace that needs to approximate the union of all preconditioned spectra. This leads to a much larger number of overall iterations, and a worse performance compared with the shifted systems approach when a wide range of frequencies is considered. Because of the less restrictive framework, however, the shift-and-invert preconditioner (5.5) can be applied inexactly which leads to improvements especially for 3D problems (experiments in Subsection 5.2.3). Moreover, the benefits of efficient block matrix-vector products when multiple sources are considered is demonstrated in [13].

- For a wide frequency range (experiments in Subsection 5.2.2) we observe that the nested algorithm 5.3-5.4 outperforms the considered alternatives with respect to measured CPU time. This is due to shorter loops in the respective Arnoldi iterations. From the summary in Table 5.12 we note that the storage requirements for the flexible outer Krylov method can be limited when m_o is small compared to m_i.

- In the numerical experiments in Subsection 5.3.1 and Subsection 5.3.2, the computational complexity of the multi-shift approach with a shift-and-invert preconditioner in two and three spatial dimensions has been studied. Therefore, multiple frequencies ω_k in the intervals $\mathcal{I}_i = 2\pi[2^i, 2^{i+1}]$Hz, $i = 0, 1, 2, ...$, have been considered. The largest frequency induces a computational grid with, hence, $n \sim 2^i/h$ grid points in each spatial direction. The parameter of the shift-and-invert preconditioner $\tau_i^*(\epsilon)$ is chosen based on the spectral analysis in Chapter 3. Due to the damping in $\tau_i^*(\epsilon)$ according to (3.16), the 2D and 3D preconditioner can be applied in optimal complexitiy, i.e. in $\mathcal{O}(n^d)$. The convergence study of preconditioned multi-shift GMRES and preconditioned nested multi-shift FOM-FGMRES has shown computational complexity $\mathcal{O}(n^d)$ if the viscous damping parameter $\epsilon > 0$, and $\mathcal{O}(n^{d+1})$ if $\epsilon = 0$, where $d \in \{2, 3\}$ is the problem dimension. Both are independent of the number of frequencies within $\mathcal{I}_i$ due to shift-invariance in both Krylov methods.

Table 5.12: *Comparison regarding memory requirements and costs-per-iteration when* (5.1) *has fixed problem size N, and N_ω distinct frequencies are present. Note that in all three algorithms, a single* `MatVec` *also requires a solve for the shift-and-invert preconditioner.*

Algorithm	leading memory requirement	# `MatVec`'s
`Gl-GMRES(m)`	$N \cdot N_\omega \cdot m$ for $\mathbf{V}_m$ (in Alg. 5.1, line 10)	$N_\omega \cdot m$
`poly-msGMRES(m,n)`	$2N \cdot m$ for V_m (in Alg. 5.2, line 10)	$(n+1) \cdot m$
`FOM(`$\mathtt{m_i}$`)-FGMRES(`$\mathtt{m_o}$`)`	$2N \cdot N_\omega \cdot m_o$ for $Z_{m_o}^{(k)}$ (in Alg. 5.4, line 11)	$m_i \cdot m_o$

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

In this dissertation, a number of new numerical methods for the efficient solution of multi-frequency wave propagation problems have been suggested. The development includes prototyping of iterative Krylov methods, spectral analysis and convergence studies, and practical implementation aspects such as memory-efficient low-rank techniques for efficiently applying the preconditioner. Many of our key findings have been demonstrated with numerical experiments towards the end of the respective thesis chapters, and in particular in chapter 5 we present detailed evaluations of our implementations. We conclude with an overview of our scientific findings.

Matrix Equation vs. Shifted Systems

A key component of this thesis work has been the efficient iterative solution of wave propagation problems in the situation when multiple linear problems arising from different angular wave frequencies ω_k are solved. Therefore, a reformulation to shifted linear systems (as stated in Problem 1.6) and a reformulation as a matrix equation (as stated in Problem 1.8) have been followed to some extend *in parallel*. The following conclusions are drawn based on theoretical and practical aspects presented in this thesis.

Applicability to a wide frequency range. We have compared both approaches for a fixed frequency range $\omega_k \in [\omega_{\min}, \omega_{\max}]$ in Section 5.2.2. When the frequency interval is increased, numerical experiments have shown a faster convergence behavior for multi-shift methods compared to block Krylov methods. Here, the linearization in Problem 1.6 yields shifts that are proportional to the angular frequencies which is favorable compared to the squared frequencies that appear in the matrix equation approach. This is moreover proven by Figure 3.2 (right) where the GMRES convergence bound grows as the ration $\omega_{\max}/\omega_{\min}$ increases.

Inexact solves of the shift-and-invert preconditioner. It has been pointed out in Section 5.1 that the shift-and-invert preconditioner in the shifted systems approach needs to be applied to full accuracy while the matrix equation framework allows inexact solves. The computational benefit of an inexact LU decomposition for the shift-and-invert preconditioner is demonstrated in Section 5.2.3. On the other hand, we want to point out that the seed shift τ^* according to (3.16) contains damping and can, therefore, be applied at optimal computational complexity. Moreover, the spectral analysis of Section 3.3 applies only in case of an exact shift-and-invert preconditioner. This implies, for instance, that the bounding circles do not hold in case of inexact solves, and the second-level rotation preconditioner for the matrix equation as demonstrated in Figure 3.4 becomes more complicated.

Memory requirements and convergence behavior. In Table 5.12, we summarize the leading memory requirements for GMRES-type implementations of the proposed algorithms. The advantage of the inner-outer multi-shift approach becomes evident especially if the number of outer iterations is small compared to the iterations of global GMRES applied to the matrix equation approach.

The Role of the Single Shift-And-Invert Preconditioner

The second focus of this work lies on the efficient implementation of a (single) preconditioner that is suitable for a sequence of elasticity problems that stem from a wide range of frequencies. This task consists of two main aspects:

- Firstly, the shift-and-invert preconditioner needs to be applied at a seed frequency τ that works well for angular wave frequencies ω_k from within a range $[\omega_{\min}, \omega_{\max}]$. In chapter 3, this parameter is optimized based on a GMRES convergence bound for the preconditioned spectrum. We have furthermore shown that, in the presence of viscous damping, this bound depends on the ratio of the frequency interval boundaries.

- Secondly, we address the fast application of the shift-and-invert preconditioner at the seed value τ^* which is optimal in the sense of (3.16) in two and three spatial dimensions. For two-dimensional problems, we developed an inexact block LU preconditioner based on multilevel sequentially semiseparable matrix computations. Numerical experiments in Section 4.3.2 and Section 5.3.2 show that the damped problem can be inverted efficiently at a very low off-diagonal rank. For three-dimensional problems, we suggest a block SSOR preconditioner with an additive coarse grid correction. Again, numerical experiments for the damped problem show optimal computational complexity when applying the 3D preconditioner, cf. Section 5.6.

The approaches in [6, 119] suggest the application of multiple shift-and-invert preconditioners. In the present setting, however, where the number of wave frequencies is relatively small, we conclude that it is most efficient to apply a single preconditioner.

Overall Computational Complexity

Efficient multi-shift Krylov methods and the fast application of the shift-and-invert preconditioner at an optimal seed frequency are the two main building blocks of the developed algorithms. Numerical experiments have shown that the damped shift-and-invert preconditioner can be applied at optimal computational complexity in 2D and

3D. Together with the consequences of Lemma 3.8 and Corollary 3.12, the following conclusion for the multi-shift algorithm can be drawn.

Consider a set of angular frequencies $\{\omega_k\}_{k=1}^{N_\omega}$ within the intervals $\mathcal{I}_i := 2\pi[2^i, 2^{i+1}]\text{Hz}$, $i = 0, 1, 2, ...$, i.e. the situation where $\omega_{\max}/\omega_{\min} = \text{const.} \equiv 2$. For a constant number of points per wavelength this implies an increasing computational grid of $n \sim 2^i/h$ points in each spatial direction. Let $\tau_i^*(\epsilon, \omega_{\min}, \omega_{\max})$ be chosen *optimally* according to Lemma 3.8. The numerical experiments in Chapter 5 show that multi-shift Krylov methods with a single shift-and-invert preconditioner applied at τ_i^* yield computational complexity of $\mathcal{O}(n^d)$ if $\epsilon > 0$, and of $\mathcal{O}(n^{d+1})$ if $\epsilon = 0$; where $d \in \{2, 3\}$ is the problem dimension. In particular, this complexity is independent of the number of frequencies N_ω within $\mathcal{I}_i$.

An iteration number proportional to the frequency has been observed many times in the single-frequency case for Helmholtz [112] and elasticity problems [113]. The main contribution of our work is, therefore, the fact that problems from multiple frequencies can be solved simultaneously with the same complexity.

Time vs. Frequency Domain Approach

In Remark 1.4, we have stated that the computational complexity of an explicit time-domain approach is $n_t\mathcal{O}(n^d)$ and of an iterative frequency-domain approach is $n_{it}\mathcal{O}(n^d)$, and that both approaches have shown $\mathcal{O}(n^{d+1})$ complexity in practice since the iteration number n_{it} and the number of time steps n_t are both proportional to the largest frequency. Together with the conclusions of the previous section, this generally shows that both approaches are computationally comparable for solving the elastic forward problem which has been the focus of this thesis. A complete conclusion is, therefore, only possible if also modeling aspects and the performance of the inverse problem, i.e. the quality of the objective function for the full-waveform inversion (1.35)-(1.36), are taken into account.

6.2 Recommendations and Open Questions

Scientific research begins and ends with open questions. The research on seismic wave simulations in the frequency-domain addressed in this thesis is, therefore, neither complete nor finished. We present some possible future research directions and, moreover, give some recommendations to closely related scientific areas from a computational mathematics point-of-view.

Model-Order Reduction

The key problem considered in this thesis is to solve the multi-frequency problem $(K + i\omega_k C - \omega_k^2 M)\mathbf{x}_k = \mathbf{b}$ at a given set of frequencies $\omega_k \in [\omega_{\min}, \omega_{\max}]$ where k is from a known index set, i.e. $k \in \mathcal{I}$. Suppose there is a rectangular matrix U such that the approximation, $\mathbf{x}_\ell \approx U\tilde{\mathbf{x}}_\ell, \ell \notin \mathcal{I}$, holds where the numerical solution $\mathbf{x}_\ell$ is associated with frequency ω_ℓ within the same interval, but $\ell \notin \mathcal{I}$. We here assume that $\tilde{\mathbf{x}}_\ell$ is of much smaller dimension compared to $\mathbf{x}_\ell$. If a POD-type approach [154] is followed, the columns of the matrix U are the leading left singular vectors of a *snapshot matrix* of the form $[\mathbf{x}(\omega_1), ..., \mathbf{x}(\omega_{N_\omega})]$. This requires the solution at multiple frequencies during the so-called *offline phase*. The reduced-order model is then given by the projected systems,

$$(\tilde{K} + i\omega_\ell \tilde{C} - \omega_\ell^2 \tilde{M})\tilde{\mathbf{x}}_\ell = \tilde{\mathbf{b}}, \quad \ell \notin \mathcal{I},\ \omega_\ell \in [\omega_{\min}, \omega_{\max}], \tag{6.1}$$

with matrices of reduced dimensions,

$$\tilde{K} := U^{\mathsf{H}} K U, \quad \tilde{C} := U^{\mathsf{H}} C U, \quad \tilde{M} := U^{\mathsf{H}} M U \quad \text{and} \quad \tilde{\mathbf{b}} := U^{\mathsf{H}} \mathbf{b}.$$

Note that the structure of (6.1) is unchanged compared to the discrete full-order model. The reduced-order approach is in particular interesting because seismic measurements are often only available on the surface (upper boundary) of the computational domain and, therefore, the objective function (1.35) in a full-waveform approach requires the displacement only at certain points. For the reduced model this implies that only $\mathbf{y}_\ell := \Psi \mathbf{x}_\ell$ has to be approximated well by $\tilde{\mathbf{y}}_\ell := \Psi U \tilde{\mathbf{x}}_\ell$, i.e. $\|\mathbf{y}_\ell - \tilde{\mathbf{y}}_\ell\| \to \min$, where the matrix Ψ here maps the state variable to the system's output. A key open question in this approach would be to investigate which set of frequencies are relevant for the offline phase in order to derive a reduced-order model (6.1) of satisfying quality.

A similar approach has been suggested in the context of quadratic eigenvalue problems by [89]. Efficient reduced-order models for wave propagation problems at multiple frequencies are subject to current research [32, 162].

Subspace Recycling

In order to design an efficient Krylov method for multi-frequency wave propagation problems, we rely on a simultaneous solution process. This is reflected by the shift-invariance property for Problem 1.6 and by the block approach for Problem 1.8. An alternative approach for the solution of a sequence of shifted systems is the sequential solution of some systems and then to recycle information from these solves in order to improve the solution process of the remaining systems.

The approach in [133] follows a recycling strategy for restarted multi-shift GMRES based on an augmented search space using harmonic Ritz vectors. It is clear that shifted linear systems share the same eigenspace. To what extend approximate eigenvectors (Ritz vectors) can be used for elasticity problems at different wave frequencies requires further research. When the recycled information corresponds to solution vectors of previous solves, a preconditioner similar to the POD-based deflation approach of [23] can be of interest.

Absorbing Boundary Conditions

It is beyond the scope of this thesis to evaluate the quality of the Sommerfeld radiation conditions for absorbing boundaries. The derivations presented in Section 1.1.1 are based on characteristics of the outwards traveling waves and leads to a Robin boundary condition that includes the first temporal derivative of the displacement. In frequency-domain, such a boundary condition yields a term proportional to the angular wave frequency ω. In contrast, an absorbing boundary condition in terms of the acceleration $\ddot{\mathbf{u}}$ or the displacement $\mathbf{u}$ would result in frequency-domain in a term of the form, $C(\omega) = C_1 - \omega^2 C_2$, that could be combined in a straightforward way with the stiffness and mass matrix to,

$$K + C(\omega) - \omega^2 M = (K + C_1) - \omega^2 (M + C_2).$$

From a computational point-of-view, this has the advantage that the reformulation approaches in Section 1.1.4 can be simplified, and, for instance, no doubling of dimensions in the shifted systems approach (Problem 1.6) is necessary.

In [31], a frequency-*in*dependent perfectly matched layer (PML) boundary condition is derived for electromagnetic wave propagation. To our knowledge, the extension to the elastic case is still an open task. When the interval $[\omega_{\min}, \omega_{\max}]$ is split equidistantly on a logarithmic scale according to Corollary 3.12, a separate PML can be prescribed for each subinterval. The quality of a frequency-independent PML in this framework yields another open question subject to future research.

Geometric Multigrid

In Section 5.3, we present the effect of an additive coarse grid correction applied to the damped elastic operator for large three-dimensional problems. Numerical experiments have shown that a grid-independent convergence can be achieved if the dimension of the coarse grid is chosen appropriately; in other words, the additive coarse grid correction yields a constant number of preconditioned GMRES iterations. This implies optimal computational complexity in term of the fine grid dimension. The coarse grid correction, however, can be seen as a two-grid method and requires in our current framework a direct solve of the coarse operator. In order to make this solve more feasible, a V-cycle, i.e. a multigrid method [18, 140], with several levels of coarsening can be investigated. Moreover, an approximate solution of the coarse grid operator can be considered and its effect on the (outer) GMRES convergence behavior requires further analysis.

Geometric multigrid has been analyzed theoretically for the acoustic wave equation. Fourier analysis for the Helmholtz equation at high frequencies [73] as well as for the damped Complex Shifted Laplacian [45] has shown the importance of damping for the multigrid solution of the preconditioner. A local-mode analysis for the two-dimensional elastic wave equation with damping is performed in [113]. A strict proof for the three-dimensional case as well as an efficient multigrid implementation for damped 3D elastic problems remains an open task.

High-Performance Computing

In Corollary 3.12 and the corresponding numerical experiment 3.19 we address the question of distributing subintervals from a given frequency range to multiple parallel processors. Our proposed distribution is load-balanced in such a way that an equal number of Krylov iterations is expected in the sense of the GMRES convergence bound in Corollary 3.7. Many other important aspects of high-performance computing are not addressed in this thesis. The parallelization approach for iterative methods applied to elastic problems in [80] concludes a potential improvement in computational complexity compared to the time-domain approach. A parallel implementation of the multi-shift conjugate gradient method is presented in [51] but has not been applied to multi-frequency wave propagation problems. Concerning the low-rank approach described in Section 4.3 we note that the hierarchically semiseparable matrix format [21] can be an attractive alternative for parallel programming.

Appendix A

Spline-based Finite Element Discretization for the Vector-Valued Elastic Wave Equation

A finite element method [36, 39, 56] gives some freedom with respect to the choice of basis functions for the trial and test spaces. In this section we give some implementation details concerning the finite element discretization described in Section 1.1.2 and in more detail in Section 4.1.2. We use B-spline basis functions of degree $p \in \mathbb{N}^+$ according to the Cox-de Boor recursion formula [24]. Consider, first, the one-dimensional case ($d = 1$), i.e. $\Omega = [0, L_x] = [0, x_1] \cup [x_1, x_2] \cup ... \cup [x_{n_x - 1}, x_{n_x}]$, with $x_{n_x} \equiv L_x$.

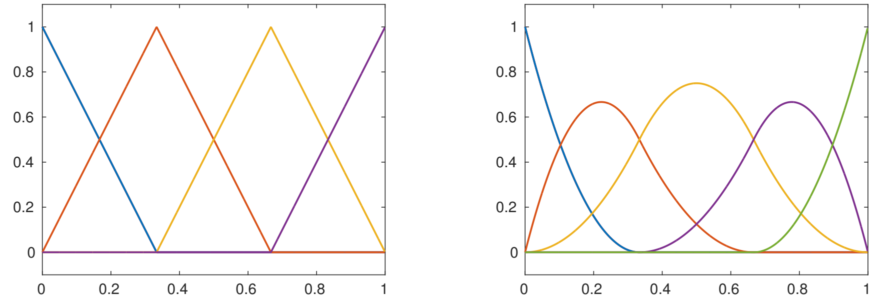


Figure A.1: *B-spline basis functions of degree* $p = 1$ *(left) and* $p = 2$ *(right) on the one-dimensional unit interval* $[0, 1]$.

B-splines of degree p are then defined recursively via,

$$\phi_i^p(x) = \frac{x - x_i}{x_{i+p} - x_i}\phi_i^{p-1}(x) + \frac{x_{i+p+1} - x}{x_{i+p+1} - x_{i+1}}\phi_{i+1}^{p-1}(x), \quad p = 1, 2, ..., \tag{A.1}$$

with piecewise constant functions ϕ_i^0 equal to one within $[x_i, x_{i+1}]$, and zero otherwise. In Figure A.1, we plot the set of basis functions ϕ_i^p, $i = 1, ..., (n_x - 1 + p)$, for $p = 1$ and $p = 2$ when $n_x = 4$. Note that for $p = 1$ the knots x_i in (A.1) coincide with grid points, cf. [24] for more details.

For $d = \{2, 3\}$, the elastic wave equation is vector-valued and we make the ansatz (4.7) for the displacement vector,

$$\mathbf{u}_k(\mathbf{x}) \approx \sum_{i=1}^{n_{\text{dofs}}} u_k^i \boldsymbol{\varphi}_{\mathbf{i}}(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad u_k^i \in \mathbb{C} \quad \text{and} \quad \boldsymbol{\varphi}_{\mathbf{i}} : \mathbb{R}^d \to \mathbb{R}^d. \tag{A.2}$$

The basis functions $\boldsymbol{\varphi}_{\mathbf{i}}$ are defined via the tensor expansion,

$$\mathbb{R}^d \ni \boldsymbol{\varphi}_{\mathbf{i}}(x, y, z) := \big(\phi_i(x)\xi_j(y)\eta_k(z)\big)\mathbf{e}_\ell, \quad \ell \in \{1, ..., d\}, \tag{A.3}$$

where $\mathbf{e}_\ell$ is the ℓ-th unit vector in $\mathbb{R}^d$. For $d = 2$ or $d = 3$, the set of functions ξ_j and η_k are defined according to (A.1) along the y-axis and z-axis, respectively. We exemplify the index numbering in (A.3) for $p = 1$,

$$\begin{aligned} \mathbf{i} &:= (n_x n_y n_z)(\ell - 1) + (n_y n_z)i + n_z j + k, \quad \text{with} \\ 0 &\le i \le n_x - 1, \quad 0 \le j \le n_y - 1, \quad 0 \le k \le n_z - 1. \end{aligned}$$

This lexicographic numbering is used in the finite element software Nutils [149] used for our numerical experiments, and renumbering yields a discretization matrix of smaller bandwidth, cf. Sections 4.3.2 and 4.3.3.

If we define $n := n_x n_y n_z$, we obtain $n_{\text{dofs}} = dn$ degrees of freedom for linear B-spline basis functions, cf. (4.8) for the general case when $p > 1$. The spy pattern for the stiffness matrix derived from the weak form (1.15) at different dimensions d and different spline basis degrees p is shown in Figure A.2. The matrix size equals the number of degrees of freedom in (A.2).

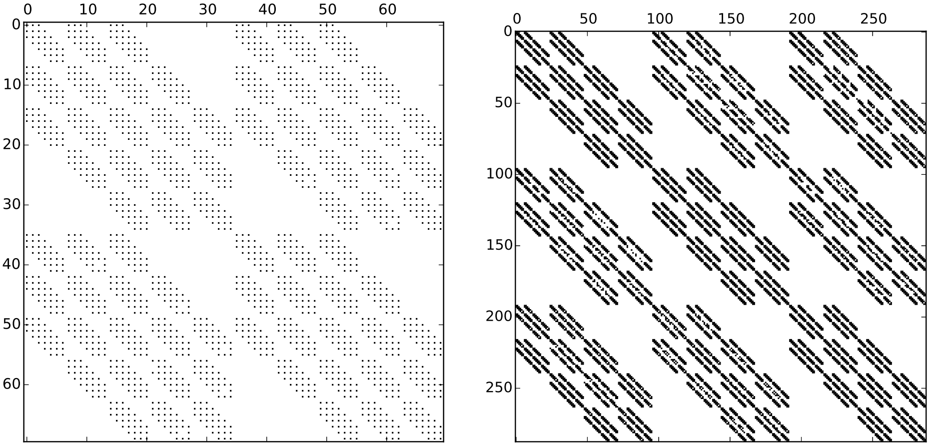


Figure A.2: *Spy pattern of the stiffness matrix K (4.10) for $d = 2, p = 2$ (left) and $d = 3, p = 1$ (right). Lexicographic ordering of the unknown yields a $d \times d$ block matrix with non-zeros resulting from p neighbors in each spatial direction.*

Appendix B

Inversion of an SSS Matrix Corresponding to a 1D Discretization

This appendix serves two purposes: We illustrate two basic SSS matrix operations used at 1D level by means of an example computation. At the same time, we complete Algorithm 4.2. For simplicity, we consider the case $n = 4$ in Definition 4.5,

$$A = \begin{bmatrix} D_1 & U_1V_2^{\mathsf{H}} & U_1W_2V_3^{\mathsf{H}} & U_1W_2W_3V_4^{\mathsf{H}} \\ P_2Q_1^{\mathsf{H}} & D_2 & U_2V_3^{\mathsf{H}} & U_2W_3V_4^{\mathsf{H}} \\ P_3R_2Q_1^{\mathsf{H}} & P_3Q_2^{\mathsf{H}} & D_3 & U_3V_4^{\mathsf{H}} \\ P_4R_3R_2Q_1^{\mathsf{H}} & P_4R_3Q_2^{\mathsf{H}} & P_4Q_3^{\mathsf{H}} & D_4 \end{bmatrix},$$

and refer to standard literature [19, 110, 150] for the more general case.

Inversion of a lower/upper diagonal SSS matrix

A lower diagonal SSS matrix in generator form is given by

$$L = SSS(P_s, R_s, Q_s, D_s, 0, 0, 0), \quad 1 \leq s \leq n, \tag{B.1}$$

and we denote L^{-1} via,

$$L^{-1} = SSS(\underline{P}_s, \underline{R}_s, \underline{Q}_s, \underline{D}_s, 0, 0, 0), \quad 1 \leq s \leq n.$$

Clearly, for $n = 4$, the matrix (B.1) yields,

$$L = \begin{bmatrix} D_1 & 0 & 0 & 0 \\ P_2Q_1^{\mathsf{H}} & D_2 & 0 & 0 \\ P_3R_2Q_1^{\mathsf{H}} & P_3Q_2^{\mathsf{H}} & D_3 & 0 \\ P_4R_3R_2Q_1^{\mathsf{H}} & P_4R_3Q_2^{\mathsf{H}} & P_4Q_3^{\mathsf{H}} & D_4 \end{bmatrix},$$

and we immediately conclude $\underline{D}_s = D_s^{-1}, s = 1, ..., 4$, for all diagonal generators of L^{-1}. In Lemma 4.7, we claim that L^{-1} can be computed without increase of the off-diagonal rank, and we illustrate this fact by computing the generators at entry $(2, 1)$:

$$P_2Q_1^{\mathsf{H}}\underline{D}_1 + D_2\underline{P}_2\underline{Q}_1^{\mathsf{H}} = 0 \quad \Leftrightarrow \quad \underline{P}_2\underline{Q}_1^{\mathsf{H}} \equiv (-D_2^{-1}P_2)(D_1^{-\mathsf{H}}Q_1)^{\mathsf{H}}.$$

The computation of U^{-1} in Algorithm 4.2 can be done analogously, and we refer to [19, Lemma 2] for the complete algorithm and the case $n \neq 4$.

Matrix-matrix multiplication in SSS structure

In the final step of Algorithm 4.2 we perform, based on an LU factorization, the matrix-matrix multiplication $A^{-1} = U^{-1} \cdot L^{-1}$ with U^{-1} and L^{-1} given in upper/lower diagonal SSS format (see previous appendix section). Here, we illustrate how to perform the SSS matrix-matrix multiplication $C = A \cdot B$ when $n = 4$ and A and B are given as,

$$\begin{bmatrix} D_1^A & U_1 V_2^{\mathsf{H}} & U_1 W_2 V_3^{\mathsf{H}} & U_1 W_2 W_3 V_4^{\mathsf{H}} \\ 0 & D_2^A & U_2 V_3^{\mathsf{H}} & U_2 W_3 V_4^{\mathsf{H}} \\ 0 & 0 & D_3^A & U_3 V_4^{\mathsf{H}} \\ 0 & 0 & 0 & D_4^A \end{bmatrix}, \begin{bmatrix} D_1^B & 0 & 0 & 0 \\ P_2 Q_1^{\mathsf{H}} & D_2^B & 0 & 0 \\ P_3 R_2 Q_1^{\mathsf{H}} & P_3 Q_2^{\mathsf{H}} & D_3^B & 0 \\ P_4 R_3 R_2 Q_1^{\mathsf{H}} & P_4 R_3 Q_2^{\mathsf{H}} & P_4 Q_3^{\mathsf{H}} & D_4^B \end{bmatrix}.$$

The SSS matrix C can then be computed by appropriate block multiplications of the respective generators. For example, the $(3, 2)$ entry of the product yields,

$$\begin{aligned} C_{32} &= 0 \cdot D_2^B + D_3^A P_3 Q_2^{\mathsf{H}} + U_3 V_4^{\mathsf{H}} P_4 R_3 Q_2^{\mathsf{H}} \\ &= (D_3^A P_3 + U_3 V_4^{\mathsf{H}} P_4 R_3) Q_2^{\mathsf{H}} \equiv P_3^C (Q_2^C)^{\mathsf{H}} \end{aligned}$$

The above computation illustrates on the one hand that the off-diagonal rank (of C_{32}) does not increase due to the lower/upper diagonal SSS structure of the matrices A and B. On the other hand, we note that in general the off-diagonal rank of C will increase due to the non-vanishing term that contains the full-rank generator D_2^B. Matrix-matrix multiplication in SSS form is presented in [19, Theorem 1].

Appendix C

Numerical Results for the Elastic Wedge Problem in Three Dimensions

We perform the numerical tests presented in Section 5.3.1 and Section 5.3.2 for the 3D elastic wedge problem as described in Figure 1.4. The problem sizes considered are tabulated in Table 1.1, and the computational domain for this test case is $\Omega = [0, 600] \times [0, 600] \times [0, 1000] \subset \mathbb{R}^3$, i.e. larger than in the test case considered in Chapter 5.

Again, we first demonstrate the performance gain in CPU time of an additive coarse grid correction (CGC) applied to the (damped) preconditioner. In more detail, we apply preconditioned GMRES to the damped elastic problem $K + i\tau^* C - (\tau^*)^2 M$, with τ^* chosen as a function of the considered frequency range, and the viscous damping parameter ϵ, cf. Section 3.2. The preconditioner is,

$$\mathcal{P}_h(\tau^*) = P_{\text{SSOR}}^{-1} + \alpha \cdot P\mathcal{P}_H(\tau^*)^{-1}R, \quad \text{with } \alpha \in \{0, 1\}, \tag{C.1}$$

with second term being the additive coarse grid correction, $P_{\text{CGC}} := P\mathcal{P}_H(\tau^*)^{-1}R$, as described in detail in Section 5.3. For large 3D problems, the block-SSOR term resembles a sequence of n_z 2D problems that are solved efficiently based on MSSS techniques described in Section 4.3.2. In Table C.1, we report the results of adding a coarse grid correction to PGMRES (compare Table 5.6 in Section 5.3.1). The

Table C.1: *Iteration numbers and CPU times for PGMRES when the preconditioner* (5.19) *is applied to the damped 3D problem. We report the effect of an additive CGC at the absence of viscous damping, i.e.* $\epsilon = 0$.

$h_x = h_y = h_z$ freq. range	40m $[0.5, 1]$Hz	20m $[1, 2]$Hz	10m $[2, 4]$Hz
$n_{\text{dofs}} =$	$16,875$	$135,000$	$1,080,000$
$\mathcal{P}_H(\tau^*)^{-1}$ at	$H = 100$m	$H = 50$m	$H = 25$m
P_{SSOR}^{-1}	38 (1.58)	65 (22.7)	106 (318.6)
$P_{\text{SSOR}}^{-1} + P_{\text{CGC}}$	39 (1.76)	39 (15.8)	42 (162.9)

offline costs for computing the *LU* factorization[†] of $\mathcal{P}_H(\tau^*)$ are not reported as the factors are computed *once* in the overall multi-shift algorithm. However, the costs for applying the coarse grid operator increase which is the reason for a factor slightly higher than 8 recorded in CPU time between the different experiments, even when a constant PGMRES iteration number in the presence of the CGC yields optimal complexity in terms of n_{dofs}. When $\epsilon > 0$, similar results have been obtained, see also Table 5.7.

Next, the shift-and-invert preconditioner with additive CGC (case $\alpha = 1$ in (C.1)) is applied within multi-shift GMRES. In the same way as in Table 5.11, we see a constant iteration number in the presence of viscous damping, and an iteration number proportional to the largest frequency in the case of $\epsilon = 0$. Again, the factor in CPU timings is suboptimal due to the increasing work for applying the coarse operator, and a multigrid approach needs to be applied when the size of the coarse grid increases.

Table C.2: *Multi-shift GMRES with a block-SSOR and additive CGC preconditioner* (5.19) *applied to the 3D* wedge *problem shown in Figure 1.4 at* $N_\omega = 10$ *equally spaced frequencies.*

$h_x = h_y = h_z$ freq. range	40m $[0.5, 1]$Hz	20m $[1, 2]$Hz	10m $[2, 4]$Hz
$\epsilon = 0.5$	7 (23.4)	8 (272.4)	9 (3,192.6)
$\epsilon = 0.1$	9 (30.6)	14 (469.2)	20 (7,491.1)
$\epsilon = 0.05$	10 (33.1)	17 (551.8)	24 (8,894.3)
$\epsilon = 0.0$	10 (31.0)	21 (577.5)	44 (13,802.3)

[†] As in Section 5.3, we use SuperLU [79] on the coarse level.

Bibliography

[1] M. I. Ahmad, D. B. Szyld, and M. B. van Gijzen. Preconditioned multishift BiCG for $\mathcal{H}_2$-optimal model reduction. *SIAM. J. Matrix Anal. Appl.*, 38:401–424, 2017.

[2] T. Airaksinen, A. Pennanen, and J. Toivanen. A damping preconditioner for time-harmonic wave equations in fluid and elastic material. *J. Comput. Phys.*, 228(5): 1466–1479, 2009.

[3] P. Amestoy, C. Ashcraft, O. Boiteau, A. Buttari, J.-Y. L'Excellent, and C. Weisbecker. Improving Multifrontal Methods by Means of Block Low-Rank Representations. *SIAM J. Sci. Comput.*, 37:A1451–A1474, 2015.

[4] P. Amestoy, R. Brossier, A. Buttari, J.-Y. L'Excellent, T. Mary, L. Métivier, A. Miniussi, and S. Operto. Fast 3D frequency-domain full-waveform inversion with a parallel block low-rank multifrontal direct solver: Application to OBC data from the North Sea. *Geophysics*, 81(6):R363–R383, 2016.

[5] R. Astudillo and M. B. van Gijzen. Induced Dimension Reduction Method for Solving Linear Matrix Equations. *Procedia Comput. Sci.*, 80:222–232, 2016. ISSN 1877-0509.

[6] T. Bakhos, P. K. Kitanidis, S. Ladenheim, A. K. Saibaba, and D. B. Szyld. Multipreconditioned GMRES for shifted systems. *SIAM J. Sci. Comput.*, 39:222–247, 2017.

[7] H. T. Banks, S. Hu, and Z. R. Kenz. A Brief Review of Elasticity and Viscoelasticity for Solids. *Adv. Appl. Math. Mech.*, 3(01):1–51, 2011.

[8] M. Baumann and M. B. van Gijzen. Nested Krylov methods for shifted linear systems. *SIAM J. Sci. Comput.*, 37(5):S90–S112, 2015.

[9] M. Baumann and M. B. van Gijzen. An Efficient Two-Level Preconditioner for Multi-Frequency Wave Propagation Problems. Technical Report 17-03, Delft University of Technology, 2017.

[10] M. Baumann and M. B. van Gijzen. Efficient iterative methods for multi-frequency wave propagation problems: A comparison study. *Procedia Comput. Sci.*, 108:645–654, 2017.

[11] M. Baumann, J. Heiland, and M. Schmidt. Discrete Input/Output Maps and their Relation to Proper Orthogonal Decomposition. In Peter Benner, Matthias Bollhöfer, Daniel Kressner, Christian Mehl, and Tatjana Stykel, editors, *Numerical Algebra,*

Matrix Theory, Differential-Algebraic Equations and Control Theory, pages 585–608. Springer International Publishing, 2015.

[12] M. Baumann, P. Benner, and J. Heiland. Space-time Galerkin POD with application in optimal control of semi-linear parabolic partial differential equations. Technical report, arXiv:1611.04050, 2016.

[13] M. Baumann, R. Astudillo, Y. Qiu, E. Y. M. Ang, M. B. van Gijzen, and R.-É. Plessix. An MSSS-preconditioned matrix equation approach for the time-harmonic elastic wave equation at multiple frequencies. *Comput. Geosci.*, 2017. doi: 10.1007/s10596-017-9667-7.

[14] E. Bécache, S. Fauqueux, and P. Joly. Stability of perfectly matched layers, group velocities and anisotropic waves. *J. Comput. Phys.*, 188(2):399–433, 2003.

[15] M. Benzi and D. Bertaccini. Approximate inverse preconditioning for shifted linear systems. *BIT Numerical Mathematics*, 43:231–244, 2003.

[16] M. Berljafa and S. Güttel. Generalized rational Krylov decompositions with an application to rational approximation. *SIAM J. Matrix Anal. Appl.*, 36(2):894–916, 2015.

[17] M. Berljafa and S. Güttel. The RKFIT algorithm for nonlinear rational approximation. MIMS EPrint 2015.38, Manchester Institute for Mathematical Sciences, The University of Manchester, 2015.

[18] W. L. Briggs, V. Emden Henson, and S. F. McCormick. *A Multigrid Tutorial, Second Edition.* Society for Industrial and Applied Mathematics, 2000.

[19] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A.-J. van der Veen. Fast Stable Solvers for Sequentially Semi-separable Linear Systems of Equations. Technical report, Lawrence Livermore National Laboratory, 2003.

[20] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.J. van der Veen, and D. White. Some fast algorithms for sequentially semiseparable representations. *SIAM J. Matrix Anal. Appl.*, 27(2):341–364, 2005.

[21] S. Chandrasekaran, P. Dewilde, M. Gu, W. Lyons, and T. Pals. A fast solver for HSS representations via sparse matrices. *SIAM J. Matrix Anal. Appl.*, 29(1):67–81, 2007.

[22] 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 J. Sci. Comput.*, 39(2): A438–A478, 2017.

[23] G. B. Diaz Cortes, C. Vuik, and J. D. Jansen. On POD-based deflation vectors for DPCG applied to porous media problems. *J. Comput. Appl. Math.*, 2017. doi: 10.1016/j.cam.2017.06.032.

[24] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis. Towards integration of CAD and FEA.* John Wiley & Son, Ltd., 2009.

[25] D. Darnell, R. B. Morgan, and W. Wilcox. Deflated GMRES for systems with multiple shifts and multiple right-hand sides. *Linear Algebra Appl.*, 429(10):2415–2434, 2008.

[26] T. A. Davis. *Direct Methods for Sparse Linear Systems.* SIAM, Philadelphia, 2006.

[27] J. De Basabe. *High-Order Finite Element Methods for Seismic Wave Propagation.* PhD thesis, The University of Texas at Austin, 2009.

[28] A.T. De Hoop. *Handbook of Radiation and Scattering of Waves.* Academic Press, London, United Kingdom, 1995.

[29] E. de Sturler. Truncation Strategies for Optimal Krylov Subspace Methods. *SIAM J. Numer. Anal.*, 36(3):864–889, 1999.

[30] P. Dewilde and A.J. Van der Veen. *Time-varying systems and computations.* Kluwer Academic Publishers, Boston, 1998.

[31] V. Druskin and R. Remis. A Krylov Stability-Corrected Coordinate-stretching method to simulate wave propagation in unbounded domains. *SIAM J. Sci. Comput.*, 35(2): B376–B400, 2013.

[32] V. Druskin, R. Remis, and M. Zaslavsky. An extended Krylov subspace model-order reduction technique to simulate wave propagation in unbounded domains. *J. Comput. Phys.*, 272:608–618, 2014.

[33] L. Du, T. Sogabe, and S.-L. Zhang. IDR(s) for solving shifted nonsymmetric linear systems. *Int. J. Comput. Appl. Math.*, 274:35–43, 2014.

[34] O. Dubois, M. J. Gander, S. Loisel, A. St-Cyr, and D. B. Szyld. The Optimized Schwarz Method with a Coarse Grid Correction. *SIAM J. Sci. Comput.*, 34(1):A421–A458, 2012.

[35] Y. Eidelman and I. Gohberg. On generators of quasiseparable finite block matrices. *Calcolo*, 42(3):187–214, 2005.

[36] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics.* Numerical Mathematics and Scientific Computation. Oxford University Press, 2014.

[37] B. Engquist and A. Majda. Absorbing Boundary Conditions for the Numerical Simulation of Waves. *Math. Comp.*, 31:629–651, 1977.

[38] B. Engquist and A. Majda. Radiation boundary conditions for acoustic and elastic wave calculations. *Comm. Pure Appl. Math.*, 32(3):313–357, 1979.

[39] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations.* Cambridge University Press, 2nd edition, 1996.

[40] Y. A. Erlangga. *A robust and efficient iterative method for the numerical solution of the Helmholtz equation.* PhD thesis, Delft University of Technology, 2005.

[41] Y. A. Erlangga and R. Nabben. On a multilevel Krylov mehtod for the Helmholtz equation preconditioned by shifted Laplacian. *Electron. Trans. Numer. Anal.*, 31: 403–424, 2008.

[42] Y. A. Erlangga, C. W. Oosterlee, and C. Vuik. A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.*, 27:1471–1492, 2006.

[43] Y. A. Erlangga, L. Garcia Ramos, and R. Nabben. The Multilevel Krylov-Multigrid Method for the Helmholtz Equation Preconditioned by the Shifted Laplacian. In D. Lahaye, J. Tang, and C. Vuik, editors, *Modern Solvers for Helmholtz Problems*, pages 113–139. Springer International Publishing, 2017.

[44] Y.A. Erlangga, C. Vuik, and C.W. Oosterlee. On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math.*, 50(3-4):409–425, 2004.

[45] Y.A. Erlangga, C. Vuik, and C.W. Oosterlee. Comparison of multigrid and incomplete LU shifted-Laplace preconditioners for the inhomogeneous Helmholtz equation. *Appl. Numer. Math.*, 56:648–666, 2006.

[46] V. Etienne, E. Chaljub, J. Virieux, and N. Glinsky. An hp-adaptive discontinuous Galerkin finite-element method for 3-D elastic wave modelling. *Geophys. J. Int.*, 183 (2):941–962, 2010.

[47] B. Fischer and R.W. Freund. On adaptive weighted polynomial preconditioning for Hermitian positive definite matrices. *SIAM J. Sci. Comput.*, 15(2):408–426, 1994.

[48] R. Freund. Solution of shifted linear systems by quasi-minimal residual iterations. In R. S. Varga L. Reichel, A. Ruttan, editor, *Numerical linear algebra: Proceedings of the Conference in Numerical Linear Algebra and Scientific Computation*, pages 101–121, 1993.

[49] A. Frommer. BiCGStab(ℓ) for Families of Shifted Linear Systems. *Computing*, 70: 87–109, 2003.

[50] A. Frommer and U. Glässner. Restarted GMRES for Shifted Linear Systems. *SIAM J. Sci. Comput.*, 19(1):15–26, 1998.

[51] R. Galvez and G. van Anders. Accelerating the solution of families of shifted linear systems with CUDA. Technical report, arXiv:1102.2143, 2011.

[52] M. J. Gander and H. Zhang. Iterative Solvers for the Helmholtz Equation: Factorizations, Sweeping Preconditioners, Source Transfer, Single Layer Potentials, Polarized Traces, and Optimized Schwarz Methods. Technical report, arXiv:1610.02270, 2017.

[53] 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? *Numer. Math.*, 131(3):567–614,, 2015.

[54] A. Gaul. *Recycling Krylov subspace methods for sequences of linear systems.* PhD thesis, Technical University of Berlin, 2014.

[55] M.G.D. Geers. Fundamentals of deformation and linear elasticity. Eindhoven University of Technology. Lecture notes - course 4A450, 2001.

[56] M. S. Gockenbach. *Understanding And Implementing the Finite Element Method.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[57] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 3rd edition, 1996.

[58] R. W. Graves. Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences. *Bull. Seismol. Soc. Amer.*, 86(4):1091–1106, 1996.

[59] G.-D. Gu, X.L. Zhou, and L. Lin. A flexible preconditioned Arnoldi method for shifted linear systems. *J. Comput. Math.*, 25:522–530, 2007.

[60] M.H. Gutknecht and J.-P.M. Zemke. Eigenvalue computations based on IDR. *SIAM J. Matrix Anal. Appl.*, 34(2):283–311, 2013.

[61] G.Wu, H. Pang, and J. Sun. Preconditioning the Restarted and Shifted Block FOM Algorithm for Matrix Exponential Computation. Technical report, arXiv:1405.0707, 2014.

[62] I. Harari and U. Albocher. Studies of FE/PML for exterior problems of time-harmonic elastic waves. *Comput. Methods Appl. Mech. Engrg.*, 195:3854–3879, 2006.

[63] J. Heiland, M. Baumann, A. Walle, V. Mehrmann, and M. Schäfer. Simulation and Control of Drop Size Distributions in Stirred Liquid/Liquid Systems. In *Proc. 4th International Conference on Population Balance Modelling*, 2010.

[64] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49(6):409, 1952.

[65] R. L. Higdon. Absorbing boundary conditions for elastic waves. *Geophysics*, 56:231–241, 1991.

[66] N. J. Higham, D. S. Mackey, F. Tisseur, and S. D. Garvey. Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems. *Internat. J. Numer. Methods Engrg.*, 73(3):344–360, 2008.

[67] C. Hofreither, B. Jüttler, G. Kiss, and W. Zulehner. Multigrid methods for isogeometric analysis with THB-splines. *Comp. Meth. Appl. Mech. Engrg.*, 308:96–112, 2016.

[68] K. Jbilou, A. Messaoudi, and H. Sadok. Global FOM and GMRES algorithms for matrix equations. *Appl. Numer. Math.*, 31:49–63, 1999.

[69] B. Jegerlehner. Krylov space solvers for shifted linear systems. Technical Report IUHET-353, Indiana University, 1996. hep-lat/9612014.

[70] A. Kavcic and J.M.F. Moura. Matrices with banded inverses: inversion algorithms and factorization of Gauss-Markov processes. *IEEE Trans. Inform. Theory*, 46(4): 1495–1509, 2000. ISSN 0018-9448.

[71] B. Kennett. *Seismic Wave Propagation in Stratified Media.* ANU E Press. The Australian National University, 2013.

[72] M.E. Kilmer and E. de Sturler. Recycling subspace information for diffuse optical tomography. *SIAM J. Sci. Comput.*, 27:2140–2166, 2006.

[73] S. Kim and S. Kim. Multigrid Simulation for High-Frequency Solutions of the Helmholtz Problem in Heterogeneous Media. *SIAM J. Sci. Comput.*, 24(2):684–701, 2002.

[74] S. Kirchner. IDR-Verfahren zur Lösung von Familien geshifteter linearer Gleichungssysteme. Master's thesis, Bergische Universität Wuppertal, 2011 (in German).

[75] H. Knibbe, C. Vuik, and C. W. Oosterlee. Reduction of computing time for least-squares migration based on the Helmholtz equation by graphics processing units. *Comput. Geosci.*, 20(2):297–315, 2016. ISSN 573-1499.

[76] H.P. Knibbe. *Reduction of computing time for seismic applications based on the Helmholtz equation by Graphics Processing Units.* PhD thesis, Delft University of Technology, 2015.

[77] D. Lahaye and C. Vuik. How to Choose the Shift in the Shifted Laplace Preconditioner for the Helmholtz Equation Combined with Deflation. In D. Lahaye, J. Tang, and C. Vuik, editors, *Modern Solvers for Helmholtz Problems*, pages 85–112. Springer International Publishing, 2017.

[78] D. Lahaye, J. Tang, and C. Vuik, editors. *Modern Solvers for Helmholtz Problems*, volume 1. Birkhäuser Basel, 2017.

[79] X. S. Li. An overview of SuperLU: algorithms, implementation, and user interface. *ACM Trans. Math. Software*, 31:302–325, 2005.

[80] Y. Li, L. Métivier, R. Brossier, B. Han, and J. Virieux. 2D and 3D frequency-domain elastic wave modeling in complex media with a parallel iterative solver. *Geophysics*, 80(3):T101–T118, 2015.

[81] J. Liesen and Z. Strakos. *Krylov Subspace Methods: Principles and Analysis.* Numerical Mathematics and Scientific Computation. OUP Oxford, 2012.

[82] E. Liu, Z. Zhang, J. Yue, and A. Dobson. Boundary Integral Modelling of Elastic Wave Propagation in Multi-Layered 2D Media with Irregular Interfaces. *Commun. Comput. Phys.*, 3:52–62, 2008.

[83] G. D. Manolis and P. S. Dineva. Elastic waves in continuous and discontinuous geological media by boundary integral equation methods: A review. *Soil Dyn. Earthq. Eng.*, 70:11 – 29, 2015.

[84] T. A. Manteuffel and S. V. Parter. Preconditioning and Boundary Conditions. *SIAM J. Numer. Anal.*, 27(3):656–694, 1990.

[85] G. S. Martin, K. J Marfurt, and S. Larsen. Marmousi-2: an updated model for the investigation of AVO in structurally complex areas. In *72nd Annual International Meeting, SEG, Expanded Abstract, 1979–1982*, 2002.

[86] Z. Martinec. Continuum mechanics. Charles University in Prague, Department of Geophysics. Lecture notes, 2011.

[87] K. Meerbergen. The solution of parametrized symmetric linear systems. *SIAM J. Matrix Anal. Appl.*, 24:1038–1059, 2003.

[88] K. Meerbergen and Z. Bai. The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides. *SIAM J. Matrix Anal. Appl.*, 4:1642–1662, 2010.

[89] V. Mehrmann and C. Schröder. Nonlinear eigenvalue and frequency response problems in industrial practice. *J. Math. Ind.*, 1(1):7, 2011.

[90] J. Meng, P. Zhu, and H. Li. QMRCGstab Algorithm for Families of Shifted Systems. In *Ninth International Conference on Computational Intelligence and Security*, 2013.

[91] L. Métivier, R. Bossier, S. Operto, and J. Virieux. Full Waveform Inversion and the Truncated Newton Method. *SIAM Review*, 59:153–195, 2017.

[92] S. Mönkölä, E. Heikkola, A. Pennanen, and T. Rossi. Time-harmonic elasticity with controllability and higher-order discretization methods. *J. Comput. Phys.*, 227(11): 5513–5534, 2008.

[93] R. B. Morgan. Implicitly Restarted GMRES and Arnoldi Methods for Nonsymmetric Systems of Equations. *SIAM J. Matrix Anal. Appl.*, 21(4):1112–1135, 2000.

[94] W. A. Mulder and R.-É. Plessix. How to choose a subset of frequencies in frequency-domain finite-difference migration. *Geophys. J. Int.*, 158:801–812, 2004.

[95] R. Nabben and C. Vuik. A comparison of Deflation and Coarse Grid Correction applied to porous media flow. *SIAM J. Numer. Anal.*, 42:1631–1647, 2004.

[96] Y. Notay and A. Napov. Further comparison of additive and multiplicative coarse grid correction. *Appl. Numer. Math.*, 65:53–62, 2013.

[97] S. Operto, J. Virieux, P. Amestoy, J.-Y. L'Excellent, L. Giraud, and H. Ben Hadj Ali. 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study. *Geophysics*, 72(5):SM195–SM211, 2007.

[98] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. Recycling Krylov Subspaces for Sequences of Linear Systems. *SIAM J. Sci. Comput.*, 28(5):1651–1674, 2006.

[99] L. F. Pavarino and E. Zampieri. Overlapping Schwarz and Spectral Element Methods for Linear Elasticity and Elastic Waves. *J. Sci. Comput.*, 27(1-3):51–73, 2006.

[100] P. V. Petrov and G. A. Newman. Three-dimensional inverse modelling of damped elastic wave propagation in the Fourier domain. *Geophys. J. Int.*, 198:1599–1617, 2014.

[101] R.-É. Plessix. A Helmholtz iterative solver for 3D seismic-imaging problems. *Geophysics*, 72(5):SM185–SM194, Sep 2007. ISSN 1942-2156.

[102] R.-É. Plessix. Three-dimensional frequency-domain full-waveform inversion with an iterative solver. *Geophysics*, 74:WCC149–WCC157, 2009.

[103] R.-É. Plessix. Some Computational Aspects of the Time and Frequency Domain Formulations of Seismic Waveform Inversion. In D. Lahaye, J. Tang, and C. Vuik, editors, *Modern Solvers for Helmholtz Problems*, pages 159–187. Springer International Publishing, 2017.

[104] R.-É. Plessix and W.A. Mulder. Separation-of-variables as a preconditioner for an iterative Helmholtz solver. *Appl. Numer. Math.*, 44:383–400, 2003.

[105] R.-É. Plessix and C. A. Pérez Solano. Modified surface boundary conditions for elastic waveform inversion of low-frequency wide-angle active land seismic data. *Geophys. J. Int.*, 201(3):1324–1334, 2015.

[106] R.G. Pratt. Seismic waveform inversion in the frequency domain, Part 1: Theory and verification in a physical scale mode. *Geophysics*, 64(3):888–901, 1999.

[107] Y. Qiu, M. B. van Gijzen, J.-W. van Wingerden, M. Verhaegen, and C. Vuik. Efficient Preconditioners for PDE-Constrained Optimization Problems with a Multilevel Sequentially SemiSeparable Matrix Structure. *Electron. Trans. Numer. Anal.*, 44:367–400, 2015.

[108] Y. Qiu, M. B. van Gijzen, J.-W. van Wingerden, M. Verhaegen, and C. Vuik. Evaluation of multilevel sequentially semiseparable preconditioners on CFD benchmark problems using incompressible flow and iterative solver software. *Math. Methods Appl. Sci.*, 38, 2015.

[109] A. Quarteroni, A. Tagliani, and E. Zampieri. Generalized galerkin approximations of elastic waves with absorbing boundary conditions. *Comput. Methods Appl. Mech. Engrg.*, 163(1-4):323–341, 1998.

[110] J. Rice and M. Verhaegen. Distributed Control: A Sequentially Semi-Separable Approach for Spatially Heterogeneous Linear Systems. *IEEE Trans. Automat. Control*, 54(6):1270–1283, 2009.

[111] J. K. Rice. *Efficient Algorithms for Distributed Control: a Structured Matrix Approach.* PhD thesis, Delft University of Technology, 2010.

[112] C.D. Riyanti, Y.A. Erlangga, R.-E. Plessix, W.A. Mulder, C. Vuik, and C. Osterlee. A new iterative solver for the time-harmonic wave equation. *Geophysics*, 71(5):E57–E63, 2006.

[113] G. Rizzuti and W.A. Mulder. Multigrid-based 'shifted-Laplacian' preconditioning for the time-harmonic elastic wave equation. *J. Comput. Phys.*, 317:47–65, 2016.

[114] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra Appl.*, 58:391–405, 1984.

[115] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14:461–469, 1993.

[116] Y. Saad. SPARSEKIT: a basic tool kit for sparse matrix computations. Technical report, University of Minnesota, Minneapolis, 1994.

[117] Y. Saad. *Iterative Methods for Sparse Linear Systems: Second Edition.* Society for Industrial and Applied Mathematics, 2003.

[118] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7(3):856–869, 1986.

[119] A. Saibaba, T. Bakhos, and P. Kitanidis. A flexible Krylov solver for shifted systems with application to oscillatory hydraulic tomography. *SIAM J. Sci. Comput.*, 35:3001–3023, 2013.

[120] A. H. Sheikh. *Development of the Helmholtz Solver based on a Shifted Laplace Preconditioner and a Multigrid Deflation technique.* PhD thesis, Delft University of Technology, 2014.

[121] A. H. Sheikh, D. Lahaye, and C. Vuik. A scalable Helmholtz solver combining the shifted Laplace preconditioner with multigrid deflation. Technical report, DIAM report 11-01, 2011.

[122] A. H. Sheikh, D. Lahaye, and C. Vuik. On the convergence of shifted Laplace preconditioner combined with multilevel deflation. *Numer. Linear Algebra Appl.*, 20(4): 645–662, 2013.

[123] A. H. Sheikh, D. Lahaye, L. Garcia Ramos, R. Nabben, and C. Vuik. Accelerating the shifted Laplace preconditioner for the Helmholtz equation by multilevel deflation. *J. Comput. Phys.*, 322:473–490, 2016.

[124] V. Simoncini. Restarted full orthogonalization method for shifted linear systems. *BIT Numerical Mathematics*, 43:459–466, 2003.

[125] V. Simoncini and F. Perotti. On the numerical solution of $(\lambda^2 A + \lambda B + C)x = b$ and application to structural dynamics. *SIAM J. Sci. Comput.*, 23:1875:1897, 2002.

[126] V. Simoncini and D. Szyld. Flexible inner-outer Krylov subspace methods. *SIAM J. Numer. Anal.*, 40:2219–2239, 2003.

[127] V. Simoncini and D. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numer. Linear Algebra Appl.*, 14:1–59, 2007.

[128] G. L. G. Sleijpen and H. A. van der Vorst. Maintaining convergence properties of BiCGstab methods in finite precision arithmetic. *Numer. Algorithms*, 10:203–223, 1995.

[129] G. L. G. Sleijpen, P. Sonneveld, and M. B. van Gijzen. Bi-CGSTAB as an induced dimension reduction method. *Appl. Numer. Math.*, 60:1100–1114, 2010.

[130] 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 J. Sci. Comput.*, 31 (2):1035:1062, 2008.

[131] K. M. Soodhalter. Two recursive GMRES-type methods for shifted linear systems with general preconditioning. *Electron. Trans. Numer. Anal.*, 45:499–523, 2016.

[132] K. M. Soodhalter. Block Krylov Subspace Recycling for Shifted Systems with Unrelated Right-Hand Sides. *SIAM J. Sci. Comput.*, 38(1):A302–A324, 2016.

[133] K. M. Soodhalter, D. B. Szyld, and F. Xue. Krylov subspace recycling for sequences of shifted linear systems. *Appl. Numer. Math.*, 81C:105–118, 2014.

[134] J. M. Tang. *Two-Level Preconditioned Conjugate Gradient Methods.* PhD thesis, Delft University of Technology, 2008.

[135] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga. Comparison of Two-Level Preconditioners Derived from Deflation, Domain Decomposition and Multigrid Methods. *J. Sci. Comput.*, 39(3):340–370, 2009.

[136] J. M. Tang, S. P. MacLachlan, R. Nabben, and C. Vuik. A Comparison of Two-Level Preconditioners Based on Multigrid and Deflation. *SIAM J. Matrix Anal. Appl.*, 31 (4):1715–1739, 2010.

[137] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM review*, 43 (2):235–286, 2001.

[138] L.N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM: Society for Industrial and Applied Mathematics, 1997.

[139] F. Tröltzsch. *Optimal Control of Partial Differential Equations - Theory, Methods and Applications.* Graduate Studies in Mathematics, Vol. 112. American Mathematical Society, 2010.

[140] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid.* Academic Press, 2001. ISBN 9780127010700.

[141] P. Tsuji, J. Poulson, B. Engquist, and L. Ying. Sweeping preconditioners for elastic wave propagation with spectral element methods. *ESAIM Math. Model. Num.*, 48(2): 433–447, 2014.

[142] J. van den Eshof and G. L. G. Sleijpen. Accurate Conjugate Gradient Methods for Families of Shifted Systems. *Appl. Numer. Math.*, 49(1):17–37, 2004.

[143] J. van den Eshof, G. L. G. Sleijpen, and M. B. van Gijzen. Relaxation strategies for nested Krylov methods. *Int. J. Comput. Appl. Math.*, 177:347–365, 2005.

[144] H. A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 13(2):631–644, 1992.

[145] M. B. van Gijzen. A polynomial preconditioner for the GMRES algorithm. *J. Comput. Appl. Math.*, 59(1):91 –107, 1995.

[146] M. B. van Gijzen and P. Sonneveld. Algorithm 913: An Elegant IDR(s) Variant that Efficiently Exploits Bi-orthogonality Properties. *ACM Trans. Math. Software*, 38(1): 5:1–5:19, 2011.

[147] M. B. van Gijzen, Y. A. Erlangga, and C. Vuik. Spectral Analysis of the Discrete Helmholtz Operator Preconditioned with a Shifted Laplacian. *SIAM J. Sci. Comput.*, 29(5):1942–1958, 2007.

[148] M. B. van Gijzen, G. L. G. Sleijpen, and J.-P. M. Zemke. Flexible and multi-shift Induced Dimension Reduction algorithm for solving large sparse linear systems. *Numer. Linear Algebra Appl.*, 1:1–25, 2014.

[149] G. van Zwieten, C. Verhoosel, J. van Zwieten, T. van Opstal, and W. Hoitinga. Nutils v2.0. Zenodo, February 2016. URL `http://doi.org/10.5281/zenodo.822381`.

[150] R. Vandebril, M. Van Barel, and N. Mastronardi. *Matrix computations and semiseparable matrices: linear systems*. Johns Hopkins University Press, Baltimore, 2007.

[151] T. Vdovina, S. E. Minkoff, and S. M.L. Griffith. A Two-Scale Solution Algorithm for the Elastic Wave Equation. *SIAM J. Sci. Comput.*, 31(5):3356–3386, 2009.

[152] J. Virieux. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51(4):889–901, 1986.

[153] J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 73(6):VE135–VE144, 2009.

[154] S. Volkwein. *Model reduction using proper orthogonal decomposition*. Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz, 2011.

[155] C. Vuik and J. Frank. A Parallel Block Preconditioner Accelerated by Coarse Grid Correction. In M. Bubak, H. Afsarmanesh, B. Hertzberger, and R. Williams, editors, *High Performance Computing and Networking: 8th International Conference, HPCN Europe 2000 Amsterdam*, pages 99–108. Springer Berlin Heidelberg, 2000.

[156] C. Vuik and H. van der Vorst. GMRESR: a Family of Nested GMRES Methods. *Numer. Linear Algebra Appl.*, 1(4):369–386, 1994.

[157] S. Wang, M. V. de Hoop, J. Xia, and X. Li. Massively parallel structured multifrontal solver for time-harmonic elastic waves in 3-D anisotropic media. *Geophys. J. Int.*, 191 (1):346–366, 2012.

[158] A. J. Wathen. Preconditioning. *Acta Numerica*, 24:329–376, 2015.

[159] D. Werner. *Funktionalanalysis*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007.

[160] G. Wu, Y.-C. Wang, and X.-Q. Jin. A preconditioned and shifted GMRES algorithm for the Pagerank problem with multiple damping factors. *SIAM J. Sci. Comput.*, 34: 2558–2575, 2012.

[161] J. Xia. Efficient structured multifrontal factorization for general large sparse matrices. *SIAM J. Sci. Comput.*, 35(2):A832–A860, 2013.

[162] M.Y. Zaslavsky, V.L. Druskin, and R.F. Remis. Model reduction approaches for solution of wave equations for multiple frequencies. In *Proceedings of the 76th EAGE Conference & Exhibition*, 2014.

Software availability

An interactive visualization using Bokeh[†] demonstrates the findings of Lemma 3.8. The visualization is purely browser-based and can be obtained from:

`http://www.manuelbaumann.de/opt_tau`

Most of the presented numerical experiments are available from the author's github account `https://github.com/ManuelMBaumann`. In particular, the following repositories correspond to a chapter of this thesis:

Chapter 2	`https://bitbucket.org/ManuelMBaumann/nestedkrylov`
Chapter 3	`https://github.com/ManuelMBaumann/opttau`
Chapter 4	`https://github.com/ManuelMBaumann/elastic_benchmarks`
Chapter 5	`https://github.com/ManuelMBaumann/freqdom_compare`

[†] Python's interactive visualization library `http://bokeh.pydata.org/`

Acknowledgements

> *"Pure Vernunft darf niemals siegen."*
> – Tocotronic (2005)

First of all, I would like to thank Jan Heiland for being my *academic older brother* for many years already! I always valued your advices a lot, and I really enjoyed working with you on model-order reduction techniques [11, 12] and on our 'early work' [63]. I wish you, Xiao Ai and both your families all the best! When thinking of my undergraduate studies at TU Berlin, mostly André Gaul, Pavel Buran and Danny Panknin come to my mind. I had a great time with you – all of you are very excellent nerds. ✌

I then studied one year at KTH Stockholm and no one represents this international study environment better than the beautifully annoying Slobodan Milovanovic (I call him Boba). Then, I moved to Delft where I met many amazing people too. I discovered race cycling with Lennard Budelmann, and we went to a great wedding of Hector & Majka. Mathias Mul was always there for me (Yes, I do have Dutch friends!) and I must admit that I am a little bit jealous on Moritz & Laura for having such a cute baby girl ♡.

Concerning this work, I first of all want to thank the developers of the open source FEM package nutils [149], and in particular Joost van Zwieten for a million Linux helps and valuable scientific discussions. I want to thank my friends and office mates Joost & Guido for a great time at the office, and my two PhD brothers Yue Qiu and Reinaldo Astudillo for a great and non-competitive atmosphere at work. Many more colleagues and friends made my time special in Delft and helped me to *survive* PhD life; in particular Gaby, Sławek, Dave, Chiara, Mohamed, Ina, Raphael, Jörn, Andreas, Jochen, Roel, Me{n,r}el, and all participants of Project baNaNa[†] and the Tuesday soccer group. I always enjoyed going on conferences and meeting new people there; the most memorable events for me were the Krylov Day 2015[‡] and the Preconditioning Conferences in Eindhoven (2015) and Vancouver (2017).

This work was made possible thanks to the financial support of Shell Global Solutions International B.V. I would like to express my gratitude to my advisor from Shell, René-Édouard Plessix, and my promoter at TU Delft, Kees Vuik, for making

---

[†] Project baNaNa – Seminar organized for-and-by PhD students of the Delft Institute of Applied Mathematics (http://projectbanana.github.io/)

[‡] Krylov Day 2015 – One-day workshop organized by the SIAM Student Chapter Delft (https://sinews.siam.org/Details-Page/european-students-gather-at-tu-delft-for-krylov-day)

this research project possible. Both of you were there when needed and gave me a lot of freedom (when needed).

The biggest 'thank you' deserves my *Doktorvater* Martin van Gijzen whose mathematical input was extremely important for me throughout the years. I am very happy that you and Marielba convinced me to stay in Delft for my PhD research. All I want to say is that I learned a lot from you – I think I never told you.

Das Wichtigste im Leben sind gute Freunde und Familie. Ich habe zum Glück beides. Die meisten meiner Erinnerungen sind eng mit dem Herrn Pischon verbunden. Er steht wie kein anderer für die Werte der SKM.

Als letztes möchte ich meinen lieben *ouders* danken und meinen beiden Omis. Es ist schön, dass es euch gibt! Auf diese Dissertation sollt auch ihr ein bisschen stolz sein.

Manuel M. Baumann
Delft, September 5, 2017

Curriculum Vitae

Manuel Baumann started his academic career at the Technical University of Berlin in 2007. He obtained his Bachelor degree in *Engineering Physics* in 2010 and in *Mathematics* in 2011, both from TU Berlin. During the last two years of his studies at TU Berlin he worked as a student research assistant in the working group *ModNumDiff* under the supervision of Volker Mehrmann and Jan Heiland. In 2011, he won the prize for the best Bachelor thesis in Mathematics at *Dies Mathematicus*. He then joined the multidisciplinary *Erasmus Mundus* master study track *Computer Simulations for Science and Engineering (COSSE)* from 2011 until 2013. He finished his Master studies in 2013 *cum laude* with a double degree in *Scientific Computing* from the Royal Institute of Technology in Stockholm, Sweden, and in *Applied Mathematics* from Delft University of Technology, The Netherlands. In 2013, he joined the *Numerical Analysis* research group of Kees Vuik as a PhD candidate at TU Delft. His research was supervised by Martin B. van Gijzen (co-promoter, TU Delft) and René-Édouard Plessix (Shell Global Solutions International B.V.). The focus of his scientific work during the last four years include Krylov subspace methods, preconditioning techniques for the elastic wave equation and model order reduction for optimal control. In 2014, he co-founded the *SIAM Student Chapter* at TU Delft and was the first president of the Chapter; in 2017 he received the *SIAM Student Chapter Certificate of Recognition* for his contributions to the Student Chapter. Next to his studies, Manuel is a skiing instructor at the *Berliner Hochschulsport*, a hobby race cyclist, and an enthusiastic supporter of the recent open science movement.

http://www.manuelbaumann.de

List of Scientific Activities

Journal Publications

- M. Baumann and M.B. van Gijzen (2015). *Nested Krylov methods for shifted linear systems.* SIAM Journal of Scientific Computing, **37**(5), S90–S112.
- M. Baumann, P. Benner, J. Heiland (2016). *Space-time Galerkin POD with application in optimal control of semi-linear parabolic partial differential equations.* Technical report, arXiv:**1611.04050**. [Currently under review at SIAM Journal of Scientific Computing]
- M. Baumann, R. Astudillo, Y. Qiu, E.Y.M Ang, M.B. van Gijzen, and R.-É. Plessix (2017). *An MSSS-Preconditioned Matrix Equation Approach for the Time-Harmonic Elastic Wave Equation at Multiple Frequencies.* Computational Geosciences, Springer, DOI: 10.1007/s10596-017-9667-7.
- M. Baumann and M.B. van Gijzen (2017). *Efficient iterative methods for multi-frequency wave propagation problems: A comparison study.* Procedia Computer Science, Vol. **108**, pp. 645–654.[†]
- M. Baumann and M.B. van Gijzen (2017). *An Efficient Two-Level Preconditioner for Multi-Frequency Wave Propagation Problems.* DIAM Technical Report **17–03**, Delft University of Technology. [Currently under review at Elsevier Applied Numerical Mathematics]

Peer-Reviewed Conference Proceedings

- M. Baumann and M.B. van Gijzen (2016). *A Fast Iterative Solution of the Time-harmonic Wave Equation with MSSS-preconditioned IDR(s).* Appeared in: Extended abstract for 78th EAGE Conference & Exhibition in Vienna, Austria, EarthDoc.
- Baumann, M., Heiland, J., and Schmidt, M. (2015). *Discrete Input/Output Maps and their Relation to Proper Orthogonal Decomposition.* In: Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory. Festschrift in Honor of Volker Mehrmann. Springer International Publishing. 585-608.

[†] The extended paper version is invited for publication in the ICCS 2017 Special Issue of the Journal of Computational Science.

- Heiland, J. and Baumann, M. and Walle, A. and Mehrmann, V. and Schäfer, M. (2010). *Simulation and Control of Drop Size Distributions in Stirred Liquid/Liquid Systems.* Appeared in: Proc. 4th International Conference on Population Balance Modelling, September 15-17 2010, Berlin, Germany.

Contributions at International Conferences

Oral Presentations (selection)

- *An Efficient Two-Level Preconditioner for Multi-Frequency Wave Propagation Problems.* Presented at: International Conference On Preconditioning Techniques For Scientific And Industrial Applications 2017, Vancouver, Canada, August 1, 2017.

- *Efficient iterative methods for multi-frequency wave propagation problems: A comparison study.* Presented at: International Conference on Computational Science (ICCS), Zürich, Switzerland, June 14, 2017.

- *An Efficient Two-Level Preconditioner for Multi-Frequency Wave Propagation Problems.* Presented at: WSC Spring Meeting, Antwerp, Belgium, May 19, 2017.

- *Nested Krylov Methods for Solving the Time-Harmonic Elastic Wave Equation at Multiple Frequencies.* Presented at: SIAM Annual Meeting, Boston, United States, July 14, 2016.

- *A Fast Iterative Solution of the Time-Harmonic Elastic Wave Equation with MSSS-preconditioned IDR(s).* Presented at: 78th EAGE Conference & Exhibition – Workshop 13, Vienna, Austria, June 3, 2016.

- *Nested Krylov methods for shifted linear systems.* Presented at: SIAM Conference on Applied Linear Algebra, Atlanta, United States, October 26, 2015.

- *Nested Krylov methods for shifted linear systems.* Presented at: International Conference On Preconditioning Techniques For Scientific And Industrial Applications 2015, Eindhoven, The Netherlands, June 19, 2015.

- *Nested Krylov methods for shifted linear systems.* Presented at: 4th IMA Conference on Numerical Linear Algebra and Optimisation, Birmingham, United Kingdom, September 3, 2014.

Scientific Posters

- *Preconditioning the elastic wave equation in 2D and 3D based on inexact MSSS matrix computations.* Presented at: International Conference On Preconditioning Techniques For Scientific And Industrial Applications, Vancouver, Canada, July 31–August 2, 2017 (joint with Y. Qiu).

- *Preconditioning the time-harmonic elastic wave equation at multiple frequencies.* Presented at: COSSE alumni workshop, Stockholm, Sweden, November 10–12, 2016; and DCSE Kick-off meeting, Delft, The Netherlands, May 23, 2017.

- *A generalized POD space-time Galerkin scheme for parameter dependent dynamical systems* Presented at: MoRePas - Model Reduction of Parametrized Systems, Trieste, Italy, October 13–16, 2015 (presenter: J. Heiland).
- *A Set of Fortran 90 and Python Routines for Solving Linear Equation with IDR(s).* Presented at SIAM Conference on Applied Linear Algebra, Atlanta, United States, October 26–30, 2015 (joint with R. Astudillo).
- *A generalization of the Proper Orthogonal Decomposition method for nonlinear model-order reduction.* Presented at: Fortieth Woudschoten Conference, Zeist, The Netherlands, October 7–9, 2015.
- *Nested Krylov methods for shifted linear systems.* Presented at: Numerical Algebra, Matrix Theory, Differential-Algebraic Equations, and Control Theory, Berlin, Germany, May 6–9, 2015.
- *Nested Krylov methods for shifted linear systems arising in frequency-domain wave simulations.* Presented at: Thirty-ninth Woudschoten Conference, Zeist, The Netherlands, October 8–10, 2014.
- *Fast iterative solution of the time-harmonic elastic wave equation at multiple frequencies.* Presented at: Thirty-eighth Woudschoten Conference, Zeist, The Netherlands, October 2–4, 2013.

Academic Theses

- M. Baumann (2013). *Nonlinear Model Order Reduction using POD/DEIM for Optimal Control of Burgers' Equation.* Master Thesis in Applied Mathematics, Delft University of Technology. Scientific supervision: Marielba Rojas and Martin B. van Gijzen.
- M. Baumann (2011). *Modellierung und Simulation von Dispersionen in turbulenten Strömungen.* Bachelor Thesis in Mathematics, Technical University of Berlin (in German). Scientific supervision: Jan Heiland and Volker Mehrmann.
- M. Baumann and P. Buran (2010). *Lösung der zweidimensionalen Wirbeltransportgleichung auf NVIDIA Grafikkarten.* Bachelor Thesis in Engineering Physics, Technical University of Berlin (in German). Scientific supervision: Günter Bärwolff and Jörn Sesterhenn.

Teaching Experiences

I was teaching assistant for the following courses given at Delft University of Technology:

- TW2060: Numerical Analysis 1 (2016-2017 Q3-Q4)
- TW2060: Numerical Analysis 1 (2015-2016 Q3-Q4)
- TB131B: Lineaire Algebra en Differentiaalvergelijkingen (2015-2016 Q2)
- TW2060: Numerical Analysis 1 (2014-2015 Q3-Q4)
- TB131A: Analyse en Differentiaalvergelijkingen (2014-2015 Q1)