



A Semi Matrix-Free Twogrid Preconditioner for the Helmholtz Equation with Near Optimal Shifts

Daniel Drzisga¹ · Tobias Köppl¹  · Barbara Wohlmuth¹

Received: 2 March 2022 / Revised: 17 March 2023 / Accepted: 20 March 2023 /
Published online: 3 May 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Due to its significance in terms of wave phenomena a considerable effort has been put into the design of preconditioners for the Helmholtz equation. One option to design a preconditioner is to apply a multigrid method on a shifted operator. In such an approach, the wavenumber is shifted by some imaginary value. This step is motivated by the observation that the shifted problem can be more efficiently handled by iterative solvers when compared to the standard Helmholtz equation. However, up to now, it is not obvious what the best strategy for the choice of the shift parameter is. It is well known that a good shift parameter depends sensitively on the wavenumber and the discretization parameters such as the order and the mesh size. Therefore, we study the choice of a near optimal complex shift such that a flexible generalized minimal residual (FGMRES) solver converges with fewer iterations. Our goal is to provide a map which returns the near optimal shift for the preconditioner depending on the wavenumber and the mesh size. In order to compute this map, a data driven approach is considered: We first generate many samples, and in a second step, we perform a nonlinear regression on this data. With this representative map, the near optimal shift can be obtained by a simple evaluation. Our preconditioner is based on a twogrid V-cycle applied to the shifted problem, allowing us to implement a semi matrix-free method in which only the coarse grid and boundary matrices need to be stored in memory. On the fine grid, only the action of the matrix applied to a vector is computed, without assembling the global matrix. This enables efficient use of computing resources and allows problems to be solved at scales that were previously limited by the available memory. The performance of our preconditioned FGMRES solver is illustrated by several benchmark problems with heterogeneous wavenumbers in two and three space dimensions.

✉ Tobias Köppl
koeppl@ma.tum.de

Daniel Drzisga
drzisga@ma.tum.de

Barbara Wohlmuth
wohlmuth@ma.tum.de

¹ Lehrstuhl für Numerische Mathematik, Fakultät für Mathematik (M2), Technische Universität München, Garching bei München, Germany

Keywords Helmholtz equation · Multigrid · Shifted Laplacian · Preconditioner · Matrix-free · Data driven · Nonlinear regression

Mathematics Subject Classification 35J05, 65N55, 65F08, 65F50

1 Introduction

The Helmholtz equation plays an important role in the mathematical modeling of wave phenomena like the propagation of sound and light. Because of its importance, the Helmholtz equation is of large interest for both, analytical and numerical research. In this work, we deal with the efficient solving of linear systems of equations resulting from standard conforming finite element discretizations of the Helmholtz equation.

According to [11, 20], there are several reasons why the numerical solution of these equation systems is a challenging task. One reason is that the solutions of the Helmholtz equation are oscillating on a scale of $1/k$, where k denotes the wavenumber. The wavenumber k is proportional to the frequency of the simulated waves. As a consequence, a large number of mesh nodes is required to resolve high frequency waves. On closer examination, it turns out that the required number of mesh nodes is proportional to k^2 . Moreover, low-order methods suffer from pollution effects, which implies that $\mathcal{O}(k^2)$ mesh nodes are not sufficient to bound the discretization error as the wavenumber k increases. This implies that very large systems of equations have to be solved if large wavenumbers are considered. A further difficulty is that for large wavenumbers, these linear systems of equations can be distinctly indefinite such that classical iterative solvers perform poorly [12]. For instance, standard multigrid methods yield unsatisfactory results, since it can be shown that the smoothers as well as the coarse grid corrections cause growing error components [8, 14]. These observations motivate the design of more effective iterative solvers. As a consequence, great effort has been made to achieve this goal. An overview of different solution methods that have been tested in this context can be found in [17, 21]. Due to the fact that classical iterative solvers like the Jacobi method and multigrid methods are not appropriate for a direct application to the Helmholtz problem [10, 38], Krylov subspace methods like GMRES [35] or BiCGSTab [16] have attracted more attention. This is motivated by the fact that these Krylov subspace methods converge even in the case of indefinite matrices. However, their convergence can be very slow without a sophisticated preconditioner [29, 31, 37, 40].

It turns out that a simple modification of the original Helmholtz problem forms the basis to derive an efficient preconditioner for a Krylov subspace method. This is achieved, e.g., by adding a complex shift to the square of the wave number resulting in a new partial differential equation (PDE). This PDE is referred to as the *shifted Laplacian problem* or the *shifted Laplacian preconditioner*. In the remainder of this work, we will use the term *shifted Laplacian problem* [2, 6, 10, 20, 28, 39].

A crucial issue in this context is to determine the optimal shift denoted by ε [10, 20]. It can be shown that for $\varepsilon \in \mathcal{O}(k)$, the shifted Laplacian problem is a preconditioner for the GMRES method with wavenumber-independent convergence. On the other hand using a shift $\varepsilon \in \mathcal{O}(k^2)$ [10, 15], the standard multigrid method shows optimal convergence for the discrete counterpart of the shifted Laplacian problem. This means that there is a gap between the choice $\varepsilon \in \mathcal{O}(k)$ and $\varepsilon \in \mathcal{O}(k^2)$. From this conclude that a shift of $\varepsilon \in \mathcal{O}(k)^\sigma$, $\sigma \in [1, 2]$ should yield a solver for the unshifted Helmholtz problem, which can be regarded as a compromise between a fast multigrid convergence and a good preconditioner which reduces the total

number of required iterations. One cannot expect, nor is it the goal, that this approach yields wavenumber independent iteration numbers. Obviously, it is of great interest to determine not only qualitatively but also quantitatively ε depending on the wave number k such that the convergence rate is minimal. However, there is no analytical formula how to choose the optimal order exponent σ , and according to our knowledge, no article has been published yet which provides a-priori optimal shifts for problems considering general heterogeneous wavenumbers.

The goal of this paper is to design a Krylov subspace method using a near optimal shift. As the outer iterative solver for the standard Helmholtz equation, we choose the FGMRES method [36]. As the preconditioner for the outer solver, we use the standard twogrid solver [41] applied to the shifted problem. The twogrid solver uses the damped Jacobi method as a smoother. This allows for a semi-matrix free implementation of the iterative solver, meaning that most of the required matrix vector products can be realized without accessing a stored global sparse matrix. The Helmholtz equation as well as the shifted Laplacian problem are discretized using standard Q_p , $p \in \{1, 2, 3\}$, finite elements. In order to circumvent the lack of knowledge on how to choose an optimal shift, we use a data driven approach such as in [23, 24, 32]. Thereby, for a given constant wavenumber k and a mesh size h , the near optimal complex shift is determined by an optimization method. For this purpose, we use the golden-section method [25]. This procedure is repeated for a large number of samples with respect to k and h . Having for each sample the optimal exponent σ_{opt} at hand, a map is constructed, such that a near optimal shift can be obtained by just evaluating the map for a given wavenumber and mesh size. By means of such a map, an optimal FGMRES solver can be constructed in an efficient way, particularly, if a highly heterogeneous distribution of the wavenumber has to be considered. In order to support our numerical findings, we perform a Local Fourier Analysis (LFA) for the Q_1 -discretization in two space dimensions. Similar to [12, 15], we determine for a certain wavenumber and meshsize exponents shifts such that the twogrid solver is convergent. LFA is a standard tool for analyzing the convergence behavior of a multigrid solver. Thus it has a long tradition in the multigrid context. Details on LFA can be found e.g. in the following references: [7, 27, 41, 43].

The rest of this paper is organized as follows: Sect. 2 contains the problem setting as well as the variational formulations of the Helmholtz equation and the shifted Laplacian equation. Furthermore, the standard finite element discretizations and theoretical results from literature are recalled. Section 3 focuses on the twogrid preconditioner. Using a LFA, we study the convergence behavior of the twogrid solver. Then in Sect. 4, the FGMRES solver combined with the twogrid solver is introduced. In Sect. 5, the generation of training data that are used to estimate the optimal shift is described. The numerical results are presented and discussed in Sect. 6. In case of the two dimensional settings, the optimal shifts are related to the results of the LFA in Sect. 3. Finally in Sect. 7, we summarize our main findings and give an outlook.

2 Problem Setting and Variational Formulations

The basic equation for modeling wave phenomena like the propagation of sound and light, is the linear wave equation

$$w_{tt} - c^2 \Delta w = \tilde{f}, \quad (1)$$

where the solution variable w represents, e.g., the intensity of sound or light. The term c denotes the propagation speed in a specific medium, and \tilde{f} incorporates external source

terms. Considering only time-harmonic solutions

$$w(\mathbf{x}, t) = u(\mathbf{x})e^{-i\omega t},$$

with an angular frequency $\omega \in \mathbb{R}$, (1) can be transformed into a stationary equation which is known as the Helmholtz equation [18]:

$$\begin{aligned} -\Delta u - k^2 u &= f && \text{in } \Omega, \\ \frac{\partial u}{\partial \mathbf{n}} - iku &= g && \text{on } \partial\Omega. \end{aligned} \tag{2}$$

The parameter k is given by $k = \omega/c$ and is referred to as the wavenumber. In the remainder of this work, we assume that it is given by a spatially varying function $k: \Omega \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, represents a tensorial bounded domain. The source term f results from the transformation of \tilde{f} . The Helmholtz equation is equipped with an impedance boundary condition, i.e., a first-order absorbing boundary condition with $g: \partial\Omega \rightarrow \mathbb{C}$. Using this notation, the *shifted Laplacian problem* can be defined as follows [10, 20]:

$$\begin{aligned} -\Delta u - (k^2 + i\varepsilon)u &= f && \text{in } \Omega, \\ \frac{\partial u}{\partial \mathbf{n}} - iku &= g && \text{on } \partial\Omega. \end{aligned} \tag{3}$$

Thereby the parameter $\varepsilon \in \mathbb{R}$ represents the imaginary shift with respect to the Helmholtz equation (2). For $\varepsilon \geq 0$, $f \in L^2(\Omega)$, $g \in L^2(\Gamma)$, and $k \in L^\infty(\Omega)$, the standard variational formulation of (3) reads as follows: Find $u \in H^1(\Omega, \mathbb{C})$ such that

$$a_\varepsilon(u, v) = F(v), \text{ for all } v \in H^1(\Omega, \mathbb{C}). \tag{4}$$

The space $H^1(\Omega, \mathbb{C})$ consists of complex valued functions, whose real and imaginary parts are in the real valued space $H^1(\Omega)$. The sesquilinear form and the linear form of the variational formulation are given by:

$$\begin{aligned} a_\varepsilon(u, v) &= \int_\Omega \nabla u \cdot \nabla \bar{v} \, dx - \int_\Omega (k^2 + i\varepsilon)u\bar{v} \, dx - i \int_{\partial\Omega} k u \bar{v} \, dx \\ &= a(u, v) - m(u, v; k, \varepsilon) - ib(u, v; k) \end{aligned}$$

and

$$F(v) = \int_\Omega f \bar{v} \, dx + \int_{\partial\Omega} g \bar{v} \, dx.$$

Note that the sesquilinear form for the original Helmholtz problem is given by $a_0(\cdot, \cdot)$. According to [33, Prop. 8.1.3] the variational formulation (4) is well-posed.

In order to solve this equation numerically, standard Q_p -elements, $p \in \{1, 2, 3\}$, are used. Our assumptions on Ω guarantee that we can easily construct suitable meshes yielding an exact presentation of the boundary $\partial\Omega$. The mesh size of the used tensorial grid is denoted by h . Further, let V_h be the finite dimensional subspace of $H^1(\Omega, \mathbb{C})$ defined by conforming Q_p -elements. Using this notation, the discrete version of (4) has the following form: Find $u_h \in V_h$ such that:

$$a_\varepsilon(u_h, v_h) = a(u_h, v_h) - m(u_h, v_h; k, \varepsilon) - ib(u_h, v_h; k) = F(v_h) \text{ for all } v_h \in V_h. \tag{5}$$

As it has been shown in [19, Prop. 2.1], the discrete variational Helmholtz formulation has a unique solution. Taking $u_h = \sum_{i=1}^N u_i \phi_i$, where $\{\phi_i\}_{i=1}^N$ is a basis for V_h , problem (5)

induces the following matrix equation for the coefficient vector $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N]^T$:

$$\mathbf{K}\mathbf{u} - \mathbf{M}(k, \varepsilon)\mathbf{u} - \mathbf{iB}(k)\mathbf{u} = \mathbf{F}, \tag{6}$$

where $K_{ij} = a(\phi_j, \phi_i)$, $M_{ij}(k, \varepsilon) = m(\phi_j, \phi_i; k, \varepsilon)$, $B_{ij}(k) = b(\phi_j, \phi_i; k)$, and $F_i = \int_{\Omega} f \overline{\phi_i} \, dx + \int_{\partial\Omega} g \overline{\phi_i} \, dx$. We define the system matrix as

$$\mathbf{A}(k, \varepsilon) = \mathbf{K} - \mathbf{M}(k, \varepsilon) - \mathbf{iB}(k). \tag{7}$$

3 Local Fourier Analysis for the Twogrid Solver in Two Space Dimensions

In this section, we study the convergence behavior of the twogrid solver applied to the discrete equation (5). Thereby only the two dimensional problem and the Q_1 -discretization is considered. With respect to the reference element $(0, 1)^2$, we use the following basis functions:

$$\begin{aligned} \varphi_1(x, y) &= x \cdot y, \quad \varphi_2(x, y) = (1 - x) \cdot y, \\ \varphi_3(x, y) &= (1 - y) \cdot x \text{ and } \varphi_4(x, y) = (1 - x) \cdot (1 - y). \end{aligned}$$

Furthermore, we investigate only the local convergence of the twogrid solver i.e. we neglect the boundary conditions and consider only the following matrix:

$$\mathbf{A}(k, \varepsilon) = \mathbf{K} - (k^2 + \mathbf{i}\varepsilon)\mathbf{M}, \tag{8}$$

where $\mathbf{M} = m(\phi_j, \phi_i; 1, 0)$. Of particular interest is the region of convergence of the twogrid solver. To analyze the convergence behavior of a twogrid solver, we use a standard technique which is referred to as Local Fourier Analysis (LFA) [7, 12, 27, 41, 43]. In this work it is used to determine for which shifts ε the twogrid solver is converging. Thereby, we restrict ourselves to shifts having the form $\varepsilon = k^\sigma$, where k is the wavenumber and $\sigma \in [1, 2]$ denotes an exponent. Therefore, we consider in the remainder of this work the system matrices $\mathbf{A}(k, k^\sigma)$, $\sigma \in [1, 2]$. The matrices $\mathbf{A}(k, k^\sigma)$ in (8) can be represented in a simplified way by means of the stencil notation:

$$L_h(\sigma) = \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} - (k^2 + \mathbf{i}k^\sigma) \frac{h^2}{36} \begin{bmatrix} 1 & 4 & 1 \\ 4 & 16 & 4 \\ 1 & 4 & 1 \end{bmatrix}.$$

The first part of the stencil $L_h(\sigma)$ corresponds to the standard stiffness matrix, while the second part represents the standard mass matrix. Let \mathbf{G}_h be the grid given by

$$\mathbf{G}_h = \{ \mathbf{x} = (x_1, x_2) = (l_1h, l_2h), \mathbf{l} = (l_1, l_2) \in \mathbb{Z}^2 \},$$

and V the index set of compact stencils

$$V = \{ \kappa = (l_1, l_2) \mid l_1, l_2 \in \{-1, 0, 1\} \} \subset \mathbb{Z}^2.$$

For simplicity, we have assumed that the Q_1 -discretization is based on a mesh consisting of squares with an edge length of h . A stencil S applied to a grid function w_h works as follows [41, Chapter 4]:

$$Sw_h(\mathbf{x}) = \sum_{\kappa \in V} s_\kappa w_h(\mathbf{x} + h\kappa), \quad \mathbf{x} \in \mathbf{G}_h.$$

The values $s_\kappa \in \mathbb{C}$ are the coefficients of a stencil S . In a next step, we introduce the notation for the operator of the twogrid solver T_h^{2h} [9]. Applying T_h^{2h} to the error function e_h^l of the l -th iteration yields:

$$e_h^{l+1} = T_h^{2h} e_h^l, T_h^{2h} = S_h^{v_2} K_h^{2h} S_h^{v_1}, K_h^{2h} = I_h - I_{2h}^h L_{2h}^{-1} I_h^{2h} L_h.$$

Thereby, S_h denotes the smoothing operator. In our work, we use the damped ω -Jacobi smoother, where ω is the damping factor and $v_j, j \in \{1, 2\}$, are the number of pre- and postsmoothing steps. It is a well known fact that the operator for the damped Jacobi is given by

$$S_h(\sigma) = I_h - \omega D_h^{-1}(\sigma) L_h(\sigma),$$

where I_h is the identity operator, and $D_h(\sigma)$ corresponds to the diagonal of the matrix in (8). A straightforward computation yields the following stencil for $S_h(\sigma)$:

$$S_h(\sigma) = \frac{\omega}{8 - \lambda h^2 \frac{4}{3}} \begin{bmatrix} 1 + \lambda h^2 \frac{1}{12} & 1 + \lambda h^2 \frac{1}{3} & 1 + \lambda h^2 \frac{1}{12} \\ 1 + \lambda h^2 \frac{1}{3} & (\frac{1}{\omega} - 1) (8 - \lambda h^2 \frac{4}{3}) & 1 + \lambda h^2 \frac{1}{3} \\ 1 + \lambda h^2 \frac{1}{12} & 1 + \lambda h^2 \frac{1}{3} & 1 + \lambda h^2 \frac{1}{12} \end{bmatrix},$$

where we have used the abbreviation $\lambda = k^2 + ik^\sigma$. Combining the smoothing operator with the correction operator K_h^{2h} , results in the twogrid operator T_h^{2h} . K_h^{2h} itself is given by a combination of L_{2h}, I_h^{2h} , and I_{2h}^h . The operators

$$I_h^{2h} : \mathbf{G}_h \rightarrow \mathbf{G}_{2h} \text{ and } I_{2h}^h : \mathbf{G}_{2h} \rightarrow \mathbf{G}_h$$

stand for the restriction and prolongation where \mathbf{G}_{2h} denotes the coarse grid.

According to [41, Chapters 2 and 4], the stencils for the prolongation, restriction and identity operator are given by:

$$I_{2h}^h = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, I_h^{2h} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \text{ and } I_h = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

L_{2h} in stencil notation has a similar shape as L_h , the only difference is that h has to be replaced by $2h$, and \mathbf{x} is an element of the coarse grid \mathbf{G}_{2h} . The inverted brackets for the stencil of the prolongation operator I_{2h}^h indicate that this stencil has to be applied in a different way as the remaining stencils, see [41, Chapter 2] for more details on the notation. Applying I_{2h}^h to a coarse grid function w_{2h} yields a fine grid function w_h using the following rule [41, Chapter 2]:

$$w_h(\mathbf{x} + h\kappa) = (I_{2h}^h)_\kappa w_{2h}(\mathbf{x}), \mathbf{x} \in \mathbf{G}_{2h},$$

where $(I_{2h}^h)_\kappa$ is the entry of the stencil I_{2h}^h belonging to the index $\kappa \in V$. By means of a LFA we determine for the twogrid operator T_h^{2h} a local convergence factor $\rho_{loc}(\sigma)$ depending on the exponent σ . Thereby, we follow the steps as described in [12] and [41, Chapter 4]. For more detailed information the interested reader is referred to ‘‘Appendix A’’.

Once calculated, the local convergence factor $\rho_{loc}(\sigma)$ helps us to determine the minimal exponent σ_c for the shift ε separating the interval $I_\sigma = [1, 2]$ into two subsets I_{conv} and $I_{div} = I_\sigma \setminus I_{conv}$, where $I_{conv} = [2, \sigma_c]$ and $\sigma_c = \operatorname{argmin}_{\sigma \in I_\sigma} \{\rho_{loc}(\sigma) < 1\}$. In other words: $I_{conv} \subset I_\sigma$ contains the exponents σ for which the twogrid method converges. Varying the smoothing steps v_1 and v_2 as well as the damping factor ω for $kh < 0.75$ and a

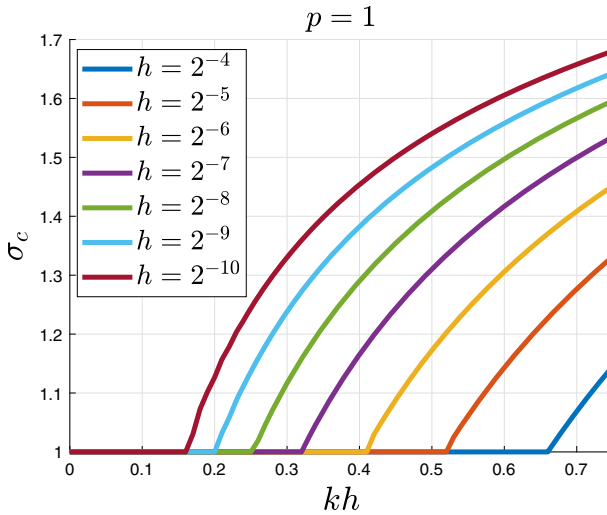


Fig. 1 Minimal exponent σ_c with respect to kh for different mesh sizes $h \in \{2^{-4}, \dots, 2^{-10}\}$

fixed mesh size h shows only little impact on the graph of σ_c . On the other hand, varying the mesh size h and keeping the remaining parameters fixed, shows a significant impact on σ_c (see Fig. 1). On closer examination, it can be observed that for a fixed wavenumber k a small mesh size h enlarges the interval I_{conv} . Reduced intervals I_{conv} arise for a fixed h and a sufficiently large k .

4 Semi Matrix-Free FGMRES with Twogrid Shifted Laplacian Preconditioner

In this section, we draw our attention to the twogrid method from the previous section being used as a preconditioner within a Krylov subspace method. Therefore, we briefly describe the building blocks of the iterative solver used for the numerical experiments in the subsequent sections. For the outer iterations of our solver, we use the preconditioned FGMRES Krylov subspace method [36] to solve the system

$$A_0 P_\varepsilon^{-1} (P_\varepsilon u) = F$$

for a matrix A_0 , a preconditioner P_ε , and a right-hand side F . In our applications, the system matrix A_0 is given by (7) without a complex shift, i.e., $A_0 = A(k, 0)$ and the corresponding right-hand side F is as in (6). The preconditioner P_ε^{-1} is a single twogrid (ν, ν) -cycle with $\nu \in \mathbb{N}$ pre- and postsmoothing steps applied to the preconditioner system matrix of the shifted problem $A_\varepsilon = A(k, \varepsilon)$.

Let $V_1 \subset V_2 = V_h$ be a given hierarchy of two discrete subspaces of V_h . To each of these subspaces, we relate a preconditioner system matrix $A_\varepsilon^{(\ell)}$ stemming from the discretization of (5) with V_ℓ for $\ell = 1, 2$. By I_ℓ , we denote the canonical interpolation or prolongation matrix, mapping a discrete function from V_1 to V_2 . Moreover, we define a damped Jacobi smoother S_ε with damping factor $\omega = \frac{2}{3}$. The Jacobi smoother may be divergent [38], but we choose it because of its straightforward matrix-free implementation in which only the inverse

diagonal of the system matrix needs to be stored in an additional vector. Additionally, the multiplication by the inverse diagonal may be easily parallelizable. Convergent smoothers like the damped Kacmarz-like smoother [11] require the conjugate transpose of the system matrix which is not available in a matrix-free method.

The application of P_ε^{-1} is implemented by performing a twogrid (ν, ν) -cycle to approximately solve the system $P_\varepsilon u = f$ for u , given some right-hand side vector f . This is achieved by calling the function $CYCLE(u, f)$ given in algorithm 1. An important aspect of using this

Algorithm 1 Twogrid (ν, ν) -cycle

```

1: function CYCLE( $u^{(2)}, F^{(2)}$ )
2:   For  $\nu$  steps  $u^{(2)} \leftarrow u^{(2)} + S_\varepsilon^{-1}(F^{(2)} - A_\varepsilon^{(2)}u^{(2)})$  ▷ Presmoothing
3:    $r^{(1)} \leftarrow I^T(F^{(2)} - A_\varepsilon^{(2)}u^{(2)})$  ▷ Restrict residual
4:   Solve  $A_\varepsilon^{(1)}u^{(1)} = r^{(1)}$  ▷ Direct solve
5:    $u^{(2)} \leftarrow u^{(2)} + Iu^{(1)}$  ▷ Add coarse grid correction
6:   For  $\nu$  steps  $u^{(2)} \leftarrow u^{(2)} + S_\varepsilon^{-T}(F^{(2)} - A_\varepsilon^{(2)}u^{(2)})$  ▷ Postsmoothing
7:   return  $u^{(2)}$ 
8: end function

```

iterative method is that on the fine level $\ell = 2$, the matrices $A_\varepsilon^{(2)}, S_\varepsilon^{(2)}$, and I never need to be formed explicitly. Only the result of their action to a vector is required. More precisely, the action of A is implemented in a matrix-free fashion by summing the results of the underlying matrices K and $M(k, \varepsilon)$. Due to software limitations, the boundary matrix corresponding to $B(k)$ was not implemented in a matrix-free way, but is assembled instead. However, since the integration in the sesquilinear form of $b(\cdot, \cdot; k)$ is performed on the boundary, the number of non-zeros in $B(k)$ is of lower order $\mathcal{O}(h^{-d+1})$ compared to the other terms with $\mathcal{O}(h^{-d})$ non-zeros. Furthermore, the system matrix $A_\varepsilon^{(1)}$ on the coarse level $\ell = 1$ is assembled and its LU factorization is stored in memory. Because of this mix of matrix-free operations and stored matrices, we denote this method as *semi matrix-free*.

Using such a semi matrix-free approach effectively saves the memory required for storing the global sparse matrices on finer grids which in turn allows solving problems for which the matrices and its factorizations do not fit in memory. The direct solve in line 4 of `alg:line:coarsesolve 1` is only performed on the coarse grid and thus requires fewer floating point operations and less memory storage. Note that the smaller memory consumption is not the only advantage of a matrix-free approach, since it also significantly reduces the memory traffic. It can be shown that using matrix-free methods instead of matrix-based approaches is beneficial for the performance and may even outperform matrix–vector multiplications with stored matrices [13, 26].

In the remainder of this article, V_2 is formed by uniformly subdividing each element in the mesh corresponding to V_1 . The mesh size corresponding to V_1 is therefore $2h$. Additionally, we do not consider any restarts in the FGMRES method.

5 Data Generation and Processing

In this section, we present our approach to obtain the near optimal complex shift exponents $\hat{\sigma}$ depending on the wavenumber k and discretization parameters $\ell \in \mathbb{N}$ with $h = 2^{-\ell}$ and $p \in \{1, 2, 3\}$. The analysis in Sect. 3 justifies the existence of a near optimal complex shift

σ for which the twogrid method converges, but this does not necessarily need to be optimal shift for which the FGMRES converges faster. The ultimate goal is to construct a map which maps the input parameters to the near optimal shift exponent $\hat{\sigma}$, i.e., $(k, \ell, p) \mapsto \hat{\sigma}$. In order to compute this map, we consider a data based approach in which we generate a set of samples $(k, \ell, p, \hat{\sigma})$ that are used in a subsequent nonlinear regression step. The number of samples that have been generated are contained in Table 1. We first generated samples for mesh sizes down to $h = 2^{-7}$. After this initial sample generation, more samples were obtained for levels $h < 2^{-7}$. Since the sample generation on finer meshes takes more time than on coarser meshes, only fewer samples were collected for $h < 2^{-7}$, which explains the smaller number of samples for these levels. However, this disparity in the number of samples per level is relativized by our special choice of objective function. The following steps describe the process of generating such a single data point:

1. Choose an order p from the set $\{1, 2, 3\}$.
2. Choose a mesh size $h = 2^{-\ell}$ by selecting an ℓ from the set $\{4, \dots, 10\}$.
3. Choose a wavenumber from the interval $[\frac{3p}{16h}, \frac{3p}{4h}]$.
4. Generate a right-hand side vector f with components laying uniformly in the interval $[-1, 1]$.
5. Find the optimal exponent $\hat{\sigma} \in [1, 2]$ of the complex shift $k^{\hat{\sigma}}$ using the gradient free golden-section search [25] with a tolerance of 10^{-2} . This involves repeatedly solving (2) on $\Omega = (0, s)^2$ with $s = 1$ and $g = 0$ using the parameters (k, ℓ, p) and f . The term optimal indicated that we are looking for exponents that are minimizing the number of outer iterations for the FGMRES.
6. Store the tuple $(k, \ell, p, \hat{\sigma})$ and go to Step 1.

Let r_0 be the residual norm at beginning of a solve in Step 5, i.e., $r_0 = \|A_0 u^{(0)} - f\|_2$. In this process, we solve until either a relative residual reduction of 10^{-8} is obtained or a maximum of 50 iterations is reached. Let $r_{\text{end}} = \|A_0 u^{(\text{end})} - f\|_2$ be the final residual at termination after end iterations. In the case without convergence after 50 iterations, the obtained sample is still included in the data, but the corresponding ρ may be close to one. The objective function for the minimization in Step 5 is then given by $\rho = (\frac{r_{\text{end}}}{r_0})^{\frac{1}{\text{end}}}$. Note that the actual residual norm and not the residual norm of the preconditioned system is used since it reflects the error of the underlying physical problem more closely. Repeating the above process N times will generate a set of training data $\mathcal{D} = \{(k_i, \ell_i, p_i, \hat{\sigma}_i)\}_{i=1}^N$. Plots of the sample points for this particular case are presented in Fig. 2 for different values of p . The corresponding average convergence rate ρ for the near optimal complex shifts are illustrated in Fig. 3. The idea of this approach is to approximate this map by a smooth function which can be evaluated for any reasonable input (k, ℓ, p) . Even if this data is obtained only for a constant wavenumber over the whole domain, we will use the approximated map to obtain optimal complex shifts for inhomogeneous wavenumbers in the numerical experiments in Sect. 6.

In this and the following section, we fixed the number of smoothing steps ν to 3, but ultimately the parameters ν and ω can also be included in the data generation in order to further augment the data set. However, the values of ν should be restricted to small values since the Jacobi smoother may be divergent. Let $N_{\ell,p} = |\{(k_i, \ell_i, p_i, \hat{\sigma}_i) \in \mathcal{D} : p_i = p\}|$ and define the weighting coefficient as $w_{\ell,p} = \frac{N_{\ell,p}}{|\mathcal{D}|}$. For a vector of coefficients $(\hat{k}_{c,0}, \hat{k}_{c,1}, \hat{\alpha}_0, \hat{\alpha}_1)^T \in \mathbb{R}^4$, let the approximated map σ_p be defined as

$$\begin{aligned} k_c(\ell) &= \hat{k}_{c,1} \cdot \exp(\hat{k}_{c,0} \cdot \ell), \\ \alpha(\ell) &= \hat{\alpha}_1 \cdot \exp(\hat{\alpha}_0 \cdot \ell), \\ \beta(k, \ell) &= 2 - \exp(-\alpha(\ell) \cdot (k - k_c(\ell))), \end{aligned}$$

Table 1 Number of samples that have been generated for the different polynomial degrees p and meshsizes h

h	Samples $p = 1$	Samples $p = 2$	Samples $p = 3$
2^{-4}	4384	4405	3811
2^{-5}	4344	4410	3723
2^{-6}	4434	4376	3780
2^{-7}	2142	2217	2067
2^{-8}	56	55	57
2^{-9}	57	74	53
2^{-10}	62	66	62

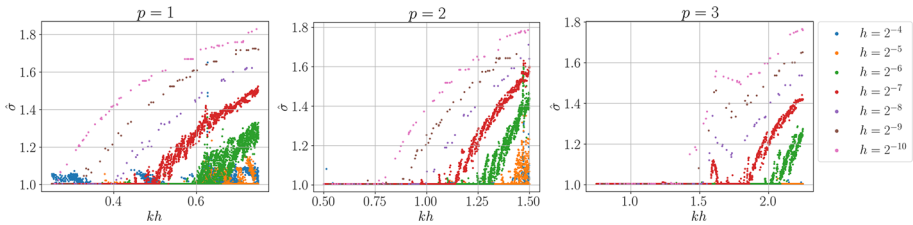


Fig. 2 Optimized shift exponents $\hat{\sigma}$ for decreasing h and $p \in \{1, 2, 3\}$ from left to right

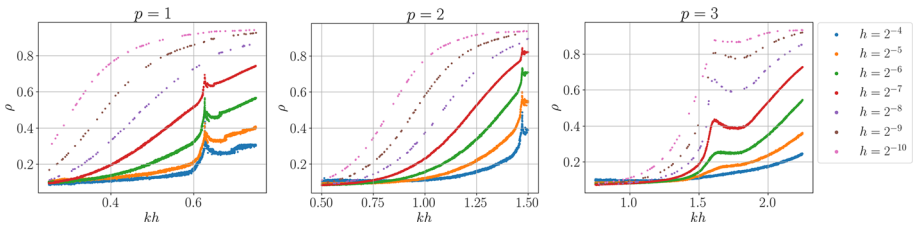


Fig. 3 Average convergence rate ρ using the optimized exponential shift for decreasing h and $p \in \{1, 2, 3\}$ from left to right

$$\sigma_p(k, \ell) = \min(\max(\beta(k, \ell), 1), 2). \tag{9}$$

The objective function of our regression for a fixed p is defined as

$$\text{loss}_p(\hat{k}_{c,0}, \hat{k}_{c,1}, \hat{\alpha}_0, \hat{\alpha}_1) = \sum_{(k_i, \ell_i, p_i, \hat{\sigma}_i) \in \mathcal{D}, p_i=p} w_{\ell_i, p}^{-2} \cdot (\hat{\sigma}_i - \sigma_p(k_i, \ell_i))^2.$$

Each summand is multiplied by the inverse weighting coefficient squared in order to relativize the disparity in the number of samples per level. For each $p \in \{1, 2, 3\}$, we train a map σ_p with the data obtained in the previous step. This is achieved by optimizing for the vector of coefficients $(\hat{k}_{c,0}, \hat{k}_{c,1}, \hat{\alpha}_0, \hat{\alpha}_1)^T \in \mathbb{R}^4$. For this purpose, we use PyTorch [34] together with the ADAM optimizer and a learning rate of 10^{-3} . The weights are initialized with $(0.1, 1.0, -0.5, 1.0)^T$ and the final weights after 50 000 epochs are depicted in Table 2 for each p . The approximated optimal shift maps together with the sampling points are illustrated in Fig. 4. These optimized values are used in the MFEM [4] solver code described in the next section.

This process of training data generation and learning of the parameters in (9) may be considered as offline computations which only need to be done once. Even if our goal is that

Table 2 Optimal weights in function σ_p for $p \in \{1, 2, 3\}$

p	$\hat{k}_{c,0}$	$\hat{k}_{c,1}$	$\hat{\alpha}_0$	$\hat{\alpha}_1$
1	0.4592788619853418	2.5790032999702346	-0.6261637288068426	1.7580549857142198
2	0.5736926870738827	2.5729974893966001	-0.6615199737374460	1.5966386518185063
3	0.6305770719029798	2.4284320222555804	-0.4465407372367102	0.1287828338493968

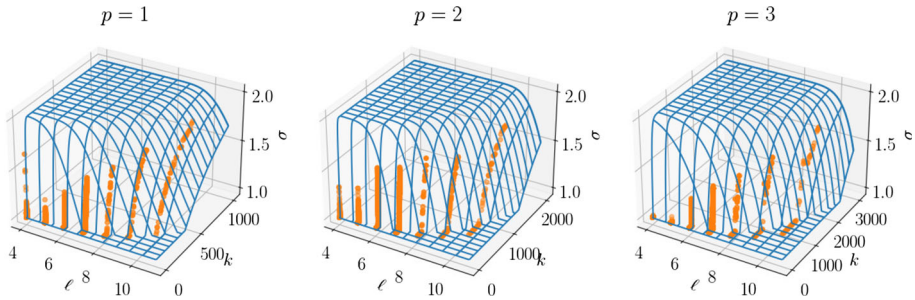


Fig. 4 Sampling points in orange and approximated optimal shift map σ_p in blue for $p \in \{1, 2, 3\}$ from left to right

users use our pretrained parameters from Table 2 and plug them directly into (9) to obtain the optimal shift exponent, we briefly assess these offline costs. Obtaining each training data sample requires solving the Helmholtz problem several times in order to find the optimal shift. We assume that generating a single sample for $\ell = 10$ and $p = 1$ requires solving the Helmholtz problem for 5 times on average. On the machine used in Sect. 6, this requires in the worst case $5 \cdot 3.5 \text{ min} = 17.5 \text{ min}$. Generating the 62 samples on the finest mesh for $p = 1$ therefore took about 18 h in total. However, since each sample can be generated independently, this process can be trivially parallelized if enough resources are available. The final learning process requires much less resources because of the relatively small number of training data and parameters. On a state-of-the-art laptop, the parameter identification required only about one minute using just the CPU.

Next, we compare the minimal exponents σ_c from Sect. 3 to the optimal shifts for the FGMRES solver. For this purpose, we gradually enlarge the computational domain in order to exclude the influence of the boundary conditions. By means of this comparison, we want to theoretically support the choice of the shift k^σ obtained by the optimization procedure described in the first part of this section.

From the LFA, we obtain the shift exponent for which it is guaranteed that the twogrid method applied to the shifted Laplacian problem converges, provided that the domain is very large and boundary effects do not play a role. This shift exponent, however, does not need to be the optimal one when used in conjunction with the outer FGMRES solver. Since our data generation provides us with near optimal shift exponents of the whole solver, we can compare them to the values from the LFA. We expect that, asymptotically, the near optimal shift exponents are larger than the LFA exponents, since these guarantee convergence.

The LFA yields results on an infinitely large domain, therefore, we need to make the near optimal shifts from the numerical experiments comparable. For this purpose, we perform two different systematic comparisons. In the first setup, we fix the parameters $p = 1$ and set $h \in \{2^{-5}, 2^{-6}, 2^{-7}\}$. In the second setup, we consider growing square domains $\Omega_s = (0, s)^2$

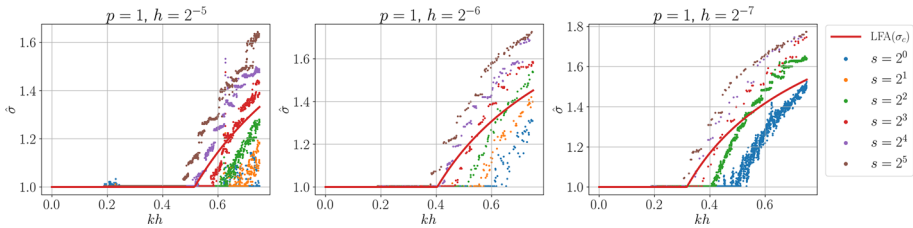


Fig. 5 Required shift exponent σ_c obtained from the LFA and numerically sampled near optimal shifts on different domains Ω_s for $p = 1$ and a fixed mesh size $h \in \{2^{-5}, 2^{-6}, 2^{-7}\}$

for $s \in \{2^i : i \in \{0, 1, \dots, 5\}\}$ and keep the mesh size fixed. For both setups, the number of degrees of freedom increases. More precisely in Fig. 5, we have an increase in the number of degrees of freedom within each picture from the blue to the brown color as well as from the left to the right picture within each color group.

We illustrate the results in Fig. 5 by plotting the required shift obtained from LFA alongside the near optimal shifts obtained by the data driven approach on different domains Ω_s and different fixed shifts h . We may observe that the near optimal shift exponents are in fact larger than the required shift exponent, if the domain is enlarged.

6 Numerical Results

In this section, we demonstrate the effectiveness of using the approximated near optimal shift exponents when applied to solving (2). For this purpose, we consider a set of synthetic and actual scenarios with heterogeneous wavenumbers. In particular, we solve (2) on $\Omega = (0, 1)^d$, $d \in \{2, 3\}$, with a source term

$$f(\mathbf{x}) = 2 \cdot \exp(-1000 \cdot \|\mathbf{x} - \mathbf{s}\|_2^2),$$

where the source location $\mathbf{s} \in \mathbb{R}^d$ is specified in each of the scenarios. The additional boundary term g is set to zero, i.e., $g = 0$. In each of the following scenarios, we normalize the given velocity profile such that its values lie in the interval $[0, 1]$ and denote this scaled profile by $\mu : \Omega \rightarrow [0, 1]$. In the following, μ is called velocity profile. Moreover, we choose a $k_{\max} \in \mathbb{R}$ and set the final heterogeneous wavenumber profile as $k(\mathbf{x}) = k_{\max} \cdot \mu(\mathbf{x})$.

The solver with the two-grid preconditioner described in Sect. 4 is implemented using the MFEM modular finite element library [4] with its support for multigrid operators. The operators on the fine grid are implemented in a matrix-free fashion by using the partial assembly approach in which only the values at quadrature points are stored. The operator applications reuse these precomputed data to compute the action of the operators on-the-fly. The coarse grid problem is solved by computing the LU decomposition with MUMPS [3] through the PETSc [5] interface within the first FGMRES iteration. Subsequent iterations reuse the factorization for an efficient inversion of the coarse grid matrix. The linear systems in each of the scenarios are solved without restarts and until a relative residual reduction 10^{-8} is achieved or the maximum number of 500 iterations is reached. The data and source code for the nonlinear regression in Sect. 5 as well as the MFEM driver source code are available in the Zenodo archive.¹

¹ <https://doi.org/10.5281/zenodo.4607330>.

All run-time measurements for the 2D examples are obtained on a machine equipped with two Intel Xeon Gold 6136 processors with a nominal base frequency of 3.0 GHz. Each processor has 12 physical cores which results in a total of 24 physical cores. The total available memory of 251 GB is split into two NUMA domains, one for each socket. All available 24 physical cores are used for each computation.

The measurements for the 3D examples are conducted on the SuperMUC-NG system equipped with Skylake nodes. The following values were taken from [30]. Each node has two Intel Xeon Platinum 8174 processors with a nominal clock rate of 3.1 GHz. Each processor has 24 physical cores which results in 48 cores per node. Each core has a dedicated L1 (data) cache of size 32 kB and a dedicated L2 cache of size 1024 kB. Each of the two processors has a L3 cache of size 33 MB shared across all its cores. The total main memory of 94 GB is split into equal parts across two NUMA domains with one processor each. We use the native Intel 19.0 compiler together with the Intel 2019 MPI library.

In the following subsections, we consider scenarios with velocity profiles stemming from different sources: an artificial wedge example and three profiles from synthetic and non-synthetic geological cross-sections. Lastly, we consider a 3D scenario in which one of the 2D cross-sections are extruded to 3D. For each possible scenario, we compare the outer FGMRES iteration numbers for different values of the complex shift $\varepsilon \in \{0, k, k^{\frac{3}{2}}, k^2, k^{\sigma_p}\}$. By k^{σ} , we denote the case in which the optimal shift exponent map σ_p from (9) is used with the respective coefficients from Table 2. Note that for heterogeneous k , the shift exponent $\sigma_p = \sigma_p(\mathbf{x})$ depends on the spatial location \mathbf{x} . The speed-ups are calculated with respect to the case without a shift. In particular, let t_{ref} be the time-to-solution for $\varepsilon = 0$ and t_{new} the time-to-solution for one of the methods with $\varepsilon \neq 0$. The respective speed-up is then calculated as $100 \cdot (\frac{t_{ref}}{t_{new}} - 1)\%$.

6.1 Wedge Example

In the first scenario, we consider an artificial velocity profile μ with three distinct values as illustrated in the left of Fig. 6. The maximum value of the source term is located at $s = (0.5, 0.55)^T$. We collect the required number of FGMRES iterations and the respective compute times for $p \in \{1, 2, 3\}$ and the maximum wavenumbers $k_{max} \in \{450, 1100, 1800\}$ in Table 3. Additionally, in the center and right of Fig. 6, we present the real and imaginary parts of the solution in the case of $p = 2$. We observe that using the near optimal shift exponent results in the smallest number of iterations throughout all the considered cases. However, due to a faster LU factorization, using a shift $k^{\frac{3}{2}}$ still results in a shorter compute time for $p = 2$ even if five more iterations are performed.

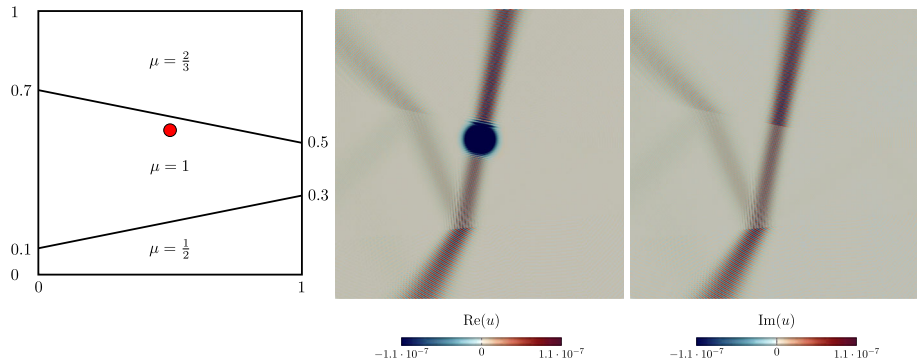


Fig. 6 Wedge example velocity profile with source location (left) and real and imaginary part of the solution in the case $p = 2$ (center and right)

Table 3 Parameter values, iteration numbers, and compute times for the 2D wedge example

Param.	Value	ε	Iter	Time [m:s]	Speed-up
p	1	0	412	3:33.68	
h	2^{-10}	k	330	2:10.38	63.89%
k_{\max}	450	$k^{\frac{3}{2}}$	96	0:32.16	564.43%
$k_{\max} \cdot h$	0.585938	k^2	> 500	–	–
DoFs	2,101,250	k^σ	93	0:31.29	582.90%
p	2	0	491	19:01.23	
h	2^{-10}	k	385	12:52.17	47.80%
k_{\max}	1100	$k^{\frac{3}{2}}$	114	3:09.02	503.76%
$k_{\max} \cdot h$	1.07422	k^2	> 500	–	–
DoFs	8,396,802	k^σ	109	3:18.40	475.22%
p	3	0	371	28:04.96	
h	2^{-10}	k	297	21:04.68	33.23%
k_{\max}	1800	$k^{\frac{3}{2}}$	111	7:13.79	288.43%
$k_{\max} \cdot h$	1.75781	k^2	> 500	–	–
DoFs	18,886,658	k^σ	106	6:53.57	307.42%

6.2 Marmousi Model

In a second scenario, we consider the velocity profile stemming from the synthetic Marmousi model devised by the Institut Français du Pétrole [42]. The corresponding scaled velocity profile is illustrated in the left of Fig. 7. Here, the maximum value of the source term is located at $s = (0.5421, 0.8946)^T$. We collect the required number of FGMRES iterations and the respective compute times for $p \in \{1, 2, 3\}$ and the maximum wavenumbers $k_{\max} \in \{600, 800, 1250, 1900\}$ in Table 4. Additionally, in the center and right of Fig. 7, we present the real and imaginary parts of the solution in the case of $p = 2$. We observe that using the near optimal shift exponent results in the smallest number of iterations for large wavenumbers. For a smaller wavenumber $k_{\max} = 800$ and $p = 2$, the trained shift still yields the minimal

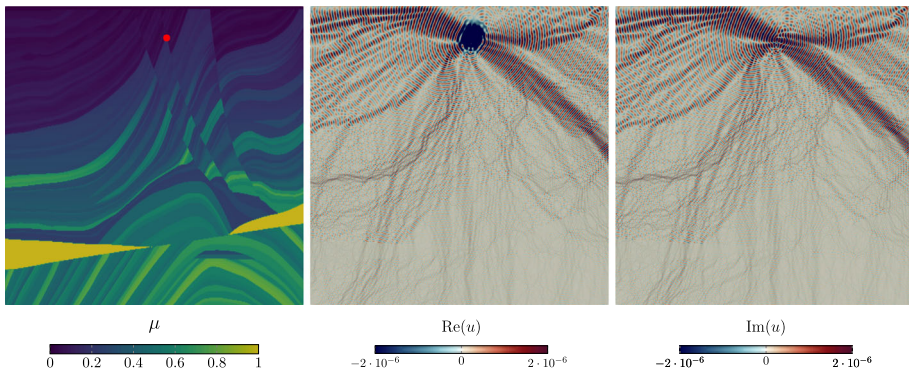


Fig. 7 Marmousi example velocity profile with source location (left) and real and imaginary part of the solution in the case $p = 2$ (center and right)

number of required iterations when compared to the other shifts. This means that using the trained shift does not worsen the performance if the wavenumber is not in the critical regime. Using a shift $k^{\frac{3}{2}}$ results in a slightly shorter compute time for $p = 1$ even if eight more iterations are performed, since the LU factorization required more time. This cannot be observed for the higher order cases with $p > 1$.

6.3 Migration from Topography

In a third scenario, we consider the velocity profile stemming from the migration from topography model representing a cross section through the foothills of the Canadian Rockies; courtesy of Amoco and BP [22]. The corresponding scaled velocity profile is illustrated in the left of Fig. 8. Here, the maximum value of the source term is located at $s = (0.2159, 0.6054)^T$. We collect the required number of FGMRES iterations and the respective compute times for $p \in \{1, 2, 3\}$ and the maximum wavenumbers $k_{\max} \in \{500, 1100, 1900\}$ in Table 5. Additionally, in the center and right of Fig. 8, we present the real and imaginary parts of the solution in the case of $p = 2$. We observe that using the optimal shift exponent results in the smallest number of iterations and the shortest compute time throughout all the considered examples. The choice $k^{\frac{3}{2}}$ yields similar small iterations when compared to using no shift or a shift of k , but using the trained shift yields the fastest solution without requiring a manual choice of the shift.

6.4 BP Statics Benchmark Model

In a fourth scenario, we consider the velocity profile stemming from the synthetic BP statics benchmark model created by Mike O'Brien and Carl Regone, provided by courtesy of Amoco and BP [1]. The corresponding scaled velocity profile is illustrated in the left of Fig. 9. Here, the maximum value of the source term is located at $s = (0.4368, 0.6852)^T$. We collect the required number of FGMRES iterations and the respective compute times for $p \in \{1, 2, 3\}$ and the maximum wavenumbers $k_{\max} \in \{450, 1100, 1900\}$ in Table 6. Additionally, in the center and right of Fig. 9, we present the real and imaginary parts of the solution in the case of $p = 2$. We observe that using the optimal shift exponent results in the smallest number of iterations and shortest compute time throughout all the considered examples. Considerably,

Table 4 Parameter values, iteration numbers, and compute times for the 2D Marmousi example

Param	Value	ε	Iter	Time [m:s]	Speed-up
p	1	0	358	3:03.67	
h	2^{-10}	k	312	1:59.62	53.54%
k_{\max}	600	$k^{\frac{3}{2}}$	147	0:49.81	268.74%
$k_{\max} \cdot h$	0.5859	k^2	> 500	–	–
DoFs	2,101,250	k^σ	139	0:50.73	262.05%
p	2	0	15	0:55.37	
h	2^{-10}	k	15	0:41.30	34.07%
k_{\max}	800	$k^{\frac{3}{2}}$	55	1:47.19	–48.34%
$k_{\max} \cdot h$	0.78125	k^2	> 500	–	–
DoFs	8,396,802	k^σ	15	0:41.41	33.71%
p	2	0	197	6:54.95	
h	2^{-10}	k	181	5:33.39	24.46%
k_{\max}	1250	$k^{\frac{3}{2}}$	125	3:44.19	85.09%
$k_{\max} \cdot h$	1.2207	k^2	> 500	–	–
DoFs	8,396,802	k^σ	109	3:21.13	106.31%
p	3	0	89	6:08.71	
h	2^{-10}	k	85	5:35.00	10.06%
k_{\max}	1900	$k^{\frac{3}{2}}$	99	6:29.95	–5.45%
$k_{\max} \cdot h$	1.85547	k^2	> 500	–	–
DoFs	18,886,658	k^σ	76	5:01.33	22.36%

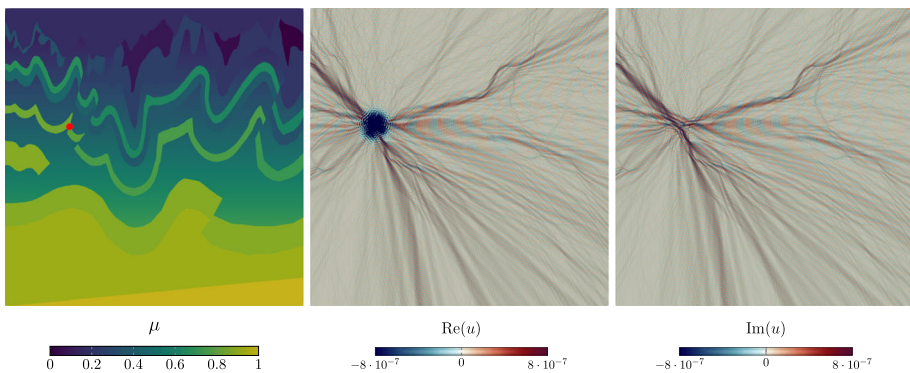


Fig. 8 Migration from topography velocity profile with source location (left) and real and imaginary part of the solution in the case $p = 2$ (center and right)

Table 5 Parameter values, iteration numbers, and compute times for the 2D migration from topography example

Param	Value	ε	Iter	Time [m:s]	Speed-up
p	1	0	421	3:38.80	
h	2^{-10}	k	345	2:18.51	57.97%
k_{\max}	500	$k^{\frac{3}{2}}$	118	0:40.02	446.73%
$k_{\max} \cdot h$	0.4883	k^2	> 500	–	–
DoFs	2,101,250	k^σ	107	0:35.98	508.12%
p	2	0	147	5:09.24	
h	2^{-10}	k	134	4:05.64	25.89%
k_{\max}	1100	$k^{\frac{3}{2}}$	91	2:36.09	98.12%
$k_{\max} \cdot h$	1.07422	k^2	> 500	–	–
DoFs	8,396,802	k^σ	73	2:18.60	123.12%
p	3	0	402	31:39.68	
h	2^{-10}	k	345	26:23.38	19.98%
k_{\max}	1900	$k^{\frac{3}{2}}$	118	7:48.05	305.87%
$k_{\max} \cdot h$	1.85547	k^2	> 500	–	–
DoFs	18,886,658	k^σ	108	7:04.33	347.69%

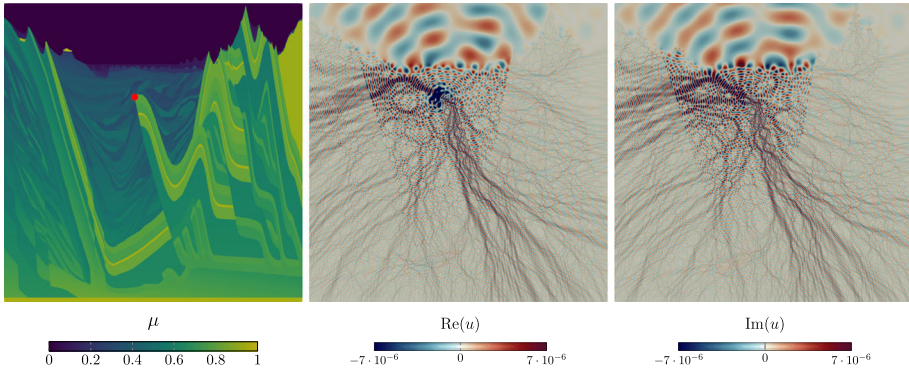


Fig. 9 BP statics benchmark model velocity profile with source location (left) and real and imaginary part of the solution in the case $p = 2$ (center and right)

for $p = 2$ and $p = 3$ the choice of $k^{\frac{3}{2}}$ results in a larger number of required iterations and a longer compute time than for the case without a shift.

6.5 Marmousi 3D Model

In the last scenario, we consider again the velocity profile stemming from the synthetic Marmousi model devised by the Institut Français du Pétrole [42] extruded along a third dimension. The corresponding scaled velocity profile is illustrated in the left of Fig. 10. Here, the maximum value of the source term is located at $s = (0.5421, 0.8946, 0.5)^T$. We collect the required number of FGMRES iterations and the respective compute times for $p \in \{1, 2\}$

Table 6 Parameter values, iteration numbers, and compute times for the 2D BP statics benchmark model example

Param	Value	ε	Iter	Time [m:s]	Speed-up
p	1	0	192	1:35.43	–
h	2^{-10}	k	154	0:59.33	60.85%
k_{\max}	450	$k^{\frac{3}{2}}$	125	0:42.00	127.21%
$k_{\max} \cdot h$	0.439453	k^2	> 500	–	–
DoFs	2,101,250	k^σ	109	0:39.22	143.32%
p	2	0	113	4:01.63	–
h	2^{-10}	k	94	2:53.50	%
k_{\max}	1100	$k^{\frac{3}{2}}$	148	4:25.31	–8.93%
$k_{\max} \cdot h$	1.07422	k^2	> 500	–	–
DoFs	8,396,802	k^σ	79	2:27.00	64.37%
p	3	0	146	9:58.49	–
h	2^{-10}	k	129	8:28.25	17.76%
k_{\max}	1900	$k^{\frac{3}{2}}$	185	12:23.17	–19.47%
$k_{\max} \cdot h$	1.85547	k^2	> 500	–	–
DoFs	18,886,658	k^σ	107	7:01.24	42.08%

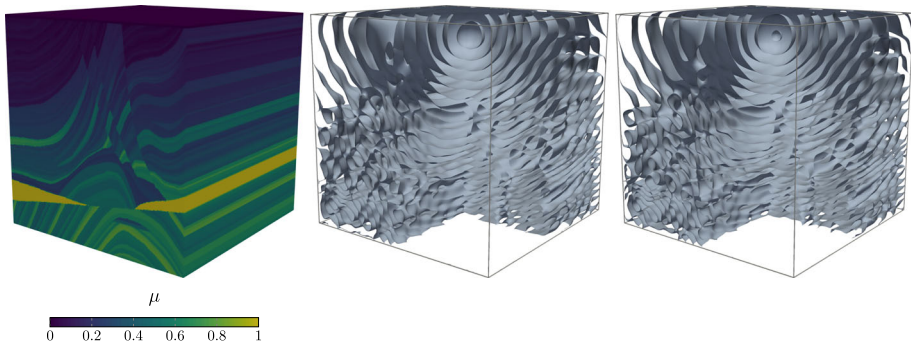


Fig. 10 Marmousi 3D model velocity profile (left) and isosurfaces at value 0 of the real and imaginary part of the solution in the case $p = 1$ (center and right)

and the maximum wavenumbers $k_{\max} = 150$ in Table 7. Additionally, in the center and right of Fig. 10, we present the real and imaginary parts of the solution for $p = 1$. The timings were obtained on the SuperMUC-NG cluster described above by using 384 cores across 16 compute nodes, i.e., 24 cores per compute node. We observe that for $p = 1$, the near optimal shift yields the smallest number of outer FGMRES iterations, but like in the 2D Marmousi case, using a shift of $k^{\frac{3}{2}}$ is faster even if six more iterations are performed. For $p = 2$, the number of iterations using the near optimal shift was larger by one compared to the smallest number of iterations obtained by using no shift or a shift by k , but the compute time was still smaller. These discrepancies in the run-time and number of iterations may be explained by the time deviations of the parallel LU decomposition performed in the first application of the preconditioner.

Table 7 Parameter values, iteration numbers, and compute times for the 3D Marmousi model example with $p = 1$

Param	Value	ε	Iter	Time [m:s]	Speed-up
p	1	0	51	12:47.05	–
h	2^{-8}	k	47	12:03.38	6.04%
k_{\max}	150	$k^{\frac{3}{2}}$	45	10:40.04	19.84%
$k_{\max} \cdot h$	0.585938	k^2	239	25:03.34	–48.98%
DoFs	33,949,186	k^σ	39	11:17.22	13.26%
p	2	0	16	10:35.21	–
h	2^{-7}	k	16	10:01.42	5.62%
k_{\max}	150	$k^{\frac{3}{2}}$	35	10:56.22	–3.20%
$k_{\max} \cdot h$	1.17188	k^2	241	24:13.33	–56.29%
DoFs	33,949,186	k^σ	17	9:59.29	5.99%

7 Conclusion

In this work, we have presented a preconditioner for the Helmholtz equation obtained from a data driven approach. The preconditioner uses near optimal complex shifts in the shifted Laplacian problem which is used as a preconditioner of the Helmholtz equation by applying a twogrid V-cycle to the discrete problem. The near optimal shifts were obtained by generating training data for different mesh sizes h , wavenumbers k , and discretization orders p , and subsequently performing a nonlinear regression to construct a near optimal shift map. Using such an approximated optimal shift map allows users to obtain near optimal shifts automatically without having to tune the required complex shifts manually. In order to solidify this approach, we have performed theoretical considerations based on a local Fourier analysis which justify this data driven approach and we have related the theoretical results to experimental data. Additionally, the twogrid method has been implemented in a semi matrix-free fashion which saves on memory storage and traffic which usually required for matrices corresponding to the finer grids. Furthermore, we have used these near optimal shifts on a set of numerical benchmarks with heterogeneous wavenumbers in 2D and 3D. It could be observed that using these near optimal shifts yielded the smallest FGMRES iteration numbers throughout almost all the examples with speed ups up to 582%.

In the data generation and the numerical experiments, we had restricted ourselves to a single $V(3, 3)$ twogrid cycle and a damped Jacobi smoother with damping factor $\omega = \frac{2}{3}$. Moreover, the complex shift had been always in the form ik^σ . Possible further work could take into account more levels of subspaces resulting in a multigrid solver for which, e.g., the number of smoothing steps per level maybe optimized for. Likewise, wavenumber coefficients of the form $\beta_1 k^{\sigma_1} + i\beta_2 k^{\sigma_2}$ with $\beta_1, \beta_2, \sigma_1, \sigma_2 \in \mathbb{R}$ are of interest as well.

Acknowledgements This work was partly supported by the German Research Foundation by Grant WO671/11-1. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre (www.lrz.de).

A Local Fourier Analysis for the Twogrid Operator

The convergence analysis by means of LFA is based on the assumption that the error function e_h^l on the fine grid \mathbf{G}_h can be represented by a linear combination of Fourier modes [41]:

$$\varphi_h(\theta, \mathbf{x}) = e^{i\theta \cdot \mathbf{x}/h} = e^{i\theta_1 x_1/h} e^{i\theta_2 x_2/h}, \quad \mathbf{x} \in \mathbf{G}_h.$$

$\theta = (\theta_1, \theta_2) \in \mathbb{R}^2$ denote the Fourier frequencies, which may be restricted to the domain $\Theta = (-\pi, \pi]^2 \subset \mathbb{R}^2$, see [12], as a consequence of the fact that

$$\varphi_h(\theta + 2\pi, \mathbf{x}) = \varphi_h(\theta, \mathbf{x}).$$

Thus the space of Fourier modes is given by:

$$E_h = \text{span} \left\{ \varphi_h(\theta, \mathbf{x}) = e^{i\theta \cdot \mathbf{x}/h} \mid \mathbf{x} \in \mathbf{G}_h, \theta \in \Theta \right\}.$$

According to [41], the Fourier modes are the formal eigenfunctions of an operator

$$H \in \left\{ T_h^{2h}, I_h, I_h^{2h}, I_{2h}^h, L_h, L_{2h}, S_h \right\}.$$

This means that:

$$H\varphi_h(\theta, \mathbf{x}) = \tilde{H}(\theta)\varphi_h(\theta, \mathbf{x}),$$

where $\tilde{H}(\theta)$ is called the Fourier symbol of the operator H . In a next step, we decompose the frequency space into a low frequency space $T^{low} = (-\frac{\pi}{2}, \frac{\pi}{2}]^2$ and a high frequency space $T^{high} = \Theta \setminus T^{low}$. For each low frequency $\theta \in T^{low} = (-\frac{\pi}{2}, \frac{\pi}{2}]^2$, we define the following four frequencies:

$$\theta^{(0,0)} = (\theta_1, \theta_2), \quad \theta^{(1,1)} = (\bar{\theta}_1, \bar{\theta}_2), \quad \theta^{(1,0)} = (\bar{\theta}_1, \theta_2), \quad \theta^{(0,1)} = (\theta_1, \bar{\theta}_2),$$

where

$$\bar{\theta}_i = \begin{cases} \theta_i + \pi & \text{if } \theta_i < 0, \\ \theta_i - \pi & \text{if } \theta_i \geq 0. \end{cases}$$

Obviously, the four frequencies fulfil

$$\varphi_h(\theta^\alpha, \mathbf{x}) = \varphi_{2h}(2\theta^{(0,0)}, \mathbf{x}), \quad \mathbf{x} \in \mathbf{G}_{2h}, \quad \alpha \in I = \{(0, 0), (1, 1), (1, 0), (0, 1)\}.$$

Fourier modes with this relationship are called *harmonic to each other*. Considering a given low frequency $\theta = \theta^{(0,0)}$, we define a four dimensional space of harmonics by

$$E_h^\theta = \text{span} \left\{ \varphi(\theta^\alpha, \cdot) \mid \alpha \in I \right\}.$$

An important feature of these spaces is that they are invariant under the twogrid operator T_h^{2h} . Applying T_h^{2h} to an arbitrary element $\Psi \in E_h^\theta$ which is represented by some coefficients A :

$$\Psi = \sum_{\alpha \in I} A\varphi(\theta^\alpha, \cdot),$$

yields new coefficients B :

$$\begin{pmatrix} B^{(0,0)} \\ B^{(1,1)} \\ B^{(1,0)} \\ B^{(0,1)} \end{pmatrix} = \hat{T}_h^{2h}(\theta, \sigma) \begin{pmatrix} A^{(0,0)} \\ A^{(1,1)} \\ A^{(1,0)} \\ A^{(0,1)} \end{pmatrix}.$$

$\hat{T}_h^{2h}(\theta, \sigma) \in \mathbb{C}^{4 \times 4}$ is a matrix depending on the frequency θ and the exponent σ . It represents the operator T_h^{2h} with respect to E_h^θ . In order to determine the convergence behavior of the twogrid method, we study how the coefficients A are transformed by $\hat{T}_h^{2h}(\theta, \sigma)$ into new coordinates B . Therefore, the local convergence factor $\rho_{loc}(\sigma)$ depending on the complex shift exponent in the twogrid operator is taken into account:

$$\rho_{loc}(\sigma) = \sup \left\{ \rho(\hat{T}_h^{2h}(\theta, \sigma)) \mid \theta \in T^{low}, \theta \notin \Lambda \right\}.$$

Thereby $\Lambda \subset (-\pi, \pi]^2$ denotes the set of frequencies for which \hat{L}_{2h} and \hat{L}_h are not invertible and $\rho(\hat{T}_h^{2h}(\theta, \sigma))$ is the spectral radius of $\hat{T}_h^{2h}(\theta, \sigma)$. This means that finding the convergence factor for a fixed frequency is reduced to finding the spectral radius of a 4×4 matrix. To obtain the convergence rate of the twogrid solver, one has to search for the maximum of this quantity in the low frequency space T^{low} . The search for the maximal spectral radius is restricted to T^{low} , since the pattern of $\rho(\hat{T}_h^{2h})$ in T^{low} is extended periodically to the whole frequency domain Θ . Furthermore, an important feature of a multigrid solver is the damping of the low frequency error components [41].

To compute $\hat{T}_h^{2h}(\theta, \sigma)$, the operators occurring in the definition of T_h^{2h} are represented with respect to the harmonic space E_h^θ by means of 4×4 matrices. Exceptions are the prolongation, restriction and L_{2h}^{-1} operator whose matrices are given by 1×4 , 4×1 and 1×1 matrices, since these are mappings related to E_h^θ and E_{2h}^θ . The latter space is the space of harmonics with respect to the coarse grid \mathbf{G}_{2h} :

$$E_{2h}^\theta = \text{span} \left\{ \varphi_{2h}(\theta^\alpha, \cdot) \mid \alpha \in \{(0, 0)\} \right\}.$$

The matrices defined with respect to the harmonic spaces are indicated by a hat:

$$\hat{T}_h^{2h}(\theta, \sigma) = \hat{S}_h^{v_2}(\theta, \sigma) \hat{K}_h^{2h}(\theta, \sigma) \hat{S}_h^{v_1}(\theta, \sigma),$$

$$\hat{K}_h^{2h}(\theta, \sigma) = \hat{I}_h - \hat{I}_{2h}^h(\theta) \hat{L}_{2h}^{-1}(2\theta, \sigma) \hat{I}_h^{2h}(\theta) \hat{L}_h(\theta, \sigma).$$

In the following, we list the matrices with respect to the harmonic spaces. Obviously, the matrix for the identity operator \hat{I}_h is given by the identity matrix. The discretization operator L_h can also be represented by a diagonal matrix [12] [41, Chapter 2]:

$$\hat{L}_h(\theta, \sigma) = \text{diag} \left(\tilde{L}_h(\theta^{(0,0)}, \sigma), \tilde{L}_h(\theta^{(1,1)}, \sigma), \tilde{L}_h(\theta^{(1,0)}, \sigma), \tilde{L}_h(\theta^{(0,1)}, \sigma) \right).$$

The symbol of L_h is given by:

$$\begin{aligned} \tilde{L}_h(\theta, \sigma) &= \frac{1}{3} (8 - 2 \cos(\theta_1) - 2 \cos(\theta_2) - 4 \cos(\theta_1) \cos(\theta_2)) \\ &\quad - \frac{\lambda h^2}{36} (16 + 8 \cos(\theta_1) + 8 \cos(\theta_2) + 4 \cos(\theta_1) \cos(\theta_2)). \end{aligned}$$

The matrix for the coarse grid operator and $\theta = \theta^{(0,0)}$ is given by:

$$\hat{L}_{2h}(\theta, \sigma) = \tilde{L}_{2h}(\theta, \sigma), \quad L_{2h}(\sigma) \varphi_{2h}(2\theta, \mathbf{x}) = \tilde{L}_{2h}(\theta, \sigma) \varphi_{2h}(2\theta, \mathbf{x}).$$

The exact formula for $\tilde{L}_{2h}(\theta, \sigma)$ reads as follows:

$$\begin{aligned} \tilde{L}_{2h}(2\theta, \sigma) &= \frac{1}{3} (8 - 2 \cos(2\theta_1) - 2 \cos(2\theta_2) - 4 \cos(2\theta_1) \cos(2\theta_2)) \\ &\quad - \frac{\lambda h^2}{9} (16 + 8 \cos(2\theta_1) + 8 \cos(2\theta_2) + 4 \cos(2\theta_1) \cos(2\theta_2)). \end{aligned}$$

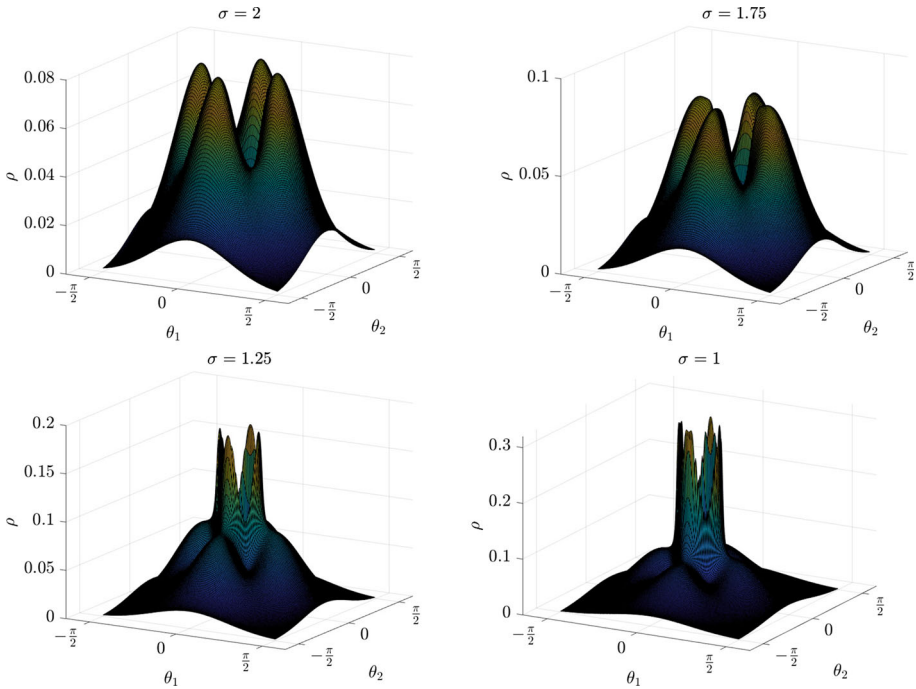


Fig. 11 Convergence factors ρ in the frequency space T^{low} in case of four different shifts $\sigma \in \{1.0, 1.25, 1.75, 2.0\}$

In a next step the matrices for the restriction operator and the prolongation operator are derived:

$$\hat{I}_{2h}^h(\theta) = \begin{pmatrix} \tilde{I}_{2h}^h(\theta^{(0,0)}) \\ \tilde{I}_{2h}^h(\theta^{(1,1)}) \\ \tilde{I}_{2h}^h(\theta^{(1,0)}) \\ \tilde{I}_{2h}^h(\theta^{(0,1)}) \end{pmatrix} = \begin{pmatrix} (1 + \cos(\theta_1))(1 + \cos(\theta_2)) \\ (1 - \cos(\theta_1))(1 - \cos(\theta_2)) \\ (1 - \cos(\theta_1))(1 + \cos(\theta_2)) \\ (1 + \cos(\theta_1))(1 - \cos(\theta_2)) \end{pmatrix} \text{ and } \hat{I}_h^{2h}(\theta) = \frac{1}{4} \cdot (\hat{I}_{2h}^h(\theta))^T.$$

Thereby the matrix entries are determined by the following equation:

$$I_{2h}^h \varphi_{2h}(2\theta, \mathbf{x}) = \sum_{\alpha} \tilde{I}_{2h}^h(\theta^\alpha) \varphi_h(\theta^\alpha, \mathbf{x}) \text{ and } I_h^{2h} \varphi_h(\theta^\alpha, \mathbf{x}) = \tilde{I}_h^{2h}(\theta^\alpha) \varphi_{2h}(2\theta^{(0,0)}, \mathbf{x}), \alpha \in I.$$

We point out that the restriction and prolongation operator are independent of k and σ . Summarizing the previous considerations, one obtains the matrix $\hat{K}_h^{2h}(\theta, \sigma)$. Finally, it remains to specify the matrix for the smoother. According to [41, Chapter 4], it is given by a diagonal matrix:

$$\hat{S}_h(\theta, \sigma) = \text{diag}(\tilde{S}_h(\theta^{(0,0)}, \sigma), \tilde{S}_h(\theta^{(1,1)}, \sigma), \tilde{S}_h(\theta^{(1,0)}, \sigma), \tilde{S}_h(\theta^{(0,1)}, \sigma)).$$

A straightforward calculation yields for $\theta = \theta^{(0,0)}$:

$$\tilde{S}_h(\theta, \sigma) = (1 - \omega) + \omega \frac{\frac{2}{3} + \lambda h^2 \frac{8}{36}}{\frac{8}{3} - \lambda h^2 \frac{16}{36}} \cos(\theta_1)$$

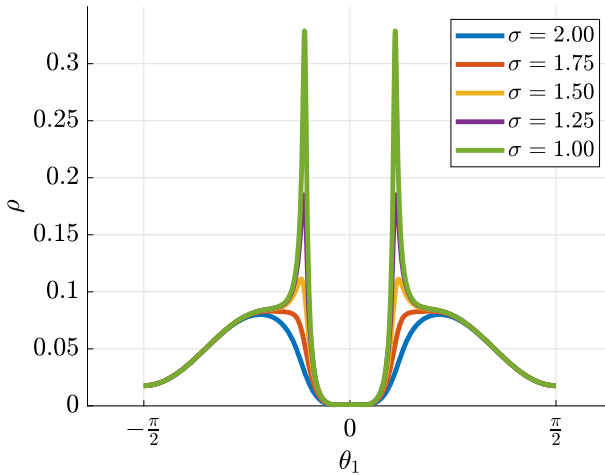


Fig. 12 Convergence factor ρ for $\theta_2 = 0$ and $\theta_1 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The exponents for the shift are chosen as follows: $\sigma \in \{1.00, 1.25, 1.50, 1.75, 2.00\}$

$$+ \omega \frac{\frac{2}{3} + \lambda h^2 \frac{8}{36}}{\frac{8}{3} - \lambda h^2 \frac{16}{36}} \cos(\theta_2) + \omega \frac{\frac{4}{3} + \lambda h^2 \frac{4}{36}}{\frac{8}{3} - \lambda h^2 \frac{16}{36}} \cos(\theta_1) \cos(\theta_2).$$

Finally, we have all ingredients at hand to assemble the matrix \hat{T}_h^{2h} so that $\rho_{loc}(\sigma)$ can be computed numerically.

In Fig. 11, we show some typical surfaces for ρ in the space consisting of low frequencies for $\nu_1 = \nu_2 = 3$, $\omega = 2/3$, $h = 2^{-5}$ and $k = 0.35/h$. It can be observed that the maximum of the spectral radius $\rho(\hat{T}_h^{2h})$, i.e., ρ_{loc} increases as σ is decreased from $\sigma = 2$ towards $\sigma = 1$. In addition to that, there is a symmetry with respect to the axes of the coordinate system. This can be explained by the fact that the entries of the Fourier symbols are composed of cosine functions (see ‘‘Appendix A’’). We further observe numerically that the maxima are located on the axes for $\theta_1 = 0$ or $\theta_2 = 0$ (see Fig. 1). Motivated by these observations, we restrict our search for the maximal spectral radius of \hat{T}_h^{2h} to the line with $\theta_2 = 0$ (see Fig. 12).

References

1. Amoco statics test. https://software.seg.org/datasets/2D/Statics_1994/ (1994). [Online]. Accessed 26 Jan 2021
2. Alvarado, A., Castillo, P.: Computational performance of LDG methods applied to time harmonic Maxwell equation in polyhedral domains. *J. Sci. Comput.* **67**(2), 453–474 (2016)
3. Amestoy, P., Buttari, A., L’Excellent, J.-Y., Mary, T.: Performance and scalability of the block low-rank multifrontal factorization on multicore architectures. *ACM Trans. Math. Softw.* **45**, 1–26 (2019)
4. Anderson, R., Andrej, J., Barker, A., Bramwell, J., Camier, J.-S., Dobrev, J.C.V., Dudouit, Y., Fisher, A., Kolev, T., Pazner, W., Stowell, M., Tomov, V., Akkerman, I., Dahm, J., Medina, D., Zampini, S.: MFEM: a modular finite element library. *Comput. Math. Appl.* **81**, 42–74 (2021)
5. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D., Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.A.,McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H.: PETSc Web page. <https://www.mcs.anl.gov/petsc> (2019)
6. Bayliss, A., Goldstein, C.I., Turkel, E.: An iterative method for the Helmholtz equation. *J. Comput. Phys.* **49**(3), 443–457 (1983)

7. Bolten, M., Rittich, H.: Fourier analysis of periodic stencils in multigrid methods. *SIAM J. Sci. Comput.* **40**(3), A1642–A1668 (2018)
8. Brandt, A., Ta'asan, S.: Multigrid method for nearly singular and slightly indefinite problems. In: *Multigrid Methods II*, pp. 99–121. Springer (1986)
9. Briggs, W.L., Henson, V.E., McCormick, S.F.: *A multigrid tutorial*. SIAM (2000)
10. Cocquet, P.-H., Gander, M.J.: 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)
11. Cocquet, P.-H., Gander, M.J., Xiang, X.: Closed form dispersion corrections including a real shifted wavenumber for finite difference discretizations of 2D constant coefficient Helmholtz problems. *SIAM J. Sci. Comput.* **43**(1), A278–A308 (2021)
12. Cools, S., Vanroose, W.: Local Fourier analysis of the complex shifted Laplacian preconditioner for Helmholtz problems. *Numer. Linear Algebra Appl.* **20**(4), 575–597 (2013)
13. Drzisga, D., Rüde, U., Wohlmuth, B.: Stencil scaling for vector-valued PDEs on hybrid grids with applications to generalized Newtonian fluids. *SIAM J. Sci. Comput.* **42**(6), B1429–B1461 (2020)
14. Elman, H.C., Ernst, O.G., O'leary, D.P.: A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. *SIAM J. Sci. Comput.* **23**(4), 1291–1315 (2001)
15. Erlangga, Y.A., Oosterlee, C.W., Vuik, C.: A novel multigrid based preconditioner for heterogeneous Helmholtz problems. *SIAM J. Sci. Comput.* **27**(4), 1471–1492 (2006)
16. Erlangga, Y.A., Vuik, C., Oosterlee, C.W.: On a class of preconditioners for solving the Helmholtz equation. *Appl. Numer. Math.* **50**(3–4), 409–425 (2004)
17. Ernst, O.G., Gander, M.J.: Why it is difficult to solve Helmholtz problems with classical iterative methods. In: *Lecture Notes in Computational Science and Engineering*, pp. 325–363. Springer, Berlin (2011)
18. Ernst, O.G., Gander, M.J.: Multigrid methods for Helmholtz problems: a convergent scheme in 1D using standard components. In: *Direct and Inverse Problems in Wave Propagation and Applications*. De Gruyter, pp. 135–186 (2013)
19. Esterhazy, S., Melenk, J.M.: An analysis of discretizations of the Helmholtz equation in L2 and in negative norms. *Comput. Math. Appl.* **67**(4), 830–853 (2014)
20. Gander, M.J., Graham, I.G., Spence, E.A.: 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)
21. Gander, M.J., Zhang, H.: A class of iterative solvers for the Helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods. *SIAM Rev.* **61**(1), 3–76 (2019)
22. Gray, S.H., Marfurt, K.J.: Migration from topography: improving the near-surface image. *Can. J. Explor. Geophys.* **31**(1–2), 18–24 (1995)
23. Greenfeld, D., Galun, M., Basri, R., Yavneh, I., Kimmel, R.: Learning to optimize multigrid PDE solvers. In: *Proceedings of Machine Learning Research*, vol. 97, pp. 2415–2423, Long Beach, California, USA (2019). PMLR
24. Heinlein, A., Klawonn, A., Lanser, M., Weber, J.: Machine learning in adaptive domain decomposition methods—predicting the geometric location of constraints. *SIAM J. Sci. Comput.* **41**(6), A3887–A3912 (2019)
25. Kiefer, J.: Sequential minimax search for a maximum. *Proc. Am. Math. Soc.* **4**(3), 502 (1953)
26. Kronbichler, M., Ljungkvist, K.: Multigrid for matrix-free high-order finite element computations on graphics processors. *ACM Trans. Parallel Comput.* **6**(1), 1–32 (2019)
27. Kumar, P., Rodrigo, C., Gaspar, F.J., Oosterlee, C.W.: On local Fourier analysis of multigrid methods for PDEs with jumping and random coefficients. *SIAM J. Sci. Comput.* **41**(3), A1385–A1413 (2019)
28. Li, C., Qiao, Z.: A fast preconditioned iterative algorithm for the electromagnetic scattering from a large cavity. *J. Sci. Comput.* **53**(2), 435–450 (2012)
29. Liu, F., Ying, L.: Recursive sweeping preconditioner for the three-dimensional Helmholtz equation. *SIAM J. Sci. Comput.* **38**(2), A814–A832 (2016)
30. LRZ. Hardware of SuperMUC-NG. <https://doku.lrz.de/display/PUBLIC/Hardware-of+SuperMUC-NG>. Accessed 25 Feb 2020
31. Lu, P., Xu, X.: A robust multilevel preconditioner based on a domain decomposition method for the Helmholtz equation. *J. Sci. Comput.* **81**(1), 291–311 (2019)
32. Luz, I., Galun, M., Maron, H., Basri, R., Yavneh, I.: Learning algebraic multigrid using graph neural networks. In: *International Conference on Machine Learning*, pp. 6489–6499. PMLR (2020)
33. Melenk, J.M.: On generalized finite element methods. PhD thesis, research directed by Dept. of Mathematics. University of Maryland at College Park (1995)

34. Paszke, A. et al.: PyTorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035. Curran Associates, Inc. (2019)
35. Ramos, L.G., Nabben, R.: A two-level shifted Laplace preconditioner for Helmholtz problems: field-of-values analysis and wavenumber-independent convergence. arXiv preprint [arXiv:2006.08750](https://arxiv.org/abs/2006.08750) (2020)
36. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.* **14**(2), 461–469 (1993)
37. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **7**(3), 856–869 (1986)
38. Stolk, C.C., Ahmed, M., Bhowmik, S.K.: A multigrid method for the Helmholtz equation with optimized coarse grid corrections. *SIAM J. Sci. Comput.* **36**(6), A2819–A2841 (2014)
39. Sun, Y., Wu, S., Xu, Y.: A parallel-in-time implementation of the Numerov method for wave equations. *J. Sci. Comput.* **90**(1), 1–31 (2022)
40. Taus, M., Zepeda-Núñez, L., Hewett, R.J., Demanet, L.: L-Sweeps: a scalable, parallel preconditioner for the high-frequency Helmholtz equation. *J. Comput. Phys.* **420**, 109706 (2020)
41. Trottenberg, U., Oosterlee, C.W., Schuller, A.: *Multigrid*. Elsevier (2000)
42. Versteeg, R.: The Marmousi experience: velocity model determination on a synthetic complex data set. *Lead. Edge* **13**(9), 927–936 (1994)
43. Wienands, R., Joppich, W.: *Practical Fourier Analysis for Multigrid Methods*. Chapman and Hall/CRC (2004)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.