A matrix-free parallel two-level deflation preconditioner for two-dimensional heterogeneous Helmholtz problems

Chen, Jinqiang; Dwarka, Vandana; Vuik, Cornelis

DOI
10.1016/j.jcp.2024.113264

Publication date
2024

Document Version
Final published version

Published in
Journal of Computational Physics

Journal of Computational Physics 514 (2024) 113264

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A matrix-free parallel two-level deflation preconditioner for two-dimensional heterogeneous Helmholtz problems

Jinqiang Chen *, Vandana Dwarka, Cornelis Vuik

Delft University of Technology, Numerical Analysis, Mekelweg 4, Delft, 2628 CD, the Netherlands

A R T I C L E I N F O

MSC:
68Q25
68R10
68U05

Keywords:
Parallel computing
Matrix-free
CSLP
Deflation
Scalable
Helmholtz equation

A B S T R A C T

We propose a matrix-free parallel two-level deflation method combined with the Complex Shifted Laplacian Preconditioner (CSLP) for two-dimensional heterogeneous Helmholtz problems encountered in seismic exploration, antennas, and medical imaging. These problems pose challenges in terms of accuracy and convergence due to scalability issues with numerical solvers. Motivated by the limitations imposed by excessive computational time and memory constraints when employing a sequential solver with constructed matrices, we parallelize the two-level deflation method without constructing any matrices. Our approach utilizes preconditioned Krylov subspace methods and approximates the CSLP preconditioner with a parallel geometric multigrid V-cycle. For the two-level deflation, standard inter-grid deflation vectors and further high-order deflation vectors are considered. As another main contribution, the matrix-free Galerkin coarsening approach and a novel re-discretization scheme as well as high-order finite-difference schemes on the coarse grid are studied to obtain wavenumber-independent convergence. The optimal settings for an efficient coarse-grid problem solver are investigated. Numerical experiments of model problems show that the wavenumber independence has been obtained for medium wavenumbers. The matrix-free parallel framework shows satisfactory weak and strong parallel scalability.

1. Introduction

The Helmholtz equation, describing the phenomena of time-harmonic wave scattering in the frequency domain, finds applications in many scientific fields, such as seismic problems, advanced sonar devices, medical imaging, and many more. To solve the Helmholtz equation numerically, we discretize it and obtain a linear system $Ax = b$. The matrix of the linear system is sparse, symmetric, complex-valued, non-Hermitian, and indefinite. Instead of a direct solver, iterative methods and parallel computing are commonly considered for a large-scale linear system resulting from a practical problem. However, the indefiniteness of the linear system brings a great challenge to the numerical solution method, especially for large wavenumbers. The convergence rate of many iterative solvers is affected significantly by increasing wavenumber. An increase in the wavenumber leads to a dramatic increase in iterations. Moreover, the general remedy for minimizing the so-called pollution error, driven by numerical dispersion errors due to discrepancies between the exact and numerical wavenumber, is to refine the grid such that the condition $k^3h^2 < 1$ is satisfied [1], where k is the

* Corresponding author.
E-mail addresses: j.chen-11@tudelft.nl (J. Chen), v.n.s.r.dwarka@tudelft.nl (V. Dwarka), c.vuik@tudelft.nl (C. Vuik).

https://doi.org/10.1016/j.jcp.2024.113264
Received 23 November 2023; Received in revised form 2 July 2024; Accepted 3 July 2024

Available online 8 July 2024

wavenumber and h is the mesh size. Therefore, the research problem of how to solve the systems efficiently and economically while at the same time maintaining a high accuracy by minimizing pollution error arises in this field. A parallel scalable iterative method with wavenumber-independent convergence could significantly enhance the corresponding research in electromagnetics, seismology, and acoustics.

Many efforts have been made to solve the problem in terms of accuracy and scalable convergence behavior. One of the main concerns is the spectrum and eigenvectors of the preconditioned system matrix, which are closely related to the convergence of Krylov subspace methods. When dealing with indefinite problems solved with the GMRES method, the convergence rate is influenced not only by the distribution of the eigenvalues but also by the eigenvectors of the preconditioned matrix, although the eigenvalues are usually more important [2]. The preferable idea is to preprocess the system with a preconditioner. By applying a preconditioner to the linear system, the solution remains the same, but the coefficient matrix has a more favorable distribution of eigenvalues. Many preconditioners have been proposed for the Helmholtz problem so far, such as incomplete Cholesky (IC)/incomplete LU (ILU) factorization, shifted Laplacian preconditioners [3,4], and so on. The Complex Shifted Laplace Preconditioner (CSLP) [4,5] does show good properties for medium wavenumbers. Nevertheless, the eigenvalues of the preconditioned system shift toward the origin as the wavenumber increases. The convergence speed of Krylov-based iterative solvers is adversely affected by the presence of near-zero eigenvalues. Although spectral analysis for a nonnormal matrix becomes less meaningful in assessing convergence properties, numerical experiments show that this notion is also true [6].

The deflation method was developed to eliminate unfavorable eigenvalues by projecting them to zero. Initially proposed independently by Nicolaides [7] and Dostál [8] as a means to accelerate the standard conjugate gradient method for symmetric positive definite (SPD) systems, the deflation method has been further investigated by [9–12]. The deflation preconditioner is sensitive to approximations of the coarse-grid system, but obtaining the exact inverse of the coarse-grid matrix for large-scale problems is not always feasible. In the absence of an accurate approximation, the projection step can introduce numerous new eigenvalues close to zero, leading to adverse effects. To address this limitation, Erlangga et al. [13] extended the method to handle systems with nonsymmetric matrices. Instead of projecting the small eigenvalues to zero, they deflated the smallest eigenvalues to match the maximum eigenvalue. Erlangga et al. demonstrated that their modified projection method was less sensitive to approximations of the coarse-grid system, allowing for the utilization of multilevel projected Krylov subspace iterations. In a recent development, Dwarka and Vuik [6] introduced higher-order approximation schemes to construct deflation vectors. Due to the alignment of the near-zero eigenvalues of A_h and A_{2h}, this advanced two-level deflation method exhibits convergence that is nearly independent of the wavenumber. The authors further extend the two-level deflation method to a multilevel deflation method [14]. By using higher-order deflation vectors, they show that up to the level where the coarse-grid linear systems remain indefinite, the near-zero eigenvalues of these coarse-grid operators remain aligned with the near-zero eigenvalues of the fine-grid operator. Combining this with the well-known CSLP preconditioner, they obtain a scalable solver for highly indefinite linear systems.

The development of scalable parallel Helmholtz solvers is also ongoing. One approach is to parallelize existing advanced algorithms. Kononov and Riyanti [15,16] first developed a parallel version of multigrid-based CSLP. Gordon and Gordon [17] parallelized their so-called CARP-CG algorithm (Conjugate Gradient acceleration of Component-Averaged Row Projections) blockwise. The block-parallel CARP-CG algorithm shows improved scalability as the wavenumber increases. Calandra et al. [18,19] proposed a geometric two-grid preconditioner for 3D Helmholtz problems, which shows the strong scaling property in a massively parallel setup.

Another approach is the Domain Decomposition Method (DDM) [20], which originates from the early Schwarz methods. DDM has been widely used to develop efficient preconditioners and parallel solution methods for Helmholtz problems. For comprehensive surveys, we refer the reader to [21–25] and references therein.

Based on the parallel framework of the CSLP-preconditioned Krylov subspace methods [26], this paper proposes a matrix-free parallel implementation of the two-level-deflation preconditioner for the two-dimensional Helmholtz problems. Besides matrix-free parallelization of each component of the deflation technique, as one of our main contributions, we extensively studied and compared several matrix-free implementations of coarse-grid operators. To compare the scalability of our matrix-free parallel solver with the benchmark results in [6], which used an exact solver on the coarse-grid system, we first solve the coarse-grid system close to machine precision using a default tolerance of 10^{-12}. This allows us to study the impact of the coarse-grid operators on the convergence of the two-level deflation preconditioner while minimizing the potential influence of an inexact coarse-system solver. Upon establishing the convergence behavior, we then focus on finding an efficient and practical setting for the coarse-system solver to optimize the performance of the proposed method. We show that the coarse-grid re-discretization schemes derived from the Galerkin coarsening approach yield convergence properties that are nearly independent of the wavenumber. Furthermore, we investigate the weak and strong scaling properties of the present parallel solution method in a massively parallel setting.

It is important to note that the present study focuses on the Helmholtz equation discretized on regular 2D finite difference grids in quadrilateral domains. The extension of the method to more complex geometries and unstructured grids requires further investigation. However, the 2D method presented in this paper serves as a foundation and proof of concept for the proposed parallel matrix-free implementation of the two-level deflation preconditioner. In principle, the key ideas and techniques can be generalized to 3D problems, although such extensions are non-trivial and require significant additional work. The development of a parallel matrix-free deflation preconditioner for 3D Helmholtz problems is an ongoing research effort and will be the subject of future publications.

The paper is organized as follows. The mathematical models and numerical methods are in section 2 and section 3. Section 4 introduces the parallel framework and matrix-free operators in detail. The experimental results will be discussed in section 5, and the conclusions follow in section 6.

2. Mathematical models

In this paper, we mainly consider the following Helmholtz equation. Suppose that the domain Ω is rectangular with boundary $\Gamma = \partial\Omega$. The Helmholtz equation reads

$$-\Delta \mathbf{u} - k(x,y)^2 \mathbf{u} = \mathbf{b}, \text{ on } \Omega$$

supplied with one of the following boundary conditions:

Dirichlet: $\mathbf{u} = \mathbf{g}$, on $\partial\Omega$

first-order Sommerfeld: $\frac{\partial \mathbf{u}}{\partial \vec{n}} - \mathrm{i}k(x,y)\mathbf{u} = \mathbf{0}$, on $\partial\Omega$

where i is the imaginary unit. $\vec{n}$ and $\mathbf{g}$ represent the outward normal and the given data of the boundary, respectively. $\mathbf{b}$ is the source function. $k(x,y)$ is the wavenumber on Ω. Suppose the frequency is f, the speed of propagation is $c(x,y)$, they are related by

$$k(x,y) = \frac{2\pi f}{c(x,y)}$$

2.1. Discretization

Structural vertex-centered grids are used to discretize the computational domain. Suppose the mesh width in x and y direction are both h. A second-order finite difference scheme for a 2D Laplace operator has the following stencil

$$\left[-\Delta_h\right] = \frac{1}{h^2}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

The discrete Helmholtz operator A_h can be obtained by subtracting the diagonal matrix $\mathcal{I}(k_{i,j}^2)_h$ to the Laplacian operator $-\Delta_h$, *i.e.*

$$A_h = -\Delta_h - \mathcal{I}\left(k_{i,j}^2\right)_h$$

Therefore, the stencil of the discrete Helmholtz operator is

$$\left[A_h\right] = \frac{1}{h^2}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 - k_{i,j}^2 h^2 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{1}$$

In case the Sommerfeld radiation boundary condition is used, the discrete schemes for the boundary points need to be defined. Ghost points located outside the boundary points can be introduced. For instance, suppose $u_{0,j}$ is a ghost point on the left of $u_{1,j}$, the normal derivative can be approximated by

$$\frac{\partial u}{\partial \vec{n}} - \mathrm{i}k(x,y)u = \frac{u_{0,j} - u_{2,j}}{2h} - \mathrm{i}k_{1,j}u_{1,j} = 0 \tag{2}$$

We can rewrite it as

$$u_{0,j} = u_{2,j} + 2h\mathrm{i}k_{1,j}u_{1,j}$$

With the elimination of the ghost point, the stencil for the boundary points $u_{1,j}$ becomes

$$\frac{1}{h^2}\begin{bmatrix} 0 & -1 & 0 \\ 0 & 4 - k_{1,j}^2 h^2 - 2\mathrm{i}k_{1,j}h & -2 \\ 0 & -1 & 0 \end{bmatrix} \tag{3}$$

For corner points, we can do the above step in both x- and y- directions. The resulting stencil is

$$\frac{1}{h^2}\begin{bmatrix} 0 & -2 & 0 \\ 0 & 4 - k_{1,1}^2 h^2 - 4\mathrm{i}k_{1,1}h & -2 \\ 0 & 0 & 0 \end{bmatrix} \tag{4}$$

Discretization of the partial differential equation on the finite-difference grids results in a system of linear equations

$$A_h \mathbf{u}_h = \mathbf{b}_h$$

With first-order Sommerfeld radiation boundary conditions, the resulting matrix is sparse, symmetric, complex-valued, indefinite, and non-Hermitian.

Note that kh is an important parameter indicating how many grid points per wavelength are needed. The mesh width h can be determined by the guidepost of including at least N_{pw} (e.g. 10 or 30) grid points per wavelength. They have the following relationships

$$kh = \frac{2\pi h}{\lambda} = \frac{2\pi}{N_{pw}}$$

For example, if at least 10 grid points per wavelength are required, we can maintain $kh = 0.625$. In cases of heterogeneous media, where the wavenumber k is not constant throughout the computational domain, kh is computed based on the maximum wavenumber present. In this paper, we mainly employ the standard second-order finite difference discretization on the fine grid. Our matrix-free parallel framework is designed to solve the Helmholtz problem by utilizing a sufficiently fine grid to obtain pollution-free solutions. Thus, we will mainly use 10 or more grid points per wavelength. While investigating the performance of the proposed method with a lower number of grid points per wavelength is a valuable research direction [27], it is beyond the scope of the current study.

2.2. Model problem - constant wavenumber

We first consider a 2D problem with constant wavenumber in a rectangular homogeneous domain $\Omega = [0, 1]$. A point source is given by

$$b(x, y) = \delta\left(x - x_0, y - y_0\right), \ \Omega = [0, 1]$$

where $\delta(x, y)$ is a Dirac delta function in the following form

$$\delta(x, y) = \begin{cases} +\infty & x = 0, \ y = 0 \\ 0 & x \neq 0, \ y \neq 0 \end{cases}$$

which satisfies

$$\int\int \delta^2(x, y)\, dx dy = 1$$

In this problem, the point source is imposed at the center $(x_0, y_0) = (0.5, 0.5)$. The wave propagates outward from the center of the domain. The Dirichlet boundary conditions (denoted as MP-1a) or the first-order Sommerfeld radiation conditions (denoted as MP-1b) are imposed, respectively.

2.3. Model problem - Wedge problem

Most physical problems of geophysical seismic imaging describe a heterogeneous medium. The so-called Wedge problem [28] is a typical problem with a simple heterogeneity. It mimics three layers with different velocities hence different wavenumbers. As shown in Fig. 1, the rectangular domain $\Omega = [0, 600] \times [-1000, 0]$ is split into three layers. Suppose the wave velocity c is constant within each layer but different from each other. A point source is located at $(x, y) = (300, 0)$.

The problem is given by

$$\begin{cases} -\Delta u(x, y) - k(x, y)^2 u(x, y) = b(x, y), & \text{on } \Omega = (0, 600) \times (-1000, 0) \\ b(x, y) = \delta(x - 300, y) & x, y \in \Omega \end{cases}$$

where $k(x, y) = \frac{2\pi f}{c(x,y)}$. The wave velocity $c(x, y)$ is shown in Fig. 1. f is the frequency. The first-order Sommerfeld radiation boundary conditions are imposed on all boundaries.

2.4. Model problem - Marmousi problem

For industrial applications, the third model problem is the so-called Marmousi problem [29], a well-known benchmark problem. The geometry of the problem stems from the section of the Kwanza Basin through North Kungra. It contains 158 horizontal layers in the depth direction, making it highly heterogeneous.

The problem is given by

$$\begin{cases} -\Delta u(x, y) - k(x, y)^2 u(x, y) = b(x, y), & \text{on } \Omega = (0, 9200) \times (-3000, 0) \\ b(x, y) = \delta(x - 6000, y) & x, y \in \Omega \end{cases}$$

where $k(x, y) = \frac{2\pi f}{c(x,y)}$. The wave velocity $c(x, y)$ over the domain is shown in Fig. 2. The first-order Sommerfeld radiation boundary conditions are imposed on all boundaries.

3. Numerical methods

The Krylov subspace methods are popular for solving large and sparse linear systems. For non-Hermitian linear systems, it is beneficial for the coefficient matrix to have a spectrum in a bounded region that excludes the origin in the complex plane, as this can improve the convergence rate of Krylov subspace methods [30]. We can incorporate preconditioning and deflation to enhance the

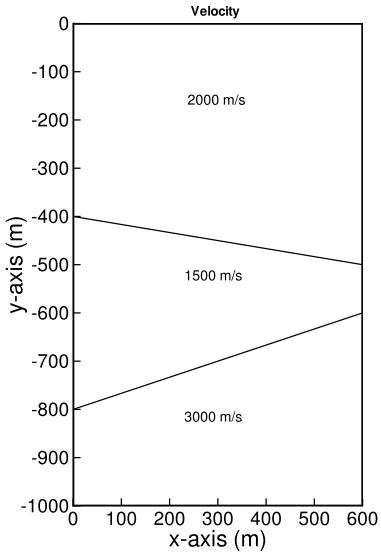


Fig. 1. The velocity distribution of the Wedge problem.

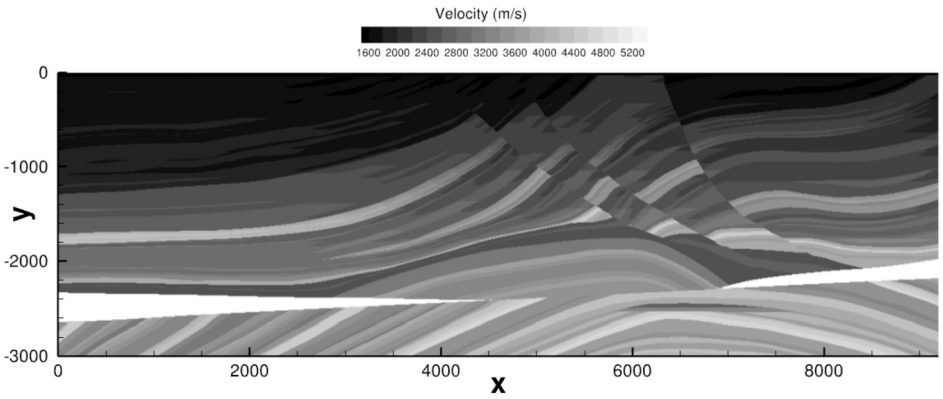


Fig. 2. The velocity distribution of the Marmousi problem.

convergence of Krylov subspace methods. This section will specify every component of the preconditioned Krylov subspace methods we used to solve the linear system resulting from the finite-difference discretization. The multigrid-based Complex shifted Laplace preconditioner (CSLP) combined with two-level deflation will be considered in this work.

3.1. Complex shifted Laplace preconditioner

We focus on the CSLP due to its superior performance and easy setup. The CSLP is defined by

$$M_{h,(\beta_1,\beta_2)} = -\Delta_h - (\beta_1 + \mathrm{i}\beta_2)\, \mathcal{I}\left(k_{i,j}^2\right)_h \tag{5}$$

where $\mathrm{i} = \sqrt{-1}$, $(\beta_1, \beta_2) \in [0, 1]$, and $\mathcal{I}(k_{i,j}^2)_h$ represents a diagonal matrix with $k_{i,j}^2$ as its diagonal elements. The parameter β_2, known as the complex shift, is crucial for the effectiveness of the shifted Laplacian preconditioner [31,32]. Standard multigrid methods will not be effective if the complex shift is too small, and the computational cost of approximating the preconditioner increases significantly [5,33]. In this paper, we use $\beta_1 = 1$ and $\beta_2 = -0.5$. A multigrid V-cycle, consisting of one step of pre-/post-damped Jacobi smoother with a relaxation parameter of 0.8, full weighting restriction, bilinear interpolation, and a General Minimal RESidual method (GMRES) [34] for the coarsest-grid problem, is used to approximate the inverse of CSLP. The choice of smoothers and relaxation parameters can be flexible within a certain range [35,36]. These parameters chosen were found to work well for our test problems [26,37]. However, the optimal relaxation parameter may vary depending on the specific problem and discretization employed.

3.2. Deflation

Suppose a general linear system $Au = b$, where $A \in \mathbb{R}^{n\times n}$, and a projection subspace matrix, $Z \in \mathbb{R}^{n\times m}$, with $m < n$ and full rank are given. Assume that $E = Z^T AZ$ is invertible, the projection matrix P can be defined as

$$P = I - AQ, \quad Q = ZE^{-1}Z^T, \quad E = Z^T AZ$$

By applying the deflation operator, the linear system to be solved becomes

$$PA\hat{u} = Pb \tag{6}$$

Krylov subspace method can be employed to solve the resulting linear system. Since eq. (6) is singular, to obtain a unique solution, we need to update u by

$$u = Qb + P^T\hat{u}$$

3.2.1. Deflation vectors for Helmholtz

The choice of the deflation vectors is one of the base components of the deflation preconditioner. For the Helmholtz problem, a good and necessarily sparse approximation of the eigenvectors corresponding to the small eigenvalues is ideal. However, it is computationally expensive. Originating from the observation that the multigrid inter-grid operators highlight the small frequencies and reserve them on the coarser level, we employ the geometrically constructed multigrid vectors as the deflation vectors. The standard linear interpolation along one dimension is given by

$$I_{2h}^h u_i^{2h} = \begin{cases} u_{i/2}^{2h} & \text{if } i \text{ is even} \\ \frac{1}{2}\left(u_{(i-1)/2}^{2h} + u_{(i+1)/2}^{2h}\right) & \text{if } i \text{ is odd} \end{cases} \tag{7}$$

for $i = 1, 2, \ldots, n-1$ and the restriction operator is

$$I_h^{2h} u_i^h = \frac{1}{2}\left(u_{2i-1}^h + 2u_{2i}^h + u_{2i+1}^h\right) \tag{8}$$

for $i = 1, 2, \ldots, n/2$. In two-dimension, the so-called full weighting restriction operator has a stencil given by

$$[I_h^{2h}] = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h^{2h}$$

and the stencil of the bilinear interpolation operator is given by

$$[I_{2h}^h] = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_{2h}^{h}$$

If we set the coarse to fine grid interpolation operator as the deflation subspace

$$Z = I_{2h}^h$$

where $h \to 2h$ indicates the standard coarsening method. Then Z^T will be the full-weighting restriction operator. The deflation preconditioner can be rewritten as

$$P_h = I_h - A_h Q_h, \quad Q_h = I_{2h}^h A_{2h}^{-1} I_h^{2h}, \quad A_{2h} = I_h^{2h} A_h I_{2h}^h$$

Recently, Dwarka and Vuik [6] used higher-order Bézier curves to construct deflation vectors and obtained wavenumber-independent convergence for very large wavenumbers. The 1D interpolation and restriction operator is given by

$$I_{2h}^h u_i^{2h} = \begin{cases} \frac{1}{8}\left(u_{(i-2)/2}^{2h} + 6u_{i/2}^{2h} + u_{(i+2)/2}^{2h}\right) & \text{if } i \text{ is even} \\ \frac{1}{2}\left(u_{(i-1)/2}^{2h} + u_{(i+1)/2}^{2h}\right) & \text{if } i \text{ is odd} \end{cases}$$

$$I_h^{2h} u_i^h = \frac{1}{8}\left(u_{2i-2}^h + 4u_{2i-1}^h + 6u_{2i}^h + 4u_{2i+1}^h + u_{2i+2}^h\right)$$

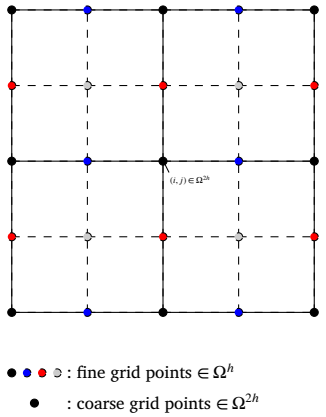


Fig. 3. The allocation map of interpolation operator.

For two-dimensional cases, as shown in Fig. 3, the higher-order interpolation can be implemented as follows.

$$
I_{2h}^{h} u_{i_c,j_c}^{2h} =
\begin{cases}
\frac{1}{64}\Big(6u_{i_c-1,j_c}^{2h} + 36u_{i_c,j_c}^{2h} + 6u_{i_c+1,j_c}^{2h} \\
\quad + u_{i_c-1,j_c-1}^{2h} + 6u_{i_c,j_c-1}^{2h} + u_{i_c+1,j_c-1}^{2h} \\
\quad + u_{i_c-1,j_c+1}^{2h} + 6u_{i_c,j_c+1}^{2h} + u_{i_c+1,j_c+1}^{2h}\Big), & \bullet \\
\frac{1}{16}\Big(u_{i_c,j_c-1}^{2h} + 6u_{i_c,j_c}^{2h} + u_{i_c,j_c+1}^{2h} \\
\quad + u_{i_c+1,j_c-1}^{2h} + 6u_{i_c+1,j_c}^{2h} + u_{i_c+1,j_c+1}^{2h}\Big), & \bullet \\
\frac{1}{16}\Big(u_{i_c-1,j_c}^{2h} + 6u_{i_c,j_c}^{2h} + u_{i_c+1,j_c}^{2h} \\
\quad + u_{i_c-1,j_c+1}^{2h} + 6u_{i_c,j_c+1}^{2h} + u_{i_c+1,j_c+1}^{2h}\Big), & \bullet \\
\frac{1}{4}\Big(u_{i_c,j_c}^{2h} + u_{i_c,j_c+1}^{2h} + u_{i_c+1,j_c}^{2h} + u_{i_c+1,j_c+1}^{2h}\Big), & \bullet
\end{cases}
$$

where $(i_c, j_c) \in \Omega^{2h}$. The higher-order interpolation can then be represented by the following stencil notation

$$
[I_{2h}^{h}] = \frac{1}{64}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}_{2h}^{h}
$$

The higher-order restriction can be implemented in a matrix-free way as follows.

$$
\begin{aligned}
&I_{h}^{2h} u_{2i_c-1,2j_c-1}^{h} = \\
&\frac{1}{64}\Big[u_{2i_c-3,2j_c+1}^{h} + 4u_{2i_c-2,2j_c+1}^{h} + 6u_{2i_c-1,2j_c+1}^{h} + 4u_{2i_c,2j_c+1}^{h} + u_{2i_c+1,2j_c+1}^{h} \\
&+ 4u_{2i_c-3,2j_c}^{h} + 16u_{2i_c-2,2j_c}^{h} + 24u_{2i_c-1,2j_c}^{h} + 16u_{2i_c,2j_c}^{h} + 4u_{2i_c+1,2j_c}^{h} \\
&+ 6u_{2i_c-3,2j_c-1}^{h} + 24u_{2i_c-2,2j_c-1}^{h} + 36u_{2i_c-1,2j_c-1}^{h} + 24u_{2i_c,2j_c-1}^{h} + 6u_{2i_c+1,2j_c-1}^{h} \\
&+ 4u_{2i_c-3,2j_c-2}^{h} + 16u_{2i_c-2,2j_c-2}^{h} + 24u_{2i_c-1,2j_c-2}^{h} + 16u_{2i_c,2j_c-2}^{h} + 4u_{2i_c+1,2j_c-2}^{h} \\
&+ u_{2i_c-3,2j_c-3}^{h} + 4u_{2i_c-2,2j_c-3}^{h} + 6u_{2i_c-1,2j_c-3}^{h} + 4u_{2i_c,2j_c-3}^{h} + u_{2i_c+1,2j_c-3}^{h}\Big]
\end{aligned}
$$

where $(i_c, j_c) \in \Omega^{2h}$. The higher-order restriction can then be represented by the following stencil notation

$$[I_h^{2h}] = \frac{1}{64}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}_h^{2h}$$

3.2.2. The deflation preconditioner

Firstly, one may want to deflate the CSLP-preconditioned Helmholtz operator, as the eigenvalues of $M_{h,(\beta_1,\beta_2)}^{-1}A_h$ begin to shift to the origin as the wavenumber increases. These small eigenvalues can be projected to zero by the preconditioner given by

$$\tilde{P}_h = I_h - \tilde{A}_h\tilde{Q}_h, \quad \tilde{Q}_h = I_{2h}^h \tilde{A}_{2h}^{-1} I_h^{2h}, \quad \tilde{A}_{2h} = I_h^{2h}\tilde{A}_h I_{2h}^h$$

where $\tilde{A}_h = M_{h,(\beta_1,\beta_2)}^{-1}A_h$. One can notice that the coarse operator $\tilde{A}_{2h}$ needs to be inverted. Since the idea of deflation is to project the small eigenvalues to zero, the exact inversion of $\tilde{A}_{2h}$ is necessary. As the direct solver is not easy to parallelize and it is impractical for large problems, one has to turn to approximate solvers. However, an approximation of $\tilde{A}_{2h}$ will lead to a cluster of near-zero values in the preconditioned spectrum. Therefore, we choose to deflate toward the largest eigenvalues of the preconditioned system by adding a term. The so-called Two-Level Krylov method (TLKM) preconditioner [38] reads as

$$\tilde{P}_{h,\gamma} = \tilde{P}_h + \gamma\tilde{Q}_h = \left(I_h - \tilde{A}_h\tilde{Q}_h\right) + \gamma\tilde{Q}_h$$

with $\gamma = 1$ in our implementation, as the CSLP preconditioner results in eigenvalues of the preconditioned system being bounded by 1 in modulus. The preconditioned linear system to be solved is

$$\tilde{P}_{h,\gamma}\tilde{A}_h u_h = \tilde{P}_{h,\gamma}\tilde{b}_h$$

where $\tilde{b}_h = M_{h,(\beta_1,\beta_2)}^{-1}b_h$. Compared to the original Helmholtz system, the TLKM preconditioner is equivalent to

$$P_{h,TLKM} = \tilde{P}_{h,\gamma}M_{h,(\beta_1,\beta_2)}^{-1}$$

One can observe that TLKM needs an application of $M_{h,(\beta_1,\beta_2)}^{-1}$ on the fine grid for every coarse grid iteration. This makes it too expensive to use.

Alternatively, one can deflate the Helmholtz operator. The preconditioner becomes

$$P_{h,\gamma} = P_h + \gamma Q_h = \left(I_h - A_h Q_h\right) + \gamma Q_h$$

Combined with the standard preconditioner CSLP, which is commonly used in the Helmholtz system, a robust two-level preconditioned method, Adapted Deflation Variant 1 (A-DEF1) [39] reads as

$$P_{h,A-DEF1} = M_{h,(\beta_1,\beta_2)}^{-1}P_h + Q_h$$

The preconditioned linear system to be solved becomes

$$P_{h,A-DEF1}A_h u_h = P_{h,A-DEF1}b_h. \tag{10}$$

Note that $P_{h,A-DEF1}A_h$ is nonsingular, so eq. (10) has a unique solution. When the higher-order deflation vector is employed, it will be denoted as Adapted Preconditined Deflation (APD) [6].

We use the deflation preconditioner to enhance the convergence of GMRES-type methods. That is to pre-multiply the linear system with a preconditioning matrix P. It can be proven that there is no essential difference between right- and left-preconditioning methods with respect to convergence behavior [40]. Besides the full GMRES method, the preconditioned Generalized Conjugate Residual method (GCR) [41] will also be considered as it allows for variable preconditioning inherently.

4. Parallel matrix-free implementation

To develop a parallel scalable iterative solver for Helmholtz problems, we intend to parallelize the deflated Krylov subspace methods in a matrix-free framework, which allows us to tackle larger-scale problems.

We employ a parallel framework based on the MPI Cartesian topology, which partitions the computational domain blockwise. The parallel multigrid iteration is implemented on the global grid, with coarse grid parameters determined by the corresponding fine grid parameters using the index correspondence between coarse and fine grids. Restriction and interpolation operations are performed using stencils based on the index correspondence between coarse and fine grids, enabling a matrix-free implementation. The coarsening process stops when a predefined coarsest grid size is reached, and the coarsest grid problem is solved in parallel using GMRES. The coarsest grid size is chosen to ensure that each subdomain contains a sufficient number of grid points for efficient parallel computation. For example, we may set the coarsest grid size such that each subdomain contains at least 2×2 grid points. However, this approach leads to efficiency loss. The communication overhead between subdomains becomes significant.

4.1. Matrix-free Helmholtz and CSLP operator

In the Krylov subspace methods, one of the main operations is matrix-vector multiplication. They are the Helmholtz operator $y = Ax$ and the deflation operator $y = Px$, where y and x are arbitrary vectors. Besides, the inverse of the CSLP preconditioner is required in APD.

The Helmholtz operator is implemented using computational stencils on finite-difference grids. The multipliers for grid points are adjusted accordingly when physical boundary conditions are encountered. The matrix-free matrix-vector multiplication is performed using Algorithm 1.

The CSLP preconditioner M defined by (5) has a similar computational stencil to the Helmholtz operator. All applications of CSLP on fine or coarse grids use the matrix-free method proposed in [26]. The CSLP preconditioner is approximately inverted by one parallel geometric multigrid V-cycle using a re-discretization approach to obtain the coarse-grid operators.

Algorithm 1 Matrix-free Matrix-Vector Multiplication $\mathbf{v} = A\mathbf{u}$.

Input: Array $\mathbf{u}$; Output: Results of $\mathbf{v} = A\mathbf{u}$;
Initiate a_p, a_w, a_e, a_s and a_n;
Exchange interface boundaries data;
for $j = 2, ..., ny - 1$ **do**
  for $i = 2, ..., nx - 1$ **do**
    $v(i,j) = a_p u(i,j) + a_e u(i+1,j) + a_w u(i-1,j) + a_n u(i,j+1) + a_s u(i,j-1)$;
  end for
end for
for physical boundary grid points **do**
  Reset ap, aw, ae, as or an;
  $v(i,j) = a_p u(i,j) + a_e u(i+1,j) + a_w u(i-1,j) + a_n u(i,j+1) + a_s u(i,j-1)$;
end for
Return $\mathbf{v}$;

4.2. Matrix-free coarse-grid operator

The deflation operator can be applied in a matrix-free manner by leveraging the matrix-free application of its individual components, namely the inter-grid transfer operators and the CSLP preconditioner. The main matrix-vector multiplication of the coarse-grid problem, $y = A_{2h}x$, where x and y denote arbitrary vectors on the coarse grid, is a key operation in the deflation process. Generally, the coarse-grid operator A_{2h} can be built by Galerkin coarsening. However, in this way, one needs to construct the matrices and perform matrix-matrix multiplications, which do not fit our matrix-free framework.

In this section, we discuss various possibilities for implementing the coarse-grid operator in a matrix-free way. As one of the main contributions of this paper, we propose a novel finite-difference scheme called "Re-discretized scheme from Galerkin coarsening (ReD-Glk)" for the re-discretization of the coarse-grid problem. The ReD-Glk method approximates the results of the Galerkin coarsening approach while overcoming its limitations within the matrix-free framework. We also discuss other approaches, including the straightforward matrix-free Galerkin coarsening operator (str-Glk), the re-discretization approach employing the standard explicit second-order and fourth-order schemes, and a fourth-order compact finite-difference scheme.

4.2.1. Re-discretized scheme from Galerkin coarsening (ReD-Glk)

Combining the idea of the Galerkin coarsening and re-discretization approach, we propose to use the stencils from the results of the Galerkin coarsening operator as the finite-difference scheme to perform re-discretization on the coarse grid. In terms of the interior grid points, since the inter-grid operations and the Helmholtz operator can be represented by the so-called stencil notations, it is possible to express the result of $I_h^{2h} A_h I_{2h}^h$ by using a stencil notation.

Remark 1 *(Stencil Operations)*. Suppose $A \in \mathbb{C}^{n\times n}$ is symmetric and can be represented by the stencil

$$\begin{bmatrix} a_l & a_0 & a_r \end{bmatrix}$$

Matrix $B \in \mathbb{R}^{m\times n}$ can be represented by the stencil

$$\begin{bmatrix} b_l & b_0 & b_r \end{bmatrix}$$

and $C = BA$. Then the resulting stencil of C can be obtained by the following stencil operation,

$$\begin{aligned} [C] &= [B] \boxplus [A] \\ &= \begin{bmatrix} b_l & b_0 & b_r \end{bmatrix} \boxplus \begin{bmatrix} a_l & a_0 & a_r \end{bmatrix} \\ &= \begin{bmatrix} a_l b_r & a_l b_0 + a_0 b_r & a_l b_l + a_0 b_0 + a_r b_r & a_0 b_l + a_r b_0 & a_r b_l \end{bmatrix} \end{aligned}$$

where $\boxplus$ is denoted as the stencil operation symbolically.

Laplace operator If we only consider the internal grid points, using the higher-order deflation vectors I_h^{2h} and I_{2h}^h, we have

$$
\begin{aligned}
&[I_h^{2h}] \boxplus [-\Delta_h] \boxplus [I_{2h}^h] = \\
&\frac{1}{64}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}_h^{2h} \boxplus \frac{1}{h^2}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}_h \boxplus \frac{1}{64}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}_{2h}^{h} \\
&\Rightarrow [-\Delta_{2h}] = \frac{1}{(2h)^2} \cdot \frac{1}{256}\begin{bmatrix} -3 & -44 & -98 & -44 & -3 \\ -44 & -112 & 56 & -112 & -44 \\ -98 & 56 & 980 & 56 & -98 \\ -44 & -112 & 56 & -112 & -44 \\ -3 & -44 & -98 & -44 & -3 \end{bmatrix}_{2h}
\end{aligned} \tag{11}
$$

where $\boxplus$ indicates the operation between stencils. Taylor series expansion analysis shows that the finite difference scheme based on the stencil eq. (11) is a second-order approximation of 2D Laplacian, that is,

$$
-\Delta_{2h} u_{2h} = 4\left[-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} - \left(\frac{13}{48}\frac{\partial^4 u}{\partial x^4} + \frac{1}{2}\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{13}{48}\frac{\partial^4 u}{\partial y^4}\right)(2h)^2\right] + \mathcal{O}(h^4)
$$

Note that the higher-order deflation vectors enlarge the size of the stencil but maintain the accuracy of finite-difference discretization.

Constant wavenumber For the diagonal wavenumber matrix, a similar derivation can be made

$$
\begin{aligned}
&[I_h^{2h}] \boxplus [\mathcal{I}(k_{i,j}^2)_h] \boxplus [I_{2h}^h] = \\
&\frac{1}{64}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}_h^{2h} \boxplus \begin{bmatrix} 0 & 0 & 0 \\ 0 & k(i,j)^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}_h \boxplus \frac{1}{64}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}_{2h}^{h}
\end{aligned}
$$

where $(i,j) \in \Omega^h$. Note that, for non-constant wavenumber, the result will be a 5×5 stencil, of which each element contains the wavenumber on up to 25 fine grid points. This is complicated to implement and leads to extra work.

In order to obtain a simple stencil, suppose the wavenumber is a constant k, we will have

$$
[\mathcal{I}(k^2)_{2h}] = \frac{k^2}{64^2}\begin{bmatrix} 1 & 28 & 70 & 28 & 1 \\ 28 & 784 & 1960 & 784 & 28 \\ 70 & 1960 & 4900 & 1960 & 70 \\ 28 & 784 & 1960 & 784 & 28 \\ 1 & 28 & 70 & 28 & 1 \end{bmatrix}_{2h} \tag{12}
$$

Non-constant wavenumber The stencil of the wavenumber operator needs to be slightly modified for non-constant wavenumber cases. For each node of the stencil, the wavenumber on the corresponding coarse grid point is used. Then the stencil of the coarse-grid operator from Galerkin coarsening is obtained by

$$
[A_{2h}] = [-\Delta_{2h}] - [\mathcal{I}(k_{i_c,j_c}^2)_{2h}]
$$

Since the computational stencil is extended to 5×5, it is important to note a drawback of using a wider stencil. The memory access pattern for wider stencils becomes less efficient compared to 3×3 stencils, and additional communication and computation overhead will be required. While the wider stencils result in some performance decreases, our approach leverages the benefits of the matrix framework and scalable outer iterations to compensate for these potential drawbacks. The close-to wavenumber independent convergence presented in our paper demonstrates that the overall performance is not significantly affected by the use of wider stencils, and the potential improvements in convergence outweigh the additional overhead.

Boundary conditions Since they are 5×5 stencils, two grid points on the boundary should be considered. We can make use of the ghost grid point. Specifically, we can calculate the value of a ghost point using the method described in eq. (2) for Sommerfeld radiation boundary conditions. As for Dirichlet boundary conditions, one can approximate the value of the ghost point $u_{0,j}$ by the following relationship

$$
u_{1,j} = \frac{u_{0,j} + u_{2,j}}{2}
$$

Then, the aforementioned 5×5 stencils will be applicable on the second grid point of the boundary. It should be noted that one can set the wavenumber of the ghost point to zero without making any approximation, which is needed by eq. (12). The first point of the boundary can be discretized using the second-order method mentioned in Section 2.1. However, this simplification on the boundary may result in an extra number of iterations although it may keep the wavenumber-independent convergence, which will be reflected in the numerical results in the next section.

4.2.2. Re-discretization using standard finite-difference schemes (ReD)

An alternative is to obtain the operator A_{2h} by re-discretizing the Helmholtz operator on the coarse grid. Although it leads to the consequence that P_h is no longer a projection, this will not break the convergence in practical application. The natural way to re-discretize is to use the same discretization method as on the fine grid, that is, use the second-order finite difference scheme to discretize the Laplace operator, and then subtract the diagonal matrix of wavenumbers. In this way, we will get the same computational stencil as eq. (1) (denoted as **ReD-$\mathcal{O}$2**).

In addition, inspired by high-order deflation vectors [6], one can consider discretizing the Laplace operator using a fourth-order finite difference scheme (**ReD-$\mathcal{O}$4**) for the internal grid points (x_i, y_j), i.e.,

$$\frac{-u_{i-2,j} + 16u_{i-1,j} - 30u_{i,j} + 16u_{i+1,j} - u_{i+2,j}}{12h^2} = \frac{\partial^2 u}{\partial x^2}\left(x_i, y_j\right) + \mathcal{O}(h^4)$$

where $u_{i,j}$ denotes the value of function $u(x, y)$ on the grid point (x_i, y_j) (and similar for y-derivative). The complexity of the computational stencils obtained by using the re-discretization approach will be less than that of stencils from Galerkin coarsening. For example, if a fourth-order finite difference scheme is used, the stencil for the coarse-grid operator becomes

$$\left[A_{2h}\right] = \frac{1}{12 \cdot (2h)^2}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -16 & 0 & 0 \\ 1 & -16 & 60 - k_{2h}^2(i_c, j_c)(2h)^2 & -16 & 1 \\ 0 & 0 & -16 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}_{2h}$$

where $(i_c, j_c) \in \Omega^{2h}$.

As for the boundary, since the stencil is extended to 5×5, more boundary grid points need to be considered. In this paper, the stencils for the boundary grid points will be changed to that of the second-order re-discretization method, for instance, the Sommerfeld radiation boundary condition eq. (3) and eq. (4).

Additional experiments show that a higher-order re-discretization scheme for the boundaries will not make any improvement in the convergence compared to the second-order method.

Instead of using the explicit central finite-difference schemes, which do not fully use the entire template and the wavenumber information of adjacent grid points, one may think of a compact finite-difference approximation of the Helmholtz operator that has the following 3×3 stencil at grid point (i_c, j_c):

$$[A_{2h}] = \frac{1}{(2h)^2}\begin{bmatrix} a_c & a_s & a_c \\ a_s & a_0 & a_s \\ a_c & a_s & a_c \end{bmatrix} - \begin{bmatrix} b_c & b_s & b_c \\ b_s & b_0 & b_s \\ b_c & b_s & b_c \end{bmatrix} k^2 \tag{13}$$

where the coefficients a and b are non-zeros.

The fourth-order finite difference scheme introduced by Singer and Turkel [42] has a similar form as eq. (13). From [42], we have

$$a_0 = \frac{10}{3}, \quad a_s = -\frac{2}{3}, \quad a_c = -\frac{1}{6}$$
$$b_0 = \frac{2}{3} + \frac{\gamma}{36}, \quad b_s = \frac{1}{12} - \frac{\gamma}{72}, \quad b_c = \frac{\gamma}{144}$$

with γ an arbitrary constant. Here we will use $\gamma = 1$. We denote this approach as **ReD-cmp$\mathcal{O}$4**.

As for the boundary, we can use the fourth-order approximations of the Sommerfeld radiation boundary condition given by [43]. For instance, suppose $u_{0,j}$ is a ghost point on the left of $u_{1,j}$, it can be approximated by

$$u_{0,j} = 2ikh\left(1 - \frac{k^2h^2}{6}\right)u_{1,j} + u_{2,j}$$

The relationship between numerical dispersion and order of accuracy plays a crucial role in the performance of solvers for the Helmholtz equation. Turkel et al. [44] proposed a sixth-order compact scheme for a non-constant wavenumber and showed the connection between the order of accuracy and numerical dispersion. Therefore, higher-order accuracy is advantageous in terms of efficiency, as it allows for maintaining accuracy with fewer grid points for larger wave numbers. Chen et al. [45] investigated minimal dispersion when using the CSLP preconditioner and found that, for certain step sizes and large wavenumbers, a second-order scheme with minimal numerical dispersion may compete with a sixth-order scheme in terms of accuracy. Stolk [27] demonstrated that matching the phase slowness errors between the fine and coarse grid discretizations is essential for efficient multigrid convergence. A careful design of the discretization scheme (13) and exploring the impact of matching phase slowness errors between fine and

coarse grids on the overall convergence of our method could provide valuable insights and potentially lead to further improvements in performance. This avenue of research merits exploration in future studies.

4.2.3. Straightforward Galerkin coarsening operator (str-Glk)

The Galerkin coarsening operation can be computed in a matrix-free manner by performing the interpolation, fine-grid Helmholtz operation, and restriction sequentially, as shown in (14).

$$\text{Interpolation: } v_{i1} = I_{2h}^{h} v_i \tag{14a}$$

$$\text{Fine-grid operator: } v_{i2} = A_h v_{i1} \tag{14b}$$

$$\text{Restriction: } \hat{v}_i = I_{h}^{2h} v_{i2} \tag{14c}$$

This approach is essentially equivalent to the matrix-based Galerkin coarsening method presented in [6], but it is implemented here in a matrix-free framework. However, it should be noted that this method requires the application of the fine-grid Helmholtz operator A_h at every Krylov iteration on the coarse level, which can be computationally expensive.

5. Experimental results

The numerical experiments are conducted on the Linux supercomputer DelftBlue [46]. DelftBlue runs on the Red Hat Enterprise Linux 8 operating system. Each compute node is equipped with two Intel Xeon E5-6248R processors with 24 cores at 3.0 GHz, 192 GB of RAM, a memory bandwidth of 132 GByte/s per socket, and a 100 Gbit/s InfiniBand card. In our experiments, the solver is developed in Fortran 90. On DelftBlue, the code is compiled using GNU Fortran 8.5.0 with the compiler options `-O3` for optimization purposes. Open MPI library (version 4.1.1) is employed for message passing.

In the numerical experiments, the convergence test is based on the l_2-norm of the residual. Unless mentioned, the preconditioned relative residual is reduced to 10^{-6} by the deflated GMRES algorithm. Since it is a two-level method, it may still be expensive to solve the coarse-grid problem directly. The (CSLP-preconditioned) GMRES is employed for approximating the inverse of the coarse-grid operator. The (preconditioned) relative residuals on the coarse level are reduced to a certain tolerance. It is important to note that GMRES has the disadvantage of increasing the cost of each iteration as the algorithm proceeds. This can be a concern when a large number of iterations is required, as observed in some of our numerical experiments. We have explored the use of alternative solvers, such as Bi-CGSTAB, for the coarse-grid system. However, our experiments have shown that Bi-CGSTAB often requires significantly more iterations than GMRES to converge and may even diverge for model problems with large wavenumbers. Thus, a full GMRES solver is employed for the inner iterations to ensure stability and to enable a fair comparison with the exact coarse-grid solver. Moreover, to study the convergence properties of the deflation-preconditioned Krylov-subspace method for solving the Helmholtz problems, we aimed to obtain the approximation of A_{2h}^{-1} in the deflation process as accurately as possible. Therefore, we set the tolerance for the coarse-grid solver to 10^{-12}. Furthermore, we will explore the optimal tolerance for the coarse-grid solver in our study.

The inverse of the CSLP preconditioner is approximated by a multigrid V-cycle. According to [26], the stopping criterion for the coarsest grid preconditioner solver should be 2-3 orders of magnitude smaller than the stopping criterion for the outer iteration. Thus, a distinct full-GMRES is used to reduce the relative residuals to 10^{-8} on the predefined coarsest level of the V-cycle.

This section will mainly illustrate the numerical performance of our solver for the model problems on DelftBlue. The number of iterations (denoted as #iter) will be the main metric to estimate the convergence. The speedup and parallel efficiency are used to study the parallel performance. The Wall-clock time for the preconditioned Krylov solver to reach the stopping criterion is denoted as t_w. The speedup S_p is defined by

$$S_p = \frac{t_{w,r}}{t_{w,p}}$$

where $t_{w,r}$ is the Wall-clock time for the reference case and $t_{w,p}$ is the Wall-clock time for parallel computation. The parallel efficiency E_p is given by

$$E_P = \frac{S_p}{np/np_r} = \frac{t_{w,r} \cdot np_r}{t_{w,p} \cdot np}$$

where np (np_r) is the (reference) number of processors.

5.1. Complexity and spectral analysis of the coarse grid operators

Recalling from the previous section, there are the following possible coarse-grid operators in matrix-free:

- **str-Glk**: Straightforward Galerkin coarsening approach
- **ReD-$\mathcal{O}2$**: Re-discretization using the second-order finite-difference scheme
- **ReD-$\mathcal{O}4$**: Re-discretization using the fourth-order finite-difference scheme
- **ReD-cmp$\mathcal{O}4$**: Re-discretization using a scheme derived from the fourth-order compact finite difference of the Helmholtz equation

Table 1
Upper bound to the number of non-zero elements of each relevant matrix and the approximate FLOPs as well as the elapsed CPU time for performing the coarse grid operator 100 times. The coarse grid size is 7681×7681.

Methods	str-Glk			ReD-$\mathcal{O}2$	ReD-$\mathcal{O}4$	ReD-cmp$\mathcal{O}4$	ReD-Glk
Operators	I_h^{2h}	A_h	I_{2h}^h	A_{2h}	A_{2h}	A_{2h}	A_{2h}
Max. nnz per row	25	5	9	5	9	9	25
FLOPs	$50M + 10N + 18N$			$10M$	$18M$	$18M$	$50M$
CPU time	587.95			29.49	42.30	43.62	187.37

Table 2
The index of the near-zero eigenvalues for the fine- and coarse-grid operators. MP-1a with wavenumber $k = 50$ and $kh = 0.625$.

	A_h	A_{2h}				
Methods	FD-$\mathcal{O}2$	str-Glk	ReD-$\mathcal{O}2$	ReD-$\mathcal{O}4$	ReD-cmp$\mathcal{O}4$	ReD-Glk
$j_{\min}$	186, 187	186, 187	206, 207	188, 189	184, 185	186, 187

- **ReD-Glk**: Re-discretization using a scheme derived from the Galerkin coarsening approach (A ghost point is included for the second boundary grid point and ReD-$\mathcal{O}2$ is only for the first boundary grid point)

To give a preliminary comparison of the above methods for obtaining coarse grid operators, we conduct a rough complexity analysis to quantify the FLOPs for performing the coarse grid operator $y = A_{2h}x$ by different methods.

Suppose the number of fine grid points is N and that of the coarse grid is M, if the matrices are constructed explicitly, we will have $A_h \in \mathbb{R}^{N\times N}$, $A_{2h} \in \mathbb{R}^{M\times M}$, $I_h^{2h} \in \mathbb{R}^{M\times N}$ and $I_{2h}^h \in \mathbb{R}^{N\times M}$. In Table 1, we list the *nnz* of the relevant matrices and the approximate FLOPs of $A_{2h}x$ for different methods. Since $4M \approx N$, the total FLOPs of the straightforward Galerkin coarsening operator is around $162M$. To verify the results of the complexity analysis, we also test the elapsed CPU time for performing the coarse grid operators derived from the Galerkin coarsening approach. One can find the ratios between the elapsed CPU time are fairly consistent with the estimated FLOPs. From the complexity analysis of the coarse grid operator, without considering the overall convergence characteristics, the re-discretization approach seems to be preferable in the frame of matrix-free implementation.

To better understand the behavior of the proposed re-discretization approaches, we have performed a numerical spectral analysis for the coarse-grid operators obtained by different re-discretization approaches as well as the fine-grid operator. For problem MP-1a with wavenumber $k = 50$ and $kh = 0.625$, we numerically solve the eigenvalues of the coarse-grid operators in ascending order and find the index j_{min}, where the magnitude of eigenvalues $|\lambda^{j_{min}}(A)|$ is the smallest. As shown in Table 2, one can compare the index of the near-zero eigenvalues of the coarse-grid operators with the fine-grid operator to investigate their alignment. The results demonstrate that the proposed coarsening approaches effectively align the near-singular eigenmodes of the fine-grid and coarse-grid operators. This alignment is crucial for preventing the reappearance of near-zero eigenvalues and ensuring the effectiveness of the two-level deflation preconditioner.

5.2. Scalable convergence

As we introduced, in the context of solving the Helmholtz problem using iterative solvers, the number of iterations required increases significantly as the wavenumber increases. A recent study by Dwarka and Vuik [6] has shown promising results in achieving convergence properties that are independent of the wavenumber using high-order deflation preconditioning methods. In their work, the Galerkin coarsening approach is employed to derive the coarse-grid operator. In this subsection, we further explore various scenarios utilizing the re-discretization method for the coarse-grid operator to investigate which approach can achieve wavenumber-independent convergence. By examining different cases and comparing the outcomes, we aim to provide insights into the effectiveness of these methods.

5.2.1. Low-order deflation

To observe the possible impact on the convergence behavior of using the re-discretization approach, we first compare the convergence behavior of low-order deflation using the straightforward Galerkin coarsening and second-order re-discretization approaches. As mentioned, regardless of the computational cost, we can obtain the coarse-grid operator as eq. (14) in the frame of the Galerkin coarsening approach. Tables 3 and 4 give the number of outer iterations and Wall-clock time required to solve MP-1a using A-DEF1 preconditioned GMRES for different kh. In parentheses is the approximate number of iterations required to solve the coarse-grid problem (CGP) once by (preconditioned) GMRES. One can find the number of outer iterations is consistent with the results in [47] when the straightforward Galerkin coarsening approach is used. Since the coarse-grid problem has similar properties to the original Helmholtz problem, we would like to solve the coarse-grid problem combined with the CSLP preconditioner. The number of inner iterations shows that the CSLP preconditioner for the coarse-grid problem solver can accelerate the convergence on the coarse level, while the outer number of iterations remains the same. The Wall-clock time reflects that for large wavenumber problems, one can save significant computational cost by employing the CSLP preconditioner to solve the coarse-grid system. The idea of applying a

Table 3
The number of iterations and Wall-clock time t_w required to solve MP-1a using low-order A-DEF1 preconditioned GMRES for different k and kh. The coarse-grid problem (CGP) obtained by the str-Glk approach is solved by full-GMRES (GMRES) and CSLP preconditioned GMRES (CSLP-GMRES). In parentheses is the number of iterations required to solve the coarse grid problem once.

			(GMRES)		(CSLP-GMRES)	
Grid size	k	kh	#iter (coarse)	$t_w(s)$	#iter (coarse)	$t_w(s)$
33×33	20	0.625	8 (55)	0.03	8 (54)	0.14
65×65	40	0.625	16 (257)	1.14	16 (247)	2.09
129×129	80	0.625	39 (1058)	187.07	44 (903)	191.16
65×65	20	0.3125	6 (131)	0.14	6 (74)	0.24
129×129	40	0.3125	9 (527)	11.61	9 (256)	3.54
257×257	80	0.3125	18 (2217)	1285.90	19 (918)	325.04

Table 4
The number of iterations and Wall-clock time t_w required to solve MP-1a using low-order A-DEF1 preconditioned GMRES for different k and kh. The coarse-grid problem (CGP) obtained by ReD-$\mathcal{O}2$ approach is solved by full-GMRES (GMRES) and CSLP preconditioned GMRES (CSLP-GMRES). In parentheses is the number of iterations required to solve the coarse grid problem once.

			(GMRES)		(CSLP-GMRES)	
Grid size	k	kh	#iter (coarse)	$t_w(s)$	#iter (coarse)	$t_w(s)$
33×33	20	0.625	21 (82)	0.06	21 (61)	0.33
65×65	40	0.625	40 (328)	4.64	41 (251)	5.33
129×129	80	0.625	97 (1609)	1343.97	97 (1125)	767.11
65×65	20	0.3125	17 (123)	0.32	17 (59)	0.41
129×129	40	0.3125	20 (660)	31.37	20 (209)	5.53
257×257	80	0.3125	25 (2645)	2037.38	29 (840)	327.73

Table 5
The number of iterations and Wall-clock time t_w required to solve the 2D Wedge problem using low-order A-DEF1 preconditioned GMRES for different frequencies f and grid size. The coarse-grid problem (CGP) obtained by str-Glk or ReD-$\mathcal{O}2$ approach is solved by CSLP preconditioned GMRES. In parentheses is the number of iterations required to solve the coarse grid problem once.

			str-Glk		ReD-$\mathcal{O}2$	
Grid size	f	kh	#iter (coarse)	$t_w(s)$	#iter (coarse)	$t_w(s)$
73×121	10	0.35	8 (112)	2.91	24 (104)	5.93
145×241	20	0.35	9 (278)	19.15	31 (242)	43.88
289×481	40	0.35	12 (629)	348.25	34 (547)	642.49
577×961	80	0.35	19 (1448)	13079.7	41 (1220)	15674.43
145×241	10	0.175	7 (108)	4.57	25 (107)	12.28
289×481	20	0.175	7 (257)	54.35	31 (245)	162.96
577×961	40	0.175	8 (568)	829.33	33 (533)	2307.76

recursive strategy, where another two-level deflation is employed on the coarse level, is highly promising and aligns with our ongoing research. The scope of this paper is limited to the development and benchmarking of the parallel matrix-free two-level deflation approach, which provides a solid foundation for our future work on multilevel extensions.

As for the second-order re-discretization approach, Table 4 shows that the second-order re-discretized A_{2h} leads to an increase in the outer iterations. Table 5 confirms that our matrix-free two-level deflation methods also work for the case with a non-constant wavenumber and exhibit similar convergence behavior to the constant-wavenumber case.

A noteworthy observation from our results is the relationship between kh and the number of outer iterations when the wavenumber is fixed. The straightforward Galerkin coarsening approach (str-Glk), which is essentially equivalent to the matrix-based method, aligns with the established trend of fewer iterations for smaller kh values, as corroborated by [47]. When the second-order re-discretization approach (ReD-$\mathcal{O}2$) is employed, it consistently applies to the constant wavenumber problem, as shown in Table 4. However, for the Wedge problem with frequency $f = 10,\ 20,\ 40$ Hz in Table 5, a smaller kh does not equate to fewer outer iterations. Here we mainly aim to demonstrate the applicability of our matrix-free method to lower-order deflation and the convergence behavior versus the wavenumber, exploring the particular effects of kh across a larger variety of heterogeneity to reach a strong conclusion can be a direction for future research.

Table 6
The number of iterations required to solve MP-1b using APD-preconditioned GMRES for different k and kh. The coarse-grid operator is obtained by str-Glk, ReD-$\mathcal{O}2$, ReD-$\mathcal{O}4$, ReD-cmp$\mathcal{O}4$ and ReD-Glk, respectively. The coarse-grid problem is solved by CSLP preconditioned GMRES. In parentheses is the number of iterations required to solve the coarse grid problem once.

Grid size	k	kh	str-Glk	ReD-$\mathcal{O}2$	ReD-$\mathcal{O}4$	ReD-cmp$\mathcal{O}4$	ReD-Glk
65×65	40	0.625	7 (126)	20 (98)	17 (106)	19 (91)	9 (128)
129×129	80	0.625	7 (236)	30 (305)	18 (298)	20 (185)	9 (251)
257×257	160	0.625	7 (495)	87 (731)	19 (650)	23 (362)	9 (585)
513×513	320	0.625	7 (1030)	319 (1539)	23 (1330)	28 (690)	10(1276)
1025×1025	640	0.625	8 (2089)	1099 (3193)	34 (2662)	44 (1328)	11 (2590)
2049×2049	1280	0.625	9 (4230)	3417 (6750)	79 (5315)	109 (2660)	13 (5266)
129×129	40	0.3125	5 (142)	18 (76)	18 (81)	18 (76)	7 (144)
257×257	80	0.3125	5 (228)	19 (205)	18 (212)	18 (188)	7 (231)
513×513	160	0.3125	5 (411)	21 (438)	18 (458)	19 (395)	7 (432)
1025×1025	320	0.3125	5 (809)	28 (885)	20 (909)	20 (775)	6 (859)
2049×2049	640	0.3125	5 (1630)	53 (1722)	23 (1763)	24 (1506)	6 (1690)

Table 7
The number of iterations required to solve the 2D Wedge problem using APD-preconditioned GMRES for different frequencies f and grid size. The coarse-grid operator is obtained by str-Glk, ReD-$\mathcal{O}2$, ReD-$\mathcal{O}4$, ReD-cmp$\mathcal{O}4$ and ReD-Glk, respectively. The coarse-grid problem is solved by CSLP preconditioned GMRES. In parentheses is the number of iterations required to solve the coarse grid problem once.

Grid size	f	kh	str-Glk	ReD-$\mathcal{O}2$	ReD-$\mathcal{O}4$	ReD-cmp$\mathcal{O}4$	ReD-Glk
73×121	10	0.349	7 (145)	22 (104)	22 (108)	22 (99)	9 (138)
145×241	20	0.349	6 (301)	28 (244)	27 (243)	28 (228)	9 (303)
289×481	40	0.349	6 (580)	31 (535)	29 (519)	30 (491)	9 (585)
577×961	80	0.349	6 (1206)	37 (1200)	30 (1175)	31 (1069)	9 (1255)
1153×1921	160	0.349	6 (2799)	58 (2826)	34 (2721)	35 (2353)	8 (3032)

5.2.2. *Higher-order deflation*

To get scalable convergence with respect to the wavenumber, this section considers using the higher-order deflation vectors, *i.e.* APD.

Table 6 shows the number of iterations required to solve the constant wavenumber problem (MP-1b) using APD-preconditioned GMRES. We found that when the str-Glk approach is used for the coarse grid problem, the number of iterations required shows the wavenumber independence for the case with the wavenumber up to 10^3.

When the coarse grid operator is obtained by the ReD-$\mathcal{O}2$, it requires more outer iterations, which increase significantly with the increase of the wavenumber.

Obtaining the coarse grid operator by ReD-$\mathcal{O}4$ results in a slightly lower number of outer iterations compared to ReD-$\mathcal{O}2$. Nevertheless, it remains dependent on the wavenumber

As ReD-cmp$\mathcal{O}4$ have the same order of accuracy as ReD-$\mathcal{O}4$, little difference in outer iterations can be observed. The compact scheme (ReD-cmp$\mathcal{O}4$) offers advantages over the standard scheme (ReD-$\mathcal{O}4$) in terms of memory access and computational efficiency due to its compact stencil. The compact scheme significantly reduces the number of iterations required to solve the coarse grid problem, which is a notable benefit considering that the inner solver accounts for a significant portion of the computational cost.

As for using ReD-Glk, we can obtain considerable wavenumber independence. The number of outer iterations for using ReD-Glk is slightly more than that of str-Glk. This is due to the simplified treatment for the discretization of the boundary conditions mentioned in Section 4.2.1 as the operation on the internal grid points will be consistent with the Galerkin coarsening approach for the constant-wavenumber model.

For the non-constant wavenumber problem, the results for solving the so-called Wedge problem and the heterogeneous Marmousi problem are given in Table 7 and Table 8, respectively. The convergence behaviors exhibited by using different coarse grid operators are consistent with solving the constant wavenumber problem. For heterogeneous model problems, one can find that ReD-cmp$\mathcal{O}4$ and ReD-$\mathcal{O}4$ exhibit comparable performance. Thus, we will only present the results for ReD-$\mathcal{O}4$ in the subsequent numerical experiments. It is worth mentioning that the stencil eq. (12) in ReD-Glk is obtained based on the constant wavenumber assumption. But it also shows wavenumber independence in the non-constant wavenumber problem. This is consistent with the work of Dwarka and Vuik [6].

In terms of time consumption, see the Wall-clock time required to solve the Marmousi problem in Table 8, we found that ReD-Glk stands out as the most efficient choice among the re-discretization approaches and is comparable to that of the str-Glk method. Considering the superior complexity of the ReD-Glk method in terms of performing the coarse-grid operator one time, it will be cheaper than the str-Glk method when generalized to the potential multilevel deflation method. Thus, we consider ReD-Glk as the optimal coarse-grid re-discretization scheme. In subsequent numerical experiments, unless otherwise specified, ReD-Glk will be the default method.

Table 8
The number of iterations and Wall-clock time required to solve the 2D Marmousi problem using APD-preconditioned GMRES for different frequencies f and grid size ($kh = 0.5236$) with 12 processors. The coarse-grid operator is obtained by str-Glk, ReD-$\mathcal{O}2$, ReD-$\mathcal{O}4$, ReD-cmp$\mathcal{O}4$ and ReD-Glk, respectively. The coarse-grid problem is solved by CSLP preconditioned GMRES. In parentheses is the number of iterations required to solve the coarse grid problem once.

Grid size		737×241	1473×481	2945×961
f(Hz)		10	20	40
	#iter	7	7	7
str-Glk	(coarse)	(775)	(1858)	(5380)
	t_w (s)	120.67	1039.96	27496.67
	#iter	38	71	>200[a]
ReD-$\mathcal{O}2$	(coarse)	(748)	(1988)	(>5700)
	t_w (s)	615.54	9606.14	>432000
	#iter	30	34	50
ReD-$\mathcal{O}4$	(coarse)	(762)	(1947)	(5484)
	t_w (s)	402.79	4386.57	173650.06
	#iter	32	37	60
ReD-cmp$\mathcal{O}4$	(coarse)	(762)	(1947)	(5484)
	t_w (s)	316.53	5128.28	175379.27
	#iter	10	10	11
ReD-Glk	(coarse)	(802)	(1923)	(5592)
	t_w (s)	164.79	1376.40	46008.03

[a] ">" indicates it does not converge to the specified tolerance within the allowed maximum Wall-clock time.

Table 9
The number of APD-preconditioned GMRES iterations required to solve corresponding model problems when using different stopping criteria for the coarse-grid iterative solver.

Model Problems	10^{-1}	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-11}
MP-1b, $k = 320$, 513×513	14	10	10	10	10	10
MP-1b, $k = 320$, 1025×1025	11	7	6	6	6	6
Wedge, $f = 40$ Hz, 289×481	12	9	9	9	9	9
Marmousi, $f = 20$ Hz, 1473×481	13	10	10	10	10	10

5.3. Efficient coarse-grid solver

It appears that using the ReD-Glk format for coarse-grid re-discretization leads to near wavenumber-independent convergence. Despite this, a large number of iterations is still needed for the coarse-grid problem, especially for large wavenumbers. This is partially due to the fact that we are currently employing a two-level deflation method. Furthermore, as mentioned earlier, a strict tolerance is used for the convergence test of the coarse-grid solver. Consequently, this section explores the range of possible values for the tolerance of the coarse-grid solver.

Using higher-order deflation vectors and ReD-Glk for re-discretization of the coarse-grid system, the present APD-preconditioned GMRES is utilized to solve the linear system ($A_h u_h = b_h$), reducing the preconditioned relative residual to 10^{-6}. The stopping criteria for the coarse-grid iterative solver ($A_{2h} v_2 = v_1$) ranged from 10^{-1} to 10^{-13}. Table 9 shows that, for various model problems, the number of outer iterations required remains constant for all convergence criteria smaller than 10^{-2} on the coarse level, which is in line with the results in [12].

For example, in the scenario of the Marmousi problem ($f = 20$ Hz, grid size 1473×481), Fig. 4 displays the change in iterations required to solve a single coarse-grid problem with the CSLP-preconditioned GMRES solver, for various tolerance, along with the total Wall-clock time and the relative residual of the final solution. Note that the number of iterations for the CSLP-preconditioned GMRES solver to solve a single coarse-grid problem increases linearly as the size of the tolerance decreases. The Wall-clock time elapsed increases even faster as the tolerance becomes more stringent. Once the tolerance exceeds 10^{-5}, the relative residual of the final solution increases, while it remains constant for the tolerance less than 10^{-6}. This pattern holds for larger grid sizes.

The findings suggest that, when the present APD-preconditioned GMRES is used, maintaining the tolerance of the coarse-grid solver at the same order of magnitude as that of the outer iterations enables a consistent number of outer iterations and constant relative residual of the final solution while minimizing the total Wall-clock time. It is worth noting that for the GMRES algorithm, we employ left preconditioning and use the preconditioned relative residual as the criterion for convergence. Additional numerical experiments have been conducted to investigate the use of right preconditioning with GMRES, using the unpreconditioned relative residual as the convergence criterion. The results also demonstrate a consistent number of outer iterations across various tolerances for the coarse-grid solver, with only one more iteration compared to the results shown in Table 9, which strictly reduces the final relative residual to below 10^{-6}. Nevertheless, we can reach the same conclusion regarding the impact of the coarse-grid solver tolerance on the outer iterations and the relative residual of the solution. We attribute this to that the standard GMRES algorithm for outer

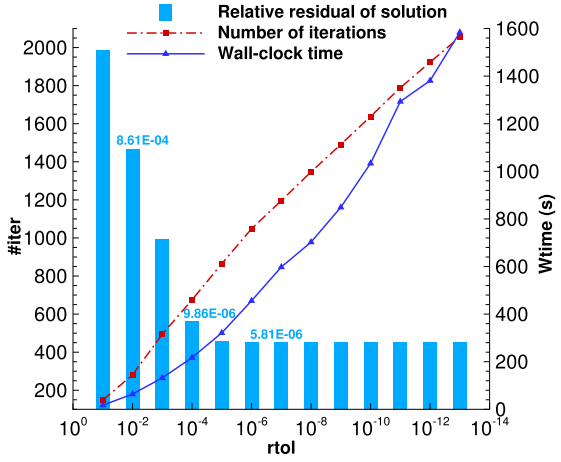


Fig. 4. Tolerance of the coarse-grid solver for Marmousi problem with $f = 20\,\text{Hz}$ and grid size 1473×481. The present APD-preconditioned GMRES is employed for outer iteration.

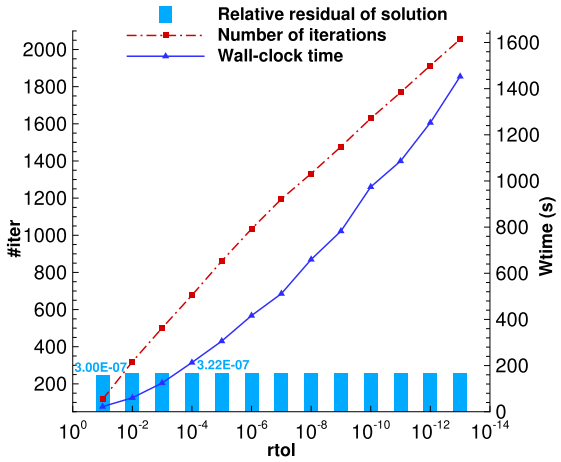


Fig. 5. Tolerance of the coarse-grid solver for Marmousi problem with $f = 20\,\text{Hz}$ and grid size 1473×481. The present APD-preconditioned GCR is employed for outer iteration.

iteration expects a constant preconditioner. However, if the tolerance of the coarse-grid solver increases beyond a certain threshold, it leads to variable preconditioning.

The results presented in Fig. 4 indicate that solving the coarse-grid problem for the Marmousi model with grid size 1473×481 and frequency $f = 20\,\text{Hz}$ requires over 1000 GMRES iterations to achieve a relative residual of 10^{-6}.

To investigate the possibility of using a larger tolerance for the coarse-grid solver and further reduce the solution time, we will proceed with the outer iteration using the GCR algorithm, which allows for a variable preconditioner. While making this transition, all other aspects of the deflation method remain unchanged, except for the use of the GCR algorithm with right preconditioning.

For the Marmousi model with grid size 1473×481 and frequency $f = 20\,\text{Hz}$, the number of outer GCR iterations remains constant at 11, when varying the tolerances of the coarse-grid solver from 10^{-2} to 10^{-12}. If the tolerance of the coarse-grid solver is set to 10^{-1}, it requires 12 outer iterations. The impact of the coarse-grid solver tolerance on the number of iterations required to solve the coarse-grid problem, the total Wall-clock time, and the relative residual of the final solution is illustrated in Fig. 5. The variation in the number of iterations for solving the coarse-grid problem and the total Wall-clock time is similar to that shown in Fig. 4. Similarly, when the tolerance of the coarse-grid solver is smaller than the tolerance of the outer iterations, the residual of the solution remains unchanged. However, when the tolerance of the coarse-grid solver exceeds the tolerance of the outer iterations, the residual of the solution remains within the tolerance instead of increasing, as observed in the standard GMRES method. The same behavior is observed in the case of the constant wavenumber model problems. Thus, using the GCR algorithm as the outer iteration method allows us to set the tolerance of the coarse-grid solver to a relatively large value 10^{-1}, while still maintaining the desired accuracy of the final solution. For the MP-1b model problem, if the APD-preconditioned GMRES is employed and the tolerance for the coarse-grid solver is set to 10^{-6}, the total Wall-clock time is 145.41 s. If the APD-preconditioned GCR is used and the tolerance for the coarse-grid solver is set to 10^{-1}, although one extra outer iteration is required, the total Wall-clock time is 9.48 s. The total computation time is reduced by around 95% of the time required when using a tolerance of 10^{-6} for the coarse-grid solver.

Table 10
The number of iterations and Wall-clock time required to solve the 2D Marmousi problem using APD-preconditioned GCR for different frequencies f and grid size ($kh = 0.5236$) with 12 processors. The coarse-grid problem is solved by CSLP preconditioned GMRES with a tolerance 10^{-1}. The coarse-grid operator is obtained by str-Glk, ReD-$\mathcal{O}2$, ReD-$\mathcal{O}4$ and ReD-Glk, respectively. In parentheses is the number of iterations required to solve the coarse grid problem once.

Grid size		737×241	1473×481	2945×961
f (Hz)		10	20	40
str-Glk	#iter	8	9	9
	(coarse)	(78)	(205)	(627)
	t_w (s)	2.47	14.00	282.82
ReD-$\mathcal{O}2$	#iter	40	71	233
	(coarse)	(34)	(171)	(373)
	t_w (s)	10.52	98.45	5359.75
ReD-$\mathcal{O}4$	#iter	33	35	41
	(coarse)	(27)	(70)	(162)
	t_w (s)	5.68	28.47	356.37
ReD-Glk	#iter	11	12	12
	(coarse)	(63)	(116)	(489)
	t_w (s)	3.25	14.62	255.38

Table 10 presents the number of iterations and wall-clock time required to solve the 2D Marmousi problem using the APD-preconditioned GCR method with different coarse-grid operators. The coarse-grid problem is solved using the CSLP-preconditioned GMRES method with a tolerance of 10^{-1}. In comparison to Table 8, there is a notable decrease in the number of iterations required for the coarse-grid problem and the total computation time when using a tolerance of 10^{-1}. The proposed ReD-Glk remains the top choice among the re-discretization approaches and now demands a similar Wall-clock time to the str-Glk method.

Additional numerical experiments have provided further evidence that regardless of whether the standard GMRES algorithm is employed with left or right preconditioning as the outer iteration algorithm, if the tolerance of the coarse-grid solver is set larger than that of the outer iterations, the residual of the final solution will increase. However, when utilizing the APD-preconditioned Flexible GMRES algorithm, which also allows for variable preconditioning, similar to the APD-preconditioned GCR approach, even with a relatively larger tolerance for the coarse-grid solver, the accuracy of the final solution can still be maintained within the specified tolerance range.

In summary, while solving the coarse-grid problem exactly is expected to provide the best convergence in the deflation method [48], the modified projection method employed in this study, as proposed by Erlangga and Nabben [13], is less sensitive to approximations of the coarse-grid system. Within a certain range of tolerances, the outer iterations do not change significantly. However, using a less strict tolerance of 10^{-1} does result in 2-3 extra outer iterations. Despite this, the computational savings achieved by solving the coarse-grid system with a looser tolerance outweigh the cost of the additional outer iterations. In the subsequent parallel performance study of the parallel preconditioned GMRES and GCR, we will set the tolerance for the coarse-grid problem solver to 10^{-6} and 10^{-1}, respectively.

To further reduce the number of iterations required to solve the coarse-grid problem, different complex shift values (β_2) for the CSLP preconditioner are investigated. Fig. 6 illustrates the impact of the complex shift of the CSLP preconditioner for solving the coarse-grid system. The results demonstrate that a shift larger than 0.5 is satisfactory, with a larger shift generally leading to a decrease in the number of iterations required to solve the coarse-grid system. This is because one multigrid V-cycle is employed to approximate the CSLP preconditioner. This observation aligns with the analysis presented in [5]. It provides a valuable insight that a shift of 1.0 for the CSLP preconditioner will save some iterations for solving the coarse-grid problem when we use the proposed matrix-free parallel two-level deflation method to solve more complex problems with larger wavenumber.

5.4. Parallel performance

In this section, we conduct a comprehensive evaluation of the parallel performance of our solver, focusing on weak scaling and strong scaling. Weak scaling refers to the solver's performance when the problem size and the number of processing elements increase proportionally. This allows us to assess our parallel solver's ability to solve large-scale heterogeneous Helmholtz problems with minimized pollution error by using smaller step size h. On the other hand, strong scaling measures the capability of our parallel solution method to solve a fixed problem size more quickly by adding more resources. Through these scaling assessments, we aim to understand the effectiveness of our solver in handling various problem sizes and resource allocations, providing crucial insights into its optimal operational parameters and potential scalability limits.

5.4.1. Weak scaling

Fig. 7 shows the results for the weak scalability test solving the MP-1b model problem with $k = 160$ from 1, 4, 16, up to 64 processes. The problem size was refined from 1025×1025, 2049×2049, 4097×4097, up to 8193×8193, ensuring that each process handled a grid size of approximately 1024×1024. One can find that, for the same problem and with the same number of processes,

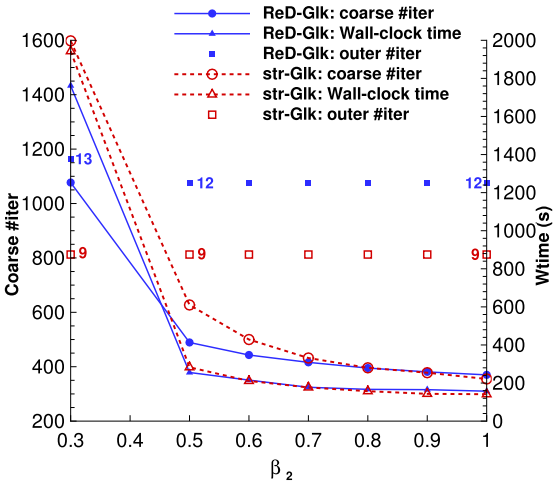


Fig. 6. Effect of the complex shift β_2. Marmousi problem with $f = 40$ Hz and grid size 2945×961. The present APD-preconditioned GCR is employed for outer iteration.

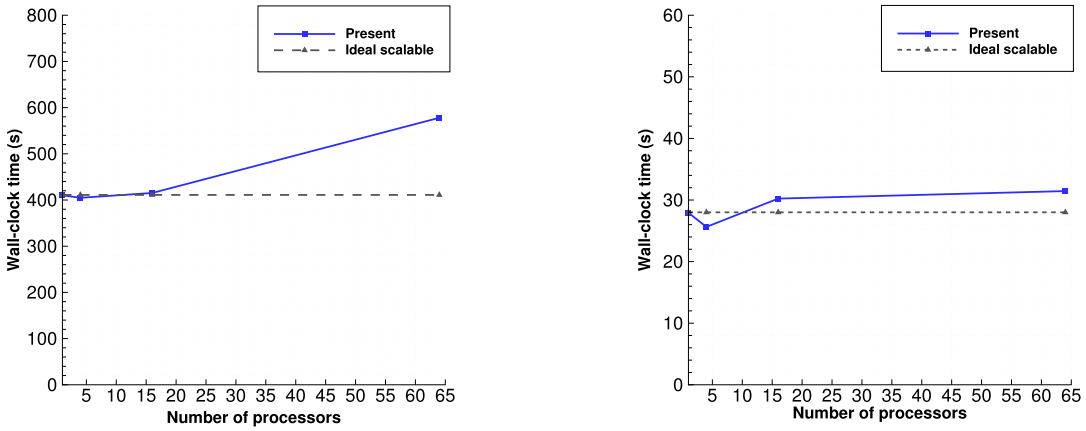


Fig. 7. Weak scaling for MP-1b with $k = 160$ and a grid size of 1024×1024 per process.

the APD-preconditioned GCR method with a coarse-grid solver tolerance of 10^{-1} exhibits a significantly reduced computational time compared to the APD-preconditioned GMRES method with a coarse-grid solver tolerance of 10^{-6}.

In Fig. 7, it appears that as the grid size and the processes increase proportionally, the required Wall-clock time does not remain perfectly constant but tends to increase slightly. This behavior may be attributed to the additional communication time required for larger problems and more processes. Additionally, the limited bandwidth also contributes to some increase in the Wall-clock time as the problem size grows larger. The same trend holds for model problems with non-constant wavenumbers, as shown in Table 11. The present weak scalability meets our requirements for minimizing pollution error by grid refinement within a certain range.

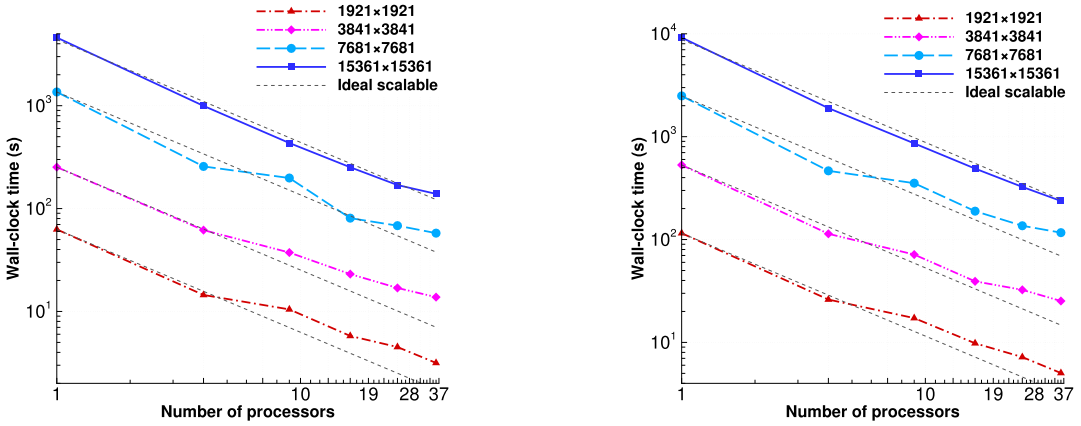
5.4.2. Strong scaling

We are also interested in the strong scalability properties of the present parallel deflation method for the Helmholtz problems. First of all, numerical experiments show that the number of external iterations required is found to be independent of the number of processes, which is a favorable property of our solution method.

We consider the MP-1b problem with a wavenumber of $k = 100$ and $k = 200$ on a growing number of processes and grid size. Fig. 8 presents the Wall-clock time versus the number of processes. The numerical experiment conducted on close to 48 processes exhibits a moderate decrease in terms of parallel efficiency. This can be partly attributed to the increased amount of communication, resulting in a significant decrease in the computation/communication ratio. If we increase the grid size, an improvement in parallel efficiency can be observed. This is because the number of ghost-grid layers used for communication remains constant, the amount of data to be communicated doubles when the number of grid points doubles in each direction. In contrast, the total number of grid points increases fourfold, resulting in a larger ratio of computational and communication time. For larger problems, a slight decrease

Table 11
Weak scaling for model problems with non-constant wavenumber. The present APD-preconditioned GMRES/GCR is employed for outer iteration.

		GMRES		GCR	
grid size	np	#iter	t_w (s)	#iter	t_w (s)
Wedge, $f = 40$ Hz					
577×961	6	9 (251)	53.41	10 (46)	4.86
1153×1921	24	10 (259)	68.59	10 (43)	5.75
Marmousi, $f = 10$ Hz					
737×241	3	10 (414)	103.92	11 (63)	10.55
1473×481	12	9 (378)	111.20	10 (58)	12.08
2945×961	48	9 (383)	156.45	10 (58)	17.72



(a) $k = 100$ (b) $k = 200$

Fig. 8. Strong scaling of APD-preconditioned GCR for MP-1b with various grid sizes.

in parallel efficiency when approaching 48 processes can be explained by the limited memory access bandwidth of a single compute node.

Furthermore, we also observe that the computational time increases nearly proportionally to the grid size, approximately by a factor of 4-5. However, the additional computational time required for data communication becomes increasingly evident. From Fig. 8, one can find that similar strong scalability holds for a larger wavenumber.

The strong scalability property of the present solver on multiple compute nodes is also investigated. Fig. 9 illustrates the time required to solve the Wedge model problem with a frequency of $f = 40$ Hz on at most 6 compute nodes (a total of 384 processes). For a grid size of 2305×3841, the numerical experiment performed on more than 96 processes exhibits a decrease in parallel efficiency. However, as the grid size increases to 4609×7681 and 9217×15361, which consequently increases the computation/communication ratio, the parallel efficiency significantly improves, with speedup even surpassing the ideal case, i.e., superlinear speedup. When parallel computing tasks can fully utilize the caches on multiple compute nodes, data access speeds are faster, thus enhancing computational efficiency. For solving the Wedge problem with a higher frequency of $f = 100$ Hz, as depicted in Fig. 9, we observe similar patterns to $f = 40$ Hz. Despite the constant number of outer iterations, the number of iterations required to solve the coarse-grid problem increases from around 45 to around 110. Consequently, the computation time also increases by a factor of 2-3. In terms of parallel efficiency, we observe similar behavior.

6. Conclusions

We propose a matrix-free parallel two-level deflation preconditioner for solving two-dimensional Helmholtz problems in both homogeneous and heterogeneous media.

By leveraging the matrix-free parallel framework and geometric multigrid-based CSLP [26], we provide a matrix-free parallelization method for the general two-level deflation preconditioner for the Helmholtz equation discretized with finite differences. Numerical experiments demonstrate that, compared with the Galerkin coarsening approach, the method of using re-discretization to obtain the coarse grid operator of deflation will slow down convergence to some extent.

To enhance the convergence properties, the higher-order approximation scheme proposed by Dwarka and Vuik [6] is employed to construct the deflation vectors. The study presents higher-order deflation vectors in two dimensions and their matrix-free parallel implementation. Various methods for implementing matrix-free coarse-grid operators are proposed and compared. The numerical

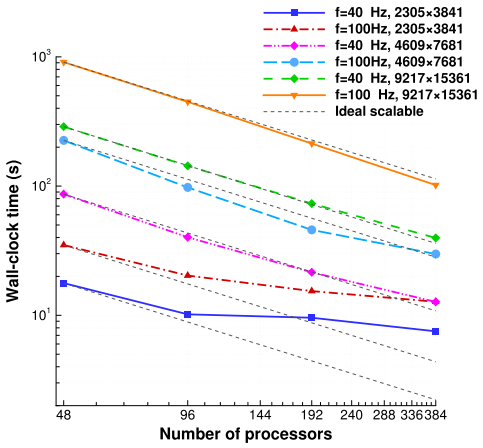


Fig. 9. Strong scaling of APD-preconditioned GCR for Wedge problem with $f = 40\,\text{Hz}$ and $f = 100\,\text{Hz}$.

experiments demonstrate the effectiveness of the proposed coarse-grid re-discretization scheme based on the Galerkin coarsening approach, which achieves wavenumber-independent convergence for both constant and non-constant wave number model problems.

Furthermore, we have studied in detail the tolerance setting of the coarse grid solver when the deflation preconditioned GMRES type methods are employed to solve the Helmholtz problems. An optimal tolerance for the coarse grid solver can effectively reduce the solution time.

Finally, the performance of the present parallel GCR solver preconditioned by higher-order deflation is studied, including both weak and strong scalability. Using a matrix-free approach to reduce memory requirements and weak scalability allows us to minimize pollution errors by refining the grid. Strong scalability allows us to solve high-frequency heterogeneous Helmholtz problems faster and more effectively in parallel. The presented two-level preconditioner serves as a benchmark for potential future developments of matrix-free parallel (multilevel) deflation preconditioners.

CRediT authorship contribution statement

Jinqiang Chen: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Vandana Dwarka:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Cornelis Vuik:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors only used ChatGPT 3.5 to improve the readability and language of the work. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

Acknowledgements

We would like to acknowledge the support of the Chinese Scholarship Council (CSC) scholarship (No. 202006230087).

References

[1] I.M. Babuska, S.A. Sauter, Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?, SIAM J. Numer. Anal. 34 (1997) 2392–2423, https://doi.org/10.1137/S0036142994269186.

[2] J.W. Pearson, J. Pestana, Preconditioners for Krylov subspace methods: an overview, GAMM-Mitt. 43 (2020) e202000015.

[3] A. Bayliss, C.I. Goldstein, E. Turkel, An iterative method for the Helmholtz equation, J. Comput. Phys. 49 (1983) 443–457, https://doi.org/10.1016/0021-9991(83)90139-0.

[4] Y.A. Erlangga, C. Vuik, C.W. Oosterlee, On a class of preconditioners for solving the Helmholtz equation, Appl. Numer. Math. 50 (2004) 409–425, https://doi.org/10.1016/j.apnum.2004.01.009.
[5] Y.A. Erlangga, C.W. Oosterlee, C. Vuik, A novel multigrid based preconditioner for heterogeneous Helmholtz problems, SIAM J. Sci. Comput. 27 (2006) 1471–1492, https://doi.org/10.1137/040615195.
[6] V. Dwarka, C. Vuik, Scalable convergence using two-level deflation preconditioning for the Helmholtz equation, SIAM J. Sci. Comput. 42 (2020) A901–A928, https://doi.org/10.1137/18M1192093.
[7] R.A. Nicolaides, Deflation of conjugate gradients with applications to boundary value problems, SIAM J. Numer. Anal. 24 (1987) 355–365, https://doi.org/10.1137/0724027.
[8] Z. Dostál, Conjugate gradient method with preconditioning by projector, Int. J. Comput. Math. 23 (1988) 315–323, https://doi.org/10.1080/00207168808803625.
[9] L. Mansfield, Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers, SIAM J. Sci. Stat. Comput. 12 (1991) 1314–1323, https://doi.org/10.1137/0912071.
[10] L.Y. Kolotilina, Twofold deflation preconditioning of linear algebraic systems. I. Theory, J. Math. Sci. 89 (1998) 1652–1689, https://doi.org/10.1007/BF02355371.
[11] Y. Saad, M. Yeung, J. Erhel, F. Guyomarc'h, A deflated version of the conjugate gradient algorithm, SIAM J. Sci. Comput. 21 (2000) 1909–1926, https://doi.org/10.1137/S1064829598339761.
[12] J.M. Tang, R. Nabben, C. Vuik, Y.A. Erlangga, Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods, J. Sci. Comput. 39 (2009) 340–370, https://doi.org/10.1007/s10915-009-9272-6.
[13] Y.A. Erlangga, R. Nabben, Multilevel projection-based nested Krylov iteration for boundary value problems, SIAM J. Sci. Comput. 30 (2008) 1572–1595, https://doi.org/10.1137/070684550.
[14] V. Dwarka, C. Vuik, Scalable multi-level deflation preconditioning for highly indefinite time-harmonic waves, J. Comput. Phys. 469 (2022) 111327, https://doi.org/10.1016/j.jcp.2022.111327.
[15] A.V. Kononov, C.D. Riyanti, S.W. de Leeuw, C.W. Oosterlee, C. Vuik, Numerical performance of a parallel solution method for a heterogeneous 2D Helmholtz equation, Comput. Vis. Sci. 11 (2007) 139–146, https://doi.org/10.1007/s00791-007-0069-6.
[16] C. Riyanti, A. Kononov, Y. Erlangga, C. Vuik, C. Oosterlee, R.-E. Plessix, W. Mulder, A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation, J. Comput. Phys. 224 (2007) 431–448, https://doi.org/10.1016/j.jcp.2007.03.033.
[17] D. Gordon, R. Gordon, Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers, J. Comput. Appl. Math. 237 (2013) 182–196, https://doi.org/10.1016/j.cam.2012.07.024.
[18] H. Calandra, S. Gratton, X. Pinel, X. Vasseur, An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media, Numer. Linear Algebra Appl. 20 (2013) 663–688, https://doi.org/10.1002/nla.1860.
[19] H. Calandra, S. Gratton, X. Vasseur, A geometric multigrid preconditioner for the solution of the Helmholtz equation in three-dimensional heterogeneous media on massively parallel computers, in: Modern Solvers for Helmholtz Problems, Springer, 2017, pp. 141–155.
[20] V. Dolean, P. Jolivet, F. Nataf, An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015.
[21] B. Engquist, L. Ying, Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers, Multiscale Model. Simul. 9 (2011) 686–710, https://doi.org/10.1137/100804644.
[22] C.C. Stolk, A rapidly converging domain decomposition method for the Helmholtz equation, J. Comput. Phys. 241 (2013) 240–252, https://doi.org/10.1016/j.jcp.2013.01.039.
[23] M.J. Gander, H. Zhang, A class of iterative solvers for the Helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods, SIAM Rev. 61 (2019) 3–76, https://doi.org/10.1137/16M109781X.
[24] M. Taus, L. Zepeda-Núñez, R.J. Hewett, L. Demanet, L-sweeps: a scalable, parallel preconditioner for the high-frequency Helmholtz equation, J. Comput. Phys. 420 (2020) 109706, https://doi.org/10.1016/j.jcp.2020.109706.
[25] E. Treister, R. Yovel, A hybrid shifted Laplacian multigrid and domain decomposition preconditioner for the elastic Helmholtz equations, J. Comput. Phys. 497 (2024) 112622, https://doi.org/10.1016/j.jcp.2023.112622.
[26] J. Chen, V. Dwarka, C. Vuik, Matrix-free parallel preconditioned iterative solvers for the 2D Helmholtz equation discretized with finite differences, in: Scientific Computing in Electrical Engineering, Springer Nature, Switzerland, 2024, pp. 61–68.
[27] C.C. Stolk, A dispersion minimizing scheme for the 3-D Helmholtz equation based on ray theory, J. Comput. Phys. 314 (2016) 618–646, https://doi.org/10.1016/j.jcp.2016.03.023.
[28] R.E. Plessix, W.A. Mulder, Separation-of-variables as a preconditioner for an iterative Helmholtz solver, Appl. Numer. Math. 44 (2003) 385–400, https://doi.org/10.1016/S0168-9274(02)00165-4.
[29] R. Versteeg, The marmousi experience: velocity model determination on a synthetic complex data set, Lead. Edge 13 (1994) 927–936, https://doi.org/10.1190/1.1437051.
[30] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, 2003.
[31] M.B. van Gijzen, Y.A. Erlangga, C. Vuik, Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted Laplacian, SIAM J. Sci. Comput. 29 (2007) 1942–1958, https://doi.org/10.1137/060661491.
[32] M.J. Gander, I.G. Graham, 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 (2015) 567–614, https://doi.org/10.1007/s00211-015-0700-2.
[33] S.P. MacLachlan, C.W. Oosterlee, Algebraic multigrid solvers for complex-valued matrices, SIAM J. Sci. Comput. 30 (2008) 1548–1571, https://doi.org/10.1137/070687232.
[34] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869, https://doi.org/10.1137/0907058.
[35] Y.A. Erlangga, A robust and efficient iterative method for the numerical solution of the Helmholtz equation, Ph.D. thesis, Delft University of Technology, 2005, http://resolver.tudelft.nl/uuid:af9be715-6ebf-4fc1-b948-ebd9d2c4167b.
[36] L.R. Hocking, C. Greif, Optimal complex relaxation parameters in multigrid for complex-shifted linear systems, SIAM J. Matrix Anal. Appl. 42 (2021) 475–502, https://doi.org/10.1137/20M1342161.
[37] J. Chen, V. Dwarka, C. Vuik, A matrix-free parallel solution method for the three-dimensional heterogeneous Helmholtz equation, Electron. Trans. Numer. Anal. 59 (2023) 270–294, https://doi.org/10.1553/etna_vol59s270.
[38] A.H. Sheikh, Development of the Helmholtz Solver based on a Shifted Laplace Preconditioner and a Multigrid Deflation technique, Thesis, TU Delft, 2014, 10.4233/uuid:1020f418-b488-4435-81ee-2b4f6a5024e1.
[39] J.M. Tang, Two-Level Preconditioned Conjugate Gradient Methods with Applications to Bubbly Flow Problem, Thesis, TU Delft, 2008, http://resolver.tudelft.nl/uuid:e8c5f63b-ee7d-4a59-90da-a8025f5f88b0.
[40] A. Sheikh, D. Lahaye, L. Garcia Ramos, R. Nabben, C. Vuik, Accelerating the shifted Laplace preconditioner for the Helmholtz equation by multilevel deflation, J. Comput. Phys. 322 (2016) 473–490, https://doi.org/10.1016/j.jcp.2016.06.025.
[41] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (1983) 345–357, https://doi.org/10.1137/0720023.

[42] I. Singer, E. Turkel, High-order finite difference methods for the Helmholtz equation, Comput. Methods Appl. Mech. Eng. 163 (1998) 343–358, https://doi.org/10.1016/S0045-7825(98)00023-1.
[43] Y. Erlangga, E. Turkel, Iterative schemes for high order compact discretizations to the exterior Helmholtz equation, ESAIM: Math. Model. Numer. Anal. 46 (2012) 647–660, https://doi.org/10.1051/m2an/2011063.
[44] E. Turkel, D. Gordon, R. Gordon, S. Tsynkov, Compact 2D and 3D sixth order schemes for the Helmholtz equation with variable wave number, J. Comput. Phys. 232 (2013) 272–287, https://doi.org/10.1016/j.jcp.2012.08.016.
[45] Z. Chen, D. Cheng, T. Wu, A dispersion minimizing finite difference scheme and preconditioned solver for the 3D Helmholtz equation, J. Comput. Phys. 231 (2012) 8152–8175, https://doi.org/10.1016/j.jcp.2012.07.048.
[46] Delft High Performance Computing Centre (DHPC), DelftBlue Supercomputer (Phase 1), https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1, 2022.
[47] A.H. Sheikh, D. Lahaye, C. Vuik, On the convergence of shifted Laplace preconditioner combined with multilevel deflation, Numer. Linear Algebra Appl. 20 (2013) 645–662, https://doi.org/10.1002/nla.1882.
[48] R. Nabben, C. Vuik, A comparison of deflation and the balancing preconditioner, SIAM J. Sci. Comput. 27 (2006) 1742–1759, https://doi.org/10.1137/040608246.