Journal of Scientific Computing (2025) 102:47
https://doi.org/10.1007/s10915-024-02786-w

Matrix-Free Parallel Scalable Multilevel Deflation Preconditioning for Heterogeneous Time-Harmonic Wave Problems

Jinqiang Chen [1] · Vandana Dwarka [1] · Cornelis Vuik[1]

Received: 1 August 2024 / Revised: 22 November 2024 / Accepted: 28 December 2024

Abstract

We present a matrix-free parallel scalable multilevel deflation preconditioned method for heterogeneous time-harmonic wave problems. Building on the higher-order deflation preconditioning proposed by Dwarka and Vuik (SIAM J. Sci. Comput. 42(2):A901-A928, 2020; J. Comput. Phys. 469:111327, 2022) for highly indefinite time-harmonic waves, we adapt these techniques for parallel implementation in the context of solving large-scale heterogeneous problems with minimal pollution error. Our proposed method integrates the Complex Shifted Laplacian preconditioner with deflation approaches. We employ higher-order deflation vectors and re-discretization schemes derived from the Galerkin coarsening approach for a matrix-free parallel implementation. We suggest a robust and efficient configuration of the matrix-free multilevel deflation method, which yields a close to wavenumber-independent convergence and good time efficiency. Numerical experiments demonstrate the effectiveness of our approach for increasingly complex model problems. The matrix-free implementation of the preconditioned Krylov subspace methods reduces memory consumption, and the parallel framework exhibits satisfactory parallel performance and weak parallel scalability. This work represents a significant step towards developing efficient, scalable, and parallel multilevel deflation preconditioning methods for large-scale real-world applications in wave propagation.

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

Mathematics Subject Classification 65Y05 · 65F08 · 35J05

✉ Jinqiang Chen
j.chen-11@tudelft.nl

Vandana Dwarka
v.n.s.r.dwarka@tudelft.nl

Cornelis Vuik
c.vuik@tudelft.nl

[1] Delft Institute of Applied Mathematics, Delft University of Technology, Delft, Netherlands

Published online: 03 January 2025

1 Introduction

The Helmholtz equation is a crucial mathematical model that describes the behavior of time-harmonic waves in various scientific fields, such as seismology, sonar technology, and medical imaging. While the classical formulation with standard Laplacian operator remains essential for many practical applications, researchers have also explored extended formulations to address specific physical phenomena. Recent studies have investigated Helmholtz equations in various forms, including elastic Helmholtz equations [43], the Stochastic Helmholtz equation [32], and Helmholtz equations with fractional Laplacian operators [1, 27]. In this work, we focus on the classical acoustics Helmholtz equation. Solving this equation numerically involves dealing with a sparse, symmetric, complex-valued, non-Hermitian, and indefinite linear system. For large-scale problems, iterative methods and parallel computing are commonly used. However, the indefiniteness of the system introduces significant difficulties, particularly when high frequencies are involved, as it limits the convergence of iterative solvers. Furthermore, to control pollution errors, it is essential to refine the grid so that $k^3h^2 < 1$ [3], where k is the wavenumber and h is the mesh size. The challenge of efficiently solving the Helmholtz equation while preserving high accuracy and minimizing pollution errors continues to be an active area of research. The development of a parallel scalable iterative method with convergence properties independent of the wavenumber could have far-reaching implications for various disciplines, including electromagnetics, seismology, and acoustics [37, 40].

A variety of preconditioners have been proposed for the Helmholtz problem, and among them, the Complex Shifted Laplace Preconditioner (CSLP) [20, 21] is one of the most popular options in the industry. The CSLP exhibits good properties for medium wavenumbers. However, the eigenvalues shift towards the origin as the wavenumber increases. As a result, the deflation method was introduced to accelerate the convergence of the CSLP-preconditioned Krylov subspace method [19, 35]. However, the number of iterations in both variations still gradually increases with the wavenumber k. In a recent development, Dwarka and Vuik [15] introduced higher-order approximation schemes to construct deflation vectors. This 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 [16]. Using higher-order deflation vectors, the authors demonstrate that the near-zero eigenvalues of the coarse-grid operators remain aligned with those of the fine-grid operator up to the level where the coarse-grid linear systems become negative indefinite. This alignment prevents the spectrum of the preconditioned system from approaching the origin. By combining this approach with the CSLP preconditioner, the authors achieved an iterative method with close to wavenumber-independent convergence for highly indefinite linear systems. Incorporating the deflation preconditioner has resulted in improved convergence; however, it has an impact on efficiency in relation to memory and computational cost. A promising branch is the use of Domain Decomposition Methods (DDM) as preconditioning techniques. These methods typically require two essential components: carefully designed transmission conditions and problem-adapted coarse spaces. Notable developments include the DtN and GenEO spectral coarse spaces [4], which utilize selected modes from local eigenvalue problems specifically tailored to the Helmholtz equation. For comprehensive surveys on DDM preconditioners for the Helmholtz problems, we refer the reader to [22] and the references therein. An alternative direction is the Multiscale Generalized Finite Element Method (MS-GFEM) [2, 30]. Recent work by Ma et al. [10] has successfully applied MS-GFEM with novel local approximation spaces to high-frequency heterogeneous Helmholtz problems. While the deflation method achieves near wavenumber-independent convergence by aligning the near-zero eigenvalues between coarse and fine-grid operators, the discrete MS-GFEM approach solves the problem in one shot without iterating based on solving some carefully-designed local problems and a global coarse problem, suggesting potential benefits in combining both approaches for future research.

Efforts are also underway to develop parallel scalable Helmholtz solvers. With well-designed parallelization strategies, domain decomposition methods have shown promise in reducing the number of iterations and improving computational efficiency [39]. Another approach is the parallelization of existing advanced algorithms. Parallel implementations of Bi-CGSTAB preconditioned by multigrid-based CSLP have been presented for 2D and 3D forward modeling by Kononov and Riyanti [26, 33], respectively. Gordon and Gordon [23] proposed the block-wise parallel extension of their so-called CARP-CG algorithm (Conjugate Gradient Acceleration of CARP). The block-parallel CARP-CG algorithm shows improved scalability as the wavenumber increases. Calandra et al. [5, 6] proposed a geometric two-grid preconditioner for 3D Helmholtz problems, which exhibits strong scaling in massively parallel setups.

Although conventional multigrid methods using standard smoothing and coarse grid corrections fail for the Helmholtz equation, their high efficiency in solving positive definite problems has motivated research into developing robust multilevel approaches for this equation [16, 17, 25, 29]. While previous works have established the theoretical foundations for multilevel deflation methods [16, 35, 36, 38], our focus is on a parallel scalable implementation of multilevel deflation for practical large-scale applications. We aim to perform comprehensive numerical experiments to validate the theoretical predictions and demonstrate the method's effectiveness in large-scale scenarios, where parallel implementation challenges often exceed idealized theoretical assumptions.

Based on the parallel framework of the CSLP-preconditioned Krylov subspace methods [7, 8], a matrix-free parallel two-level deflation preconditioning [9] has been implemented recently. To the author's knowledge, there is no existing literature on parallel multilevel deflation so far. This paper addresses this gap by proposing a matrix-free, parallel scalable multilevel deflation preconditioning method. In this work, we explore methods to extend the wavenumber-independent convergence from the two-level to a multilevel setting. This work presents significant innovations in solving large-scale Helmholtz problems. We develop novel re-discretization schemes for multilevel hierarchies, ensuring effective approximation of Galerkin coarsening operators across all levels while maintaining the matrix-free parallel framework. Through comprehensive numerical experiments, we establish optimal parameters for robust convergence across different problem scales. Furthermore, we introduce a controllable tolerance for coarse-level iterations, a previously unexplored but essential component for achieving wavenumber-independent convergence in practical multilevel deflation methods. These innovations culminate in a highly efficient parallel framework that demonstrates both wavenumber-independent convergence and excellent scaling properties in massively parallel environments, as validated by extensive numerical experiments.

The paper is organized as follows. In Sect. 2, we begin by describing our model problems. We present the matrix-free parallel variant of the multilevel deflated Krylov method in Sect. 3. Section 4 presents an optimally tuned configuration of the matrix-free parallel multilevel deflation method. Finally, we present numerical results to evaluate parallel performance in Sect. 5. Section 6 contains our conclusions.

2 Problem Description

We mainly consider the following two-dimensional mathematical model. Suppose that the domain Ω is rectangular with a boundary $\Gamma = \partial\Omega$. The Helmholtz equation reads

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

supplied with either Dirichlet or Sommerfeld radiation boundary conditions. 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

The computational domain is discretized using structural vertex-centered grids with uniform mesh width h. The discrete approximation of $u(x, y)$ is denoted as $u(i, j)$ or $u_{i,j}$, where grid points (x_i, y_j) are given by $x_i = x_1 + (i-1)h$ and $y_j = y_1 + (j-1)h$.

A second-order finite difference scheme for a 2D Laplace operator has a well-known 3 × 3 stencil. Similarly, we denote a computation stencil for the wavenumber term in the Helmholtz equation as

$$\left[\mathcal{I}(k_{i,j}^2)_h\right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_{i,j}^2 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $\mathcal{I}(k_{i,j}^2)$ represents a diagonal matrix with $k_{i,j}^2$ as its diagonal elements. The stencil of the Helmholtz operator A_h can be obtained by

$$[A_h] = [-\Delta_h] - \left[\mathcal{I}(k_{i,j}^2)_h\right]. \tag{1}$$

For boundary conditions, a ghost point located outside the boundary points can be introduced. For instance, suppose $u_{0,j}$ is a ghost point on the left of $u_{1,j}$, for Sommerfeld radiation boundary condition, we have

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

For the Dirichlet boundary condition, we have

$$u_{0,j} = 2u_{1,j} - u_{2,j}. \tag{3}$$

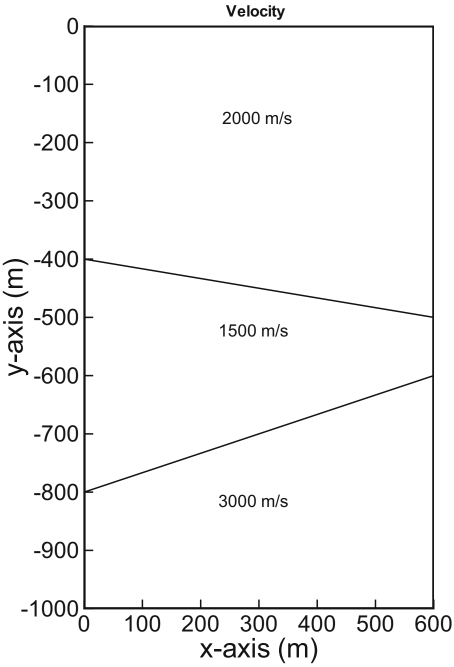
Discretization of the partial equation on the finite-difference grids results in a system of linear equations $A_h\mathbf{u}_h = \mathbf{b}_h$. With first-order Sommerfeld radiation boundary conditions, the resulting matrix is sparse, symmetric, complex-valued, indefinite, and non-Hermitian.

Note that kh is an important parameter that indicates 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 $kh = 0.625$.

Fig. 1 The velocity distribution of the Wedge problem



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 defined by a Dirac delta function 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 boundary 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 [31] is a typical problem with 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 wave velocity $c(x, y)$ is shown in Fig. 1. 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 [41], a well-known benchmark problem. It contains 158 horizontal layers in the depth direction, making it highly heterogeneous. 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.

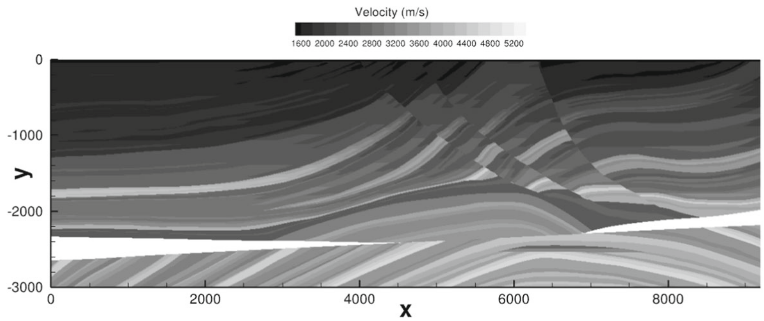


Fig. 2 The velocity distribution of the Marmousi problem

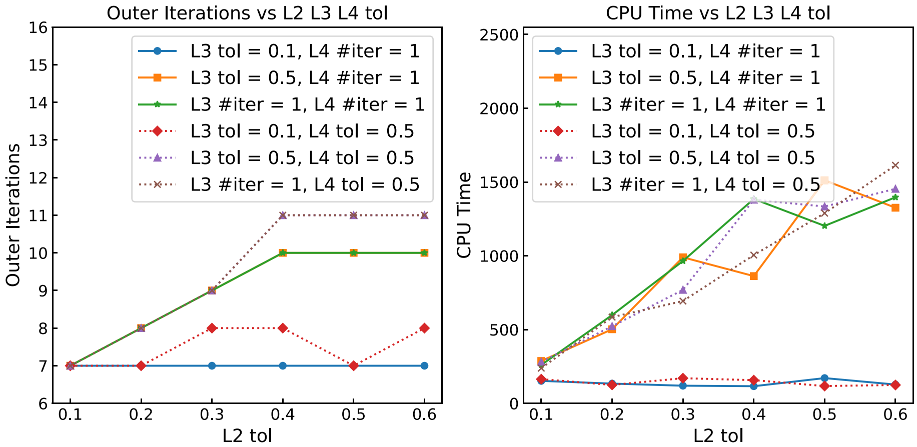


Fig. 3 Outer iterations and CPU time vary from different tolerances on the second (L2), third (L3), and fourth (L4) levels. Five-level deflation for Marmousi problem, grid size 1473 × 481, $f = 10$Hz

3 Deflated Krylov Methods

We review the two-level deflation preconditioning and a preliminary multilevel setting, then detail their adaptation to a matrix-free parallel framework.

3.1 Two-Level Deflation

Suppose a general nonsingular 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

$$P = I - AQ, \quad Q = ZE^{-1}Z^T, \quad E = Z^T AZ. \tag{4}$$

The observation that multigrid inter-grid operators emphasize small frequencies and preserve them on coarser levels leads to the possibility of using geometrically constructed multigrid vectors as deflation vectors. In such scenarios, we refer to E as the coarse grid Helmholtz operator, which exhibits similar properties to that of A.

The CSLP-preconditioner can be included to obtain even better convergence. The CSLP preconditioner $M_{(\beta_1,\beta_2)}$ is defined by

$$M_{(\beta_1,\beta_2)} = -\Delta - (\beta_1 + \mathrm{i}\beta_2)\mathcal{I}(k^2_{i,j}), \tag{5}$$

where $\mathrm{i} = \sqrt{-1}$, $(\beta_1, \beta_2) \in [0, 1]$. The solution of complex-shifted Laplacian systems, which arise from approximating the inverse of CSLP, has been extensively studied in the literature. Various approaches have been proposed, including multigrid methods [11, 20] and Krylov subspace methods [16, 24]. A recent alternative approach by [28] introduces absolute-value based preconditioners for the equivalent block real linear system formulation of the complex-shifted Laplacian system. The multigrid-based CSLP mentioned in this paper will fully adopt the matrix-free parallel framework and settings proposed in [8].

To allow approximate solvers for E^{-1}, one can prevent close-to-zero eigenvalues from obstructing the convergence of the Krylov solver by adding a term Q to deflate toward the largest eigenvalues of the preconditioned system [38]. As a result, the Adapted Deflation Variant 1 (A-DEF1) preconditioner P reads as

$$P := M^{-1}_{(\beta_1,\beta_2)} P + Q.$$

When higher-order Bezier curves are used to construct high-order deflation vectors Z, this results in the so-called Adapted Deflation Preconditioner (ADP) [15]. The preconditioned linear system to be solved becomes

$$PAu = Pb. \tag{6}$$

Note that PA is nonsingular, so Eq. (6) has a unique solution. In this work, we will stick to the use of the higher-order deflation vectors.

3.2 Multilevel Deflation

When using the two-level method in practical large-scale applications, solving the coarse-grid system remains expensive, whether solved exactly [16], or approximately by CSLP-preconditioned Krylov methods [9]. In accordance with the multigrid method, as the coarse grid system has similar properties as the original Helmholtz operator, one can obtain a multilevel framework by applying the two-level cycle recursively, as shown in Algorithm 1. The flexible subspace Krylov method such as FGMRES preconditioned by two-level deflation is applied recursively on subsequent coarse-grid systems E. Compared to the multilevel deflation proposed in [16], a few remarks are noted here.

First, Algorithm 1 does not include the process of determining the corresponding coarser-grid system E and CSLP preconditioner M on the current level l. This will be elaborated on in the next subsection.

Second, for efficient parallelization, we employ a GMRES method to solve the coarsest grid problem approximately rather than a direct solver. This method is preconditioned by CSLP, which is defined according to the coarsest grid operator. We will numerically investigate the necessary accuracy (or the number of iterations) for solving the coarsest grid problem in the next section.

Third, the multilevel deflation method requires approximating the inverse of CSLP on each level. In [16], this is accomplished using several Krylov subspace iterations (e.g., Bi-CGSTAB) on all levels. The authors set the maximum number of iterations to $C_{it}(N^l)^{\frac{1}{4}}$, where C_{it} is a constant and N^l denotes the problem size on level l. This strategy allows the

Algorithm 1 Recursive two-level deflated FGMRES: `TLADP-FGMRES(A, b)`

```
 1: Determine the current level $l$ and dimension $m$ of the Krylov subspace
 2: Initialize $u_0$, compute $r_0 = b - Au_0$, $\beta = ||r_0||$, $v_1 = r_0/\beta$;
 3: Define $\bar{H}_m \in \mathbb{C}^{(m+1)\times m}$ and initialize to zero
 4: for $j = 1, 2, \dots, m$ or until convergence do
 5:     $\hat{v}_j = Z^T v_j$                                  ▷ Restriction
 6:     if $l + 1 == ml$ then                                  ▷ Predefined coarsest level $ml$
 7:         $\tilde{v} \approx E^{-1}\hat{v}$                  ▷ Approximated by CSLP-FGMRES
 8:     else
 9:         $l \leftarrow l + 1$
10:         $\tilde{v} \leftarrow$ TLADP-FGMRES(E, $\hat{v}$)   ▷ Apply two-level deflation recursively
11:     end if
12:     $t = Z\tilde{v}$                                       ▷ Interpolation
13:     $s = At$
14:     $\tilde{r} = v_j - s$
15:     $r \approx M^{-1}\tilde{r}$                            ▷ CSLP, by multigrid method or Krylov iterations
16:     $x_j = r + t$
17:     $w = Ax_j$
18:     for $i := 1, 2, \dots, j$ do
19:         $h_{i,j} = (w, v_i)$
20:         $w \leftarrow w - h_{i,j} v_i$
21:     end for
22:     $h_{j+1,j} := ||w||_2$, $v_{j+1} = w/h_{j+1,j}$
23: end for
24: $X_m = [x_1, \dots, x_m]$, $\bar{H}_m = \{h_{i,j}\}_{1 \le i \le j+1, 1 \le j \le m}$
25: $u_m = u_0 + X_m y_m$ where $y_m = \arg\min_y ||\beta e_1 - \bar{H}_m y||$
26: Return $u_m$
```

benefits of using a small shift, resulting in a preconditioner similar to the original Helmholtz operator that retains the ability to shift indefiniteness at certain levels. However, the maximum number of iterations is positively correlated with the grid size on each level, indicating that larger grid sizes require more iterations. Considering the large-scale applications, utilizing Krylov subspace iterations on the first level (finest grid) or the second level may become computationally intensive. Therefore, we propose employing a multigrid cycle to approximate CSLP on the first or second level. Several Krylov subspace iterations can then be applied on the coarser levels. However, in the case of multigrid-based CSLP, ensuring a sufficiently large complex shift is essential. In addition to setting the maximum number of iterations, a relative tolerance as stopping criteria for the iterations is also established in this paper. This allows iterations to cease once the maximum number of iterations or the tolerance is reached.

Fourth, as shown in Algorithm 1, the number of deflated FGMRES iterations is specified by m. The cycle type of the multilevel deflation technique is determined by the number of iterations of the deflated FGMRES on each coarse level, except for the finest level. If only one iteration is allowed on the coarser levels, a V-cycle is obtained, which is similar to the V-cycle structure of multigrid when $\gamma = 1$. Correspondingly, two iterations on the coarser levels will result in a W-cycle. According to the multigrid method, the W-cycle may offer faster convergence than V-cycle but at the expense of computational efficiency. For increasingly complex model problems, striking a balance between optimal convergence and computational efficiency in the selection of m, hence determining the necessary accuracy (or the number of iterations) for the coarser levels, will be a focal point of this study.

Fifth, in Algorithm 1, all involved matrix–vector multiplications (lines 5, 7, 10, 12, 13, 15, 17) are expressed to denote the outcome of these operations. In our implementation, we compute and return the result of matrix–vector multiplication based on input variables through a linear combination. No explicit construction of any matrices takes place in our approach.

3.3 Matrix-Free Parallel Implementation

Matrix-free implementations offer a compelling alternative to standard sparse matrix data formats in large-scale computational scenarios. Besides the reduced memory consumption, matrix-free methods exhibit performance advantages and can potentially outperform matrix–vector multiplications with stored matrices [14]. Through roofline model analysis (detailed in Appendix B), we demonstrate that our approach achieves 2.35 times higher arithmetic intensity compared to traditional CSR matrix-based implementations. This theoretical advantage translates to substantial performance gains, particularly for large-scale problems. These improvements enable the solution of larger-scale Helmholtz problems previously constrained by memory limitations, while also enhancing the applicability of modern data-driven methods [13]. This section details the matrix-free implementation of operators in the multilevel deflation method.

Matrix-free matrix–vector multiplication is implemented using stencil notation. The computational stencils for both the finest-level and second-level operators (Helmholtz and CSLP preconditioner) and grid-transfer operators (higher-order interpolation and restriction) are detailed in [9]. To enable true multilevel deflation, we extend the matrix-free implementation to coarser levels. We denote the Helmholtz operators as $A_{2^{l-1}h}$ and the CSLP operators as $M_{2^{l-1}h}$, where l is the level number ($l = 1$ represents the finest grid). Starting from the second-level grid, we want to find the computational stencils for A_{4h} so that it is a good approximation to the Galerkin coarsening operator $Z^T A_{2h} Z$. Following [9], we decompose the Helmholtz operator into Laplace and wavenumber operators (assuming a constant wavenumber). By applying Galerkin coarsening operations to their stencils, we obtain the following stencils of the Laplace and wavenumber operators for interior points on the third-level coarse grid:

$$[-\Delta_{4h}] = \frac{1}{4096} \cdot \frac{1}{1024} \cdot \frac{1}{h^2} \cdot \begin{bmatrix} -3 & -534 & -5773 & -11956 & -5773 & -534 & -3 \\ -534 & -32844 & -207370 & -354088 & -207370 & -32844 & -534 \\ -5773 & -207370 & -294371 & 384244 & -294371 & -207370 & -5773 \\ -11956 & -354088 & 384244 & 2945488 & 384244 & -354088 & -11956 \\ -5773 & -207370 & -294371 & 384244 & -294371 & -207370 & -5773 \\ -534 & -32844 & -207370 & -354088 & -207370 & -32844 & -534 \\ -3 & -534 & -5773 & -11956 & -5773 & -534 & -3 \end{bmatrix}_{4h},$$

$$[\mathcal{I}(k^2)_{4h}] = \frac{1}{4096} \cdot \frac{1}{4096} \cdot k^2 \cdot \begin{bmatrix} 1 & 322 & 3823 & 8092 & 3823 & 322 & 1 \\ 322 & 103684 & 1231006 & 2605624 & 1231006 & 103684 & 322 \\ 3823 & 1231006 & 14615329 & 30935716 & 14615329 & 1231006 & 3823 \\ 8092 & 2605624 & 30935716 & 65480464 & 30935716 & 2605624 & 8092 \\ 3823 & 1231006 & 14615329 & 30935716 & 14615329 & 1231006 & 3823 \\ 322 & 103684 & 1231006 & 2605624 & 1231006 & 103684 & 322 \\ 1 & 322 & 3823 & 8092 & 3823 & 322 & 1 \end{bmatrix}_{4h}.$$

Using these stencils, the Helmholtz operator and CSLP operator on the third level can be obtained according to their definitions Eqs. (1) and (5), respectively.

Continuing this process iteratively, we can obtain stencils for the Helmholtz operator on coarser levels. It should be noted that starting from the third level, the size of the computation stencils will remain at 7 × 7. Specific stencils for the fourth to sixth levels can be found in Appendix A.

Boundary Introducing an accurate boundary scheme for the aforementioned 7 × 7 computational stencils remains an open problem. In this paper, we present a simple yet effective approach, involving the introduction of a ghost point outside the physical boundaries, as depicted in Eqs. (2) and (3). We apply standard second-order finite-difference discretization to points on the physical boundary. For points near the boundary, we set additional grid points beyond the ghost point to zero. It is important to note that the wavenumbers of the ghost points are also required. For Dirichlet boundary conditions, we determine the wavenumbers of the ghost points similar to Eq. (3). In other cases, the wavenumbers of the ghost points are uniformly set to zero. This zero-padding approach is motivated by the observation that the coefficients outside the 3 × 3 kernel become small, and thus, the influence of these points on the overall solution is expected to be minimal. By setting these points to zero, we aim to simplify the computation while maintaining the accuracy of the solution.

To develop a parallel scalable iterative solver, the matrix-free multilevel deflated Krylov subspace methods are implemented within the parallel framework presented by [8, 9].

4 Configuration

Before presenting the performance analysis of the matrix-free multilevel deflation method, we systematically tune the essential components of the algorithm to achieve an optimal balance between computational efficiency and numerical robustness. This section establishes the precise configuration that ensures wavenumber-independent convergence while minimizing computational overhead for complex numerical applications. The outer FGMRES iterations start with a zero initial guess and terminate when the relative residual in Euclidean norm reaches 10^{-6}. Note that all presented results in this section are obtained from sequential computations. In our notation, Ln represents the n-th level in the multigrid hierarchy, where L1 corresponds to the finest level.

4.1 Tolerance for Solving the Coarsest Problem

In this subsection, we explore the tolerance considerations for solving the coarsest problem. For better comparison and fewer other influencing factors, we perform a V-cycle three-level deflation approach to solve the constant wavenumber model problem with Sommerfeld radiation boundary conditions. The finest level represents the first level, and the third level corresponds to the coarsest problem, which will be addressed using GMRES preconditioned with CSLP. To ensure an accurate inverse of CSLP on each level, we employ Bi-CGSTAB iterations, reducing the relative residual to 10^{-8}, assuming that the optimal tolerance is unknown. Since the Krylov subspace method instead of the multigrid method is employed to solve the CSLP here, a small complex shift can be used. Here we will use $\beta_2 = 0.1$. The model problem with wavenumber $k = 200$ is solved with two kinds of resolution, that is $kh = 0.3125$ and 0.625, respectively. As analyzed in [16], the third level will remain indefinite for $kh = 0.3125$, while it becomes negative definite for $kh = 0.625$.

Table 1 The impact of varying tolerances for solving the coarsest problem on the number of outer iterations

kh	10^0	10^{-1}	10^{-2}	10^{-4}	10^{-6}	10^{-8}
0.3125	12 (1)	5 (8)	5 (17)	5 (31)	5 (44)	5 (58)
0.625	16 (1)	31 (32)	33 (67)	33 (133)	33 (203)	33 (256)

In parentheses is the number of iterations required to solve the corresponding coarsest problem once

Table 2 Tolerance study for CSLP approximation for $kh = 0.625$

	10^{-1}	10^{-2}	10^{-4}	10^{-6}	10^{-8}
Outer FGMRES	18	16	16	16	16
Coarsest FGMRES	1	1	1	1	1
L1 Bi-CGSTAB	69	171	408	666	780
L2 Bi-CGSTAB	15	39	84	119	177
L3 Bi-CGSTAB	29	89	174	370	544

Table 1 illustrates the impact of varying tolerances for solving the coarsest problem on the convergence behavior. Specifically, it presents the number of outer iterations required to reduce the relative residual to 10^{-6} and the corresponding number of iterations needed to solve the coarsest problem once.

From Table 1, we observe varying accuracy requirements for solving the coarsest grid problem corresponding to different values of kh. In the case of $kh = 0.3125$, a relative tolerance of 10^{-1} is necessary for maintaining the convergence of the outer iterations. Conversely, for $kh = 0.625$, a single iteration of the coarsest grid solver is sufficient. A strict tolerance in solving the coarsest grid even leads to more outer iterations.

We attribute this phenomenon to the nature of the coarsest grid system, whether it is indefinite or negative definite. According to [16], the third level remains indefinite for $kh = 0.3125$, while it becomes negative definite for 0.625. If the coarsest grid system is indefinite, the relative tolerance of the iterative solver should be ensured at 10^{-1} or smaller. On the contrary, if the coarsest grid system is negative definite, one iteration is adequate. However, it leads to more outer iterations compared to the former case.

To further validate this conclusion, we note the following numerical observations (not presented in tables or figures): using the model problem with Dirichlet boundary conditions at $kh = 0.625$, a tolerance of 10^0 results in outer iterations of 27, while 10^{-1} leads to 32. This brief numerical observation supports our finding that only one iteration suffices when the system becomes negative definite on the coarsest grid. It should be noted that, for the sake of uniformity, if a tolerance of 10^0 appears in this paper, it means that only one iteration is performed.

4.2 Tolerance for Solving CSLP

In this section, we explore the accuracy requirements for the approximate inverse of CSLP on each grid level. Consistent with the solver settings from the previous subsection, we perform one iteration on the coarsest level for $kh = 0.625$ and set the tolerance for the coarsest grid solver to 10^{-1} for $kh = 0.3125$. We vary the tolerance for the Bi-CGSTAB solver used in the approximate CSLP solution. The results are presented in Tables 2 and 3, where "Ln Bi-CGSTAB" denotes the number of Bi-CGSTAB iterations needed to achieve

Table 3 Tolerance study for CSLP approximation for $kh = 0.3125$

	10^{-1}	10^{-2}	10^{-4}	10^{-6}	10^{-8}
Outer FGMRES	6	5	5	5	5
Coarsest FGMRES	8	8	8	8	8
L1 Bi-CGSTAB	22	338	1149	1771	2344
L2 Bi-CGSTAB	10	44	131	251	300
L3 Bi-CGSTAB	34	48	114	198	272

the corresponding tolerance on the n-th level. The phrases "Outer FGMRES" and "Coarsest FGMRES" denote the number of FGMRES iterations required for the outer solvers and the coarsest problem solver, respectively.

From Tables 2 and 3, it is observed that whether the coarsest grid system remains indefinite or becomes negative definite, setting a tolerance stricter than 10^{-2} for the approximate inverse of CSLP does not necessarily result in a further reduction in the number of outer iterations. For cases with tolerances of 10^{-1} and 10^{-2}, while the number of outer iterations is reduced by $1-2$ with a tolerance of 10^{-2}, achieving this tolerance requires several times more iterations, particularly on the first and second levels, where computations are expensive. We choose to set the tolerance for solving the CSLP to 10^{-1}, striking a balance between achieving sufficient accuracy in the solution and minimizing the overall computational cost. Given that the scaling behavior of CSLP is well-established in the literature [18, 21], we limit our analysis to a single configuration and wavenumber, as this adequately demonstrates the effectiveness of our chosen tolerance.

For a tolerance on the order of 10^{-1}, where the required number of iterations is not substantial, we choose to use the more stable GMRES solver for better approximations to the inverse of CSLP. Furthermore, according to [16], while setting the tolerance at 10^{-1}, we limit the maximum number of iterations to $6(N^l)^{\frac{1}{4}}$, where N^l denotes the size of the problem on level l. This allows the iterations to cease once the maximum number of iterations or the tolerance level is reached.

4.3 On Wavenumber Independent Convergence

To achieve a robust multilevel deflation method, we expand our investigation of convergence to deeper levels, larger wavenumbers, and more complex model problems.

4.3.1 For Constant Wavenumber Problem

We employ the V-cycle multilevel deflation method with coarsening to different levels to solve the constant wavenumber model problem with increasing wavenumber k. As mentioned, GMRES instead of Bi-CGSTAB is used to solve the CSLP with complex shift $\beta_2 = 0.1$. The relative tolerance for solving CSLP is set to 10^{-1}. For the coarsest problem, one iteration is performed if it is negative definite; otherwise, CSLP-preconditioned GMRES iterations are employed to reduce the relative residual to 10^{-1}. The tolerance for the outer FGMRES iterations is 10^{-6}. We consider the scenario with $kh = 0.3125$, indicating that from the fourth level onward, the linear system becomes negative definite.

From Table 4, we observe that for the multilevel deflation method, if the coarsest grid system remains indefinite, it exhibits convergence behavior that is close to wavenumber

Table 4 The number of outer iterations required for the constant wavenumber model problems with increasing wavenumber k by the multilevel deflation combined with CSLP with complex shift $\beta_2 = 0.1$

k	Multilevel deflation		
	Three-level	Four-level	Five-level
100	6	9	8
200	6	13	12
400	7	20	20
800	7	37	37

Table 5 The number of outer iterations required for the constant wavenumber model problems with increasing wavenumber k by the multilevel deflation combined with CSLP with complex shift $\beta_2 = k^{-1}$

k	Multilevel deflation		
	Three-level	Four-level	Five-level
100	6	6	6
200	6	7	7
400	6	8	8
800	7	9	9

independent, corresponding to the three-level deflation method in the table. However, if the coarsest grid system becomes negative definite, as shown by the four-level deflation method in the table, convergence results can still be achieved, but the number of outer iterations starts to increase with the wavenumber. We also find that continuing to deeper levels, as demonstrated by the five-level deflation method in the table, does not lead to an increase in the number of outer iterations compared to the four-level deflation. While theoretically we could continue to deeper levels until the coarsest problem becomes small enough for direct solving, this approach is less favorable in massively parallel computing environments due to the increased communication costs and potential load imbalance.

As mentioned above, the Krylov subspace iterations for the CSLP allow the benefits of using a small shift, resulting in a preconditioner similar to the original Helmholtz operator that retains the ability to shift indefiniteness at certain levels. Similar to [16], one can use the inverse of the wavenumber k as the shift ($\beta_2 = k^{-1}$). As observed in Sect. 4.2, having a tolerance of 10^{-1} to approximate the inverse of CSLP leads to an increase in the number of outer iterations. For a small complex shift, the residual cannot be reduced to 10^{-1} within the maximum number of iterations given. Using the more stable GMRES often provides a relatively accurate approximation compared to the Bi-CGSTAB method. This is one of the reasons why GMRES is employed for approximating the inverse of CSLP on the coarse grid levels in this study.

As shown in Table 5, with complex shift $\beta_2 = k^{-1}$, the close-to wavenumber independent convergence is obtained even for the multilevel deflation methods where the coarsest grid problems become negative definite. Hereafter, we denote this configuration, which is mainly a parallel matrix-free implementation of the V-cycle multilevel deflation method proposed in [16], as **MADP-v1** (Matrix-free multilevel Adapted Deflation Preconditioning).

4.3.2 For Non-Constant Wavenumber Problem

In this section, we apply MADP-v1 to non-constant wavenumber problems. Note that for the model problems described in Sect. 2, due to the use of a computational domain based on actual physical dimensions rather than being scaled to a unit length, we use a complex shift

$\beta_2 = (k_{\text{dim}})^{-1}_{\text{max}}$, where k_{dim} is the so-called dimensionless wavenumber, defined as

$$k_{\text{dim}} = \sqrt{\left(\frac{2\pi f}{c}\right)^2 L_x L_y},$$

with L_x and L_y denoting the lengths of the computational domain in the x and y directions, respectively.

In Table 6, we give the results for the Wedge problem with $kh = 0.349$, indicating that the linear systems become negative definite from the fourth-level coarse grid onward. We find that the latter case requires more outer iterations and significantly more CPU time. Upon further observation of the solving process, it is observed that coarsening to negative definite levels requires a higher number of GMRES iterations to approximate the CSLP compared to the scenario of coarsening to indefinite levels. In cases where the coarsening remains on indefinite levels, the tolerance of 10^{-1} is achieved within the maximum number of iterations. However, in cases where the coarsening goes to negative definite levels, the number of iterations reaches the maximum specified value without achieving the same tolerance. For example, consider the Wedge problem with a grid size of 1153×1921 and $f = 160$Hz. On the first and second levels, the four-level deflation requires 232 and 164 GMRES iterations to approximate the CSLP per outer iteration, respectively, whereas the three-level deflation only requires 73 and 49 GMRES iterations.

Table 7 reports the number of iterations required and the time elapsed for the Marmousi problem with $kh = 0.54$. In this scenario, the linear systems become negative definite from the third-level grid onward. Despite being coarsened to deeper negative definite levels, the number of outer iterations remains constant and the computational time is comparable. However, one can find that, for such a highly heterogeneous model problem, the number of outer iterations starts increasing with the frequency. This is consistent with the results in [16].

In summary, the variant MADP-v1, based on the configuration proposed in [16], utilizes a V-cycle type and allows the combination of CSLP with a smaller complex shift $\beta_2 = (k_{\text{dim}})^{-1}_{\text{max}}$. For constant wavenumber problems, MADP-v1 achieves near wavenumber-independent convergence. However, for non-constant wavenumber model problems, as the wavenumber increases, the number of outer iterations will increase gradually, and the cost of using Krylov subspace methods to solve CSLP will become more noticeable.

For practical applications, where the wavenumber is usually non-constant within the domain, and also for better scalability, the coarsening in this multilevel deflation method should not be limited only to indefinite levels. Therefore, we will pay more attention to the common occurrence of coarsening to negative definite levels. The case of coarsening to indefinite levels will serve as a reference for the case of coarsening to negative definite levels. We aim to achieve at least similar convergence and computational efficiency in the case of coarsening to negative definite levels.

4.3.3 On the Tolerance for Coarse Levels

Sheikh et al. [36] stated that using $n2$, $n3$, and $n4$ iterations on the second, third, and fourth levels, respectively, can accelerate convergence, and their results indicate that larger $n2$ leads to better convergence for larger wavenumber. Additionally, [16] demonstrates that employing a W-cycle instead of a V-cycle for constructing the multilevel hierarchy results in a reduced number of iterations across reported frequencies. In this paper, we attribute this to the accuracy of solving on the second, third, and fourth levels.

Table 6 Number of iterations required and time elapsed for the Wedge problem with $kh = 0.349$ for the largest wavenumber k

f (Hz)	Grid size	Three-level MADP-v1				Four-level MADP-v1			
		Outer #iter	#iter for CSLP		CPU	Outer #iter	#iter for CSLP		CPU
		(Coarsest)	L1	L2	time (s)	(Coarsest)	L1	L2	time (s)
20	145×241	7 (2)	20	51	3.78	8 (1)	48	59	7.00
40	289×481	7 (3)	25	59	20.14	9 (1)	116	83	103.31
80	577×961	8 (4)	38	61	195.14	11 (1)	164	116	907.00
160	1153×1921	8 (4)	73	49	1060.50	13 (1)	232	164	5101.73

In parentheses are the number of iterations to solve the coarsest grid system. L1 and L2 represent the number of GMRES iterations required to approximate the inverse of CSLP on the first and second level, respectively

Table 7 Number of iterations required and time elapsed for the Marmousi problem with $kh = 0.54$ for the largest wavenumber k

f (Hz)	Grid size	Three-level MADP-v1				Five-level MADP-v1			
		Outer	#iter for CSLP		CPU	Outer	#iter for CSLP		CPU
		#iter	L1	L2	time (s)	#iter	L1	L2	time (s)
10	737×241	12	124	88	133.91	12	124	88	165.43
20	1473×481	16	175	124	1560.03	16	175	124	1743.97
40	2945×961	23	247	175	21557.94	23	247	175	21233.10

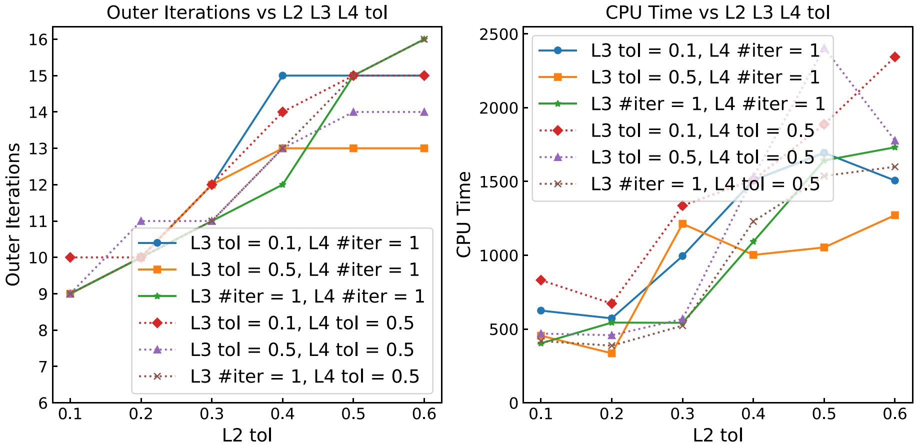


Fig. 4 Outer iterations and CPU time vary from different tolerances on the second (L2), third (L3), and fourth (L4) levels. Five-level deflation for Marmousi problem, grid size 1473 × 481, $f = 20$Hz

In contrast to the previous configuration of a single iteration on each coarser level, we introduced distinct tolerances for iterations on the second, third, and fourth levels, exploring their impact on outer iterations and CPU time. The numerical experiments utilized the Marmousi model problem with a grid size of 1473 × 961 and frequencies of 10Hz and 20Hz, corresponding to $kh = 0.27$ and 0.54, respectively. As we mentioned, for $kh = 0.27$, the linear systems of the second and third levels remain indefinite, while that of the fourth level becomes negative definite. For $kh = 0.54$, the third and fourth levels become negative definite.

It is evident in Figs. 3 and 4 that the number of outer iterations is correlated with the accuracy of solving the second-level grid system. Overall, a higher number of outer iterations usually corresponds to increased CPU time. However, we also observe that the minimum CPU time does not necessarily align with the minimum number of outer iterations, as depicted in Fig. 4. This suggests that sacrificing a few extra outer iterations may result in computational time savings. A balance between the number of outer iterations and computational time needs to be identified.

For $kh = 0.27$, it is best to set a tolerance of 10^{-1} for the second and third levels and perform one iteration for coarser levels. Conversely, for the case of $kh = 0.54$, the recommended tolerances for the second and third levels are 2×10^{-1} and 5×10^{-1}, respectively. From extensive numerical experiments across various grid sizes and multilevel deflation methods,

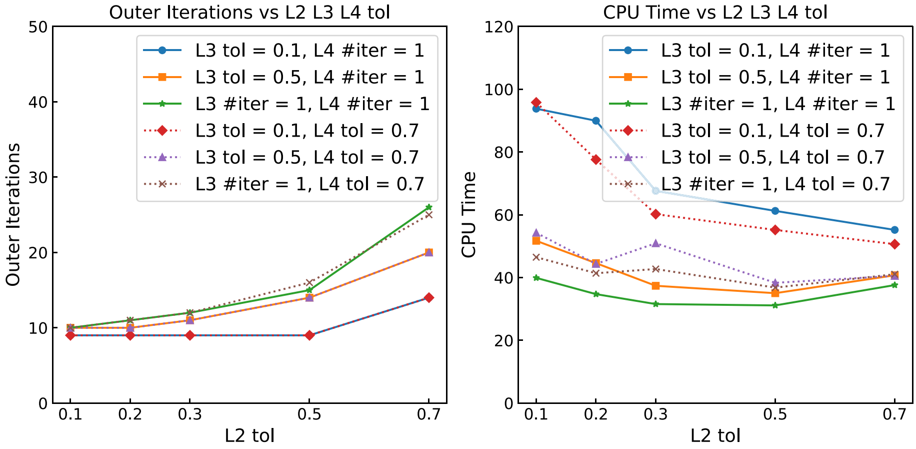


Fig. 5 Outer iterations and CPU time vary from different tolerances on the second (L2), third (L3), and fourth (L4) levels. Five-level deflation combined with multigrid-based CSLP on first and second levels. Marmousi problem, grid size 1473 × 481, $f = 10$Hz

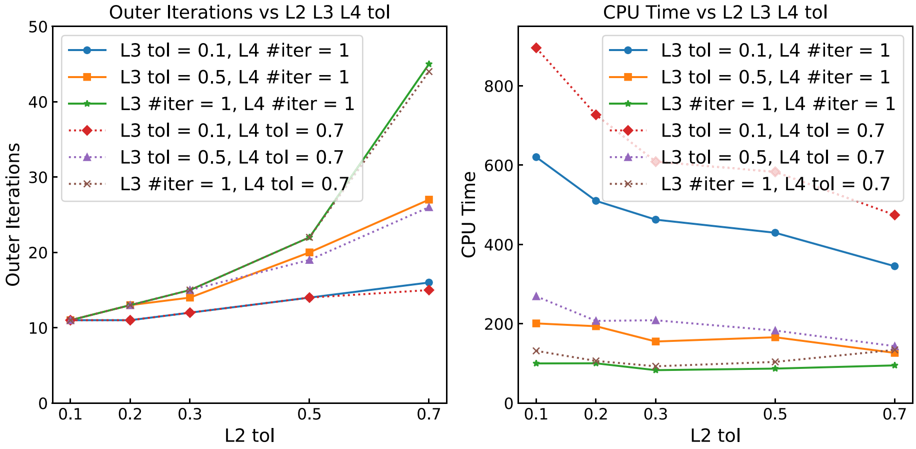


Fig. 6 Outer iterations and CPU time vary from different tolerances (tol) on the second (L2), third (L3), and fourth (L4) levels. Five-level deflation combined with multigrid-based CSLP on first and second levels. Marmousi problem, grid size 1473 × 481, $f = 20$Hz

it is observed that the optimal setting of the tolerance for solving the second, third, and fourth-level grid systems, corresponding to the minimum CPU time and the fewest outer iterations, may vary. However, on a comprehensive scale, a robust and acceptable configuration is to set a tolerance for solving the second-level grid system to 10^{-1}, while performing only one iteration on the other coarser grid levels. Let us denote this configuration as **MADP-v2**.

We applied the MADP-v2 to solve the Wedge and Marmousi problems, respectively. The results are presented in Tables 8 and 9. Compared to the corresponding results in Tables 6 and 7, MADP-v2 results in reduced computation time and a lower number of iterations across all reported frequencies, showcasing a closer-to-wavenumber-independent behavior. In addition to reduced outer iterations, we also observe that, when setting the tolerance to 10^{-1} for the second-level coarse grid system, the number of iterations required to solve CSLP on the

Table 8 Number of outer FGMRES-iterations for the Wedge problem with $kh = 0.349$ for the largest wavenumber k

f (Hz)	Grid size	Four-level MADP-v2			
		Outer #iter (L2 #iter)	#iter for CSLP L1	#iter for CSLP L2	CPU time (s)
20	145×241	6 (2)	6	59	4.69
40	289×481	6 (2)	8	83	19.26
80	577×961	7 (2)	7	116	148.05
160	1153×1921	7 (3)	15	164	1113.86

In parentheses are the number of iterations to solve the second-level grid system

first-level grid is significantly reduced. Comparing Tables 8 with 6 (three-level deflation), in the case of coarsening to negative definite levels, MADP-v2 achieves similar convergence and computational efficiency.

4.4 Combined with Multigrid-Based CSLP

Further observation reveals that, in cases where coarsening reaches negative definite levels, a significant portion of the computational time is still dedicated to approximating the inverse of CSLP on the first and second levels. Moreover, these iterations on the second level typically reach the specified maximum number of iterations $6(N^l)^{\frac{1}{4}}$ rather than achieving a tolerance of 10^{-1}. The use of GMRES iterations for solving CSLP on the first and second levels consumes a substantial amount of time, since the scale of the first- and second-level grid systems is large.

Instead of employing the GMRES or Bi-CGSTAB methods, we can utilize the multigrid method to approximate the inverse of CSLP on the first and second levels. On coarser levels, GMRES iterations are still used to approximate the inverse of CSLP. However, as known, the multigrid method requires that the complex shift should not be too small. Consequently, we cannot use $\beta_2 = (k_{\text{dim}})^{-1}_{\text{max}}$ as the complex shift for CSLP. Therefore, in the multilevel deflation methods combined with the multigrid-based CSLP, a complex shift of $\beta_2 = 0.5$ will be consistently utilized. (Additional numerical experiments have demonstrated that $\beta_2 = 0.5$ is a superior choice among other smaller complex shifts.)

Except for the choice of the complex shift for CSLP and the method used to solve CSLP on the first- and second-level coarse grid systems, the remaining settings are mostly inherited from MADP-v2. Specifically, a tolerance of 10^{-1} is set for solving the second-level grid system, and only one iteration is performed on other coarser levels. This modified configuration is denoted as **MADP-v3**. As it combines with multigrid-based CSLP on the first and second levels, this variant can be considered as an extension of the two-level deflation method proposed in [9].

The number of iterations and computation time required for solving the Wedge and Marmousi problems using MADP-v3 are presented in Tables 10 and 11, respectively. We observe that compared to MADP-v2 (as shown in Tables 8 and 9), while the number of outer iterations has increased, it exhibits nearly wavenumber-independent convergence, with computation time three times faster. Moreover, compared to the two-level deflation method proposed in [9], the current multilevel deflation method ensures a similar number of outer iterations while significantly reducing computation time.

Table 9 Number of outer FGMRES-iterations for the Marmousi problem with $kh = 0.54$ for the largest wavenumber k

f (Hz)	Grid size	Three-level MADP-v2				Five-level MADP-v2			
		Outer #iter	#iter for CSLP		CPU	Outer #iter	#iter for CSLP		CPU
		(L2 #iter)	L1	L2	time (s)	(L2 #iter)	L1	L2	time (s)
10	737×241	8 (3)	23	88	58.01	8 (3)	23	88	48.29
20	1473×481	9 (3)	25	124	422.33	9 (3)	20	124	364.21
40	2945×961	10 (3)	39	175	5267.53	10 (3)	39	175	4106.46

In parentheses are the number of iterations to solve the second-level grid system

Table 10 Number of outer FGMRES-iterations for the Wedge problem with $kh = 0.349$ for the largest wavenumber k

f (Hz)	Grid size	Four-level MADP-v3		Five-level MADP-v3	
		Outer #iter (L2 #iter)	CPU time (s)	Outer #iter (L2 #iter)	CPU time (s)
20	145×241	10 (6)	1.73	10 (6)	1.83
40	289×481	10 (10)	8.08	10 (10)	8.87
80	577×961	10 (17)	48.05	10 (18)	64.54
160	1153×1921	11 (34)	356.76	11 (34)	367.53
320	2305×3841	11 (66)	3458.14	11 (64)	3065.03

In parentheses are the number of iterations to solve the second-level grid system

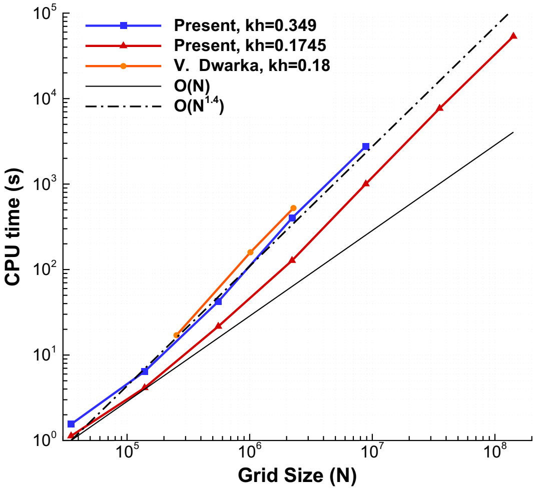


Fig. 7 Evolution of computational time versus problem size. Wedge model problem. The data in orange is extracted from [15]

Similarly to the last section, the optimal tolerance setting is studied for the second, third and fourth levels, as shown in Figs. 5 and 6. From the figures, it can be observed that performing only one iteration on the fourth level (L4) or setting a tolerance of 0.5 has little impact on the outer iterations but introduces additional computational costs. For this reason, we can keep one iteration on the fourth level. In comparison to performing only one iteration on the third level (L3), setting a smaller tolerance on L3 helps slow down the increase in the number of outer iterations but leads to more computational costs. From the perspective of computation time, performing only one iteration on L3 remains the optimal choice. With the incorporation of multigrid-based CSLP, the number of outer iterations increases as the tolerance of the second level (L2) increases, while the computation time shows a trend of decreasing first and then increasing. If one wants to minimize the computation time, choosing the tolerance of the second level as 0.3 while still performing only one iteration on other coarser levels can be the optimal option. Let us denote this configuration as **MADP**.

Tables 12 and 13 present the number of iterations required and computation time to solve the Wedge and Marmousi problems using MADP. Compared with Tables 10 and 11, it can be seen that although a few extra outer iterations are consumed, reduced computation time is obtained.

Table 11 Number of outer FGMRES-iterations for the Marmousi problem with $kh = 0.54$ for the largest wavenumber k

f (Hz)	Grid size	Two-level Deflation [9]		Three-level MADP-v3		Five-level MADP-v3	
		Outer #iter	CPU time (s)	Outer #iter (L2 #iter)	CPU time (s)	Outer #iter (L2 #iter)	CPU time (s)
10	737×241	11	23.15	11 (17)	16.53	11 (13)	18.57
20	1473×481	11	224.21	11 (30)	110.79	11 (24)	108.03
40	2945×961	12	4354.83	13 (69)	1220.61	13 (50)	1084.42

In parentheses are the number of iterations to solve the second-level grid system

Table 12 Number of outer FGMRES-iterations for the Wedge problem with $kh = 0.349$ for the largest wavenumber k

f (Hz)	Grid size	Four-level MADP		Five-level MADP	
		Outer #iter (L2 #iter)	CPU time (s)	Outer #iter (L2 #iter)	CPU time (s)
20	145×241	11 (4)	1.51	11 (4)	1.55
40	289×481	12 (6)	6.34	12 (6)	6.42
80	577×961	13 (10)	39.62	13 (10)	42.21
160	1153×1921	15 (17)	410.27	14 (18)	400.64
320	2305×3841	16 (33)	2748.70	16 (32)	2762.84

In parentheses are the number of iterations to solve the second-level grid system

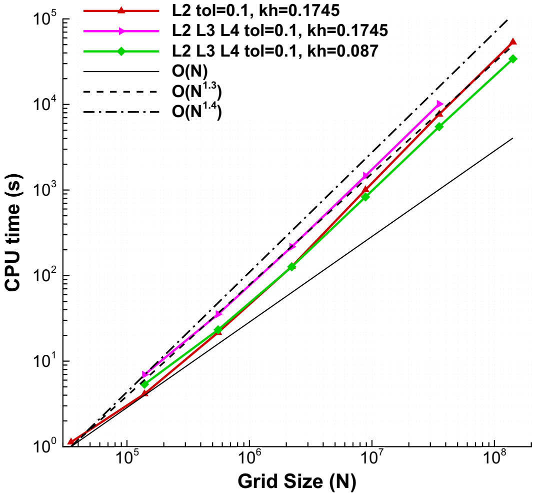


Fig. 8 Evolution of computational time versus problem size. Wedge model problem

Table 13 Number of outer FGMRES-iterations for the Marmousi problem with $kh = 0.54$ for the largest wavenumber k

f (Hz)	Grid size	Three-level MADP		Five-level MADP	
		Outer #iter (L2 #iter)	CPU time (s)	Outer #iter (L2 #iter)	CPU time (s)
10	737×241	14 (10)	12.42	13 (7)	12.67
20	1473×481	16 (19)	93.08	15 (15)	84.06
40	2945×961	19 (43)	929.62	18 (29)	816.38

In parentheses are the number of iterations to solve the second-level grid system

Therefore, we regard the variant **MADP**, which can balance both convergence and computational efficiency, as an optimal configuration of the present matrix-free multilevel deflation method. This variant employs a tolerance of 0.3 for solving the second-level grid system and performs only one iteration on other coarser levels. A multigrid V-cycle is used to solve the CSLP on the first- and second-level grid systems. On coarser levels, several GMRES iterations approximate the inverse of CSLP with a tolerance of 10^{-1}. In the subsequent sections, we will use this variant for numerical experiments.

Table 14 Number of outer FGMRES-iterations for the Wedge problem with $kh = 0.1745$

Grid size	#unknowns	f (Hz)	Outer #iter (L2 #iter)	CPU time (s)
145× 241	34945	10	10 (2)	1.13
289× 481	139009	20	11 (3)	4.14
577× 961	554497	40	12 (4)	21.64
1153× 1921	2214913	80	12 (7)	127.47
2305× 3841	8853505	160	13 (13)	1003.71
4609× 7681	35401729	320	14 (27)	7678.83
9217× 15361	141582337	640	17 (47)	53481.69

In parentheses are the number of iterations to solve the second-level grid system

Table 15 Number of outer iterations and the number of iterations on the second, third, and fourth levels when a tolerance of 10^{-1} is set on these levels

Grid size	f (Hz)	Outer #iter	L2 #iter	L3 #iter	L4 #iter
289× 481	20	10	3	2	16
577× 961	40	10	3	2	20
1153× 1921	80	10	3	2	30
2305× 3841	160	10	3	2	46
4609× 7681	320	9	3	2	75

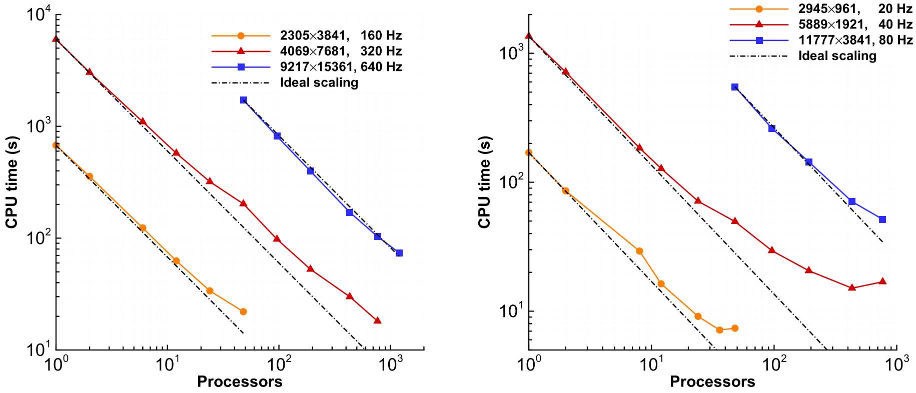
Wedge problem with $kh = 0.1745$

4.5 Complexity Analysis

We next analyze the complexity of the present multilevel deflation method in relation to the problem size or to the frequency, equivalently. In this numerical experiment, the Wedge model problem is solved using **five-level MADP**. The grid resolution, i.e. kh, is kept fixed to a specific value, while the frequency is growing from 10Hz to 160Hz for $kh = 0.349$ or even 640Hz for $kh = 0.1745$, respectively. The case of $f = 640$Hz leads to a linear system with approximately 142 million unknowns. The number of outer iterations and the number of iterations on the second level are reported in Table 14. Similarly to the results in Table 12, the number of outer iterations is rather moderate and is found to grow slightly with respect to frequency.

Figure 7 shows the evolution of computational time versus problem size for $kh = 0.349$ and $kh = 0.1745$. As the grid size increases, the computational time of the present matrix-free multilevel deflation method shows a similar trend to the matrix-based version proposed by [16]. However, with single-core sequential computing, the present method can handle grid sizes much larger than those achievable in [16]. If N represents the total number of unknowns, it has been observed that the computational time follows a behavior of $\mathcal{O}(N)$ for small grid sizes and asymptotically approaches $\mathcal{O}(N^{1.4})$. This is comparable to the geometric two-grid preconditioner in [6]. The reason for this behavior is that, as the frequency increases, the number of iterations required on the second level almost increases linearly with frequency, as shown in Table 14.

One can think about continuing a similar approach, setting a tolerance of 10^{-1} on the third level, ensuring that the number of iterations on the second level is independent of



(a) Wedge problem (b) Marmousi problem

Fig. 9 Strong scaling of the parallel multilevel ADP-FGMRES for the non-constant wavenumber model problems with various grid sizes and frequencies

Table 16 Weak scaling for the model problem with constant wavenumber

Grid size	#unknowns	np	#iter	CPU time (s)
$k = 400$				
641×641	410881	1	16	49.68
1281×1281	1640961	4	13	21.63
2561×2651	6558721	16	12	16.13
5121×5121	26224641	64	11	21.66
$k = 1600$				
2561×2561	6558721	16	20	168.26
5121×5121	26224641	64	14	100.84
10241×10241	104878081	256	13	79.69
20481×20481	419471361	1024	13	93.62

the wavenumber, and so forth. This is feasible but only limited to indefinite levels. Setting tolerance on negative definite levels, i.e., performing more than one iteration, may lead to a significant increase in outer iterations and computational time, consistent with the conclusion in Sect. 4.1. For instance, considering the Wedge model problem with $kh = 0.1745$, where the fourth-level grid system remains indefinite, turning negative definite onwards the fifth level. We can extend MADP-v3 by setting a tolerance of 10^{-1} for iterations on the third and fourth levels instead of performing one iteration. Table 15 provides the required number of outer iterations and the number of iterations on the second, third, and fourth levels. We can observe that the number of outer iterations and iterations on the second and third levels are almost independent of the wavenumber, while the number of iterations on the fourth level gradually increases with the wavenumber. However, as shown in Fig. 8, the computation time required is more than MADP. Together with the case of $kh = 0.087$ in the figure, it can be observed that there are subtle differences in the growth trend compared to that of MADP, possibly approaching $\mathcal{O}(N^{1.3})$. While it is beneficial to set a tolerance for coarser levels for problems with smaller kh, whether the present multilevel deflation method can be closer to $\mathcal{O}(N)$ remains an open problem that requires further study. To complement this study,

Table 17 Weak scaling for the Wedge model problem with $f = 320\text{Hz}$

Grid size	#unknowns	np	#iter	CPU time (s)
2305×3841	8853505	48	16	69.75
4609×7681	35401729	192	14	53.20
9217×15361	141582337	768	14	67.03

it would be interesting to perform the same complexity analysis for more levels and three-dimensional cases. These are left to a future line of research. Additionally, in the numerical experiments on current two-dimensional problems, it does not lead to a significant reduction in computation time. Therefore, we consider MADP to still be the optimal choice.

5 Parallel Performance

In this section, we aim to present both weak scalability and strong scalability. Through this analysis, our goal is to offer insight into the suitability of the present multilevel deflation method for practical large-scale applications in the context of heterogeneous time-harmonic wave problems.

The **parallel six-level MADP** preconditioned FGMRES is used as the default approach in this section to solve model problems. All numerical experiments are carried out on the Linux supercomputer DelftBlue [12], which operates on the Red Hat Enterprise Linux 8 operating system. Each compute node is furnished with two Intel Xeon E5-6248R CPUs featuring 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. The present solver is developed in Fortran 90 and is compiled using GNU Fortran 8.5.0 with the compiler options `-O3` for optimization purposes. The Open MPI library (v4.1.1) is used for data communication.

5.1 Weak Scalability

To assess the weak scalability of the proposed matrix-free parallel multilevel deflation preconditioning method, we keep the wavenumber or frequency unchanged and solve the model problems across varying problem sizes but maintain a fixed workload per processor. The computational times for different problem sizes and the corresponding number of processors are summarized in Tables 16 and 17. As the grid undergoes refinement while maintaining a constant wavenumber, the parameter kh gradually decreases. In the context of deflation preconditioning, it has been documented that a smaller kh leads to a reduction in the number of outer iterations [36]. As kh continues to diminish, the number of outer iterations tends to stabilize. Additionally, the advantages of one or two less iterations may be counteracted by the overhead of data communication. Consequently, we observe that the computational time initially decreases due to the reduced number of outer iterations, and then remains almost constant, even as the grid size expands to tens of millions with over a thousand parallel computing cores.

This behavior is highly commendable, as it allows for the efficient resolution of large linear systems within a reasonable computational timeframe on a parallel distributed memory machine. It is important to emphasize that the advantages of the suggested approach should be considered in the context of minimizing pollution error by grid refinement for real-world application of Helmholtz problems.

Table 18 Performance comparison of matrix-free and CSR matrix-based implementations for matrix–vector multiplications

Problem size N	Matrix-free (GFLOPs/s)	CSR-Matrix (GFLOPs/s)	Performance Ratio
289	4.5918	2.5298	1.82
1,089	5.1946	2.8848	1.80
4,225	6.1535	2.8094	2.19
16,641	5.0961	2.8327	1.80
66,049	6.1740	2.6128	2.36
263,169	6.1361	2.8208	2.18
1,050,625	6.2861	2.3977	2.62
4,198,401	5.9456	1.9992	2.97
16,785,409	5.5556	1.9200	2.89
67,125,249	5.4626	1.8979	2.88
268,468,225	5.4626	1.8958	2.88

5.2 Strong Scalability

We are also interested in the strong scalability properties of the present parallel multilevel deflation preconditioning method for the Helmholtz problems. In this section, we perform numerical experiments on the nonconstant-wavenumber model problems with fixed problem sizes while varying the number of processors. First of all, the numerical experiments show that the number of iterations required is found to be independent of the number of processors used for parallel computing, which is a favorable property of our multilevel deflation method.

Figure 9 plots the computational time versus the number of processors. The figures show a decrease in parallel efficiency as the number of processors increases, particularly for the Marmousi model. The analysis suggests that maintaining a minimum of around one million unknowns per processor ensures a parallel efficiency of 60% or higher. If there are fewer than 50 thousand unknowns per processor, the ratio of computation load to data communication may significantly decrease, leading to poor parallel efficiency. However, increasing the grid size for the same model problem can result in improved parallel efficiency. This is because, while 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. Meanwhile, the total number of grid points increases fourfold, resulting in a larger ratio of computation load to data communication and better parallel efficiency. Overall, as demonstrated in solving the Wedge problem with a grid size of 9217×15361 and a frequency of $f = 640$Hz, the current matrix-free multilevel deflation approach can effectively solve complex Helmholtz problems with grid sizes up to tens of millions. It demonstrates strong parallel scalability, maintaining efficiency across more than a thousand processors.

6 Conclusion

In this work, we present an advanced matrix-free parallel scalable multilevel deflation preconditioning method for solving the Helmholtz equation in heterogeneous time-harmonic wave problems, benchmarked on large-scale real-world models. Building on recent advancements in higher-order deflation preconditioning, our approach extends these techniques to a parallel implementation. The incorporation of the deflation technique with CSLP, along with higher-order deflation vectors and re-discretization schemes derived from the Galerkin coarsening approach, forms a comprehensive setup for matrix-free parallel implementation. The proposed re-discretized finite-difference schemes at each coarse level contribute to a convergence behavior similar to that of the matrix-based deflation method.

We have explored different configurations of the multilevel deflation method, conducting research and comparing various variants. We note that the performance of different cycle types of the present multilevel deflation method is impacted by whether the coarsest-level system is negative definite. We suggest that for the multilevel deflation method coarsening to negative definite levels, ensuring a certain accuracy in iterations on the second-level grid is crucial to maintain a consistent number of outer iterations. Based on the complexities revealed during our study, we propose a robust and efficient variant, MADP. This variant employs the following settings: a tolerance of 0.3 for solving the second-level grid system, with only one iteration performed on other coarser levels; a multigrid V-cycle to solve CSLP on the first- and second-level grid systems; several GMRES iterations to approximate the inverse of CSLP on coarser levels, using a tolerance of 10^{-1}. It addresses the challenges posed by negative definite coarsest-level systems and does not lead to worse complexities.

Our numerical experiments illustrate the effectiveness of the matrix-free parallel multilevel deflation preconditioner, demonstrating convergence properties that are nearly independent of the wavenumber. The reduction in memory consumption achieved through matrix-free implementation, along with satisfactory weak and strong parallel scalability, emphasizes the practical applicability of our approach for large-scale real-world applications in wave propagation.

Appendix

A Re-Discretization Scheme for Coarse Levels

The stencils of the Laplace and wavenumber operators for interior points on the fourth-, fifth- and sixth-level coarse grid read as

$$A_{8h} = \frac{1}{4096} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot \frac{1}{h^2} \cdot$$

$$\begin{bmatrix} -10395 & -887166 & -7871637 & -15491748 & -7871637 & -887166 & -10395 \\ -887166 & -39105612 & -215169378 & -348459432 & -215169378 & -39105612 & -887166 \\ -7871637 & -215169378 & -265120059 & 413761124 & -265120059 & -215169378 & -7871637 \\ -15491748 & -348459432 & 413761124 & 2809129936 & 413761124 & -348459432 & -15491748 \\ -7871637 & -215169378 & -265120059 & 413761124 & -265120059 & -215169378 & -7871637 \\ -887166 & -39105612 & -215169378 & -348459432 & -215169378 & -39105612 & -887166 \\ -10395 & -887166 & -7871637 & -15491748 & -7871637 & -887166 & -10395 \end{bmatrix},$$

$$K_{8h} = \frac{1}{4096} \cdot \frac{1}{4096} \cdot \frac{1}{1024} \cdot k^2 \cdot$$

$$\begin{bmatrix} 27225 & 3939210 & 40768695 & 83544780 & 40768695 & 3939210 & 27225 \\ 3939210 & 569967876 & 5898859542 & 12088170168 & 5898859542 & 569967876 & 3939210 \\ 40768695 & 5898859542 & 61050008889 & 125106029556 & 61050008889 & 5898859542 & 40768695 \\ 83544780 & 12088170168 & 125106029556 & 256372094224 & 125106029556 & 12088170168 & 83544780 \\ 40768695 & 5898859542 & 61050008889 & 125106029556 & 61050008889 & 5898859542 & 40768695 \\ 3939210 & 569967876 & 5898859542 & 12088170168 & 5898859542 & 569967876 & 3939210 \\ 27225 & 3939210 & 40768695 & 83544780 & 40768695 & 3939210 & 27225 \end{bmatrix},$$

$$A_{16h} = \frac{1}{4096} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot \frac{1}{h^2} \cdot$$

$$\begin{bmatrix}
-13491387 & -1011388446 & -8590720245 & -16705596516 & -8590720245 & -1011388446 & -13491387 \\
-1011388446 & -41427399756 & -220811304386 & -353095695272 & -220811304386 & -41427399756 & -1011388446 \\
-8590720245 & -220811304386 & -262703195227 & 427978620452 & -262703195227 & -220811304386 & -8590720245 \\
-16705596516 & -353095695272 & 427978620452 & 2827174335440 & 427978620452 & -353095695272 & -16705596516 \\
-8590720245 & -220811304386 & -262703195227 & 427978620452 & -262703195227 & -220811304386 & -8590720245 \\
-1011388446 & -41427399756 & -220811304386 & -353095695272 & -220811304386 & -41427399756 & -1011388446 \\
-13491387 & -1011388446 & -8590720245 & -16705596516 & -8590720245 & -1011388446 & -13491387
\end{bmatrix},$$

$$K_{16h} = \frac{1}{4096} \cdot \frac{1}{4096} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot k^2 \cdot$$

$$\begin{bmatrix}
158684409 & 19907719338 & 199630340247 & 405976871820 & 199630340247 & 19907719338 & 158684409 \\
19907719338 & 2497518765316 & 25044582577654 & 50931743533240 & 25044582577654 & 2497518765316 & 19907719338 \\
199630340247 & 25044582577654 & 251141703197401 & 510732601675060 & 251141703197401 & 25044582577654 & 199630340247 \\
405976871820 & 50931743533240 & 510732601675060 & 1038647851363600 & 510732601675060 & 50931743533240 & 405976871820 \\
199630340247 & 25044582577654 & 251141703197401 & 510732601675060 & 251141703197401 & 25044582577654 & 199630340247 \\
19907719338 & 2497518765316 & 25044582577654 & 50931743533240 & 25044582577654 & 2497518765316 & 19907719338 \\
158684409 & 19907719338 & 199630340247 & 405976871820 & 199630340247 & 19907719338 & 158684409
\end{bmatrix},$$

$$A_{32h} = \frac{1}{4096} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot \frac{1}{h^2} \cdot$$

$$\begin{bmatrix}
-14618265915 & -1063059274398 & -8934400311925 & -17322732417892 & -8934400311925 & -1063059274398 & -14618265915 \\
-1063059274398 & -42774103061580 & -226217899925314 & -360608941958056 & -226217899925314 & -42774103061580 & -1063059274398 \\
-8934400311925 & -226217899925314 & -266795319274715 & 439293870677284 & -266795319274715 & -226217899925314 & -8934400311925 \\
-17322732417892 & -360608941958056 & 439293870677284 & 2882610253296592 & 439293870677284 & -360608941958056 & -17322732417892 \\
-8934400311925 & -226217899925314 & -266795319274715 & 439293870677284 & -266795319274715 & -226217899925314 & -8934400311925 \\
-1063059274398 & -42774103061580 & -226217899925314 & -360608941958056 & -226217899925314 & -42774103061580 & -1063059274398 \\
-14618265915 & -1063059274398 & -8934400311925 & -17322732417892 & -8934400311925 & -1063059274398 & -14618265915
\end{bmatrix},$$

$$K_{32h} = \frac{1}{4096} \cdot \frac{1}{4096} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot \frac{1}{1024} \cdot k^2 \cdot$$

$$\begin{bmatrix}
706549519225 & 85722084590890 & 852977249303575 & 1731387418334860 & 852977249303575 & 85722084590890 & 706549519225 \\
85722084590890 & 10400227566028036 & 103487421517331808 & 210060490730926272 & 103487421517331824 & 10400227566028036 & 85722084590890 \\
852977249303575 & 103487421517331840 & 1029751161146567936 & 2090206046973178368 & 1029751161146568192 & 103487421517331792 & 852977249303575 \\
1731387418334860 & 210060490730926208 & 2090206046973178624 & 4242735025361522688 & 2090206046973178624 & 210060490730926208 & 1731387418334860 \\
852977249303575 & 103487421517331840 & 1029751161146568064 & 2090206046973178624 & 1029751161146567936 & 103487421517331792 & 852977249303575 \\
85722084590890 & 10400227566028036 & 103487421517331808 & 210060490730926208 & 103487421517331808 & 10400227566028036 & 85722084590890 \\
706549519225 & 85722084590890 & 852977249303575 & 1731387418334860 & 852977249303575 & 85722084590890 & 706549519225
\end{bmatrix}.$$

B Performance Analysis Using Roofline Model

This appendix presents a performance analysis of matrix–vector multiplication operations $\mathbf{v} = \mathbf{A}\mathbf{u}$ comparing our matrix-free implementation with traditional CSR matrix-based approaches. The analysis focuses specifically on the Helmholtz operator with variable wavenumber, using five-point stencil discretization. We consider the matrix–vector multiplication as it constitutes the primary computational kernel in preconditioned Krylov subspace methods, typically accounting for the majority of computational time. Our analysis employs the roofline model [42], a performance model that bounds computational kernel performance based on peak computational performance and memory bandwidth limitations. As matrix–vector multiplication is typically memory-bound, we focus on arithmetic intensity (I), defined as the ratio of floating-point operations (FLOPs) to memory accesses:

$$I = \frac{\text{Total FLOPs}}{\text{Total Bytes Accessed}}$$

B.1 Analysis of Matrix-Free Implementation

The matrix-free implementation directly applies the five-point stencil operation for the discrete Laplacian operator combined with the wavenumber term. For variable wavenumber

k, the implementation of matrix–vector multiplication can be expressed in the following computational kernel (in Fortran):

```
! Pre-computed stencil coefficients
ap = -4.d0/h**2
as = aw = ae = an = 1.d0/h**2
do j = 1, ny
    do i = 1, nx
        v(i,j) = as * u(i,j-1)              & ! South
   neighbor
                + aw * u(i-1,j)             & ! West
   neighbor
                + (ap - k(i,j)**2) * u(i,j) & ! Center point
                + ae * u(i+1,j)             & ! East
   neighbor
                + an * u(i,j+1)               ! North
   neighbor
    end do
end do
```

In analyzing the memory access patterns, we consider the memory access per grid point operation. The implementation requires reading of five vector elements (center and four neighbors), each consuming 8 bytes in double precision. Additionally, we need to access one wavenumber value (8 bytes) and write one result value (8 bytes). Therefore, the total memory access per point is bounded by 56 bytes.

The computational intensity involves five multiplications for the stencil coefficients, four additions for combining the stencil components, and two additional operations (one square operation and one subtraction) for the wavenumber term. This results in 11 floating-point operations per grid point.

Consequently, the arithmetic intensity for the matrix-free implementation is:

$$I_{MF} \geq \frac{11}{56} \approx 0.1964 \text{ FLOPs/byte}$$

B.2 Analysis of CSR Matrix-Based Implementation

The compressed sparse row (CSR) format represents the sparse matrix A using three arrays: values (`A%col_indices`), and row pointers (`A%row_ptr`) [34]. The CSR format implementation is structured as follows:

```
do i = 1, A\%nrow
    v(i) = 0.d0
    do j = A\%row_ptr(i), A\%row_ptr(i+1)-1
        v(i) = v(i) + A\%values(j) * u(A\%col_indices(j))
    end do
end do
```

The memory access pattern for CSR implementation is more complex. For each non-zero element, we must read the matrix value (8 bytes), the column index (4 bytes), and access the corresponding vector element (8 bytes, assuming the vector is too large to fit into the cache). Additional memory operations include accessing row pointers (4 bytes per row) and reading/writing the result vector (16 bytes per row). For our five-point stencil case, with five nonzero elements per row, the total memory access is bounded by 120 bytes per row.

The computation for each non-zero element requires one multiplication and one addition, resulting in 10 total FLOPs per row. Thus, the arithmetic intensity for the CSR implementation is:

$$I_{CSR} \geq \frac{10}{120} \approx 0.0833 \text{ FLOPs/byte}$$

Based on this theoretical analysis, we expect the matrix-free implementation to outperform the CSR matrix-based implementation by approximately a factor of 2.35.

B.3 Numerical Validation

To validate our theoretical analysis, we conducted extensive performance measurements comparing both implementations. For each grid size, we performed 100 consecutive matrix–vector multiplications to obtain statistically stable performance measurements. The performance metrics are reported in billions of floating-point operations per second (GFLOPs/s), calculated using the theoretical operation count for each implementation.

Table 18 presents the experimental results, which strongly support our theoretical analysis. The matrix-free implementation consistently achieves superior performance, with the advantage becoming more pronounced as the problem size increases. For larger problem sizes ($N > 10^6$), we observe performance improvements approaching a factor of 3, exceeding our theoretical prediction of 2.35. This enhanced performance can be attributed to memory hierarchy effects. The matrix-free implementation exhibits superior cache utilization, particularly for large-scale problems where the memory access patterns of the CSR format become increasingly inefficient.

The characteristics of these implementations have significant implications for parallel computing performance. The matrix-free implementation's regular memory access patterns facilitate better parallel efficiency through predictable memory access and reduced NUMA (Non-Uniform Memory Access) effects. Furthermore, in distributed memory systems, the matrix-free approach minimizes communication overhead, requiring only ghost point exchanges along subdomain boundaries. These parallel computing advantages, combined with the superior cache utilization observed in our sequential tests, suggest even more pronounced performance benefits in parallel computing environments, particularly for large-scale problems on distributed memory systems.

Funding The first author acknowledges financial support from the China Scholarship Council (CSC) under grant No. 202006230087.

Data availability The source codes supporting the findings of this study are openly available in the GitHub repository `paraMADP`, DOI 10.5281/zenodo.13143939. Additional materials or datasets used in this research can be made available upon reasonable request to the corresponding author.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Adriani, A., Sormani, R.L., Tablino-Possio, C., Krause, R., Serra-Capizzano, S.: Asymptotic spectral properties and preconditioning of an approximated nonlocal Helmholtz equation with Caputo fractional Laplacian and variable coefficient wave number \$$\mu$\$ (2024). arXiv:2402.10569

2. Babuska, I., Lipton, R.: Optimal local approximation spaces for generalized finite element methods with application to multiscale problems. Multiscale Model. Simul. **9**(1), 373–406 (2011). https://doi.org/10.1137/100791051
3. Babuska, I.M., Sauter, S.A.: Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? SIAM J. Numer. Anal. **34**(6), 2392–2423 (1997). https://doi.org/10.1137/S0036142994269186
4. Bootland, N., Dolean, V., Jolivet, P., Tournier, P.H.: A comparison of coarse spaces for Helmholtz problems in the high frequency regime. Comput. Math. Appl. **98**, 239–253 (2021). https://doi.org/10.1016/j.camwa.2021.07.011
5. Calandra, H., Gratton, S., Pinel, X., Vasseur, X.: An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media. Numer. Linear Algebra Appl. **20**(4), 663–688 (2013). https://doi.org/10.1002/nla.1860
6. Calandra, H., Gratton, S., Vasseur, X.: 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, pp. 141–155. Springer (2017). https://doi.org/10.1007/978-3-319-28832-1_6
7. Chen, J., Dwarka, V., Vuik, C.: A matrix-free parallel solution method for the three-dimensional heterogeneous Helmholtz equation. Electron. Trans. Numer. Anal. **59**, 270–294 (2023). https://doi.org/10.1553/etna_vol59s270
8. Chen, J., Dwarka, V., Vuik, C.: Matrix-free parallel preconditioned iterative solvers for the 2D Helmholtz equation discretized with finite differences. In: Scientific computing in electrical engineering, pp. 61–68. Springer Nature Switzerland (2024). https://doi.org/10.1007/978-3-031-54517-7_7
9. Chen, J., Dwarka, V., Vuik, C.: A matrix-free parallel two-level deflation preconditioner for two-dimensional heterogeneous Helmholtz problems. J. Comput. Phys. (2024). https://doi.org/10.1016/j.jcp.2024.113264
10. Chupeng, M., Alber, C., Scheichl, R.: Wavenumber explicit convergence of a multiscale generalized finite element method for heterogeneous Helmholtz problems. SIAM J. Numer. Anal. **61**(3), 1546–1584 (2023). https://doi.org/10.1137/21M1466748
11. Cocquet, P.H., Gander, M.J.: How large a shift is needed in the shifted Helmholtz preconditioner for its effective inversion by multigrid? SIAM J. Sci. Comput. **39**(2), A438–A478 (2017). https://doi.org/10.1137/15M102085X
12. Delft High Performance Computing Centre (DHPC): DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2 (2024)
13. Drzisga, D., Köppl, T., Wohlmuth, B.: A semi matrix-free twogrid preconditioner for the Helmholtz equation with near optimal shifts. J. Sci. Comput. **95**(3), 82 (2023). https://doi.org/10.1007/s10915-023-02195-5
14. Drzisga, D., Rüde, U., Wohlmuth, B.: Stencil scaling for vector-valued PDEs on hybrid grids with applications to generalized Newtonian fluids. SIAM J. Sci. Comput. **42**(6), B1429–B1461 (2020). https://doi.org/10.1137/19M1267891
15. Dwarka, V., Vuik, C.: Scalable convergence using two-level deflation preconditioning for the Helmholtz equation. SIAM J. Sci. Comput. **42**(2), A901–A928 (2020). https://doi.org/10.1137/18M1192093
16. Dwarka, V., Vuik, C.: Scalable multi-level deflation preconditioning for highly indefinite time-harmonic waves. J. Comput. Phys. **469**, 111327 (2022). https://doi.org/10.1016/j.jcp.2022.111327
17. Elman, H.C., Ernst, O.G., O'Leary, D.P.: A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations. SIAM J. Sci. Comput. **23**(4), 1291–1315 (2001). https://doi.org/10.1137/S1064827501357190
18. Erlangga, Y.A.: 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
19. Erlangga, Y.A., Nabben, R.: Multilevel projection-based nested Krylov iteration for boundary value problems. SIAM J. Sci. Comput. **30**(3), 1572–1595 (2008)
20. Erlangga, Y.A., Oosterlee, C.W., Vuik, C.: A novel multigrid based preconditioner for heterogeneous Helmholtz problems. SIAM J. Sci. Comput. **27**(4), 1471–1492 (2006). https://doi.org/10.1137/040615195
21. Erlangga, Y.A., Vuik, C., Oosterlee, C.W.: On a class of preconditioners for solving the Helmholtz equation. Appl. Numer. Math. **50**(3–4), 409–425 (2004). https://doi.org/10.1016/j.apnum.2004.01.009
22. Gander, M.J., Zhang, H.: A class of iterative solvers for the Helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods. SIAM Rev. **61**(1), 3–76 (2019). https://doi.org/10.1137/16M109781X

23. Gordon, D., Gordon, R.: Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers. J. Comput. Appl. Math. **237**(1), 182–196 (2013). https://doi.org/10.1016/j.cam.2012.07.024
24. Graham, I.G., Spence, E.A., Vainikko, E.: Domain decomposition preconditioning for high-frequency Helmholtz problems with absorption. Math. Comp. **86**(307), 2089–2127 (2017). https://doi.org/10.1090/mcom/3190
25. Kim, S., Kim, S.: Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media. SIAM J. Sci. Comput. **24**(2), 684–701 (2002). https://doi.org/10.1137/S1064827501385426
26. Kononov, A.V., Riyanti, C.D., de Leeuw, S.W., Oosterlee, C.W., Vuik, C.: Numerical performance of a parallel solution method for a heterogeneous 2D Helmholtz equation. Comput. Vis. Sci. **11**(3), 139–146 (2007). https://doi.org/10.1007/s00791-007-0069-6
27. Li, T.Y., Chen, F., Sun, H.W., Sun, T.: Preconditioning technique based on sine transformation for nonlocal Helmholtz equations with fractional Laplacian. J. Sci. Comput. **97**(1), 17 (2023). https://doi.org/10.1007/s10915-023-02332-0
28. Lin, X., Li, C., Hon, S.: Absolute-value based preconditioner for complex-shifted Laplacian systems (2024). arXiv:2408.00488
29. Lu, P., Xu, X.: A robust multilevel preconditioner based on a domain decomposition method for the Helmholtz equation. J. Sci. Comput. **81**, 291–311 (2019). https://doi.org/10.1007/s10915-019-01015-z
30. Ma, C., Scheichl, R., Dodwell, T.: Novel design and analysis of generalized finite element methods based on locally optimal spectral approximations. SIAM J. Numer. Anal. **60**(1), 244–273 (2022). https://doi.org/10.1137/21M1406179
31. Plessix, R.E., Mulder, W.A.: Separation-of-variables as a preconditioner for an iterative Helmholtz solver. Appl. Numer. Math. **44**(3), 385–400 (2003). https://doi.org/10.1016/S0168-9274(02)00165-4
32. Pulch, R., Sète, O.: The Helmholtz equation with uncertainties in the wavenumber. J. Sci. Comput. **98**(3), 60 (2024). https://doi.org/10.1007/s10915-024-02450-3
33. Riyanti, C., Kononov, A., Erlangga, Y., Vuik, C., Oosterlee, C., Plessix, R.E., Mulder, W.: A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. J. Comput. Phys. **224**(1), 431–448 (2007). https://doi.org/10.1016/j.jcp.2007.03.033
34. Saad, Y.: Iterative Methods for Sparse Linear Systems, second edn. Society for Industrial and Applied Mathematics (2003). https://doi.org/10.1137/1.9780898718003
35. Sheikh, A.H., Lahaye, D., Ramos, L.G., Nabben, R., Vuik, C.: Accelerating the shifted Laplace preconditioner for the Helmholtz equation by multilevel deflation. J. Comput. Phys. **322**, 473–490 (2016). https://doi.org/10.1016/j.jcp.2016.06.025
36. Sheikh, A.H., Lahaye, D., Vuik, C.: On the convergence of shifted laplace preconditioner combined with multilevel deflation. Numer. Linear Algebra Appl. **20**(4), 645–662 (2013). https://doi.org/10.1002/nla.1882
37. Sourbier, F., Haidar, A., Giraud, L., Ben-Hadj-Ali, H., Operto, S., Virieux, J.: Three-dimensional parallel frequency-domain visco-acoustic wave modelling based on a hybrid direct/iterative solver. Geophys. Prospect. **59**(5), 834–856 (2011). https://doi.org/10.1111/j.1365-2478.2011.00966.x
38. Tang, J.M., Nabben, R., Vuik, C., Erlangga, Y.A.: Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. J. Sci. Comput. **39**, 340–370 (2009). https://doi.org/10.1007/s10915-009-9272-6
39. Taus, M., Zepeda-Núñez, L., Hewett, R.J., Demanet, L.: L-sweeps: A scalable, parallel preconditioner for the high-frequency Helmholtz equation. J. Comput. Phys. **420**, 109706 (2020). https://doi.org/10.1016/j.jcp.2020.109706
40. Tournier, P.H., Bonazzoli, M., Dolean, V., Rapetti, F., Hecht, F., Nataf, F., Aliferis, I., El Kanfoud, I., Migliaccio, C., De Buhan, M., Darbas, M., Semenov, S., Pichot, C.: Numerical modeling and high-speed parallel computing: New perspectives on tomographic microwave imaging for brain stroke detection and monitoring. IEEE Antennas Propag. Mag. **59**(5), 98–110 (2017). https://doi.org/10.1109/MAP.2017.2731199
41. Versteeg, R.: The marmousi experience: velocity model determination on a synthetic complex data set. Lead. Edge **13**(9), 927–936 (1994)
42. Williams, S., Waterman, A., Patterson, D.: Roofline: an insightful visual performance model for multicore architectures. Commun. ACM **52**(4), 65–76 (2009). https://doi.org/10.1145/1498765.1498785
43. Yovel, R., Treister, E.: LFA-tuned matrix-free multigrid method for the elastic Helmholtz equation. SIAM J. Sci. Comput. pp. S1–S21 (2024). https://doi.org/10.1137/23M1583466