Matrix-Free Parallel Preconditioned Iterative Solvers for the 2D Helmholtz Equation Discretized with Finite Differences

Chen, Jinqiang; Dwarka, Vandana; Vuik, Cornelis

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

Matrix-Free Parallel Preconditioned Iterative Solvers for the 2D Helmholtz Equation Discretized with Finite Differences

Jinqiang Chen[✉], Vandana Dwarka, and Cornelis Vuik

Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
{j.chen-11,v.n.s.r.dwarka,c.vuik}@tudelft.nl

Abstract. We present a matrix-free parallel iterative solver for the Helmholtz equation related to applications in seismic problems and study its parallel performance. We apply Krylov subspace methods, GMRES, Bi-CGSTAB and IDR(s), to solve the linear system obtained from a second-order finite difference discretization. The Complex Shifted Laplace Preconditioner (CSLP) is employed to improve the convergence of Krylov solvers. The preconditioner is approximately inverted by multigrid iterations. For parallel computing, the global domain is partitioned blockwise. The standard MPI library is employed for data communication. The matrix-vector multiplication and preconditioning operator are implemented in a matrix-free way instead of constructing large, memory-consuming coefficient matrices. These adjustments lead to direct improvements in terms of memory consumption. Numerical experiments of model problems show that the matrix-free parallel solution method has satisfactory parallel performance and weak scalability. It allows us to solve larger problems in parallel to obtain more accurate numerical solutions.

1 Introduction

The Helmholtz equation describes the phenomena of time-harmonic wave scattering in the frequency domain. It is widely studied in computational electromagnetics, with applications in seismic exploration, sonar, antennas, and medical imaging. To solve the Helmholtz equation numerically, we discretize it and obtain a linear system $Ax = b$. The linear system matrix is sparse, symmetric, non-Hermitian, and indefinite [1]. 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 for increasing wavenumber. 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.

Many efforts have been made to solve the problem accurately with high performance. Originally derived from [2], the industry standard, also known as the Complex Shifted Laplace Preconditioner (CSLP) [3,4] does show good properties for medium

wavenumbers. Nevertheless, the eigenvalues shift to the origin as the wavenumber increases. These near-zero eigenvalues have an unfavorable effect on the convergence speed of Krylov-based iterative solvers. Recently, a higher-order approximation scheme to construct the deflation vectors was proposed to reach close to wavenumber-independent convergence [5].

The development of scalable parallel Helmholtz solvers is also ongoing. One approach is to parallelize existing advanced algorithms. Kononov and Riyanti [6,7] first developed a parallel version of Bi-CGSTAB preconditioned by multigrid-based CSLP. Gordon and Gordon [8] parallelized their so-called CARP-CG algorithm (Conjugate Gradient acceleration of CARP) blockwise. The block-parallel CARP-CG algorithm shows improved scalability as the wavenumber increases. Calandra et al. [9] proposed a geometric two-grid preconditioner for 3D Helmholtz problems, which shows strong scaling properties in a massively parallel setup. Another approach is the Domain Decomposition Method (DDM), which originates from the early Schwarz Methods. DDM, as a preconditioner mostly, has been widely used to develop parallel solution methods for the Helmholtz problems. For comprehensive surveys, we refer the reader to [10–14] and references therein.

This work describes parallel versions of Krylov subspace methods, such as the Generalized minimal residual method (GMRES), Bi-CGSTAB, and IDR(s), preconditioned by the multigrid-based CSLP for the Helmholtz equation. We consider the CSLP preconditioner because it is the first and most popular method where the number of iterations scales linearly within medium wavenumbers. Based on a block-wise domain decomposition and a matrix-free implementation, our parallel framework contributes to robust parallel CSLP-preconditioned Krylov solvers for Helmholtz problems. It is the basis for scalable parallel computing. Numerical experiments show that, compared to [5,6] that assemble matrices, the matrix-free framework allows us to solve the Helmholtz problem with a larger grid size to reduce pollution errors related to grid resolution. The parallel efficiency is up to 70%. Its weak scaling performance means that a larger problem can be solved in about the same amount of time as a smaller problem as long as the number of tasks increases proportionally.

The rest of this paper is organized as follows. Section 2 describes the mathematical model that we will discuss. All numerical methods we use are given in Sect. 3. The numerical performance is explored in Sect. 4. Finally, Sect. 5 contains our conclusions.

2   Mathematical Model

We will consider the following 2D Helmholtz equation on a rectangular domain Ω with boundary $\Gamma = \partial\Omega$. The Helmholtz equation reads

$$-\Delta u(x,y) - k^2 u(x,y) = b(x,y), \text{ on } \Omega \tag{1}$$

supplied with Dirichlet boundary conditions $u(x,y) = g(x,y)$ or first-order Sommerfeld boundary conditions $\frac{\partial u(x,y)}{\partial n} - \mathrm{i}ku(x,y) = 0$, on $\partial\Omega$. i is the imaginary unit. n and $g(x,y)$ represent the outward normal and the given data of the boundary respectively. $b(x,y)$ is the source function. k is the wavenumber. The frequency is f, the speed of propagation is c, which are related by $k = \frac{2\pi f}{c}$.

3   Numerical Methods

3.1   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 is used. The discrete Helmholtz operator A_h can be obtained by adding the diagonal matrix $-k^2 I_h$ to the Laplacian operator $-\Delta_h$, i.e. $A_h = -\Delta_h - k^2 I_h$. Therefore, the stencil of the discrete Helmholtz operator is

$$[A_h] = \frac{1}{h^2} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 - k^2 h^2 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{2}$$

In the case of the Sommerfeld radiation condition, ghost points located outside the boundary points can be introduced for the boundary points. For instance, suppose $u_{0,j}$ is a ghost point on the left of $u_{1,j}$, the normal derivative can be approximated by $\frac{\partial u}{\partial n} - iku = \frac{u_{0,j} - u_{2,j}}{2h} - iku_{1,j} = 0$. We can rewrite it as $u_{0,j} = u_{2,j} + 2hiku_{1,j}$, which can be used in the above computational stencil. The discretization of first-order Sommerfeld boundary conditions will result in a complex-valued linear system.

3.2   Preconditioned Krylov Subspace Methods

Among the representative Krylov subspace methods, GMRES and Bi-CGSTAB are suitable choices for the Helmholtz equation, as they are designed for non-singular problems. Also, the IDR(s) developed by Sonneveld and van Gijzen [17] is an efficient alternative to Bi-CGSTAB for Helmholtz problems. Compared with full GMRES, Bi-CGSTAB and IDR(s) have short recurrences and are easily parallelizable.

As for the preconditioner, we will focus on the CSLP due to its satisfactory performance and easy setup. The CSLP is defined by

$$M_h = -\Delta_h - (\beta_1 + \beta_2 i) k^2 I_h \tag{3}$$

We need to compute the inverse of preconditioner M_h in the preconditioned Krylov-based algorithms. It is usually too costly to invert a preconditioner like CSLP directly. One idea is to approximately solve the preconditioner by using the multigrid method [4]. It is necessary to choose a proper complex shift [18], since a small complex shift may affect the convergence of the multigrid method. In the numerical experiments of this paper, $\beta_1 = 1, \beta_2 = 0.5$ will be used.

A multigrid method involves several components that need a careful design to achieve excellent convergence. In this paper, damped Jacobi smoother with relaxation $\omega = 0.8$ is used. The so-called full weighting restriction operator and the bilinear interpolation operator are employed for the inter-grid transfer operations. The coarse grid operator M_{2h} is constructed by re-discretizing on the coarse mesh in the same way that the operator M_h is obtained on the fine mesh. This is known as the discretization coarse grid operator (DCG). The classical multigrid V-cycle is performed. Instead of solving the coarse-mesh problem directly, we will solve it by full GMRES.

Suppose a problem with N unknowns is solved by the CSLP-preconditioned Krylov subspace method by assembling the matrices. According to the complexity analysis in [5, 19], except the variables vectors, we need extra memory to store the sparse matrix A_h with $5N$ non-zero elements, M_h with $5N$ non-zero elements, M_{2h} with $\frac{9N}{4}$, inter-grid transfer operator Z with $\frac{9N}{4}$, etc. To minimize memory limitations and solve real-world large-scale problems, we implement the preconditioned Krylov subspace methods in a matrix-free way instead of constructing the coefficient matrices explicitly.

3.3   Matrix-Free Parallel Implementation

For the matrix-vector multiplication, like $A_h \mathbf{u}_h$ in outer iterations, $M_h \mathbf{u}_h$ in the smoother and $M_{2h} \mathbf{u}_{2h}$ in the preconditioner solver, can be replaced by stencil computations. For example, results of $A_h \mathbf{u}_h$ can be obtained by Algorithm 1. The inter-grid transfer operations can also be performed in a matrix-free way according to the linear interpolation/restriction polynomials.

Algorithm 1. Matrix-free $\mathbf{v_h} = A_h \mathbf{u_h}$.

1: Input array $\mathbf{u_h}$;
2: According to Eq. 2: $ap = \frac{4-k^2h^2}{h^2}$, $aw = ae = as = an = -\frac{1}{h^2}$;
3: Internal grid points ($i = 2 \cdots nx - 1$, $j = 2 \cdots ny - 1$):
4:     $v_h(i,j) = a_p u(i,j) + a_w u(i-1,j) + a_e u(i+1,j) + a_s u(i,j-1) + a_n u(i,j+1)$;
5: Boundary grid points ($i = 1, nx$, $j = 1, ny$): update a_p, a_w, a_e, a_s and a_n and compute $v_h(i,j)$;

6: Return $\mathbf{v_h}$.

To implement parallel computing, the standard MPI library is employed for data communications among the processors. Based on the MPI Cartesian topology, we can partition the computational domain blockwise. The partition is carried out between two grid points. One layer of overlapping grid points is introduced outward at each interface boundary to represent the adjacent grid points. In our method, the grid unknowns are stored as an array based on the grid ordering (i, j) instead of a column vector based on x-line lexicographic ordering.

We implement the parallel multigrid iteration based on the original global grid. According to the relationship between the fine grid and the coarse grid, the parameters of the coarse grid are determined by the grid parameters of the fine one. For example, point (i_c, j_c) in the coarse grid corresponds to point $(2i_c - 1, 2j_c - 1)$ in the fine grid. For a V-cycle, after reaching a manually predefined coarsest grid size, the coarsening operation will stop and solve the coarsest problem by GMRES in parallel, which may incur some efficiency loss. In this paper, the predefined coarsest global grid size is $nc_x \times nc_y = 9 \times 9$ as the maximum number of processors we use is 4×4.

4   Numerical Experiments

The solver is developed in Fortran and compiled by GNU Fortran and runs on a Linux compute node with Intel(R) Xeon(R) Gold 6152 (2.10GHz) CPUs. For outer iterations,

the L_2-norm of preconditioned relative residuals are reduced to 10^{-6}. According to pre-experiments, the stopping criterion for the coarse grid preconditioner solver should be 2–3 orders of magnitude smaller than the stopping criterion for the outer iteration. We use 10^{-8} as the stopping criterion for the coarse grid preconditioner solver. The Wall-ClockTime for the preconditioned Krylov-based solver to reach the stopping criterion is denoted by t_w. The speedup S_p is defined by $S_p = \frac{t_1}{t_p}$, where t_1 and t_p are the WallClock-Time for sequential and parallel computation, respectively. The parallel efficiency E_p is given by $E_P = \frac{S_p}{np} \times 100\%$, where np is the number of processors.

First, we consider a model problem (MP-1) with a point source described by the Dirac delta function $\delta(x,y)$, imposed at the center $(x_0, y_0) = (0.5, 0.5)$. The wave propagates outward from the center of the domain. Dirichlet boundary conditions are imposed. The analytical solution for this problem is given in [15].

Compared to the analytical solutions given by [15], our parallel preconditioned GMRES gives a fair approximation of the exact solution with relative errors (RErr.) less than 5×10^{-6}. Parallel partitioning also has no effect on the results. The main differences in the amplitude of the waves are caused by the finite-difference approximation of the Dirac function. As shown in Table 1, if we simultaneously and proportionally scale the problem size and the number of processors (in bold), the WallClockTime almost stays constant. It means our parallel framework has weak scalability. It indicates that a parallel efficiency of up to 75% is satisfactory.

Table 1. Parallel performance of CSLP preconditioned GMRES for MP1 with wavenumber $k = 100$.

grid size	np	#Iter	RErr	t_w	Speedup	E_p
161×161	1	350	1.638E−06	**7.07**	1.00	–
	4	350	1.596E−06	2.06	3.43	85.68
321×321	1	348	1.476E−06	32.18	1.00	–
	4	348	1.592E−06	**8.10**	3.97	99.34
481×481	1	358	4.319E−06	78.22	1.00	–
	9	359	4.444E−06	**9.02**	8.68	96.41
641×641	1	339	2.000E−06	121.44	1.00	–
	4	339	1.657E−06	33.75	3.60	89.97
	16	339	2.158E−06	**9.88**	12.29	76.79

Most physical problems of geophysical seismic imaging describe a heterogeneous medium. The so-called Wedge problem (MP-2) is a typical problem with simple heterogeneity. It mimics three layers with different velocities hence, different wavenumbers. The rectangular domain $\Omega = [0, 600] \times [-1000, 0]$ is split into three layers, where the wave velocity c is constant within each layer but different from each other. A point source is located at $(x_0, y_0) = (300, 0)$. The wavenumber is $k(x,y) = \frac{2\pi f}{c(x,y)}$, where f is the frequency. The distribution of wave velocity $c(x,y)$ refers to [6]. First-order Sommerfeld boundary conditions are imposed on all boundaries.

The 2D wedge problem is used to evaluate the performance of our parallel solution method for a simple heterogeneous medium. Besides, the matrix-free parallel framework is not limited to the GMRES algorithm. All the ingredients can be directly generalized to other Krylov methods like Bi-CGSTAB and IDR(s). In Table 2 we give the WallClockTime and the required number of matrix-vector multiplications of different CSLP-preconditioned Krylov methods for the wedge problem. It illustrates that our matrix-free parallel CSLP-preconditioned method is still suitable for heterogeneous Helmholtz problems. It still leads to satisfactory scalability if we increase the number of processors correspondingly while refining the grid.

Table 2. MP-2: CPU time consumed by different parallel CSLP-preconditioned Krylov methods while refining the grid, $f = 40\,\text{Hz}$. The number of matrix-vector multiplications is in parentheses.

Grid size	np	GMRES	Bi-CGSTAB	IDR(4)
385×641	2	25.34 (278)	6.20 (321)	6.10 (282)
769×1281	8	31.92 (283)	7.98 (301)	7.41 (251)

The so-called Marmousi problem [16] is a well-known benchmark problem (MP-3). It contains 158 horizontal layers in the depth direction, making it highly heterogeneous, see [6] for an illustration. In our numerical experiments, first-order Sommerfeld boundary conditions are imposed on all boundaries. The source frequency $f = 40\,\text{Hz}$, grid size 2945×961, which indicates $kh \leq 0.54$ and guarantees more than 10 grids per wavelength. In [6], the Marmousi problem with grid size 2501×751 has to be solved on at least two cores due to memory limitations. The matrix-free framework allows us to solve larger problems within even a single core.

Table 3 presents the required number of matrix-vector multiplications (denoted by #Matvec), CPU-time, and relative speedup of different CSLP-preconditioned Krylov methods for the Marmousi problem. One can find that a huge number of iterations are required. GMRES has the least number of matrix-vector multiplications but requires the most CPU time reducing the parallel efficiency to 50%. This is due to the Arnoldi process in GMRES requiring a lot of dot product operations, which need global communication in parallel computing. IDR(4) and Bi-CGSTAB exhibit higher parallel efficiency than GMRES. The results illustrate that the matrix-free parallel CSLP-preconditioned method also works for the highly heterogeneous Helmholtz problems.

Table 3. MP-3: parallel performance of different parallel CSLP-preconditioned Krylov methods, $f = 40\,\text{Hz}$, grid size 2945×961.

GMRES					Bi-CGSTAB					IDR(4)				
np	#Matvec	Time(s)	Sp	Ep	np	#Matvec	Time(s)	Sp	Ep	np	#Matvec	Time(s)	Sp	Ep
1	2872	61705.58	–	–	1	4124	2431.94	–	–	1	4438	2881.58	–	–
3	2872	21892.21	2.82	93.95	3	4435	712.34	3.41	113.80	3	4688	854.72	3.37	112.38
12	2872	10309.02	5.99	49.88	12	4513	279.09	8.71	72.61	12	4484	334.59	8.61	71.77

5   Conclusions

In this paper, we studied a matrix-free parallel solution method using preconditioned Krylov methods for Helmholtz problems. The Complex Shifted Laplace Preconditioner is used, which is approximately inverted by multigrid iterations. The matrix-free parallel framework is suitable for different Krylov methods. Numerical experiments of model problems demonstrate the robustness, satisfactory parallel performance, and weak scalability of our matrix-free parallel solution method. It allows us to solve larger problems in parallel to obtain more accurate numerical solutions.

References

1. Freund, R., Golub, G., Nachtigal, N.: Iterative solution of linear systems. Acta Numer. **1**, 57–100 (1992)
2. Bayliss, A., Goldstein, C.I., Turkel, E.: An iterative method for the Helmholtz equation. J. Comput. Phys. **49**(3), 443–457 (1983)
3. 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)
4. 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)
5. Dwarka, V., Vuik, C.: Scalable convergence using two-level deflation preconditioning for the Helmholtz equation. SIAM J. Sci. Comput. **42**(2), A901–A928 (2020)
6. 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 (2008)
7. Riyanti, C.D., et al.: A parallel multigrid-based preconditioner for the 3D heterogeneous high-frequency Helmholtz equation. J. Comput. Phys. **224**(1), 431–448 (2007)
8. 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)
9. 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: Lahaye, D., Tang, J., Vuik, K. (eds.) Modern Solvers for Helmholtz Problems. GM, pp. 141–155. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-28832-1_6
10. Engquist, B., Ying, L.: Sweeping preconditioner for the Helmholtz equation: moving perfectly matched layers. Multisc. Model. Simul. **9**(2), 686–710 (2011)
11. Chen, Z., Xiang, X.: A source transfer domain decomposition method for Helmholtz equations in unbounded domain. SIAM J. Numer. Anal. **51**(4), 2331–2356 (2013)
12. Stolk, C.C.: A rapidly converging domain decomposition method for the Helmholtz equation. J. Comput. Phys. **241**, 240–252 (2013)
13. 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)
14. 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)

15. Dwarka, V., Vuik, C.: Pollution and accuracy of solutions of the Helmholtz equation: a novel perspective from the eigenvalues. J. Comput. Appl. Math. **395**, 113549 (2021)
16. Versteeg, R.: The Marmousi experience: velocity model determination on a synthetic complex data set. Lead. Edge **13**(9), 927–936 (1994)
17. Sonneveld, P., Van Gijzen, M.B.: IDR (s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations. SIAM J. Sci. Comput. **31**(2), 1035–1062 (2009)
18. Gander, M.J., Graham, I.G., Spence, E.A.: Applying GMRES to the Helmholtz equation with shifted Laplacian preconditioning: what is the largest shift for which wavenumber-independent convergence is guaranteed? Numer. Math. **131**(3), 567–614 (2015)
19. Dwarka, V., Vuik, C.: Scalable multi-level deflation preconditioning for highly indefinite time-harmonic waves. J. Comput. Phys. **469**, 111327 (2022)