A Block Jacobi Sweeping Preconditioner for the Helmholtz Equation

Ruiyang Dai

1 Introduction

Solving Helmholtz problems using numerical methods is challenging due to the large, indefinite, and ill-conditioned linear systems that result, which cannot be solved using classical direct or iterative solvers [5]. While optimized Schwarz (OS) methods have been proposed as an alternative, the number of iterations required by Krylov methods increases with the number of subdomains, especially for layer-type domain decompositions [9]. Preconditioners, such as sweeping preconditioners, are necessary when using iterative methods to solve Helmholtz problems. Lately, there has been significant interest in sweeping preconditioners, invented by [3, 4], that achieve quasi-linear asymptotic complexity. Despite their effectiveness, sweeping preconditioners face challenges with parallel scalability due to the inherently sequential nature of their operations, as well as the need to ensure accurate and consistent information transfer between subdomains. These challenges can restrict the use of layer-type domain decompositions.

To address these challenges, recent research has focused on improving parallel performance through new sweeping strategies on checkerboard domain decompositions that can handle more general domain decompositions. Several sweeping algorithms have been proposed that improve parallelism by ensuring consistent transfer among subdomains, such as L-sweeps preconditioners [11], trace transfer-based diagonal sweeping preconditioners [7], and multidirectional sweeping preconditioners [2], with high-order transmission conditions and cross-point treatments [8].

Subdomains in sweeping algorithms can be assigned to Message Passing Interface (MPI) ranks based on rows or columns. This enables parallel application of sweeping algorithms for a single right-hand side. However, these approaches still have limitations, including long preconditioning procedures, waste of computation resources, and relatively high computation costs. To overcome these limitations, the

Ruiyang Dai
Laboratory J.L. Lions, Sorbonne Université, Paris, France, e-mail: ruiyang.dai@upmc.fr

authors propose a block Jacobi sweeping preconditioner that uses block Jacobi matrices to decompose full sweeps into several partial sweeps, which can be thought of as sweeps that operate on a subset of the subdomains. These partial sweeps can be performed concurrently. This approach enhances scalability and makes full use of resources on parallel computer architectures.

2 Notations

Let $\mathbf{i} = (i_1, i_2) \in \mathbb{N}^2$ be a multi-index denoting the subdomain number. We define the discrete l^1 norm by: $|\mathbf{i}|_1 := |i_1| + |i_2|$. We use the convention that two multi-indices $\mathbf{i}$ and $\mathbf{j}$ are equal if and only if $i_1 = j_1$ and $i_2 = j_2$.

Definition 1 The *lexicographic order* on multi-indices is the relation defined by $\mathbf{i} < \mathbf{j}$ if and only if $|\mathbf{i}|_1 < |\mathbf{j}|_1$, or $|\mathbf{i}|_1 = |\mathbf{j}|_1$ and $i_1 < j_1$.

Definition 2 The *lexicographic order* on pair multi-indices $(\mathbf{i}, \mathbf{j}) \in \mathbb{N}^2 \times \mathbb{N}^2$ is the relation defined by $(\mathbf{i}, \mathbf{j}) < (\mathbf{k}, \mathbf{l})$ if and only if $\mathbf{i} < \mathbf{k}$, or $\mathbf{i} = \mathbf{k}$ and $\mathbf{j} < \mathbf{l}$.

We define a function m that maps a pair of multi-indices which are in lexicographic order to natural numbers in a monotonically increasing fashion $m\colon \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{N}$, such that $m((1,1),(1,2)) = 1$, $m((1,1),(2,1)) = 2$, $m((1,2),(1,1)) = 3$, etc.

We consider $\Omega \subset \mathbb{R}^2$ be a square domain with boundary $\partial\Omega$, which is given by the union of the scattered boundary $\partial\Omega^{\text{sca}}$ with the external artificial boundaries Γ_i^{∞} for $i = 1, 2, 3, 4$, and its a non-overlapping checkerboard partition, which consists in a lattice of rectangular non-overlapping subdomains $\Omega_{\mathbf{i}}$ with N_1 columns and N_2 rows ($i_1 = 1, \ldots, N_1$, and $i_2 = 1, \ldots, N_2$), that is

$$\overline{\Omega} = \bigcup \overline{\Omega}_{\mathbf{i}}, \quad \text{and} \quad \Omega_{\mathbf{i}} \cap \Omega_{\mathbf{j}} = \varnothing \quad \text{for} \quad \mathbf{j} \neq \mathbf{i}.$$

And we say that $\exists \mathbf{i}$, such that $\Omega^{\text{sca}} \subseteq \Omega_{\mathbf{i}}^{\circ}$ and $\partial\Omega^{\text{sca}} \cap \partial\Omega_{\mathbf{i}} = \varnothing$. The boundary of a subdomain $\Omega_{\mathbf{i}}$ is split into two parts: the exterior part $\partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}$ and the interior part including decomposed interior interfaces $\Gamma_{\mathbf{i},\mathbf{j}} := \partial\Omega_{\mathbf{i}} \cap \partial\Omega_{\mathbf{j}}$ ($\mathbf{j} \neq \mathbf{i}$), and $\Gamma_{\mathbf{i},\mathbf{j}} = \Gamma_{\mathbf{j},\mathbf{i}}$. There are $N_{\text{dom}} = N_1 \times N_2$ subdomains, $N_e = 2N_1N_2 - N_1 - N_2$ interior interfaces. We define the number of diagonal groups $N_g := N_1 + N_2 - 1$.

3 Non-overlapping domain decomposition method

We study the 2D Helmholtz equation in Ω with an absorbing boundary condition on Γ_i^{∞}. For a more detailed description, see [2]. We seek the field $u(\mathbf{x})$ that verifies

$$\begin{cases} (-\Delta - \kappa^2)u = 0, & \text{in } \Omega, \\ (\partial_{\boldsymbol{n}_i} - \mathscr{T})u = 0, & \text{on } \Gamma_i^{\infty}, \\ u = -u^{\text{inc}}, & \text{on } \partial\Omega^{\text{sca}}, \end{cases} \tag{1}$$

where κ is the wavenumber, u^{inc} is the incident wave, $\partial_{\boldsymbol{n}}$ is the exterior normal derivative, and $\mathscr{T}$ is an impedance operator to be defined. We take the convention that the time-dependence of the fields is $e^{-\imath\omega t}$, where ω is the angular frequency and t is the time.

The domain decomposition method consists in considering the N_{dom} local subproblems coupled by the Robin conditions: Seek the field $u_{\mathbf{i}}(\mathbf{x})$ that verifies

$$\begin{cases} (-\Delta - \kappa^2)u_{\mathbf{i}} = 0, & \text{in } \Omega_{\mathbf{i}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},i}} - \mathscr{T})u_{\mathbf{i}} = 0, & \text{on } \partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})u_{\mathbf{i}} = (-\partial_{\boldsymbol{n}_{\mathbf{j},\mathbf{i}}} - \mathscr{T})u_{\mathbf{j}}, & \text{on } \Gamma_{\mathbf{i},\mathbf{j}}, \forall \mathbf{j} \in D_{\mathbf{i}}, \\ u_{\mathbf{i}} = -u^{\text{inc}}, & \text{on } \partial\Omega_{\mathbf{i}} \cap \partial\Omega^{\text{sca}}, \end{cases} \tag{2}$$

where the set $D_{\mathbf{i}} := \left\{\mathbf{j} \,|\, \mathbf{j} \neq \mathbf{i} \text{ and } \Gamma_{\mathbf{i},\mathbf{j}} \neq \varnothing\right\}$. The paper uses high-order absorbing boundary conditions (HABCs) as transmission conditions, which are effective for both layered-type and checkerboard-type domain decompositions [1, 8]. However, special treatment is required at corners in 2D cases for polygonal domains. Section 4 of the paper employs HABCs, but in the next section, the paper uses less effective boundary conditions based on the basic impedance operator to investigate the algebraic structure of the interface problem for clarity.

4 Interface problem

To derive the interface problem, let's introduce $w_{\mathbf{i}}(\mathbf{x})$ a lifting of the source: Seek $w_{\mathbf{i}}(\mathbf{x})$ that verifies

$$\begin{cases} (-\Delta - \kappa^2)w_{\mathbf{i}} = 0, & \text{in } \Omega_{\mathbf{i}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},i}} - \mathscr{T})w_{\mathbf{i}} = 0, & \text{on } \partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})w_{\mathbf{i}} = 0, & \text{on } \Gamma_{\mathbf{i},\mathbf{j}}, \forall \mathbf{j} \in D_{\mathbf{i}}, \\ u_{\mathbf{i}} = -u^{\text{inc}}, & \text{on } \partial\Omega_{\mathbf{i}} \cap \partial\Omega^{\text{sca}}, \end{cases} \tag{3}$$

By the linearity of the problem, the field $u_{\mathbf{i}}$ can be decomposed into $v_{\mathbf{i}} + w_{\mathbf{i}}$, where $v_{\mathbf{i}}$ is the field (2) after lifting the sources by (3). We introduce the local scattering operator $\mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})} : x_{m(\mathbf{i},\mathbf{k})} \to (-\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})v_{\mathbf{i}}$ where

$$\begin{cases} (-\Delta - \kappa^2)v_{\mathbf{i}} = 0, & \text{in } \Omega_{\mathbf{i}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},i}} - \mathscr{T})v_{\mathbf{i}} = 0, & \text{on } \partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{k}}} - \mathscr{T})v_{\mathbf{i}} = x_{m(\mathbf{i},\mathbf{k})}, & \text{on } \Gamma_{\mathbf{i},\mathbf{k}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{l}}} - \mathscr{T})v_{\mathbf{i}} = 0, & \text{on } \Gamma_{\mathbf{i},\mathbf{l}}, \forall \mathbf{l} \neq \mathbf{k}, \end{cases} \tag{4}$$

and $\mathbf{j}, \mathbf{k}, \mathbf{l} \in D_{\mathbf{i}}$. Using the linearity of the problem and the above scattering operator, we obtain the interface problem

$$(\partial_{\boldsymbol{n}_{\mathbf{j},\mathbf{i}}} - \mathscr{T})v_{\mathbf{j}} = \sum_{\mathbf{k}\in D_{\mathbf{i}}} \mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})}(\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{k}}} - \mathscr{T})v_{\mathbf{i}} + (-\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})w_{\mathbf{i}}, \quad \mathbf{j} \in D_{\mathbf{i}}.$$

We introduce the global scattering matrix $S \in M_{2N_e}(\mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})})$, the global additional variable vector $g \in M_{2N_e\times 1}(g_{m(\mathbf{j},\mathbf{i})})$, and the global right-hand-side vector $b \in M_{2N_e\times 1}(b_{m(\mathbf{j},\mathbf{i})})$, where

$$g_{m(\mathbf{j},\mathbf{i})} = (+\partial_{\boldsymbol{n}_{\mathbf{j},\mathbf{i}}} - \mathscr{T})v_{\mathbf{j}}, \quad b_{m(\mathbf{j},\mathbf{i})} = (-\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})w_{\mathbf{i}}.$$

We obtain that g is the solution to the global matrix

$$(I - S)g = b, \tag{5}$$

or

$$g_{m(\mathbf{j},\mathbf{i})} - \sum_{\mathbf{k}} \mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})} g_{m(\mathbf{i},\mathbf{k})} = b_{m(\mathbf{j},\mathbf{i})}, \quad \forall \mathbf{j}, \text{ for } \mathbf{i},\mathbf{k} \in D_{\mathbf{j}}. \tag{6}$$

5 Sweeping preconditioner

Let V_i be the $2N_e \times n_i$ matrix $V_i = (e_{m(\mathbf{i},\mathbf{j})})$, with $|\mathbf{i}|_1 = i$ $(i = 2, \ldots, N_g + 1)$, where each $e_{m(\mathbf{i},\mathbf{j})}$ is the $m(\mathbf{i},\mathbf{j})$-th column of the $2N_e \times 2N_e$ identity matrix, and n_i is the number of columns. One has $V_i^\top V_j = \mathbf{0}$, if $i \neq j$. Let $S_{i,j}$ be the $n_i \times n_j$ matrix $S_{i,j} = V_i^\top S V_j$.

Proposition 1 *The upper and lower triangular matrix of the global matrix $(I - S)$ can be decomposed by Gaussian elimination.*

Proof We denote the lower triangular matrix of the global matrix (5) S_L. Consider the following matrix $\prod_i (I - V_i S_{i,i-1} V_{i-1}^\top)$, $\quad i = 3, \ldots, N_g + 1$. Then, $\forall i = 3, \ldots, N_g$, we have

$$\begin{aligned}
&(I - V_i S_{i,i-1} V_{i-1}^\top)(I - V_{i+1} S_{i+1,i} V_i^\top) \\
&= I - V_i S_{i,i-1} V_{i-1}^\top - V_{i+1} S_{i+1,i} V_i^\top + V_i S_{i,i-1} V_{i-1}^\top V_{i+1} S_{i+1,I} V_i^\top \\
&= I - V_i S_{i,i-1} V_{i-1}^\top - V_{i+1} S_{i+1,i} V_i^\top
\end{aligned}$$

The last term at the 2nd line vanishs since $V_{i-1}^\top V_{i+1}$ is null. Hence, we have

$$\prod_i (I - V_i S_{i,i-1} V_{i-1}^\top) = I - \sum_i V_i S_{i,i-1} V_{i-1}^\top = I - S_L.$$

Similarily, we can proof that the upper triangular matrix of the global matrix (5) S_U can be decomposed as

$$I - S_U = \prod_i (I - V_{i-1} S_{i-1,i} V_i^\top), \quad i = N_g + 1, \ldots, 3,$$

which is a series of matrices. □

Next, we present the Symmetric Gauss-Seidel (SGS) sweeping preconditioner P_{SGS}. This matrix can then be rewritten as $P_{\mathrm{SGS}} \approx (I - S_L)(I - S_U)$, We can easily invert the matrix $P_{\mathrm{SGS}}^{-1} = (I - S_U)^{-1}(I - S_L)^{-1}$, with

$$\begin{aligned}(I - S_U)^{-1} &= \prod_i (I + V_{i-1} S_{i-1,i} V_i^\top), \quad i = 3, \ldots, N_g + 1, \\ (I - S_L)^{-1} &= \prod_i (I + V_i S_{i,i-1} V_{i-1}^\top), \quad i = N_g + 1, \ldots, 3.\end{aligned} \tag{7}$$

Observing Eqn. (7), we notice that it consists of a sequential process, in which there are $2(N_g - 1)$ sequential *steps* in total.

6 Block Jacobi sweeping preconditioner

Let W_{Li}, W_{Ui} be the $2N_e \times s_i$ matrix

$$\begin{aligned}W_{Li} &= (e_{m(\mathbf{i},\mathbf{j})}), \quad 2 + (i-1)N_1 \le |\mathbf{i}|_1 \le 2 + iN_1, \\ W_{Ui} &= (e_{m(\mathbf{i},\mathbf{j})}), \quad (1 + N_g) - iN_1 \le |\mathbf{i}|_1 \le (1 + N_g) - (i-1)N_1,\end{aligned}$$

where s_i is the number of columns $(e_{m(\mathbf{i},\mathbf{j})})$. Let S_{Li}, S_{Ui} be the $s_i \times s_i$ matrix

$$S_{Li} = W_{Li}^\top S_L W_{Li}, \quad S_{Ui} = W_{Ui}^\top S_U W_{Ui}.$$

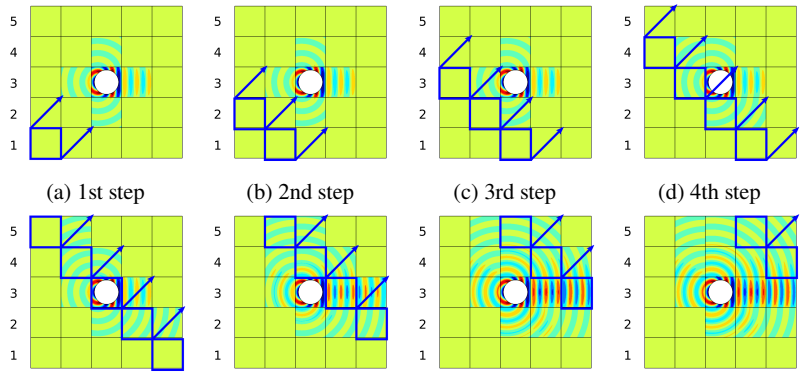
According to the *additive projection processes* [10], the next iterate can be defined as

$$\begin{aligned}g^{(k+1/2)} &= g^{(k)} + \sum_{i=1}^{p} W_{Li}(I_i - S_{Li})^{-1} W_{Li}^\top r^{(k)}, \\ g^{(k+1)} &= g^{(k+1/2)} + \sum_{i=1}^{p} W_{Ui}(I_i - S_{Ui})^{-1} W_{Ui}^\top r^{(k+1/2)}.\end{aligned}$$

$I - S_L$ and $I - S_U$ are quasi-equivalent to p blocks $I_i - S_{Li}$ and p blocks $I_i - S_{Ui}$, respectively, which form the forward and backward block Jacobi preconditioner. The block Jacobi preconditioner can be decomposed into a series of matrices, as stated in Proposition 1. The forward and backward block Jacobi preconditioner consists of the upper and lower block diagonals of $I - S$ and involves $2N_1$ sequential steps. This decomposition enhances the parallel performance of the sweeping preconditioner.

7 Numerical results

In this part, the block Jacobi sweeping preconditioner (BSP) is studied by considering a two-dimensional benchmark with a high-order finite element method and



(a) 1st step (b) 2nd step (c) 3rd step (d) 4th step

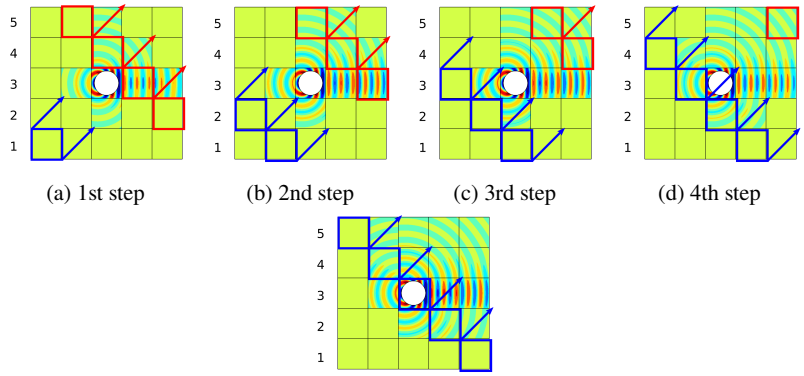
(e) 5th step (f) 6th step (g) 7th step (h) 8th step

Fig. 1 Scattering model in 2D ($k = 2\pi$) with a snapshot of the solution at different steps of the first GMRES iteration using the full sweeping preconditioner. Each row of subdomains is assigned to one MPI rank, and processors are identified by the numbers on the left side. Subdomains processed in parallel are highlighted in blue.

compared to the full sweeping preconditioner (SP). The proposed approaches and the computational results presented in this paper are implemented in parallel by MPI on a single multi-core computer. The linear systems arising from the sub-problems are solved by a sparse direct solver. The mesh generation, mesh decomposition, and post-processing are credited by Gmsh [6]. The parallelism of our approach is realized by assigning subdomains to MPI ranks in a row-based fashion such that the i-th row of the checkerboard partition is processed by rank i.

The test case is a homogeneous scattering problem in free space within a rectangle geometry ($\Omega = [-1.25, 2.50 \cdot N_1 - 1.25] \times [-1.25, 2.50 \cdot N_2 - 1.25]$), which is decomposed into $N_1 \times N_2$ rectangular subdomains. An incident plane wave is generated by a sound-soft circular cylinder of radius equal to 1 which is located at the Origin. On the circular cylinder, the Dirichlet boundary condition $u(\mathbf{x}) = -\exp^{ikx}$ is prescribed at the boundary of the sound-soft scatterer. The Padé-type HABC is prescribed on the exterior boundaries and the interior interfaces used as the absorbing boundary conditions and the transmission boundary conditions, respectively. The compatibility conditions are prescribed at the corners and the cross-points treatment is prescribed at the cross-points. The parameters of the HABC operator are $N_{\text{pade}} = 8$ and $\phi = \pi/3$. The following numerical setting are considered: $P7$ finite elements with 3 elements per wavelength ($h \approx 1/21$).

Figures 1 and 2 show snapshots of the solutions at different steps of forward sweep (sweep starts from the bottom-left corner to the top-right corner) of the 1st GMRES iteration with different sweeping preconditioners. Although the forward sweep in Fig. 1 goes through the whole computational domain from the bottom-left corner to the top-right corner, it takes 8 steps. If we take the backward sweep into account, there are 16 steps of the preconditioning procedure at each iteration. In the second situation, it only takes 5 steps in the forward partial sweeps (see Fig. 2).



(a) 1st step (b) 2nd step (c) 3rd step (d) 4th step

(e) 5th step

Fig. 2 Scattering model in 2D ($k = 2\pi$). Snapshot of the solution at different steps of forward sweep of 1st GMRES iteration with the block Jacobi sweeping preconditioner. The numbers at left side are processors' identities. Each row of subdomains is assigned to one MPI rank. Subdomains processed in parallel have same blue and red color, which represent two partial sweeps.

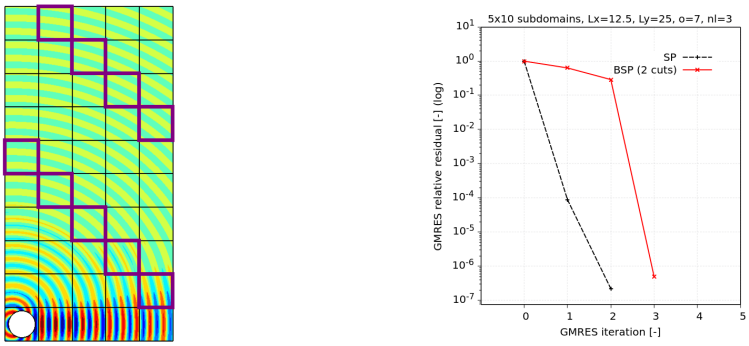


Fig. 3 Scattering model in 2D ($k = 2\pi$). The computational domain is decomposed into $N_1 \times N_2 = 5 \times 10$. Residual history with SP and BSP with two cuts. In this context, two cuts imply that $p = 3$.

Figure 3 shows snapshots of the solutions and residual histories of GMRES with the different preconditioners for the partition $N_1 \times N_2 = 5 \times 10$. All forward/backward (partial) sweeps of these preconditioners start from the bottom-left/top-left to the top-right/bottom-left. The violet boxes indicate the cut location which separates partial sweeps.

The residual histories obtained with the two different preconditioners in Fig. 3, where the relative residual suddenly drops in residual history at the first iteration when a full sweeping preconditioner is used. With the block Jacobi sweeping preconditioner used, it happens at the third iteration, which corresponds to the number of partial sweeps, that is to say, there are two partial sweeps.

Table 1 Scattering model in 2D ($k = 20\pi$). Number of iterations and runtime in seconds with the two different preconditioners for different domain partitions. "ni" stands for the number of iterations, "ns" the number of steps per iteration, and "t" time. The number of MPI ranks is equal to N_2.

$N_1 \times N_2$	SP (ni)	SP (ns)	SP (t)	BSP (ni)	BSP (ns)	BSP (t)
5×5	3	16	32.6 s	3	10	25.4 s
5×10	3	26	49.0 s	4	10	33.0 s
5×15	4	36	90.8 s	6	10	51.3 s
5×20	5	46	147.4 s	8	10	70.9 s

The number of GMRES iterations and the runtime to reach a relative residual 10^{-6} with the two different preconditioners are given in Table 1. The runtime corresponds to the GMRES resolution phase.

References

1. Boubendir, Y., Antoine, X., and Geuzaine, C. A quasi-optimal non-overlapping domain decomposition algorithm for the helmholtz equation. *Journal of Computational Physics* **231**(2), 262–280 (2012).
2. Dai, R., Modave, A., Remacle, J.-F., and Geuzaine, C. Multidirectionnal sweeping preconditioners with non-overlapping checkerboard domain decomposition for helmholtz problems. *Journal of Computational Physics* **453**, 110887 (2022).
3. Engquist, B. and Ying, L. Sweeping preconditioner for the helmholtz equation: hierarchical matrix representation. *Communications on Pure and Applied Mathematics* **64**(5), 697–735 (2011).
4. Engquist, B. and Ying, L. Sweeping preconditioner for the helmholtz equation: moving perfectly matched layers. *Multiscale Modeling & Simulation* **9**(2), 686–710 (2011).
5. Ernst, O. G. and Gander, M. J. Why it is difficult to solve helmholtz problems with classical iterative methods. In: *Numerical Analysis of Multiscale Problems*, 325–363. Springer (2011).
6. Geuzaine, C. and Remacle, J.-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* **79**(11), 1309–1331 (2009).
7. Leng, W. and Ju, L. Trace transfer-based diagonal sweeping domain decomposition method for the helmholtz equation: Algorithms and convergence analysis. *Journal of Computational Physics* **455**, 110980 (2022).
8. Modave, A., Royer, A., Antoine, X., and Geuzaine, C. A non-overlapping domain decomposition method with high-order transmission conditions and cross-point treatment for helmholtz problems. *Computer Methods in Applied Mechanics and Engineering* **368**, 113162 (2020).
9. Nataf, F., Rogier, F., and de Sturler, E. Optimal interface conditions for domain decomposition methods. Tech. rep., CMAP Ecole Polytechnique (1994).
10. Saad, Y. *Iterative Methods for Sparse Linear Systems*. SIAM (2003).
11. Taus, M., Zepeda-Núñez, L., Hewett, R. J., and Demanet, L. L-sweeps: A scalable, parallel preconditioner for the high-frequency helmholtz equation. *Journal of Computational Physics* **420**, 109706 (2020).