

Inexact Subdomain Solves Using Deflated GMRES for Helmholtz Problems

Bootland, N.; Dwarka, V.; Jolivet, P.; Dolean, V.; Vuik, C.

DOI

[10.1007/978-3-030-95025-5_11](https://doi.org/10.1007/978-3-030-95025-5_11)

Publication date

2022

Document Version

Final published version

Published in

Domain Decomposition Methods in Science and Engineering XXVI

Citation (APA)

Bootland, N., Dwarka, V., Jolivet, P., Dolean, V., & Vuik, C. (2022). Inexact Subdomain Solves Using Deflated GMRES for Helmholtz Problems. In S. C. Brenner, A. Klawonn, J. Xu, E. Chung, J. Zou, & F. Kwok (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI* (pp. 127-135). (Lecture Notes in Computational Science and Engineering; Vol. 145). Springer. https://doi.org/10.1007/978-3-030-95025-5_11

Important note

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

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Inexact Subdomain Solves Using Deflated GMRES for Helmholtz Problems

N. Bootland, V. Dwarka, P. Jolivet, V. Dolean, and C. Vuik

1 Introduction

In recent years, domain decomposition based preconditioners have become popular tools to solve the Helmholtz equation. Notorious for causing a variety of convergence issues, the Helmholtz equation remains a challenging PDE to solve numerically. Even for simple model problems, the resulting linear system after discretisation becomes indefinite and tailored iterative solvers are required to obtain the numerical solution efficiently. At the same time, the mesh must be kept fine enough in order to prevent numerical dispersion ‘polluting’ the solution [4]. This leads to very large linear systems, further amplifying the need to develop economical solver methodologies.

Domain decomposition (DD) techniques combined with Krylov solvers provide a way to deal with these large systems [6]. While the use of two-level deflation and DD techniques have been explored before, their expedience has primarily been measured in terms of providing a way to add a coarse space to obtain a two-level DD

Niall Bootland

University of Strathclyde, Department of Mathematics and Statistics, Glasgow, UK
e-mail: niall.bootland@strath.ac.uk

Vandana Dwarka

Delft University of Technology, Delft Institute of Applied Mathematics, Delft, The Netherlands
e-mail: v.n.s.r.dwarka@tudelft.nl

Pierre Jolivet

University of Toulouse, CNRS, IRIT, Toulouse, France
e-mail: pierre.jolivet@enseeiht.fr

Victorita Dolean

University of Strathclyde, Department of Mathematics and Statistics, Glasgow, UK
Université Côte d’Azur, CNRS, Laboratoire J.A. Dieudonné, Nice, France
e-mail: work@victoritadolean.com

Cornelis Vuik

Delft University of Technology, Delft Institute of Applied Mathematics, Delft, The Netherlands
e-mail: c.vuik@tudelft.nl

preconditioner [1, 2]. Without a coarse space, DD methods typically do not scale with the number of subdomains. Moreover, subdomain sizes need to be relatively small in order to optimally use local memory and direct solution methods on subdomains.

In this work we focus on the subdomain solves. Instead of using a direct solution method, we solve the local systems using GMRES preconditioned by a two-level deflation approach. As a result, similar to the inclusion of a coarse space on the fine-level, we obtain a two-level method on each subdomain as well. The inexact solve on the subdomains will allow for larger subdomains by reducing computing and memory requirements, especially in 3D. In order to allow for inexact subdomain solves we require a flexible wrapper for the outer iteration—the application of the DD preconditioner—and so we will use FGMRES. Our local subdomain solves will then employ a preconditioned GMRES iteration. It is well known that the cost of GMRES increases with each iteration. Thus, in order to mitigate the number of iterations at the subdomain level, we use a two-level deflation preconditioner [5, 8].

The techniques proposed here will feature as the topic of future research on large-scale 3D applications using pollution-free meshes. In this work we introduce the key ideas and begin an initial exploration by considering a simple 2D model problem.

2 Model problem, discretisation and preconditioning strategies

Our model problem consists of the Helmholtz equation posed on the unit square:

$$-\Delta u - k^2 u = f \quad \text{in } \Omega = (0, 1)^2, \quad (1a)$$

$$u = 0 \quad \text{on } \partial\Omega. \quad (1b)$$

Here, the parameter k denotes the wave number. Problem (1) is well-posed so long as k^2 is not a Dirichlet eigenvalue of the corresponding Laplace problem. Solving the problem with Dirichlet conditions provides a more robust test for the solver, as there is no shift keeping the spectrum away from the origin [3, 8]. In this work we will assume the problem, and any sub-problems, are well-posed. To discretise (1) we use piecewise linear (P1) finite elements on a uniform grid with mesh spacing given by $h = 2\pi k^{-1} n_{\text{ppwl}}^{-1}$, where n_{ppwl} is the number of (grid) points per wavelength (hereinafter referred to as “ppwl”). To test the solver performance we initially ensure $n_{\text{ppwl}} \approx 10$. We then double this to approximately 20 ppwl to obtain more accurate numerical solutions. Letting $V^h \subset H_0^1(\Omega)$ denote the space of piecewise linear functions on our finite element mesh T^h of Ω , our discrete solution $u_h \in V^h$ satisfies the weak formulation $a(u_h, v_h) = F(v_h) \forall v \in V^h$, where

$$a(u, v) = \int_{\Omega} \left(\nabla u \cdot \nabla v - k^2 uv \right) dx \quad \text{and} \quad F(v) = \int_{\Omega} f v dx. \quad (2)$$

With the standard basis for V^h , we can write the weak formulation as finding the solution to the linear system $\mathbf{A}u = \mathbf{f}$. We now consider how to solve such systems.

The global matrix A is preconditioned by a one-level domain decomposition method. To construct the decomposition, we define an overlapping partition $\{\Omega_j\}_{j=1}^N$ of Ω together with a restriction operator R_j to move from the global level to the subdomain level. We only consider Cartesian (rectangular) subdomains. Using this decomposition, the restricted additive Schwarz (RAS) preconditioner is defined by

$$M_{RAS}^{-1} = \sum_{j=1}^N R_j^T D_j A_j^{-1} R_j, \quad (3)$$

where D_j are diagonal matrices representing a partition of unity ($\sum_{j=1}^N R_j^T D_j R_j = I$) and $A_j = R_j A R_j^T$ are the local Dirichlet matrices. Each subdomain solve requires the solution of a local auxiliary linear system, which we denote by $A_j \tilde{\mathbf{u}}_j = \tilde{\mathbf{f}}_j$ as a general case. We solve these systems using a two-level deflation approach, that is, deflation is used to accelerate the convergence of GMRES by removing the near-zero eigenvalues. For normal matrices it has been shown that convergence can be directly related to the behaviour of these near-zero eigenvalues [7].

The two-level deflation preconditioner is defined as a projection operator P_j which leads to solving $P_j A_j \tilde{\mathbf{u}}_j = P_j \tilde{\mathbf{f}}_j$, where

$$P_j = I - A_j Q_j \quad \text{with} \quad Q_j = Z_j E_j^{-1} Z_j^T \quad \text{and} \quad E_j = Z_j^T A_j Z_j. \quad (4)$$

The rectangular matrix Z_j in this particular setting is called the deflation matrix and its columns span the deflation space. The choice of Z_j strongly dictates the overall convergence behaviour. Here, we use quadratic rational Bézier curves, as they have been shown to provide satisfactory convergence [5]. Consequently, if we let $\tilde{u}_j^i$ represent the i -th degree of freedom (DOF) on subdomain Ω_j , then in 1D Z_j maps these nodal approximations onto their coarse-grid counterpart as follows

$$[Z_j \tilde{u}_j]^i = \frac{1}{8} \left(\tilde{u}_j^{2i-2} + 4 \tilde{u}_j^{2i-1} + 6 \tilde{u}_j^{2i} + 4 \tilde{u}_j^{2i+1} + \tilde{u}_j^{2i+2} \right). \quad (5)$$

As such, Z_j can be constructed using the following 1D stencil $\frac{1}{8} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$. The dimension of Z_j will then be $n_j \times \frac{n_j}{2}$, where n_j is the size of the local 1D system.

In 2D on Cartesian grids this can be naturally extended by using the Kronecker product. For a rectangular subdomain, letting Z_j^x denote the 1D deflation matrix in the x -direction and Z_j^y that in the y -direction, the 2D deflation matrix is given by

$$Z_j = Z_j^y \otimes Z_j^x, \quad (6)$$

assuming lexicographic ordering running through x coordinates first.

3 Numerical results

We now provide numerical results for our model problem on the unit square. We take the right-hand side f to be given by a point-source at the centre of the domain. Unless stated otherwise, 10 ppwl are used to construct the mesh and we let n_{glob} be the number of DOFs along each edge of the square. For the domain decomposition, we use a uniform decomposition into N (square) subdomains. Overlap is added by appending one layer of mesh elements in a Cartesian manner (note that this means subdomains touching only one edge of Ω are rectangular rather than square).

For the outer solve we use preconditioned FGMRES with the one-level RAS preconditioner (3). The tolerance for the relative residual has been set at 10^{-6} . For the inner solve on the subdomain level we use preconditioned GMRES with the two-level deflation preconditioner (4) instead of a direct solver. Note that subdomains systems are decoupled and so can be solved locally in parallel. We will vary the inner tolerance for the relative residual between 10^{-10} and 10^{-2} in order to assess an appropriate level of accuracy needed when solving the subdomain problems. The solver is equipped to deal with both symmetric (Dirichlet) and non-symmetric systems (Sommerfeld), as we use the RAS preconditioner together with GMRES, which do not require symmetry.

All matrices are constructed using FreeFem (<http://freefem.org/>) while the solvers are then implemented using PETSc (<http://www.mcs.anl.gov/petsc/>). Computations are carried out on a laptop with an i7-10850H processor having 6 cores (12 threads).

3.1 Direct subdomain solves

We start by constructing a benchmark where we use a direct solution method for the subdomain solves, namely via an LU -decomposition. Table 1 shows that the number of iterations does not scale as the number of subdomains N increases, in agreement with the literature. The number of iterations also rapidly increases with the wave number k . The inclusion of a coarse space to improve both k - and N -scalability on top of inexact subdomain solves will be explored in future research. An interesting observation is that increasing n_{ppwl} leads to a higher iteration count. The opposite effect has been observed when using a two-level deflation preconditioner [5, 8]. There, a finer mesh leads to a smaller number of iterations as the mapping of the eigenvectors from the fine- and coarse-grid becomes more accurate.

Table 1: FGMRES iteration counts using the one-level RAS preconditioner with direct subdomain solves.

10 ppwl						
		N				
k	n_{glob}	4	9	16	25	
20	30	20	27	48	45	
40	60	31	60	85	101	
80	120	64	133	191	216	
160	240	159	262	365	495	

20 ppwl						
		N				
k	n_{glob}	4	9	16	25	
20	60	20	40	42	59	
40	120	37	66	89	115	
80	240	76	131	189	255	
160	480	130	289	398	520	

3.2 Inexact subdomain solves

The direct subdomain solves remain feasible for medium-size problems. Once we move to high-frequency 3D problems the subdomain systems become larger and the direct solver will start to become inefficient and consume more computing power and memory. In order to assess the feasibility of inexact solves, we will use the benchmarks from Section 3.1 and compare with our iterative method. The aim of these experiments is twofold. First, we want to examine the scalability with respect to the number of subdomains once we substitute the direct solution method. Secondly, we want to observe what level of accuracy is needed at the subdomain level such that the outer number of iterations remains within a satisfactory range.

High-tolerance: 10^{-10}

We start with a tolerance of 10^{-10} with results given in Table 2. This case is the closest to the use of a direct solver (see Table 1). Comparing, we observe that the results are almost identical when using 4 subdomains. Once we increase the number of subdomains N , the number of FGMRES iterations increases for both 10 and 20 ppwl. However, the increase is more noticeable when using 10 ppwl. For example, when $k = 160$ and $N = 9$ a direct solver on the subdomains leads to 262 iterations while the inexact approach converges in 301 outer iterations. However, when we double n_{ppwl} to 20, we go from 289 to 292 outer iterations. In all cases, as expected, the number of outer iterations increases with the wave number k . A similar yet slower increase in iteration counts is observed for the average number of inner iterations required by the deflated GMRES approach on the subdomains. As mentioned previously, the deflation preconditioner becomes more efficient on finer meshes. This can also be observed in our results: while the number of outer FGMRES iterations (using RAS) increases when moving from 10 ppwl to 20 ppwl, the number of inner GMRES iterations (using two-level deflation) decreases as the local subdomain systems become larger. Additionally, in this case, the number of inner iterations appears to be scaling better with the wave number k .

Table 2: FGMRES iteration counts using the one-level RAS preconditioner with subdomain problems solved inexactly to a relative tolerance of 10^{-10} using GMRES with a two-level deflation preconditioner. In parentheses we display the average number of GMRES iterations per subdomain solve.

		10 ppwl						20 ppwl			
		N						N			
k	n_{glob}	4	9	16	25	k	n_{glob}	4	9	16	25
20	30	20 (23)	27 (25)	56 (24)	46 (21)	20	60	20 (30)	43 (26)	42 (25)	59 (23)
40	60	32 (35)	62 (29)	86 (26)	101 (25)	40	120	37 (31)	66 (32)	93 (27)	112 (27)
80	120	65 (45)	137 (32)	192 (32)	221 (29)	80	240	75 (36)	132 (30)	191 (30)	268 (27)
160	240	160 (63)	301 (58)	373 (36)	518 (33)	160	480	131 (53)	292 (47)	407 (31)	530 (28)

Medium-tolerance: 10^{-5}

In Table 3 we report the results when lowering the inner tolerance to 10^{-5} . We compare with the results reported in Table 2. A general observation is that as k increases, so does the number of outer FGMRES iterations. Naturally, lowering the inner tolerance ensures we require less iterations to converge on the subdomains.

For the largest wave number reported and 9 subdomains, we needed 262 outer iterations when using a direct solver. For the inexact approach with tolerance 10^{-5} the number of outer iterations increases to 301, which is the same as when using a tolerance of 10^{-10} . However, for the finer mesh with 20 ppwl the number of outer iterations goes up from 292 to 308. At the same time, the number of inner iterations reduces accordingly: from 58 to 40 for 10 ppwl and from 47 to 31 for 20 ppwl.

If we increase the number of subdomains from 9 to 25, the outer number of FGMRES iterations increases more rapidly. If we use 20 ppwl, the direct local solves lead to 520 outer iterations. This goes up to 555 when we use the iterative approach and a tolerance of 10^{-5} . Note that the extra outer iterations compared to a tolerance of 10^{-10} is surmountable as relaxing the tolerance by 5 orders of magnitude leads to an increase of 25 iterations (from 530 to 555). Moreover, if we compare the number of inner iterations, a finer mesh works better with two-level deflation preconditioned GMRES on the subdomains, since we now need 15 iterations on average.

Similarly, on the finer mesh the number of inner iterations scales better with increasing wave number k by adding more subdomains. Contrary to the results for $k < 160$, moving from 10 ppwl to 20 ppwl with 25 subdomains leads to less outer iterations. Thus, for larger wave numbers, using a finer mesh with more subdomains leads to a smaller number of outer and inner iterations. This effect is not observed with respect to the direct solves on the subdomains and/or the use of the tolerance 10^{-10} (see Table 1 and Table 2): here as we go from 10 ppwl to 20 ppwl, the number of outer iterations always increases.

Table 3: FGMRES iteration counts using the one-level RAS preconditioner with subdomain problems solved inexactly to a relative tolerance of 10^{-5} using GMRES with a two-level deflation preconditioner. In parentheses we display the average number of GMRES iterations per subdomain solve.

10 ppwl						20 ppwl					
		N						N			
k	n_{glob}	4	9	16	25	k	n_{glob}	4	9	16	25
20	30	20 (12)	27 (14)	65 (14)	46 (12)	20	60	20 (15)	50 (13)	43 (13)	59 (12)
40	60	34 (20)	76 (15)	95 (14)	122 (14)	40	120	37 (16)	76 (18)	104 (14)	118 (14)
80	120	72 (26)	154 (17)	210 (19)	262 (17)	80	240	86 (19)	148 (16)	218 (16)	314 (14)
160	240	175 (44)	301 (40)	398 (19)	572 (20)	160	480	144 (34)	308 (31)	431 (16)	555 (15)

Low-tolerance: 10^{-2}

Finally, we reduce the inner tolerance to just 10^{-2} and report results in Table 4. While the overall observations follow a similar trend to the previous case, the number of inner iterations are reduced drastically. This comes at the expense of a higher number of outer iterations as the wave number and number of subdomains increase.

The most noticeable result is again for the highest wave number, $k = 160$. If we use 20 ppwl, the direct local solves lead to 520 outer iterations. This goes up to 584 when we use the iterative approach with a tolerance of 10^{-2} . The extra outer iterations compared to a tolerance of 10^{-10} is again surmountable as relaxing the tolerance by 8 orders of magnitude leads to an increase of 54 outer iterations (from 530 to 584). Meanwhile, the average inner iterations goes from 28 (for 10^{-10}) to 15 (for 10^{-5}), and finally to 6 iterations when using a tolerance of 10^{-2} .

Analogous to the case where we set the tolerance to 10^{-5} , we again observe that, as an exception to the rule that increasing n_{ppwl} leads to more outer iterations, the number of outer iterations actually decreases when using 20 ppwl instead of 10 ppwl. This effect is only observed for the iterative approach on the subdomains in combination with a sufficiently low tolerance, here 10^{-5} or 10^{-2} .

An important take-away message here is that the outer iteration of the one-level RAS preconditioned FGMRES method is able to reach convergence even when the subdomain systems are solved only to a relatively low level of accuracy.

Table 4: FGMRES iteration counts using the one-level RAS preconditioner with subdomain problems solved inexactly to a relative tolerance of 10^{-2} using GMRES with a two-level deflation preconditioner. In parentheses we display the average number of GMRES iterations per subdomain solve.

10 ppwl						20 ppwl					
		N						N			
k	n_{glob}	4	9	16	25	k	n_{glob}	4	9	16	25
20	30	20 (5)	27 (6)	73 (7)	50 (5)	20	60	21 (7)	59 (5)	49 (5)	72 (5)
40	60	42 (8)	87 (7)	109 (7)	126 (6)	40	120	44 (7)	84 (8)	124 (6)	134 (6)
80	120	84 (13)	172 (8)	241 (10)	298 (8)	80	240	94 (9)	154 (7)	229 (8)	333 (6)
160	240	211 (30)	332 (22)	451 (9)	1007 (8)	160	480	154 (20)	327 (19)	450 (7)	584 (6)

To provide some perspective on these results, we repeat the 10 ppwl experiment but now use GMRES preconditioned by ILU(0) as the subdomain solution method (with tolerance 10^{-2}). The results in Table 5 show that the average number of inner iterations drastically increases. Further, for $k = 160$ simulation run times were noticeably increased. Note that, for $k = 160$ and $N = 25$, using the two-level deflation preconditioner with 20 ppwl leads to both a lower inner and outer iteration count.

Table 5: FGMRES iteration counts using the one-level RAS preconditioner with subdomain problems solved inexactly to a relative tolerance of 10^{-2} using GMRES with an ILU(0) preconditioner. In parentheses we display the average number of GMRES iterations per subdomain solve. Here we use 10 ppwl.

		N			
k	n_{glob}	4	9	16	25
20	30	28 (21)	42 (12)	74 (10)	53 (7)
40	60	44 (64)	80 (36)	110 (25)	131 (17)
80	120	89 (250)	177 (121)	239 (83)	303 (55)
160	240	221 (983)	344 (502)	475 (260)	658 (199)

4 Conclusions

In this work we examined the utility of the one-level RAS preconditioner together with FGMRES to solve the 2D homogeneous Helmholtz equation when using an inexact solution method for the subdomain solves. Our results support the notion that the direct solve can be substituted by an efficient iterative solver. By using two-level deflation as a local preconditioner, we are able to keep the number of inner iterations on the subdomains low and scalable with respect to the wave number k .

The next step would be to include a coarse space and experiment with a two-level RAS preconditioner combined with inexact solves on the subdomains. Adding a coarse space would reduce the impact on the number of outer iterations when substituting direct solves for inexact solves on the subdomains. The trade-off between a higher number of outer iterations and a fast and memory efficient local subdomain solve needs to be analysed in large-scale applications to determine the break-even point in terms of wall-clock time, identifying where the iterative approach can be beneficial. Especially in high-frequency 3D applications, the inclusion of the coarse space can become a bottleneck. To reduce the outer iteration count, we either need to solve with a large coarse space or on larger subdomains. Both options can be costly when using a direct method and so an inexact solver is likely more suitable. Further, in the iterative approach the inner Krylov solvers for the subdomain problems may also benefit from the use of recycling techniques, which could further reduce the number of inner iterations and increase efficiency.

References

1. N. Bootland and V. Dolean. On the Dirichlet-to-Neumann coarse space for solving the Helmholtz problem using domain decomposition. In: *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lect. Notes Comput. Sci. Eng., vol. 139, pp. 175–184. Springer, Cham (2021).
2. N. Bootland, V. Dolean, P. Jolivet and P.H. Tournier. A comparison of coarse spaces for Helmholtz problems in the high frequency regime. *Comput. Math. Appl.* **98**, 239–253 (2021).
3. N. Bootland, V. Dolean, A. Kyriakis and J. Pestana. Analysis of parallel Schwarz algorithms for time-harmonic problems using block Toeplitz matrices. *Electron. Trans. Numer. Anal.* **55**, 112–141 (2022).
4. A. Deraemaeker, I. Babuška and P. Bouillard. Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions. *Internat. J. Numer. Methods Engrg.* **46**(4), 471–499 (1999).
5. V. Dwarka and C. Vuik. Scalable convergence using two-level deflation preconditioning for the Helmholtz equation. *SIAM J. Sci. Comput.* **42**(2), A901–A928 (2020).
6. M.J. Gander and H. Zhang. A class of iterative solvers for the Helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods. *SIAM Rev.* **61**(1), 3–76 (2019).
7. G. Meurant and J. Duintjer Tebbens. The role eigenvalues play in forming GMRES residual norms with non-normal matrices. *Numer. Algorithms* **68**(1), 143–165 (2015).
8. A.H. Sheikh, D. Lahaye, L. Garcia Ramos, R. Nabben and C. Vuik. Accelerating the shifted Laplace preconditioner for the Helmholtz equation by multilevel deflation. *J. Comput. Phys.* **322**, 473–490 (2016).