Algebraic Multilevel Preconditioner For the Helmholtz Equation In Heterogeneous Media

Matthias Bollhöfer* Marcus J. Grote† Olaf Schenk‡

Abstract

An algebraic multilevel (ML) preconditioner is presented for the Helmholtz equation in heterogeneous media. It is based on a multi-level incomplete LDL^T factorization and preserves the inherent (complex) symmetry of the Helmholtz equation. The ML preconditioner incorporates two key components for efficiency and numerical stability: symmetric maximum weight matchings and an inverse–based pivoting strategy. The former increases the block diagonal dominance of the system, whereas the latter controls $\|L^{-1}\|$ for numerical stability. When applied recursively, their combined effect yields an algebraic coarsening strategy, similarly to algebraic multigrid methods, even for highly indefinite matrices. The ML preconditioner is combined with a Krylov subspace method and applied as a "black-box" solver to a series of challenging two and three-dimensional test problems, mainly from geophysical seismic imaging. The numerical results demonstrate the robustness and efficiency of the ML preconditioner, even at higher frequency regimes.

Keywords. Helmholtz equation, inhomogeneous media, symmetric indefinite matrix, algebraic multilevel preconditioning, graph-pivoting, inverse-based pivoting

AMS specification. 65N55, 65F10, 65N22, 78A45, 86A15

1 Introduction

The efficient simulation of acoustic, electromagnetic and elastic wave phenomena is of fundamental importance in a wide range of engineering applications such as ultrasound tomography, wireless communication, or geophysical seismic imaging. When the problem of interest is linear and the time dependence is harmonic, the unknown wave field u typically satisfies the Helmholtz equation

$$-\Delta u - k^2 u = f \tag{1}$$

in a bounded domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, supplemented with appropriate physical or radiation boundary conditions, which guarantee the well-posedness of the boundary value problem. Here the *wave number* $k = \omega/c$ represents the ratio of the (constant) angular frequency, ω, and the speed of propagation c. If the medium is heterogenous, $c = c(x)$ varies in space and so does $k = k(x)$.

Higher values of k imply shorter wave lengths, $\lambda = 2\pi/k$, and thus require a smaller mesh size, h, in any standard numerical method. For second-order finite differences or continuous piecewise linear finite elements, the rule-of-thumb of at least "10 grid points per wave length", that is

$$kh \simeq \frac{2\pi}{10}, \tag{2}$$

*Institute of Computational Mathematics, TU Braunschweig, D-38106 Braunschweig, Germany (m.bollhoefer@tu-bs.de).

†Department of Mathematics, University of Basel, Rheinsprung 21, CH-4051 Basel, Switzerland (marcus.grote@unibas.ch).

‡Department of Computer Science, University of Basel, Klingelbergstrasse 50, CH-4056 Basel Switzerland (olaf.schenk@unibas.ch).

yields reasonable accuracy. When the computational domain, Ω, extends over many wave lengths, phase errors due to numerical dispersion tend to accumulate and induce an additional "pollution" error of order k^3h^2 [4, 7]. To control it, an even finer mesh (or a high-order discretization) is required. We shall ignore this effect here, that is choose h for a given k according to (2), but remark that the smaller kh, the easier the numerical solution for a fixed problem size.

Discretization of (1) by finite differences or finite elements leads to a linear system of equations

$$Ax = b, \tag{3}$$

where A is a large, sparse, ill-conditioned, complex symmetric (not Hermitian) $N \times N$ matrix; moreover, at higher wave numbers, A becomes increasingly indefinite – here we always assume A to be nonsingular, which is guaranteed for h sufficiently small [46].

According to (2), larger values of k imply even larger values of $N \sim h^{-d}$, thereby making the problem even harder to solve. In two space dimensions, the fill-in needed by direct sparse solvers scales as $N \log N$; hence, they are quasi-optimal in terms of storage and belong to the fastest and most robust solution methods in practice. In three space dimensions, however, the fill-in scales as N^2 [28], and iterative methods thus become competitive. Yet the indefiniteness introduced by the deceptively simple diagonal shift of the Laplacian in (1) has prevented most classical preconditioners for elliptic problems, such as multigrid, domain decomposition, or incomplete factorization based methods, from being as efficient as in the positive-definite case.

Classical multigrid methods rely on two key ingredients: smoothing and coarse grid correction, and both have difficulties at higher wave numbers. Indeed, standard Jacobi or Gauß-Seidel smoothers become unstable, while the coarser grids must remain sufficiently fine due to the oscillatory nature of the solution [5, 33]. To overcome some of these difficulties, Elman, Ernst and O'Leary [19] proposed GMRES smoothing together with flexible GMRES acceleration. For constant k, Brandt and Livshits [11] overcame the inherent difficulty in defining a meaningful coarse problem by augmenting the standard V-cycle with ray grids, where the oscillatory error components are represented as products of smooth envelope functions and oscillatory lattice principal components. Although their method converges independently of k, it does not easily generalize to unstructured grids or heterogeneous media – see also [36] for a similar approach.

Alternatively, instead of applying a multigrid iteration directly to the Helmholtz equation (1), one can apply it to a different nearby problem, where it is more effective, and then use it to precondition an outer iteration. Due to its simplicity, this class of "shifted Laplacian" preconditioners has recently received much attention. In 1983 Bayliss, Goldstein, and Turkel [6] used a single SSOR sweep to approximately invert the Laplacian as a preconditioner for the CG methods applied to the normal equations. More recently, Laird and Giles [35] enforced coercivity of the preconditioner by reverting the sign in front of the zeroth-order term. Erlangga, Vuik, and Osterlee [20, 21] extended this approach by introducing a complex shift, thus proposing as a preconditioner a standard multigrid V-cycle applied to

$$-\Delta u - (1 - i\beta)k^2 u = f, \qquad i = \sqrt{-1}, \tag{4}$$

where β is a free parameter. Both (1) and (4) are discretized with centered finite differences on the same equidistant grid and with identical boundary conditions. The complex shift induced by β moves the spectrum away from the origin and corresponds to damping in the time domain. The larger β, the faster the multigrid convergence of the perturbed problem is, yet clearly the perturbation must remain sufficiently small to yield a useful preconditioner for the original problem. In practice, $\beta = 0.5$ was found to be effective [20, 43, 52]; for constant kh, the number of outer iterations then increases only linearly with k. Airaksinen et al. developed a similar algebraic multigrid preconditioner coupled with a full GMRES outer iteration [1].

Domain decomposition (DD) methods reduce the solution of (3) to a succession of smaller local problems, which can be solved by direct methods, for instance, and in parallel. To achieve a convergence rate independent of the number of subdomains, a coarse space correction must be included. Cai and Widlund established the convergence of a two-level overlapping Schwarz method for (1) with constant k if the coarse grid is fine enough [14]. Later Cai et al. improved

the convergence of overlapping Schwarz algorithms by adding Sommerfeld-like conditions at the boundaries of the subdomains and GMRES acceleration [13].

Non-overlapping DD methods are ineffective if Dirichlet or Neumann transmission conditions are used at the interfaces, as they lead to local resonance. The use of complex (Sommerfeld-like) Robin transmission conditions, however, yields convergent non-overlapping DD methods [8, 16, 27, 32]. In [23], Farhat, Macedo, and Lesoinne extended the popular FETI (Finite Element Tearing and Interconnecting) DD method to (1) by introducing two key ingredients: a complex diagonal interface matrix to regularize the problem in each subdomain and an auxilliary coarse problem based on plane waves. Recently, an improved dual-primal variant (FETI-DPH) was devised and used to solve large problems from engineering applications in a parallel environment [22].

Incomplete LU (ILU) factorization based preconditioners yield fast "black-box" preconditioners for a wide range of problems [45]. Although fairly robust in practice, ILU preconditioners often fail on (3) at higher k or generate prohibitively large fill-in. Indeed, small or nonzero diagonal entries can lead to unstable, highly ill-conditioned incomplete factors with a crippling effect on the preconditioner; again, a diagonal complex shift prior to the incomplete factorization improves the spectrum of the preconditioned system [38]. Alternatively, permutation and scaling strategies applied to (3) as a preprocessing step yield an equivalent linear system, which is more diagonally dominant, and thus reduce the need for partial pivoting.

In 1996 Olshowka and Neumaier introduced weigthed matchings as a static approximation to the pivoting order in Gaussian elimination [41]. The subsequent fast algorithms by Duff and Koster [17] for sparse matrices triggered a dramatic improvement of modern sparse direct solvers. In particular, borrowing from classical Bunch and Kaufmann [12] pivoting for symmetric matrices, Gärtner and Schenk [49] and Duff and Pralet [18] developed sparse direct solvers for symmetric indefinite systems that are orders of magnitude faster and more memory efficient than previous methods. Today weighted matching techniques are regularly used in modern sparse direct solvers [37, 49, 50]. Their positive impact on preconditioning has also been recognized both in the unsymmetric [9] and in the symmetric highly indefinite case [30]. Recently, the combination of fast weighted matching techniques with an inverse-based coarsening strategy from [10] enabled Schenk, Bollhöfer, and Römer [47] to compute a few interior eigenvalues and eigenvectors of large, sparse, real symmetric, and highly indefinite matrices with millions of rows and columns [48].

Here we develop a fast and robust algebraic multilevel preconditioner for (1) which combines two key ideas: a graph–pivoting strategy based on weighted graph matchings and an inverse based coarsening process, which drives the algebraic multilevel factorization. By using the symmetrized version [18,49] of maximum weight matchings [17,41], we preorder on every level the linear system, so that the largest entries in magnitude appear essentially in the tridiagonal part of the matrix. Next, we compute a partial incomplete LDL^T decomposition on every level, while rejecting pivots whenever $\|L^{-1}\|$ exceeds a prescribed bound. When applied recursively, this procedure eventually leads to our algebraic multilevel ML preconditioner for (3), which is then solved with the SQMR (Symmetric Quasi-Minimal Residual) Krylov subspace method [25, 26].

The outline of our paper is as follows. In Section 2 we introduce symmetric maximum weighted matchings as a first key pre-processing step to our preconditioner. Next, we present the inverse-based pivoting strategy in Section 3 and show why it can be interpreted as an algebraic coarsening strategy from the point of view of traditional multigrid methods. We also analyze the eigenvalue structure on two subsequent levels and show that the eigenvalues closest to the origin are revealed by the coarsening process and that these smallest eigenvalues in modulus have their counter parts in the coarse grid system. Finally in Section 4, we present numerical results for large-scale Helmholtz problems both in two and three space dimensions to demonstrate the robustness and efficiency of the ML preconditioner.

2 The graph–pivoting strategy

We will now introduce a graph–pivoting strategy that identifies large entries in the coefficient matrix A which yield acceptable pivots for the (multilevel) incomplete factorization process, when

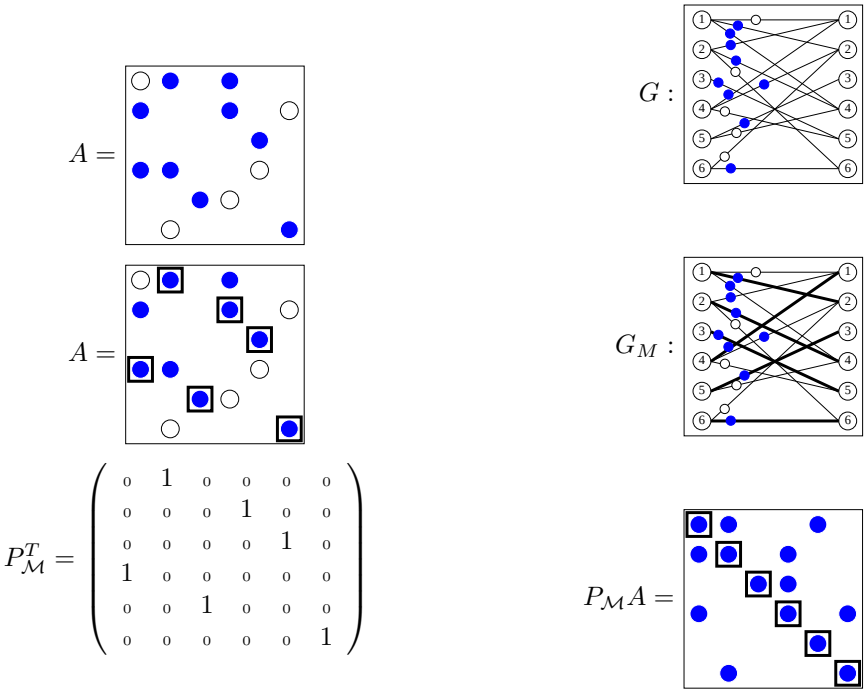

Figure 1: Nonsymmetric graph–pivoting. A small numerical value is indicated by a ∘-symbol and a large numerical value by an •-symbol. Top: a symmetric indefinite matrix A and its bipartite graph G. Middle: maximum weighted matched entries in $\mathcal{M}$ are denoted with squares and the corresponding edges in boldface. Bottom: the permutation matrix $P_{\mathcal{M}} = (e_4; e_1; e_5; e_2; e_3; e_6)$ and its effect on A.

permuted on or next to the main diagonal. This strategy is based on maximum weighted matchings in a bipartite graph associated with A and usually it improves the quality of the incomplete factors in a fashion complementary to more traditional pivoting techniques.

2.1 Graph–pivoting for nonsymmetric matrices

For any matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, its nonzero entries define a weighted bipartite graph with edges $\mathcal{E} = \{(i,j) : a_{ij} \neq 0\}$ of ordered pairs of row and column indices. A subset $\mathcal{M} \subset \mathcal{E}$ is called a matching, or a transversal, if every row index i and every column index j appears at most once in $\mathcal{M}$. A matching $\mathcal{M}$ is called perfect if its cardinality is n; it then defines a permutation matrix $P_{\mathcal{M}} = (p_{ij})$ with

$$p_{ij} = \begin{cases} 1 & (j,i) \in \mathcal{M}, \\ 0 & \text{else.} \end{cases} \tag{5}$$

For any nonsingular matrix A, a (nonunique) perfect matching always exists.

We now seek a matching $\mathcal{M}$ which increases the diagonal dominance of the permuted matrix $P_{\mathcal{M}}A$, in particular, we seek a permutation σ which maximizes the product of the diagonal values of $P_{\mathcal{M}}A$,

$$\prod_{i=1}^{n} |a_{\sigma(i)i}|. \tag{6}$$

This maximization problem, known in combinatorial optimization as the linear sum assignment problem or the bipartite weighted matching problem, is solved indirectly by a sparse variant of the Kuhn–Munkres algorithm [34,40]. Its complexity is $O(n^3)$ for $n\times n$ full matrices and $O\left(n^{1+\alpha}\log n\right)$

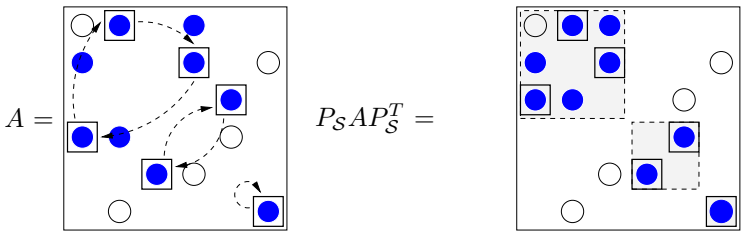

Figure 2: Symmetric graph pivoting. Left: the matrix A from Figure 1 for the cycle permutation $P_S = (e_1; e_2; e_4)(e_3; e_5)(e_6)$. Right: reordering the cycles symmetrically permutes the largest elements into diagonal blocks.

with $\alpha < 1$ for sparse matrices arising from finite-difference or finite-element discretizations [29] – for instance, $\alpha = 0.5$ for a standard second-order finite-difference discretization in the unit square. However in practice, the algorithm typically scales linearly with n [49].

In Figure 1 we observe the effect of permuting a matrix A using the permutation $P_{\mathcal{M}}$ associated with the maximum weighted matching $\mathcal{M}$. Clearly, $P_{\mathcal{M}}A$ is nonsymmetric but now has its largest nonzero entries on the main diagonal.

2.2 Graph–pivoting for symmetric indefinite matrices

Numerical discretizations of the Helmholtz equation usually yield a (complex) symmetric matrix A. While we seek to increase the diagonal dominance of A, we also wish to preserve that inherent symmetry; hence, a symmetric permutation PAP^T of A is needed. However, any symmetric permutation will leave zero or small diagonal entries on the main diagonal, which then often lead to instability in the incomplete factorization.

To circumvent the difficulties associated with small diagonal entries and yet preserve symmetry, we thus consider 2×2 diagonal block pivots,

$$\begin{pmatrix} a_{ii} & a_{i,i+1} \\ a_{i+1,i} & a_{i+1,i+1} \end{pmatrix}, \qquad a_{i,i+1} = a_{i+1,i} \ .$$

Whenever a_{ii} and/or $a_{i+1,i+1}$ is small, the corresponding off-diagonal entry $a_{i,i+1}$ must be large to guarantee a suitable 2×2 block pivot. Therefore we seek a permutation $P_{\mathcal{S}}$ that permutes large off-diagonal elements a_{ij} close to the main diagonal of $P_{\mathcal{S}}AP_{\mathcal{S}}^T$. To do so, we exploit the cycle structure of the permutation $P_{\mathcal{M}}$ associated with the nonsymmetric maximum weighted matching $\mathcal{M}$. By reordering the rows of $P_{\mathcal{M}}$ following individual cycles in $\mathcal{M}$, we obtain a new permutation $P_{\mathcal{S}}$ which permutes the largest entries into diagonal blocks [29]. For instance, the permutation $P_{\mathcal{M}} = (e_4; e_1; e_5; e_2; e_3; e_6)$ from Figure 1 yields the cycle representation $P_{\mathcal{S}} = (e_1; e_2; e_4)(e_3; e_5)(e_6)$, shown in Figure 2. As we symmetrically permute A using $P_{\mathcal{S}}$, we observe in Figure 2 how the largest entries of $A_{\mathcal{S}} = P_{\mathcal{S}}AP_{\mathcal{S}}^T$ are now located inside diagonal blocks. Since longer cycles result in larger diagonal blocks, which would generate prohibitively large fill-in, all longer cycles are subsequently broken down into disjoint 2×2 or 1×1 cycles.

In summary, the above graph-pivoting based on symmetric maximum weigthed matching permutes and scales A such that all entries are at most one in modulus; moreover, every diagonal block is either a 1×1 scalar with $|a_{ii}| = 1$ (in exceptional cases we will have $a_{ii} = 0$), or a 2×2 block with $|a_{i+1,i}| = |a_{i,i+1}| = 1$ and $|a_{ii}|, |a_{i+1,i+1}| < 1$. If A is complex valued we apply graph pivoting to $|A|$ instead. Furthermore, to keep the fill-in introduced during the elimination process minimal, we apply a nested dissection reordering [31] to the compressed graph of $A_{\mathcal{S}}$, thereby perserving the 1×1 and 2×2 block structure determined above.

Numerical experiments [18, 30, 48, 49] indicate that graph-pivoting based on symmetric maximum weight matchings typically waives the need for dynamic pivoting strategies as in [12]. Nonetheless, in the event of a nearly singular 2×2 diagonal block, the corresponding rows and

columns are permuted into the Schur complement and are handled on coarser levels subsequently, as described in Section 3.

3 Inverse–based pivoting

The ML preconditioner is based on a multilevel incomplete LDL^T factorization which uses the recent *inverse-based pivoting* strategy from [10,48] during the elimination process to (approximately) control the norm of the inverse triangular factor, $\|L^{-1}\|$, a crucial prerequisite for stability and accuracy. After describing the pivoting strategy, we shall deduce three important consequences from it: its relation to block-diagonal dominance, its connection to approximate inverses, and its similarity to algebraic multigrid methods.

3.1 The inverse-based pivoting strategy

Suppose the matrix A in (3) has been rescaled and reordered using the graph–pivoting strategy from Section 2. The tridiagonal part of A is thus sufficiently dominant to perform a block LDL^T decomposition, with the exception of a few nearly singular 2×2 block pivots, whose corresponding rows and columns are permuted into the Schur complement. This process eventually yields the partial decomposition,

$$\Pi^T A \Pi = \begin{pmatrix} B & F^T \\ F & C \end{pmatrix} = \begin{pmatrix} L_B & 0 \\ L_F & I \end{pmatrix} \begin{pmatrix} D_B & 0 \\ 0 & S_C \end{pmatrix} \begin{pmatrix} L_B^T & L_F^T \\ 0 & I \end{pmatrix}, \tag{7}$$

where D_B is block-diagonal, L_B is block-lower triangular, and Π is a permutation matrix that moves the "bad" pivots into the Schur complement

$$S_C = C - L_F D_B L_F^T. \tag{8}$$

Recursive application of this approach to S_C, including the graph–pivoting strategy from Section 2, eventually leads to our multilevel preconditioner.

Both the stability of the lower triangular factor in (7),

$$L = \begin{pmatrix} L_B & 0 \\ L_F & I \end{pmatrix}, \tag{9}$$

and the accuracy of the preconditioner hinge upon the size of L^{-1}. The pivoting strategy therefore ought to control $\|L^{-1}\|$, at least approximately, during the (incomplete) factorization process. We now describe in detail that crucial step which we refer to as *inverse-based pivoting* [10].

Ignoring the permutation Π in (7), we are faced with the partial decomposition

$$A = \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & I & 0 \\ L_{31} & 0 & I \end{pmatrix} \begin{pmatrix} D_{11} & 0 & 0 \\ 0 & S_{22} & S_{23} \\ 0 & S_{32} & S_{33} \end{pmatrix} \begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & I & 0 \\ L_{31} & 0 & I \end{pmatrix}^T,$$

at some intermediate stage of the factorization process. Here L_{11}, L_{21}, L_{31}, and D_{11} refer to the already computed part of the LDL^T decomposition whereas S_{22}, S_{23}, S_{32}, and S_{33} correspond to the Schur complement; the block S_{22} is either 1×1 or 2×2, and we simply select the partitioning that minimizes $\|S_{32}S_{22}^{-1}\|_\infty$.

For a given κ, we accept the pivot S_{22} if

$$\left\| \left(\begin{pmatrix} L_{11} & 0 & 0 \\ L_{21} & I & 0 \\ L_{31} & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & S_{32}S_{22}^{-1} & I \end{pmatrix} \right)^{-1} \right\| \lessapprox \kappa. \tag{10}$$

Otherwise, we reject it and symmetrically permute the corresponding rows and columns to the end of the linear system. In particular, whenever S_{22} is ill–conditioned or singular, the inverse

bound (10) on $S_{32}S_{22}^{-1}$ forces the factorization process to skip the corresponding rows and thus represents a true pivoting strategy. Doing so step-by-step finally yields the partial decomposition (7), with

$$\|L^{-1}\| \lessapprox \kappa. \tag{11}$$

Clearly, we cannot strictly enforce the above upper bound as any exact computation of $\|L^{-1}\|$ would be far too expensive. Instead we estimate $\|L^{-1}\|$ by an inexpensive sparse forward substitution "on the fly", which is essentially based on the classical approach from [15], for further details see also [10]. In order to let the inverse-based pivoting strategy act as a coarsening strategy in the sense of algebraic multigrid, we prescribe a fairly tight inverse bound κ, such as $\kappa = 3$, which tends to postpone a significant portion of the linear system to coarser levels. Even smaller values for κ, such as $\kappa = 1$, would result in a diagonal L.

To illustrate the effect of inverse-based pivoting when applied to (1) on the coarsening process, we now consider the following simple but typical two-dimensional wave guide problem from [35]:

$$\begin{aligned} -\Delta u(x,y) - k^2 u(x,y) &= 0 && (x,y) \in [0,1]^2, \\ \frac{\partial u(x,y)}{\partial n} &= 0 && y = 0 \text{ and } y = 1, \\ \frac{\partial u(x,y)}{\partial n} &= e^{-\frac{1}{2}(y-\frac{1}{2})^2} && x = 0, \\ \frac{\partial u(x,y)}{\partial n} + iku(x,y) &= 0, && x = 1. \end{aligned} \tag{12}$$

Here a horizontal sinusoidal Gaussian beam enters the computational domain from the left at $x = 0$ and propagates across the wave guide until it reaches the right boundary $x = 1$, where we impose a Sommerfeld-like absorbing boundary condition for simplicity. We choose $k = 80$ and discretize (12) with second-order finite differences on an equidistant 128×128 mesh, thereby satisfying (2).

To give an impression of the number and distribution of the nodes postponed from one level to the next, we identify in Figure 3 precisely those nodes inside the 20×20 lower left portion of the grid at two subsequent stages of the coarsening process. For $\kappa = 3$ we observe that inverse-based pivoting excludes bad pivots from the fine grid, thereby postponing their treatment to coarser levels, much like algebraic multigrid methods.

3.2 Inverse–based pivoting generalizes block-diagonal dominance

Diagonal dominance generally waives the need for pivoting in LDU decompositions and immediately leads to bounded matrices L^{-1} and U^{-1}. Borrowing from [24], we now prove that block-diagonal dominance indeed allows to predict the norm of the inverse triangular factors without ever computing or estimating them. In contrast, the inverse-based pivoting strategy circumvents the limited situation of diagonal dominance by directly controlling the inverse triangular factors.

Lemma 1 *Let $A \in \mathbb{R}^{n\times n}$ be partitioned as $A = (A_{ij})_{i,j=1,\dots,m}$ such that the diagonal blocks A_{ii} are square and nonsingular. Let $|\bullet|$ denote some fixed matrix norm with $|I| = 1$, and suppose there exists $\kappa > 1$ such that*

$$\sum_{j:\, j\neq i} |A_{ii}^{-1}A_{ij}| \leqslant \frac{\kappa-1}{\kappa}, \quad \sum_{i:\, i\neq j} |A_{ij}A_{jj}^{-1}| \leqslant \frac{\kappa-1}{\kappa}, \qquad \forall i,j = 1,\dots,m. \tag{13}$$

Then there exists a block triangular decomposition $A = LDU$ with the same partitioning as induced by A such that L and U^T are unit lower triangular matrices and D is a block diagonal matrix. Furthermore, with respect to this partitioning, the diagonal blocks of L and U are identity matrices. The inverse triangular factors L^{-1} and U^{-1} satisfy

$$\|L^{-1}\|_1 \leqslant \kappa, \quad \|U^{-1}\|_\infty \leqslant \kappa, \tag{14}$$

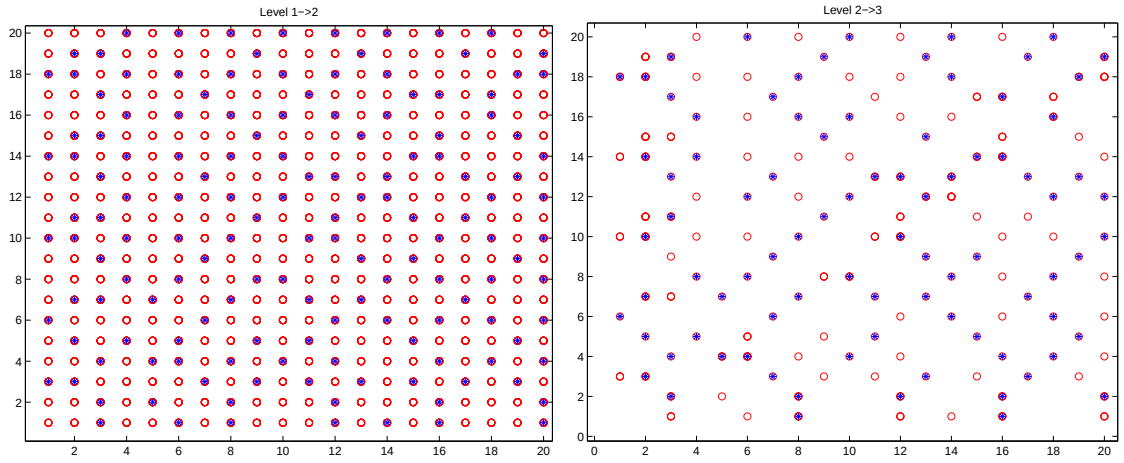

Figure 3: Two-dimensional wave guide with inverse–based pivoting recursively applied over the first two levels: dark nodes are postponed to the subsequent level and hence permuted into the Schur complement S_C.

where both $\| \bullet \|_1$ *and* $\| \bullet \|_\infty$ *are defined by blocks, i.e.,*

$$\| (M_{ij})_{ij} \|_1 := \max_{j=1,\ldots,n} \sum_{i=1}^{n} |M_{ij}|, \quad \| (M_{ij})_{ij} \|_\infty := \max_{i=1,\ldots,n} \sum_{j=1}^{n} |M_{ij}|.$$

Proof:
Let $\gamma = (\kappa - 1)/\kappa$, $0 < \gamma < 1$. One step of a block LU decomposition leads to

$$A = \begin{pmatrix} I & 0 \\ \hat{L} & I \end{pmatrix} \begin{pmatrix} A_{11} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I & \hat{U} \\ 0 & I \end{pmatrix},$$

where

$$\hat{L} = \left(A_{i1}A_{11}^{-1}\right)_{i\geqslant 2}, \quad \hat{U} = \left(A_{11}^{-1}A_{1j}\right)_{j\geqslant 2}, \quad S = \left(A_{ij} - A_{i1}A_{11}^{-1}A_{1j}\right)_{i,j\geqslant 2}.$$

From (13) we obtain for $i = 2, \ldots, m$ that

$$\begin{aligned}
\sum_{j\notin\{1,i\}} \left|(A_{ii} - A_{i1}A_{11}^{-1}A_{1i})^{-1}\left(A_{ij} - A_{i1}A_{11}^{-1}A_{1j}\right)\right| &\leqslant \\
\left|(I - A_{ii}^{-1}A_{i1}A_{11}^{-1}A_{1i})^{-1}\right| \sum_{j\notin\{1,i\}} \left(\left|A_{ii}^{-1}A_{ij}\right| + \left|A_{ii}^{-1}A_{i1}\right|\left|A_{11}^{-1}A_{1j}\right|\right) &\leqslant \\
\left|(I - A_{ii}^{-1}A_{i1}A_{11}^{-1}A_{1i})^{-1}\right| \left(\sum_{j\notin\{1,i\}} \left|A_{ii}^{-1}A_{ij}\right| + \left|A_{ii}^{-1}A_{i1}\right| \sum_{j\notin\{1,i\}} \left|A_{11}^{-1}A_{1j}\right|\right) &\leqslant \\
\left|(I - A_{ii}^{-1}A_{i1}A_{11}^{-1}A_{1i})^{-1}\right| \left(\gamma - \left|A_{ii}^{-1}A_{i1}\right| + \left|A_{ii}^{-1}A_{i1}\right|\left(\gamma - \left|A_{11}^{-1}A_{1i}\right|\right)\right). &\quad (15)
\end{aligned}$$

Next, we use Neumann series [39] to bound the first term in (15) as

$$\left|(I - A_{ii}^{-1}A_{i1}A_{11}^{-1}A_{1i})^{-1}\right| \leq (1 - \left|A_{ii}^{-1}A_{i1}A_{11}^{-1}A_{1i}\right|)^{-1} \leq (1 - \left|A_{ii}^{-1}A_{i1}\right|\ \left|A_{11}^{-1}A_{1i}\right|)^{-1}.$$

By using twice the fact that $0 < \gamma < 1$, the second term in (15) is bounded by

$$(\gamma - 1)\left|A_{ii}^{-1}A_{i1}\right| + \gamma - \left|A_{ii}^{-1}A_{i1}\right|\left|A_{ii}^{-1}A_{i1}\right| \quad \leqslant \quad \gamma\left(1 - \left|A_{ii}^{-1}A_{i1}\right|\left|A_{11}^{-1}A_{1i}\right|\right).$$

This implies that

$$\sum_{j\notin\{1,i\}} \left|(A_{ii} - A_{i1}A_{11}^{-1}A_{1i})^{-1}\left(A_{ij} - A_{i1}A_{11}^{-1}A_{1j}\right)\right| \leqslant \gamma.$$

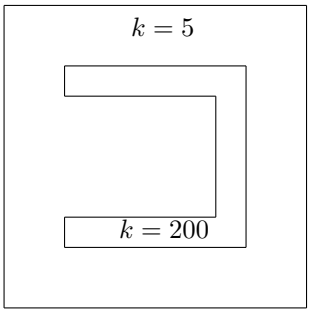

Figure 4: Two-dimensional wave guide problem (12) with varying wave number $k = k(x, y)$.

Similar arguments lead to

$$\sum_{j \notin \{1,i\}} \left| \left(A_{ij} - A_{i1}A_{11}^{-1}A_{1j}\right) \left(A_{jj} - A_{j1}A_{11}^{-1}A_{1j}\right)^{-1} \right| \leqslant \gamma.$$

Therefore, the Schur complement S satisfies the diagonal dominance criterion (13) too, with the same constant γ. Repeated use of the above argument to the Schur complement inductively implies that the resulting factors $L = (L_{ij})$ and $U = (U_{ij})$ satisfy

$$\sum_{i:\, i>j} |L_{ij}| \leqslant \gamma, \quad \sum_{j:\, j>i} |U_{ij}| \leqslant \gamma. \tag{16}$$

To prove the upper bounds on L^{-1} and U^{-1} in (14), we first write

$$L = I - E_L, \quad U = I - E_U,$$

where E_L and E_U correspond to the strict lower and upper triangular parts of $-L$ and $-U$, respectively. From (16), we then immediately find that $\|E_L\|_1 \leqslant \gamma$ and $\|E_U\|_\infty \leqslant \gamma$. Again we apply Neumann series to obtain

$$\|L^{-1}\|_1 \leqslant \frac{1-\gamma^n}{1-\gamma}, \quad \|U^{-1}\|_\infty \leqslant \frac{1-\gamma^n}{1-\gamma}.$$

Since

$$\frac{1-\gamma^n}{1-\gamma} \leqslant 1 + \frac{\gamma}{1-\gamma} = \kappa,$$

this concludes the proof of Lemma 1. □

Lemma 1 shows that the partial block-diagonal dominance of A in (13) immediately implies the upper bound (14) on the inverse triangular factors. For the linear system (3), however, the diagonal dominance criterion (13) usually requires the removal of too many columns and rows even for a small submatrix of A to fulfill (13). In contrast, if we directly impose (14), that is $\|L^{-1}\| \lessapprox \kappa$, we are no longer restricted to the rather limiting case of block-diagonal dominance while still keeping $\|L^{-1}\|$ at least approximately under control. In this sense inverse-based pivoting generalizes block diagonal dominance.

From its link to block-diagonal dominance, we expect inverse-based pivoting to select more coarse grid nodes precisely where A is less diagonally dominant. To illustrate the effect of local diagonal dominance, or lack thereof, on the coarsening process, we now consider again the example in (12) but with varying $k = k(x, y)$, as depicted in Figure 4. Indeed, Figure 5 shows how the coarsening process resulting from inverse-based pivoting moves significantly more nodes to the coarse grid precisely where k is large, that is where A displays little or no diagonal dominance. Since the resulting coarse linear systems become increasingly indefinite – see also Theorem 2 below – the graph–pivoting techniques using 2×2 block pivots from Section 2 are truly necessary for numerical stability.

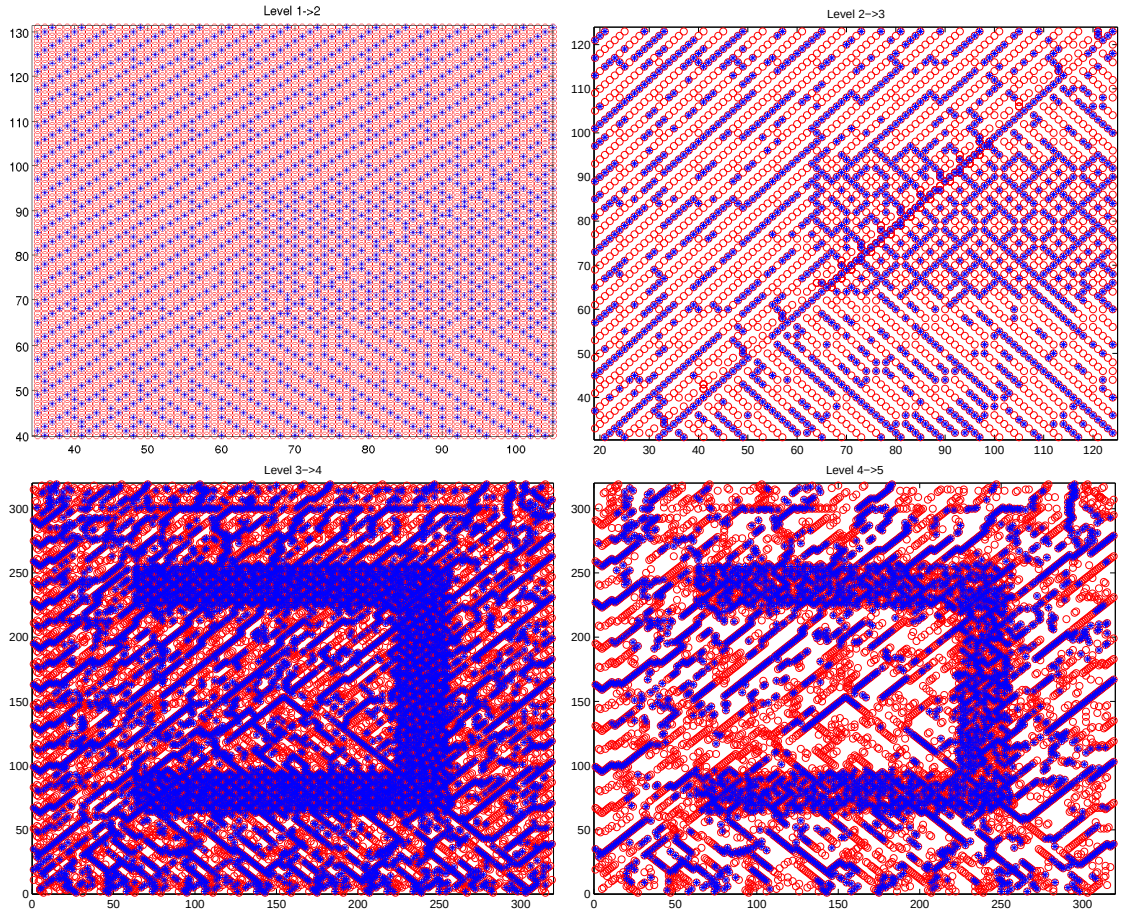

Figure 5: Inverse–based pivoting recursively applied over four levels. Dark nodes refer to postponed nodes that become part of S_C. The upper two frames show only a subsection of the grid in the lower left part of the domain.

3.3 Inverse-based pivoting returns sparse approximate inverses

For small κ, such as $\kappa = 3$, the constraint $\|L^{-1}\| \lessapprox \kappa$ implies that

$$\sum_{i=j+1}^{n} \|(L^{-1})_{ij}\| \lessapprox 2.$$

In theory, any particular (block-) row of L^{-1} might have $n - j$ blocks of equal magnitude but, in practice, only a few blocks $\|(L^{-1})_{ij}\|$ are typically of order 1, while the remaining entries essentially vanish; thus, we expect L^{-1} to be approximately sparse.

To illustrate the (approximate) sparsity of L^{-1}, we consider again the two-dimensional wave guide problem in (12) with varying $k = k(x, y)$, see Figure 4, and concentrate on the lower left part of

$$L^{-1} = \begin{pmatrix} L_B^{-1} & 0 \\ -L_F L_B^{-1} & I \end{pmatrix}.$$

Figure 6 shows the sparsity structure of $L_F L_B^{-1}$ computed by an exact factorization using inverse–based pivoting. Although the sub-block is as large as $29,167 \times 72,594$, we find only $614,009$ entries larger than 10^{-2} in magnitude, i.e. as little as 8.5 nonzero entries per column, and $180,780$ entries larger than 10^{-1} in magnitude, i.e. only about 2.5 nonzero entries per column.

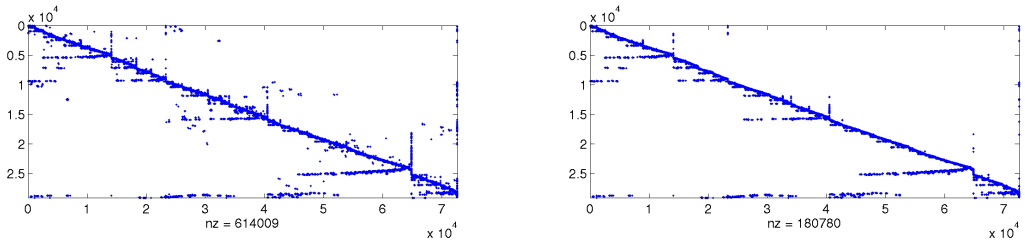

Figure 6: Approximate inverse as computed by inverse-based pivoting. The lower left part of L^{-1} is shown for the thresholds $\tau = 10^{-2}$ (left picture, 614,009 nonzeros) and $\tau = 10^{-1}$ (right picture, 180,780 nonzeros).

The two-level decomposition (7) allows us to rewrite A^{-1} as

$$(\Pi^T A \Pi)^{-1} = \begin{pmatrix} (L_B D_B L_B^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -L_B^{-T} L_F^T \\ I \end{pmatrix} S_C^{-1} \begin{pmatrix} -L_F L_B^{-1} & I \end{pmatrix}. \tag{17}$$

If we interpret (17) from the point of view of multigrid methods, we find that it formally yields an interpolation operator

$$I_h = \begin{pmatrix} -L_B^{-T} L_F^T \\ I \end{pmatrix}.$$

Therefore, the constraint $\|L^{-1}\| \leq \kappa$ is essentially equivalent to $\|I_h\| \leq \kappa$, and hence the coarsening process dictated by inverse-based pivoting constructs a bounded interpolation operator I_h. Moreover, I_h is sparse because it essentially consists of the lower left block of L^{-1} – see Figure 6. Hence, the coarse grid system given by S_C in (8) satisfies

$$S_C = I_h^T A I_h \tag{18}$$

and can be well approximated by a sparse matrix

$$\tilde{S}_C = \tilde{I}_h^T A \tilde{I}_h. \tag{19}$$

3.4 Inverse-based pivoting selects the right coarse grid

When applied to standard elliptic problems it is well-known that multigrid methods decompose the solution space into a fine- and a coarse-grid space, associated with the high and low (smoother) frequency components of the problem, respectively. From a purely algebraic point of view, multigrid methods distinguish between large and small eigenvalues of the underlying (symmetric positive definite) linear system. As long as the matrix is diagonalizable, the associated eigenvectors thereby define two complementary spaces, the fine- and the coarse-grid space.

For general matrices, the singular value decomposition

$$A = [U_1 U_2] \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} [V_1 V_2]^T$$

is more appropriate. Here we assume that the diagonal entries of Σ_1 are of the order of $\|A\|$ in magnitude, whereas those of Σ_2 contain all remaining smaller ones. As a consequence,

$$A^{-1} = \underbrace{V_1 \Sigma_1^{-1} U_1^T}_{\text{fine space}} + \underbrace{V_2 \Sigma_2^{-1} U_2^T}_{\text{coarse space}} \equiv A_F + A_C \tag{20}$$

where the first term A_F associated with the fine space is bounded by $1/\|A\|$. Thus, it is the second part A_C which reveals $\|A^{-1}\|$. In fact, the inverse–based decomposition constructs the multilevel factorization that can be split similar to (20), see also Theorem 2 below. I.e.,

$$(\Pi^T A \Pi)^{-1} = \underbrace{\begin{pmatrix} (L_B D_B L_B^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix}}_{A_F} + \underbrace{I_h S_C^{-1}\ I_h^T}_{A_C},$$

where $\|A_F\|$ is moderately bounded and $\|A_C\| \approx \mathcal{O}(\|A^{-1}\|)$.

Since the original matrix has been scaled with the graph–pivoting strategy beforehand and $\|L^{-1}\| \lessapprox \kappa$, we expect from (17) that

$$\beta = \|D_B^{-1}\| \tag{21}$$

is small. For $\varepsilon > 0$ we now let

$$\Pi^T A \Pi v = \varepsilon w, \quad \|v\| = \|w\| = 1, \qquad v, w \in \mathbb{C}^n,$$

for any pair of vectors v, w, such as eigenvectors or singular vectors, that satisfy $\|\Pi^T A \Pi v\| \ll \|v\|$; singular vectors in fact lead to the smallest possible ε. On the one hand we have $\|(\Pi^T A \Pi)^{-1} w\| = 1/\varepsilon$, and on the other hand by (17) we have

$$I_h S_C^{-1}\ I_h^T w = \frac{1}{\varepsilon} v - \begin{pmatrix} (L_B D_B L_B^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix} w. \tag{22}$$

Neglecting the smaller right-most term in (22) of order $\kappa^2 \beta$, we find that

$$I_h S_C^{-1}\ I_h^T w \approx \frac{1}{\varepsilon} v.$$

Since $\|I_h\| \leq \kappa$, we conclude $S_C^{-1}\ I_h^T w$ must be at least of order ε^{-1}. Thus, the graph-pivoting strategy combined with the bound (14) on the inverse triangular factors in (7) tends to relegate the smallest eigenvalues (or singular values) to the coarser space. In the following theorem we rigorously establish this fact when A is real symmetric, for simplicity, as in a situation without damping or absorbing boundary conditions.

Theorem 2 *Let $A \in \mathbb{R}^{n,n}$ be a nonsingular real symmetric matrix. Consider the two-level factorization (7) and suppose that $\|D_B^{-1}\|_2 \leq \beta$ and $\|L^{-1}\|_2 \leq \kappa$.*

1. *Let $\frac{1}{\lambda_p} \leqslant \cdots \leqslant \frac{1}{\lambda_1}$ be the p largest positive eigenvalues of A^{-1} and let $\frac{1}{\mu_p} \leqslant \cdots \leqslant \frac{1}{\mu_1}$ be the p largest positive eigenvalues of S_C^{-1}. Then*

$$\frac{1}{\mu_j} \leqslant \frac{1}{\lambda_j}, \quad \forall j = 1, \ldots, p. \tag{23}$$

If in addition $\kappa^2 \beta < \frac{1}{\lambda_p}$, then

$$\frac{1}{\kappa^2} \frac{1}{\lambda_j} - \beta \leqslant \frac{1}{\mu_j}, \quad \forall j = 1, \ldots, p. \tag{24}$$

2. *Let $\frac{1}{\lambda_1} \leqslant \cdots \leqslant \frac{1}{\lambda_p}$ be the p smallest negative eigenvalues of A^{-1} and let $\frac{1}{\mu_1} \geqslant \cdots \geqslant \frac{1}{\mu_p}$ be the p smallest negative eigenvalues of S_C^{-1}. Then*

$$\frac{1}{\lambda_j} \leqslant \frac{1}{\mu_j}, \quad \forall j = 1, \ldots, p. \tag{25}$$

If in addition $\frac{1}{\lambda_p} < -\kappa^2 \beta$, then

$$\frac{1}{\mu_j} \leqslant \frac{1}{\kappa^2} \frac{1}{\lambda_j} + \beta, \quad \forall j = 1, \ldots, p. \tag{26}$$

Proof. Let $V \in \mathbb{R}^{n,j}$ be a matrix of j eigenvectors, $j \leq p$, such that

$$(\Pi^T A \Pi)\, V = V \mathrm{diag}\,(\lambda_1, \ldots, \lambda_j) \equiv V\Lambda, \qquad V^T V = I.$$

Next we define $V_C := I_h^T V$ and let $V_C = Q_C R_C$ denote its QR factorization, where $Q_C^T Q_C = I$ and R_C is $j \times j$ upper triangular. Therefore $\|R_C\|_2 \leqslant \kappa$, because $\|V_C\|_2 \leqslant \kappa$. By the Courant–Fischer max-min principle, μ_j and λ_j can be characterized as

$$\frac{1}{\mu_j} = \max_{\dim\, \mathcal{W}=j} \min_{y \in \mathcal{W}\setminus\{0\}} \frac{y^T S_C^{-1} y}{y^T y}, \quad \frac{1}{\lambda_j} = \max_{\dim\, \mathcal{V}=j} \min_{x \in \mathcal{V}\setminus\{0\}} \frac{x^T (\Pi^T A \Pi)^{-1} x}{x^T x}.$$

Now, we first prove the lower bound (24). For any $y = V_C x \neq 0$, we have

$$\begin{aligned}
\frac{y^T S_C^{-1} y}{y^T y} &= \frac{x^T R_C^T Q_C^T S_C^{-1} Q_C R_C x}{x^T R_C^T R_C x} \\
&\geqslant \frac{1}{\kappa^2} \frac{x^T R_C^T Q_C^T S_C^{-1} Q_C R_C x}{x^T x} = \frac{1}{\kappa^2} \frac{x^T V^T I_h S_C^{-1} I_h^T V x}{x^T x} \\
&= \frac{1}{\kappa^2} \frac{x^T V^T \left(\Lambda^{-1} - \begin{bmatrix} (L_B D_B L_B^T)^{-1} & 0 \\ 0 & 0 \end{bmatrix} \right) V x}{x^T x} \\
&\geqslant \frac{1}{\kappa^2} \left(\min_{x \neq 0} \frac{x^T V^T \Lambda^{-1} V x}{x^T x} - \max_{x \neq 0} \frac{x^T V^T \begin{bmatrix} (L_B D_B L_B^T)^{-1} & 0 \\ 0 & 0 \end{bmatrix} V x}{x^T x} \right) \\
&\geqslant \frac{1}{\kappa^2} \left(\frac{1}{\lambda_j} - \kappa^2 \beta \right).
\end{aligned}$$

Hence the max-min property implies (24).

Suppose that $S_C \in \mathbb{R}^{m,m}$. To prove the upper bound (23), we note that by the Courant-Fischer max-min principle we can always find an $m \times j$ matrix W such that

$$\frac{1}{\mu_j} = \min_{x \neq 0} \frac{x^T W^T S_C^{-1} W x}{x^T x}, \qquad W^T W = I.$$

Since S_C^{-1} is a submatrix of $(\Pi^T A \Pi)^{-1}$, there exists a projection matrix Q such that $S_C^{-1} = Q^T (\Pi^T A \Pi)^{-1} Q$. Thus,

$$\frac{x^T W^T S_C^{-1} W x}{x^T x} = \frac{x^T W^T Q^T (\Pi^T A \Pi)^{-1} Q W x}{x^T x},$$

and taking first the minimum and then the maximum on both sides yields (23). We remark that for the proof of (23) the assumption $\lambda_p < 1/\kappa^2\beta$ is not required.

The proof of (25) and (26) is analogous and therefore omitted here. □

According to Theorem 2, the smallest eigenvalues in magnitude not only move away from the origin but also get amplified at most by a factor κ^2 when transferred to the coarse grid, as illustrated in Figure 7. Since we assume $|\lambda_p| \ll 1/\kappa^2\beta$ in the proof of Theorem 2, κ ought not to be too large for the the upper bound (24) to be useful.

Even when A in (3) is real, it is highly indefinite although most of its eigenvalues are positive. Theorem 2 implies that those eigenvalues closest to the origin are revealed by the coarse-grid system. Since the size of the linear system is greatly reduced by the inverse-based coarsening process, we conclude that the majority of the remaining positive larger eigenvalues will not be captured by the coarse system. As a consequence, the linear systems necessarily become increasingly indefinite from one level to the next, until most of the large positive eigenvalues have disappeared.

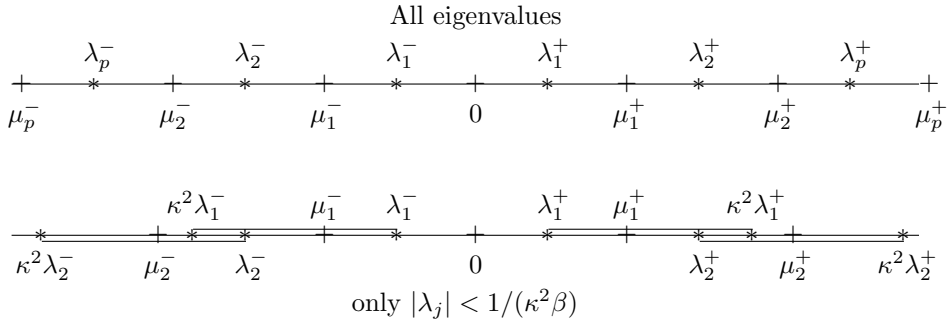

Figure 7: The eigenvalues λ_j and μ_j of A and S_C, respectively.

To illustrate this effect, we consider again Example 12 and display in Figure 8 the 200 hundred eigenvalues of A and S_C closest to the origin – the eigenvalues were computed using MATLAB's **eigs** function. In fact, Theorem 2 strictly does not apply, because A is complex symmetric here. However, the Sommerfeld radiation condition induces only a low rank complex perturbation, as corroborated by the imaginary parts of the eigenvalues in Figure 8, much smaller than their respective real parts. Indeed as predicted by Theorem 2, we observe that the smallest eigenvalues in modulus of S_C remain between $|\lambda_j|$ and $\kappa^2|\lambda_j|$.

3.5 Approximate inverse–based factorization

In the previous sections we have discussed inverse–based pivoting when used as a direct method and have concentrated on the selection of coarse-grid nodes during the elimination process. To turn this approach into a true algebraic multilevel preconditioner, however, we further need to drop small entries, while preserving key features for numerical stability. Thus we now examine the influence of inverse–based coarsening when the LDL^T factorization is only computed *approximately*. Instead of (7), we then obtain

$$\begin{aligned} \Pi^T A \Pi = \begin{pmatrix} B & F^T \\ F & C \end{pmatrix} &= \begin{pmatrix} L_B & 0 \\ L_F & I \end{pmatrix} \begin{pmatrix} D_B & 0 \\ 0 & S_C \end{pmatrix} \begin{pmatrix} L_B^T & L_F^T \\ 0 & I \end{pmatrix} \\ &\quad + \underbrace{\begin{pmatrix} E_B \\ E_F \end{pmatrix} \begin{pmatrix} D_B & 0 \end{pmatrix} + \begin{pmatrix} D_B \\ 0 \end{pmatrix} \begin{pmatrix} E_B^T & E_F^T \end{pmatrix}}_{=:E} \end{aligned} \tag{27}$$

as in [10], where E_B, E_F refer to those elements dropped from L and S_C as in (8). We denote by τ the drop tolerance used for the removal of the entries $|l_{ij}| \leqslant \tau$ from L.

Since the inverse-based pivoting startegy inherently excludes small pivots, D_B is typically well-conditioned and we can assume β in $\|D_B^{-1}\| \leq \beta$ to be quite small. Moreover, the bound $\|L^{-1}\| \lessapprox \kappa$, which we enforce during the incomplete LDL^T decomposition, guarantees the numerical stability of the inverse triangular factors. When A is preconditioned by $(LDL^T)^{-1}$, say using right preconditioning, the remaining critical part therefore is the inversion of S_C in

$$\Pi^T A \Pi L^{-T} \begin{pmatrix} D_B & 0 \\ 0 & S_C \end{pmatrix}^{-1} L^{-1} = I + E L^{-T} \begin{pmatrix} D_B & 0 \\ 0 & S_C \end{pmatrix}^{-1} L^{-1}.$$

As Theorem 2 implies that

$$\frac{1}{\kappa^2}\|A^{-1}\| - \beta \leqslant \|S_C^{-1}\| \leqslant \|A^{-1}\|,$$

the error of the preconditioned linear system is at worst on the order of

$$\tau\kappa^2\|A^{-1}\|. \tag{28}$$

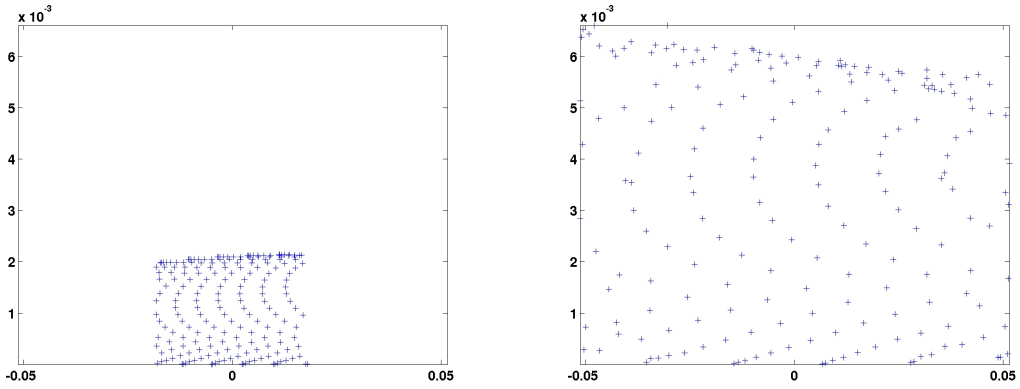

Figure 8: Two-dimensional wave guide problem (12). The 200 eigenvalues closest to the origin are shown for $\kappa = 3$. Left: A; right: the Schur complement S_C.

From (28) we infer yet again that a small κ in (14) is essential to keep the condition number of the preconditioned linear system reasonably small.

In addition to those entries of order τ dropped from L, we further drop entries in the Schur complement S_C thereby replacing it by the sparse approximation $\tilde{S}_C$ in (19). For the removal of small entries from S_C, we typically use a smaller drop tolerance, such as $\tau/10$, which adds the perturbation

$$\begin{pmatrix} 0 & 0 \\ 0 & E_S \end{pmatrix}$$

to the right side of (27).

4 Numerical experiments

To evaluate the performance of our algebraic multilevel (ML) preconditioner, we shall apply it to a series of challenging test problems in two and three space dimensions, in particular from geophysical seismic imaging. In all cases, the matrix A in (3) results from a second-order centered finite-difference discretization of (1) on a regular grid.

The preconditioned linear system is solved with the SQMR iterative method [25, 26], which requires one matrix-vector multiply per iteration and little extra storage. The initial guess, x_0, is always zero, and we stop the iteration when the relative residual satisfies

$$\frac{\|r_m\|}{\|b\|} \leqslant 10^{-7}, \qquad r_m = b - Ax_m.$$

We compare our ML preconditioner[1] with the sparse direct solver PARDISO[1]. To do so, we monitor the number of SQMR iterations, the fill-in of the ML perconditioner relative to that of A, and both the set-up and the execution times. All numerical experiments were performed on a Intel Xeon server (2.2 GHz) with 32 GB of memory.

The two parameters κ and τ, which control the norm of the inverse and the drop tolerance in the incomplete factorization process (see Section 3), determine the fill-in and hence the work and storage required for the ML preconditioner. Larger values of τ or κ lead to a cheaper but maybe ineffective preconditioner, whereas smaller values can lead to prohibitively high fill-in. We shall refrain from optimizing τ and κ and fix their values in all numerical experiments to

$$\kappa = 3, \qquad \tau = 3 \times 10^{-3}.$$

[1] available at `http://www.pardiso-project.org`.

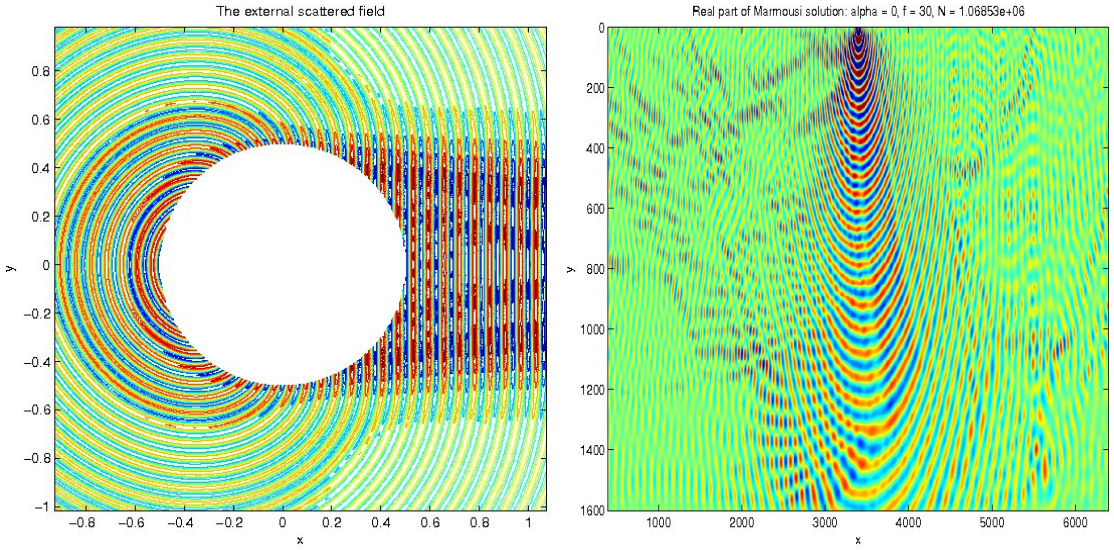

Figure 9: Two-dimensional problems. Left: scattered field from a cylinder with $k = 32 \times \pi$. Right: Marmousi problem for $\nu = 30$ Hz and $\alpha = 0$.

In general, the optimal parameter choice is problem dependent and could be adjusted for improved performance within a particular class of problems.

The ML preconditioner is computed either directly from A or from the matrix $A + i\beta k^2 I$ that results from the discretization of (4). Recall that a small complex diagonal shift typically improves on the spectrum of the preconditioned system [20, 21, 38]. Thus we shall also vary β to study its effect on the ML preconditioner; for $\beta = 0$, the ML preconditioner is computed from the original matrix A.

4.1 Two-dimensional examples

We first consider an exterior scattering problem where a plane wave $\exp(ikx)$ impinges upon a "sound-soft" infinite cylinder aligned with the z-axis. As the scattered field, u, is then independent of z, the computational domain Ω reduces to the two-dimensional annulus $\Omega = \{(r, \vartheta) | \, r_0 < r < R, 0 \leq \vartheta < 2\pi\}$, where (r, ϑ) denote polar coordinates. At the surface of the cylinder, $r = r_0$, we set $u = -\exp(ikx)$. At the outer (artificial) boundary, $r = R$, we impose the Sommerfeld radiation condition

$$\frac{\partial u}{\partial n} + iku = 0, \tag{29}$$

where n denotes the outward normal. In Ω we discretize (1) with constant k on an equidistant polar grid; here, the asymmetry in the discrete Laplacian induced by the r dependent first-order term, $r\partial_r$, is removed by diagonal rescaling of A.

In Figure 9 the real part of the scattered field is shown for $r_0 = 1/2$ and $R = 2$ in the vicinity of the cylinder. Starting at $k = 8\pi$, we progressively increase k while refining the mesh to keep the number of grid points per wavelength fixed, according to (2). Hence, when the wave number k doubles, the mesh size h is halved whereas the problem size N quadruples. Here we compare the ML preconditioner also to the standard (nonsymmetric) ILU(0) and ILUT preconditioners [45] (Matlab) combined with the BICGSTAB [51] iterative method.

In Table 1 we compare the relative fill-in and set-up times for the ML preconditioner to those of the direct solver PARDISO and of the ILU(0) , ILUT preconditioners. For zero or small complex shift β, the fill-in tends to increase with k. However, with $\beta \geq 0.25$ the fill-in remains quite low

Table 1: Scattering from a cylinder. The fill-in and set-up time in seconds (in parentheses) for the ML preconditioner, with or without complex diagonal shift β, are compared to those of the direct solver PARDISO and of the ILU(0) , ILUT preconditioners. '‡' indicates that the time limit ($> 10,000$ sec.) has been reached.

k/ grid points	PARDISO	Matlab BICGSTAB ILU(0)	Matlab BICGSTAB ILUT (0.003)	SQMR ML $\beta = 0$	SQMR ML $\beta = 0.5$	SQMR ML $\beta = 0.25$	SQMR ML $\beta = 0.1$
$8 \times \pi$	9.1	1	6.5	5.6	3.8	4.1	4.5
120×960	(2)	(446)	(64)	(6)	(5)	(5)	(5)
$16 \times \pi$	11.1	1	‡	7.1	3.3	3.6	7.0
240×1920	(15)	(9128)	‡	(58)	(24)	(30)	(56)
$32 \times \pi$	13.0	‡	‡	9.1	3.1	3.7	10.1
480×3840	(93)	‡	‡	(286)	(99)	(123)	(335)

Table 2: The multilevel grid hierarchy and fill-in on each level for the ML preconditioner, with or without complex diagonal shift β. The fill-in given below is taken relative to the number of nonzeros of A.

k/ grid points	ML multilevel grid hierarchy $\beta = 0$				$\beta = 0.25$			
$8 \times \pi$	Matrix A:	114'240	nnz:	341'760	Matrix A:	114'240	nnz:	341'760
120×960	Level 0:	114'240	Fill-in:	0.8	Level 0:	114'240	Fill-in:	0.6
	Level 1:	51'951	Fill-in:	1.1	Level 1:	51'951	Fill-in:	1.0
	Level 2:	22'297	Fill-in:	0.7	Level 2:	22'244	Fill-in:	0.5
	Level 3:	11'441	Fill-in:	0.4	Level 3:	11'347	Fill-in:	0.4
	Level 4:	6'810	Fill-in:	0.2	Level 4:	6'655	Fill-in:	0.2
	Level 5:	4'654	Fill-in:	2.2	Level 5:	4'317	Fill-in:	1.4
			Total:	5.6			Total:	4.1
$16 \times \pi$	Matrix A:	458'880	nnz:	1'374'720	Matrix A:	458'880	nnz:	1'374'720
240×1920	Level 0:	458'880	Fill-in:	0.8	Level 0:	458'880	Fill-in:	0.8
	Level 1:	209'449	Fill-in:	1.1	Level 1:	209'449	Fill-in:	1.0
	Level 2:	89'972	Fill-in:	0.7	Level 2:	89'954	Fill-in:	0.5
	Level 3:	46'310	Fill-in:	0.4	Level 3:	46'076	Fill-in:	0.3
	Level 4:	28'065	Fill-in:	0.2	Level 4:	27'489	Fill-in:	0.2
	Level 5:	19'438	Fill-in:	0.1	Level 5:	18'500	Fill-in:	0.2
	Level 6:	15'185	Fill-in:	0.1	Level 6:	12'191	Fill-in:	0.3
	Level 7:	12'897	Fill-in:	0.1	Level 7:	6'040	Fill-in:	0.2
	Level 8:	11'273	Fill-in:	3.6	Level 8:	1'050	Fill-in:	0.1
			Total:	7.1			Total:	3.6
$32 \times \pi$	Matrix A:	1'839'360	nnz:	5'514'240	Matrix A:	1'839'360	nnz:	5'514'240
480×3840	Level 0:	1'839'360	Fill-in:	0.7	Level 0:	1'839'360	Fill-in:	0.8
	Level 1:	841'557	Fill-in:	1.0	Level 1:	841'557	Fill-in:	1.0
	Level 2:	361'971	Fill-in:	0.5	Level 2:	186'207	Fill-in:	0.5
	Level 3:	187'378	Fill-in:	0.3	Level 3:	112'543	Fill-in:	0.3
	Level 4:	114'282	Fill-in:	0.2	Level 4:	76'619	Fill-in:	0.4
	Level 5:	80'005	Fill-in:	0.2	Level 5:	51'145	Fill-in:	0.3
	Level 6:	63'492	Fill-in:	0.1	Level 6:	25'734	Fill-in:	0.2
	Level 7:	53'713	Fill-in:	0.2	Level 7:	4'974	Fill-in:	0.2
	Level 8:	47'185	Fill-in:	0.1			Total:	3.7
	Level 9:	41'934	Fill-in:	5.7				
			Total:	9.1				

independently of k, while the set-up time increases only linearly with problem size. For $\beta = 0.5$ the set-up time at high wave numbers is comparable to that of PARDISO, yet the fill-in is four times smaller. Both ILU(0) and ILUT become prohibitively slow at higher wave numbers.

The multilevel grid hierarchy and fill-in on each level for the ML preconditioner with or without complex diagonal shift β is shown in Table 2. Table 3 displays the iteration counts and CPU times

Table 3: Scattering from a cylinder. The total number of iterations and CPU time in seconds (in parentheses) for the ML preconditioner, with or without complex diagonal shift β, are compared to those of the direct solver PARDISO and of the ILU(0) , ILUT preconditioners. '†' indicates divergence ($> 5,000$ iterations) and '‡' that the time limit ($> 10,000$ sec.) has been reached.

k/ grid points	PARDISO	Matlab BICGSTAB ILU(0)	Matlab BICGSTAB ILUT (0.003)	SQMR ML $\beta = 0$	SQMR ML $\beta = 0.5$	SQMR ML $\beta = 0.25$	SQMR ML $\beta = 0.1$
$8 \times \pi$	1	1084	†	25	89	46	31
120×960	(0.2)	(467)	†	(5)	(15)	(8)	(6)
$16 \times \pi$	1	2338	‡	54	197	96	63
240×1920	(0.7)	(7161)	‡	(48)	(139)	(71)	(57)
$32 \times \pi$	1	‡	‡	182	334	227	137
480×3840	(3.1)	‡	‡	(663)	(602)	(430)	(548)

achieved by the different preconditioners, either with or without complex diagonal shift β. As expected, the direct solver PARDISO outperforms all iterative solvers. The ML preconditioner leads to convergence with or without complex shift, while the number of iterations increases linearly with k. A slight complex shift $\beta = 0.25$ has a positive impact both on the fill-in and the number of iterations and thus yields the overall best performance of the ML preconditioner.

Next, we consider a geophysical migration problem from seismic imaging which requires the solution of

$$-\Delta u - (1 - i\alpha)k^2 u = f \qquad \text{in} \quad \Omega. \tag{30}$$

Here the parameter α, $0 \leq \alpha \ll 1$, represents a small fraction of physical damping in the medium; for $\alpha = 0$, (30) reduces to (1). The computational domain Ω corresponds to a 6000m $\times$ 1600m vertical slice of the Earth's subsurface, which we truncate by imposing the Sommerfeld radiation condition (29) at its boundary. The point source f ("single shot") is located in the center of the upper boundary, which corresponds to the surface of the Earth. The highly heterogeneous velocity profile $c(x, y)$ stems from the Marmousi model [42] and varies irregularly between 1500 m/s and 4000 m/s throughout Ω.

We now progressively increase the frequency $\nu = \omega/2\pi$ from 10 Hz to 30 Hz, while refining the grid according to (2). In Figure 9 the real part of the wavefield is shown for $\nu = 30$ Hz on the finest 2001×534 grid, which yields at least 17 grid points per wave length.

In Table 4 we observe that the fill-in in the ML preconditioner slowly increases with ν without complex shift. For a slightly positive value of β, however, the fill-in remains quite low, independently of k and problem size. The inclusion of physical damping α has but little effect on the fill-in, whereas the set-up times tend to decrease.

The total number of iterations and CPU times are shown in Table 5. The ML preconditioner always achieves convergence regardless of k, α, or problem size. Again a small complex shift is beneficial to the preconditioned system, as the number of iterations then increases essentially linearly with frequency ν. Yet any further increase in β typically leads to higher iteration counts, as the preconditioned system moves even farther away from the original problem. Again, the inclusion of physical damping greatly reduces the number of iterations, which corroborates previous findings [33].

4.2 Three-dimensional examples

We first consider a 3D wedge model, where the velocity profile mimics a simple geophysical situation with three distinct layers. The computational domain, $\Omega = (0, 1)^3$, is divided into three regions by the two tilted planes f_1 and f_2, as shown in Figure 10. We denote by k_{ref} the (constant) wave number inside the intermediate layer; in the upper and lower layers the wave number equals $k = 1.2 \times k_{\text{ref}}$ and $k = 1.5 \times k_{\text{ref}}$, respectively. Again, we solve (30) with the Sommerfeld radiation

Table 4: Fill-in and set-up time in seconds (in parentheses) of the multilevel preconditioner and the direct solver PARDISO for the 2D Marmousi problem.

ν	grid points	damping α	PARDISO	ML $\beta = 0$	ML $\beta = 0.5$	ML $\beta = 0.25$	ML $\beta = 0.1$
10	751×201	0.0	10.0 (2)	3.9 (4)	2.9 (2)	3.8 (3)	4.3 (4)
		0.025	10.0 (2)	3.7 (2)	2.8 (2)	3.6 (4)	4.1 (4)
		0.050	10.0 (2)	3.7 (5)	2.8 (2)	3.9 (3)	4.1 (4)
20	1501×401	0.0	11.1 (15)	6.5 (70)	2.9 (9)	3.2 (10)	4.4 (38)
		0.025	11.1 (14)	5.3 (46)	2.8 (8)	3.2 (10)	3.8 (26)
		0.050	11.1 (15)	4.0 (23)	2.8 (8)	3.1 (15)	3.6 (14)
30	2001×534	0.0	12.1 (32)	8.3 (523)	3.1 (28)	3.6 (25)	5.0 (71)
		0.025	12.1 (32)	7.3 (251)	3.0 (26)	3.5 (23)	4.6 (47)
		0.050	12.1 (32)	6.2 (106)	3.0 (24)	3.4 (54)	4.3 (38)

Table 5: Total number of SQMR iterations and CPU time in seconds (in parentheses) for the 2D Marmousi problem. The results are shown for the algebraic multilevel preconditioner ML and for the direct solver PARDISO.

ν	grid points	damping α	PARDISO	ML $\beta = 0$	ML $\beta = 0.5$	ML $\beta = 0.25$	ML $\beta = 0.1$
10	751×201	0.0	1 (0.1)	98 (20)	95 (18)	69 (13)	59 (12)
		0.025	1 (0.1)	51 (10)	63 (12)	46 (9)	39 (8)
		0.050	1 (0.1)	37 (7)	43 (8)	32 (6)	29 (5)
20	1501×401	0.0	1 (0.7)	595 (867)	248 (190)	212 (169)	210 (160)
		0.025	1 (0.7)	189 (171)	99 (76)	89 (70)	85 (66)
		0.050	1 (0.7)	85 (72)	62 (47)	52 (40)	53 (44)
30	2001×534	0.0	1 (1.1)	853 (2343)	342 (483)	331 (482)	385 (656)
		0.025	1 (1.1)	287 (468)	103 (145)	88 (130)	89 (159)
		0.050	1 (1.1)	65 (160)	62 (87)	49 (71)	47 (75)

condition (29) imposed at the boundary of Ω; the point source, f, is located at $(0.5, 0.5, 0)$. Now, we progressively increase k from 20 up to 60, while refining the mesh according to (2), that is with about ten grid points per wavelength.

In Figure 10, the relative amounts of fill-in needed by the sparse direct solver PARDISO and the ML preconditioner are shown. As expected, the fill-in required by PARDISO is now significantly higher and exceeds 200 for $k_{\text{ref}} = 60$; this confirms the well-known fact that sparse direct solvers are usually no longer competitive for three-dimensional problems. The fill-in required by the ML preconditioner is smaller, but for $\beta = 0$ still increases with k. In contrast, with β as little as $\beta = 0.1$, the amout of fill-in is even smaller and remains essentially constant throughout the entire range of k_{ref}.

Table 6 displays the iteration counts and CPU times for varying k_{ref}. Despite the size of this three-dimensional problem, the ML preconditioner performs well as the number of iterations only increases linearly with k_{ref}. Nonetheless the larger fill-in observed in Figure 10 for $\beta = 0$ at higher wave numbers results in a significant increase in CPU time with k_{ref} and problem size. For a complex shift as small as $\beta = 0.1$, however, the fill-in is moderate independently of k_{ref} and, as a consequence, the execution time reduced by one order of magnitude.

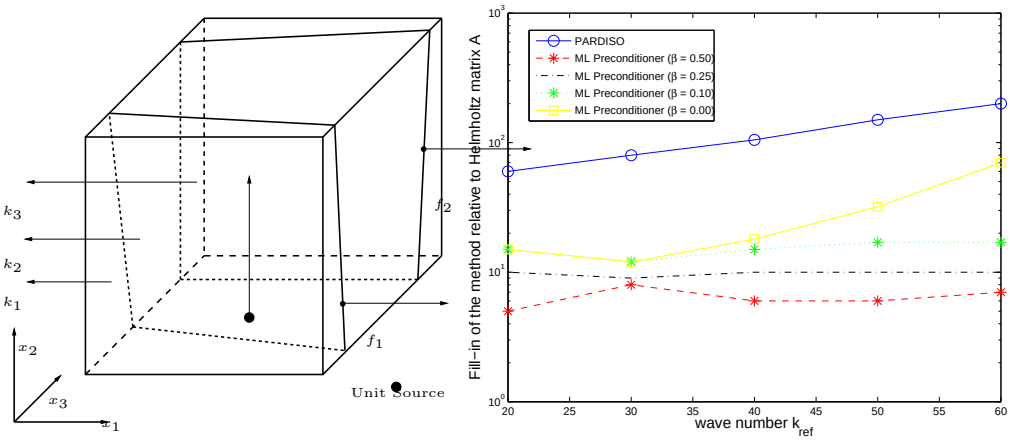

Figure 10: Left: Wedge problem with $f_1 = 0.5x_1 + 2.5x_2 + 0.375x_3 - 1 = 0$, $f_2 = -\frac{1}{6}x_1 + \frac{5}{3}x_2 - \frac{1}{3}x_3 - 1 = 0$, Right: Relative fill-in in the algebraic multilevel preconditioner ML, with or without complex diagonal shift β, and the direct solver PARDISO for the 3D wedge problem.

Table 6: Total number of SQMR iterations and CPU time in seconds (in parentheses) for the 3D wedge problem, with or without complex diagonal shift β.

k_{ref}	grid points	damping α	PARDISO	ML $\beta = 0$	ML $\beta = 0.5$	ML $\beta = 0.25$	ML $\beta = 0.1$
20	32^3	0.0	1 (4)	35 (2)	29 (2)	19 (1)	12 (1)
		0.025	1 (4)	35 (2)	30 (2)	19 (1)	12 (1)
		0.050	1 (4)	36 (3)	30 (2)	20 (1)	12 (1)
30	48^3	0.0	1 (39)	51 (15)	42 (14)	26 (10)	16 (6)
		0.025	1 (39)	63 (17)	42 (15)	26 (10)	16 (6)
		0.050	1 (39)	74 (19)	43 (13)	27 (9)	16 (6)
40	64^3	0.0	1 (213)	87 (86)	54 (41)	33 (30)	18 (22)
		0.025	1 (213)	99 (90)	55 (41)	33 (29)	19 (23)
		0.050	1 (213)	120 (100)	56 (39)	34 (30)	19 (22)
50	80^3	0.0	1 (1'012)	134 (432)	67 (108)	40 (78)	21 (62)
		0.025	1 (1'012)	150 (425)	67 (108)	41 (84)	21 (59)
		0.050	1 (1'012)	192 (458)	68 (99)	41 (78)	22 (59)
60	95^3	0.0	1 (3'467)	175 (1'505)	79 (204)	46 (158)	24 (130)
		0.025	1 (3'467)	228 (1'833)	80 (202)	47 (154)	24 (134)
		0.050	1 (3'467)	220 (1'712)	81 (210)	48 (161)	25 (137)

Finally, we consider a large-scale seismic imaging problem from subsurface geology. The computational domain Ω is 20 km wide, 20 km long, and 4 km deep. The highly heterogeneous velocity profile $c(x, y, z)$, shown in Figure 11, stems from the SEG/EAGE overthrust model [3] and varies irregularly between 2179 m/s and 6000 m/s throughout Ω. Again, we solve (30) with the Sommerfeld radiation condition (29) imposed at the boundary of Ω. The pressure source f is located at the center of the upper boundary. In Figure 11 the real part of the pressure response is shown for $\nu = 10$ Hz on the finest $401 \times 401 \times 94$ grid.

We now progressively increase the frequency ν, while refining the mesh size according to (2). The iteration counts and CPU times are shown for varying ν, β, and α in Table 7. Without complex diagonal shift, the ML preconditioner fails for this choice of parameters κ and τ. A complex diagonal shift, as little as $\beta = 0.1$, yet again leads to moderate fill-in and rapid convergence in spite of the difficulty and sheer size of this problem with over 15,000,000 complex unknowns. Although the number of iterations increases linearly with the frequency ν, it remains indeed remarkably small and thus leads to an efficient and robust black-box solver for three-dimensional

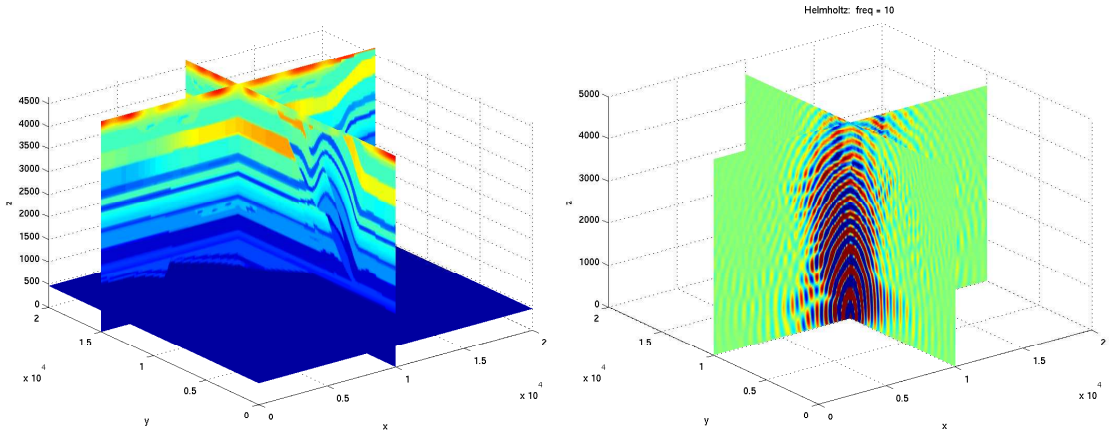

Figure 11: SEG/EAGA overthrust model. Left: the velocity profile $c(x, y, z)$ in m/s; right: real part of the pressure response for $\nu = 10$ Hz and $\alpha = 0$.

seismic imaging.

Table 7: Total number of SQMR iterations and CPU time in seconds (in parentheses) for the SEG/EAGA overthrust model. '†' denotes divergence of the problem ($> 2,000$ iterations) or that there was no residual reduction within 200 iterations.

ν	grid points	damping α	ML $\beta = 0$	ML $\beta = 0.5$	ML $\beta = 0.25$	ML $\beta = 0.1$
2.5	$101 \times 101 \times 24$	0.0	20 (31)	53 (60)	31 (33)	17 (25)
		0.025	17 (20)	40 (58)	24 (22)	14 (22)
		0.050	17 (17)	32 (21)	20 (20)	12 (15)
5	$201 \times 201 \times 48$	0.0	†	137 (1200)	75 (798)	35 (410)
		0.025	209 (4'151)	66 (501)	40 (402)	21 (321)
		0.050	134 (1'522)	45 (322)	28 (301)	17 (268)
10	$401 \times 401 \times 94$	0.0	†	202 (8'403)	102 (5'452)	78 (4'243)
		0.025	†	92 (4'062)	68 (3'534)	54 (3'188)
		0.050	290 (28'090)	59 (2'851)	37 (1'958)	30 (1'602)

5 Concluding remarks

We have presented an algebraic multilevel preconditioner for the heterogeneous Helmholtz equation (1), where the wave number k may vary arbitrarily. Our ML preconditioner is based on an incomplete LDL^T factorization and combines weighted matching reordering with an inverse–based multilevel incomplete factorization. As it preserves the (complex) symmetry inherent to the original problem, storage requirements are kept minimal.

We have applied the ML preconditioner to a series of challenging two and three-dimensional test problems, mainly from geophysical seismic imaging, where standard ILU preconditioners fail. The preconditioned system is solved with the SQMR Krylov subspace iterative method. During all numerical experiments we refrain from optimizing the two key parameters, τ and κ, which control the fill-in and work of the ML preconditioner, but instead use the ML preconditioner as a "black-box" solver. Our numerical results show that the number of iterations grows linearly with k, if the mesh size h is refined simultaneously to ensure a constant number of grid points per wave length. In addition, when a small complex shift is first applied to the linear system, the fill-in becomes essentially independent of k.

Since the preconditioner is fully algebraic, it immediately applies to high-order finite element discretizations, unstructured grids, or more sophisticated absorbing boundary conditions or perfectly matched layers. Parallelization of the ML preconditioner can be achieved by using parallel incomplete factorizations [2] and parallel bipartite matchings [44]. Alternatively, one may consider domain decomposition techniques, where the ML solver can be used either as a local solver in every subdomain or for the coarse grid solution.

Acknowledgements

We thank Yogi Erlangga for useful comments and suggestions, and both René-Edouard Plessix and Wim A. Mulder at Shell International Exploration and Production, Rijswijk, The Netherlands, for providing the data from geophysical applications.

References

[1] T. Airaksinen, E. Heikkola, A. Pennanen, and J. Toivanen, *An algebraic multigrid based shifted-laplacian preconditioner for the Helmholtz equation*, J. Comp. Phys., 226 (2007), pp. 1196–1210.

[2] J. Aliaga, M. Bollhöfer, A. Martín, and E. Quintana-Ortí, *Parallelization of multilevel preconditioners constructed from inverse-based ILUs on shared-memory multiprocessors*, in Parallel Computing: Architectures, Algorithms and Applications, vol. 38, C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr, F. Peters, John von Neumann Institute for Computing, Jülich, NIC Series, 2008, pp. 287–294.

[3] F. Aminzadeh, J. Brac, and T. Kunz, *3-D Salt and Overthrust Models*, SEG/EAGE 3-D Modeling Series, 1 (1997).

[4] I. M. Babuška and S. A. Sauter, *Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers?*, SIAM Review, 42 (2000), pp. 451–484.

[5] R. E. Bank, *A comparison of two multilevel iterative methods for nonsymmetric and indefinite elliptic finite element equations*, SIAM J. Numerical Analysis, 18 (1981), pp. 724–743.

[6] A. Bayliss, C. I. Goldstein, and E. Turkel, *An iterative method for the Helmholtz equation*, J. Comp. Phys., 49 (1983), pp. 443–457.

[7] ——, *On accuracy conditions for the numerical computation of waves*, J. Comp. Phys., 59 (1985), pp. 396–404.

[8] J.-D. Benamou and B. Després, *A domain decomposition method for the Helmholtz equation and related optimal control problems*, J. Comp. Phys., 136 (1997), pp. 62–82.

[9] M. Benzi, J. Haws, and M. Tůma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Scientific Computing, 22 (2000), pp. 1333–1353.

[10] M. Bollhöfer and Y. Saad, *Multilevel preconditioners constructed from inverse–based ILUs*, SIAM J. Scientific Computing, (2006), pp. 1627–1650.

[11] A. Brandt and I. Livshits, *Wave-ray multigrid method for standing wave equations*, Electr. Trans. Num. Anal., 6 (1997), pp. 162–181.

[12] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Mathematics of Computation, 31 (1977), pp. 163–179.

[13] X.-C. Cai, M. A. Casarin, J. F. W. Elliott, and O. B. Widlund, *Overlapping schwarz algorithms for solving Helmholtz's equation*, Contemporary Mathematics, 218 (1998), pp. 391–399.

[14] X.-C. Cai and O. B. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Scientific Computing, 13 (1992), pp. 243–258.

[15] A. Cline, C. B. Moler, G. Stewart, and J. Wilkinson, *An estimate for the condition number of a matrix*, SIAM J. Numerical Analysis, 16 (1979), pp. 368–375.

[16] F. Collino, S. Ghanemi, and P. Joly, *Domain decomposition method for harmonic wave propagation: a general presentation*, Comput. Meth. Appl. Mech. Engnrg., 184 (2000), pp. 171–211.

[17] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Analysis and Applications, 20 (1999), pp. 889–901.

[18] I. S. Duff and S. Pralet, *Strategies for scaling and pivoting for sparse symmetric indefinite problems*, SIAM J. Matrix Analysis and Applications, 27 (2005), pp. 313–340.

[19] H. C. Elman, O. Ernst, and D. P. O'Leary, *A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equations*, SIAM J. Scientific Computing, 23 (2001), pp. 1291–1315.

[20] Y. Erlangga, C. Oosterlee, and C. Vuik, *A novel multigrid based preconditioner for heterogeneous Helmholtz problems*, SIAM J. Sci. Comput., 27 (2006), pp. 1471–1492.

[21] Y. A. Erlangga, C. Vuik, and C. W. Oosterlee, *On a class of preconditioners for solving the helmholtz equation*, Appl. Numer. Math., 50 (2004), pp. 409–425.

[22] C. Farhat and J. Li, *An iterative domain decomposition method for the solution of a class of indefinite problems in computational structural dynamics*, Appl. Num. Math., 54 (2005), pp. 150–166.

[23] C. Farhat, A. Macedo, and M. Lesoinne, *A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems*, Numerische Mathematik, 85 (2000), pp. 283–308.

[24] D. Feingold and R. Varga, *Block diagonally dominant matrices and generalization of the gershgorin circle theorem*, Pacific J. Math., 12 (1962), pp. 1241–1250.

[25] R. Freund and F. Jarre, *A QMR–based interior–point algorithm for solving linear programs*, Mathematical Programming, Series B, 76 (1997), pp. 183–210.

[26] R. Freund and N. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math., 19 (1995), pp. 319–341.

[27] M. J. Gander, F. Magoulès, and F. Nataf, *Optimized Schwarz methods without overlap for the Helmholtz equation*, SIAM J. Scientific Computing, 24 (2002), pp. 38–60.

[28] A. George, *Nested dissection of a regular finite element mesh*, SIAM Journal on Numerical Analysis, 10 (1973), pp. 345–363.

[29] A. Gupta and L. Ying, *A fast maximum-weight-bipartite-matching algorithm for reducing pivoting in sparse Gaussian elimination*, Tech. Rep. RC 21576 (97320), IBM T. J. Watson Research Center, Yorktown Heights, NY, October 1999.

[30] M. Hagemann and O. Schenk, *Weighted matchings for the preconditioning of symmetric indefinite linear systems*, SIAM J. Scientific Computing, (2006), pp. 403–420.

[31] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Scientific Computing, 20 (1998), pp. 359–392.

[32] S. Kim, *Parallel multidomain iterative algorithms for the Helmholtz wave equation*, Appl. Num. Math., 17 (1995), pp. 411–429.

[33] S. Kim and S. Kim, *Multigrid simulations for high-frequency solutions of the Helmholtz problem in heterogeneous media*, SIAM J. Scientific Computing, 24 (2002), pp. 684–701.

[34] H. W. Kuhn, *The Hungarian method for solving the assignment problem*, Naval Res. Logist. Quart, (1955), pp. 83–97.

[35] A. L. Laird and M. B. Giles, *Preconditioned iterative solution of the 2D Helmholtz equation*, technical report 02/12, Oxford University Computing Laboratory, 2002.

[36] B. Lee, T. A. Manteuffel, S. F. McCormick, and J. Ruge, *First-order system least-squares for the Helmholtz equation*, SIAM J. Scientific Computing, 21 (2000), pp. 1927–1949.

[37] X. S. Li and J. W. Demmel, *SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems*, ACM Trans. Math. Softw., 29 (2003), pp. 110–140.

[38] M. M. M. Made, *Incomplete factorization based preconditionings for solving the Helmholtz equation*, Int J. Numerical Methods in Engineering, 50 (2001), pp. 1077–1101.

[39] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM Publications, 2000.

[40] J. Munkres, *Algorithms for the assignment and transportation problems*, Journal of SIAM, 5 (1957), pp. 32–38.

[41] M. Olschowka and A. Neumaier, *A new pivoting strategy for gaussian elimination*, Linear Algebra and its Applications, 240 (1996), pp. 131–151.

[42] C. Riyanti, Y. Erlangga, R.-E. Plessix, W. Mulder, C. Vuik, and C. Oosterlee, *A new iterative solver for the time-harmonic wave equation*, Geophysics, 71 (2006), pp. E57–E63.

[43] C. Riyanti, A. Kononov, Y. Erlangga, C. Vuik, C. Oosterlee, R.-E. Plessix, and M. W.A, *A parallel multigrid-based preconditioner for the 3d heterogeneous high-frequency Helmholtz equation*, J. Comp. Phys., 224 (2007), pp. 431–448.

[44] D. Ruiz, I.S.Duff, and B. Ucar, *Computung a class of bipartite matchings in parallel.* SIAM Conference on Parallel Processing for Scientific Computing, March 2008. talk.

[45] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM Publications, Philadelphia, second ed., 2003.

[46] A. H. Schatz, *An observation concerning Ritz-Galerkin methods with indefinite bilinear forms*, Mathematics of Computation, 28 (1974), pp. 959–962.

[47] O. Schenk, M. Bollhöfer, and R. A. Römer, *On large scale diagonalization techniques for the Anderson model of localization*, SIAM J. Scientific Computing, 28 (2006), pp. 963–983.

[48] ——, *On large scale diagonalization techniques for the Anderson model of localization*, SIAM Review, (2008), pp. 91–112.

[49] O. Schenk and K. Gärtner, *On fast factorization pivoting methods for symmetric indefinite systems*, Electr. Trans. Num. Anal., 23 (2006), pp. 158–179.

[50] O. Schenk, S. Röllin, and A. Gupta, *The effects of unsymmetric matrix permutations and scalings in semiconductor device and circuit simulation*, IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems, 23 (2004).

[51] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Scientific Computing, 13 (1992), pp. 631–644.

[52] M. van Gijzen, Y. Erlangga, and C. Vuik, *Spectral analysis of the discrete Helmholtz operator preconditioned with a shifted laplacian*, SIAM J. Scientific Computing, 29 (2007), pp. 1942–1958.