Preconditioning Helmholtz linear systems

Daniel Osei-Kuffuor[*] Yousef Saad[*]

April 15, 2009

Abstract

Linear systems which originate from the simulation of wave propagation phenomena can be very difficult to solve by iterative methods. These systems are typically complex valued and they tend to be highly indefinite, which renders the standard ILU-based preconditioners ineffective. This paper presents a study of ways to enhance standard preconditioners by altering the diagonal by imaginary shifts. Prior work indicates that modifying the diagonal entries during the incomplete factorization process, by adding to it purely imaginary values can improve the quality of the preconditioner in a substantial way. Here we propose simple algebraic heuristics to perform the shifting and test these techniques with the ARMS and ILUT preconditioners. Comparisons are made with applications stemming from the diffraction of an acoustic wave incident on a bounded obstacle (governed by the Helmholtz Wave Equation).

1 Introduction

The linear systems which originate from Maxwell and Helmholtz equations are known to be among the most difficult to solve by iterative techniques. These systems are typically complex valued. What makes them hard to solve is that they can be highly indefinite, and this leads to some difficulties when attempting to extract effective preconditioners. In this paper we examine just two preconditioners which are of an algebraic nature. The first is the standard Incomplete LU factorization with Threshold (ILUT) and the second is a version of the Algebraic Recursive Multilevel Solver (ARMS). In both cases, we explore a technique for modifying the diagonal in order to improve the quality of the preconditioner. The idea of diagonal perturbation was first used by Kershaw [11] to eliminate unstable pivots during an incomplete cholesky factorization. In the specific case of the Helmholtz equation, earlier work [1, 8, 9, 25, 16] has shown that a simple complex shift added to the Laplace operator can yield an improved preconditioner. These papers all consider the problem at the operator level, i.e., they motivate the approach by considering the Partial Differential Operator and show evidence that shifting the Laplacean yields an effective preconditioner. The question one may ask is *whether or not similar results can be achieved by adapting standard incomplete LU factorization preconditioners in an analogous fashion.* Based on stability arguments, it is clear that one should be able to improve stability

[*] Department of Computer Science and Engineering, University of Minnesota, 200 Union Street S.E, Minneapolis, MN 55455. Email: {dosei, saad}@cs.umn.edu. This work was supported by DOE under grants DE-FG02-03ER25585 and DE-FG-08ER25841 and by the Minnesota Supercomputer Institute.

of the ILU factorization by adding small terms to the diagonal. However, what is remarkable is that purely imaginary shifts have the effect of clustering eigenvalues close to a circle on the right half-plane, with an accumulation point near one. This can be explained easily by considering a simple model problem.

This paper considers adaptations of the idea of using complex shifts in the context of standard ILU factorizations. A number of strategies will be proposed for improving the stability of ILU preconditioners for complex indefinite matrices. These strategies will be tested with the ARMS and ILUT preconditioners and comparisons will be made using an example issued from the diffraction of an acoustic wave incident on a bounded obstacle (governed by the Helmholtz Wave Equation).

In section 2, we briefly discuss the Helmholtz equation application area considered in our examples. Section 3 introduces the solution methods and preconditioning techniques used in our experiments. In Section 4, we discuss briefly the motivation for the use of complex diagonal shifts for indefinite systems, and describe two strategies for selecting these shifts. We give some experimental results and discussion in section 5, and conclude in section 6.

2 Application Area

The Helmholtz equation is a partial differential equation of the form

$$(\Delta + k^2)\Phi = f, \tag{1}$$

which governs the propagation of waves in media. In the above equation, f represents a harmonic source, and k represents the wave number. The numerical solution to the Helmholtz equation at high wave numbers has been the subject of extensive research. At high wave numbers, the system matrix tends to be very indefinite, causing problems for many numerical methods.

Our application problem is based on the simulation of the diffraction of an acoustic wave originating from infinity through an open medium, incident on a bounded obstacle with boundary $\Gamma = \partial\Omega$, of circular shape. The corresponding boundary value problem (BVP) characterized by the Helmholtz equation is as follows:

$$\left\{ \begin{array}{rcl} \Delta u + k^2 u & = & f, \text{ in } \Omega \in \mathbb{R}^2 \\ u & = & \delta, \text{ on } \Gamma \\ \lim\limits_{r\to\infty} \sqrt{r}\left(\dfrac{\partial u}{\partial n} - iku\right) & = & 0 \end{array} \right. \tag{2}$$

where f represents a harmonic source, k is the wave number and δ is determined by the use of Bessel functions. The last equation in the above BVP is referred to as the Sommerfeld radiation condition and it models the non-reflecting condition at the boundary, thereby guaranteeing a unique solution to the above problem.

This BVP is however not suitable for solution via the finite element method, primarily because the condition that the incident wave originates from infinity prescribes the problem in an infinite domain. The problem is therefore reformulated to introduce an artificial boundary, by enforcing the so-called Dirichlet-to-Neumann (DtN) technique. Thus, the radiation condition at infinity is replaced by a boundary condition on the artificial boundary, Γ_{art}. The resulting problem becomes:

$$\left\{\begin{array}{rcl} \Delta u + k^2 u & = & f, \text{ in } \Omega \in \mathbb{R}^2 \\ u & = & \delta, \text{ on } \Gamma \\ \frac{\partial u}{\partial n} & = & -Bu \text{ on } \Gamma_{art} \end{array}\right. \tag{3}$$

where B denotes the DtN operator. Discretization of this resulting BVP was done via variational formulation, using the Galerkin Least-Squares discretization scheme. The discretized system is of the form

$$([K] - k^2[M] + \imath k[C])\{u\} = \{f\} = \{0\},$$

where [K] is the stiffness matrix, [M] is the mass matrix, and C is the boundary matrix responsible for the introduction of complex terms in the discretization. The coefficient matrix is complex, symmetric but not Hermitian, and generally not diagonally dominant. It is also very indefinite for the higher values of the wave number k. Details of the reformulation of the BVP and the finite element discretization of the result may be found in the paper by Kechroud at al. [10] and references therein.

3 Iterative solution methods and preconditioners

Linear systems which arise from wave propagation phenomena are known to be highly indefinite, in the sense that eigenvalues of the discretized operator are on both sides of the imaginary axis. In the past, iterative methods have not been too successful with such problems and for this reason, direct methods have often been used. As problems are now often formulated in 3-D geometries, the use of direct methods is becoming prohibitive. Two classes of methods have attracted the interest of researchers in recent years. First, is the class of multigrid methods which do not work well for indefinite problems and for which various adaptations have been brought, see, e.g., [7, 1, 9]. Second, is the class of preconditioned Krylov subspace methods. Here, work has focussed mainly on improving the preconditioning, see, e.g., [8, 25]. Earlier work about complex shifting was based on dealing directly with the operator of the Partial Differential Equation (PDE). In this paper we consider this technique and aim at retrofitting it to general algebraic preconditioners. There are several reasons why it is important to consider this viewpoint. First, there are complex indefinite systems in other areas of science and engineering for which the operator is not that of Helmholtz. The Maxwell equations are similar to Helmholtz in nature but lead to more complicated operators. Second, a purely algebraic strategy, may lead to an automatic selection of the shift.

We will consider here only two incomplete factorization techniques, though the ideas of complex shifts can be applied to virtually any type of ILU. These two preconditioners will be described briefly in the next two sections.

3.1 Incomplete LU Factorization with Threshold Dropping (ILUT)

ILUT is an ILU-based preconditioner which implements dropping strategies based on some threshold parameters [18, 19]. A common approach is the dual truncation technique - ILUT(τ,p)

- in which dropping during the factorization is based on two parameters: the drop tolerance (droptol $= \tau$), and the fill level (lfil $= p$).

The preconditioner is constructed based on the following rules:

- During the elimination for a particular row i, an entry $a_{i,k}$ is dropped if $|a_{i,k}| < \tau_i$, where τ_i, the relative tolerance, is the product of τ and the original norm of the i-th row.
- After a row has been updated, apply the previous dropping rule once again to discard all updated elements that are less than the relative tolerance. Then keep only the p largest elements in the L and U parts of the row respectively.

The drop tolerance, τ, serves to reduce computational cost, while the parameter lfil, p, reduces memory usage, by controlling the number of entries kept per row [19].

It is possible for the ILUT approach to fail for a number of reasons. The most obvious of these is when the original matrix A, contains a zero pivot (i.e. a zero diagonal entry). A remedy to this problem is to implement column pivoting. This leads us to a column pivoting variant of the ILUT called ILUTP [19]. ILUTP will usually result in a more robust preconditioner than ILUT, but this often comes with an additional cost in memory usage as ILUTP will tend to generate more fill-in. Since the issue is only to study the effect of using complex shifts, we will not not consider the pivoting variant ILUTP here. Regarding the choice of parameters for ILUT, a practical strategy is to take a large lfil value, and use the droptol to control the amount of fill-in. This generally yields good results without compromising memory efficiency.

3.2 Algebraic Recursive Multilevel Solver (ARMS)

ARMS [22] is a multilevel ILU preconditioner (a combination of ILU-based techniques with multilevel techniques), which implements a recursive solution to the construction of the preconditioner. The general mode of operation in the construction of a multilevel preconditioner is to first separate the points in the matrix into two subsets corresponding to the "coarse" set and the "fine" set. A block factorization of the matrix is then obtained from this partitioning and the process is continued recursively on the Schur complement associated with the coarse set. A brief sketch of the algorithm is now given.

In what follows, lev is the current level of the factorization: $0 \leq lev \leq last_{lev}$. A key step in ARMS is to find an independent set ordering P which permutes a matrix into the form

$$PAP^T = \begin{pmatrix} B & F \\ E & C \end{pmatrix},$$

where B is a block diagonal matrix corresponding to an independent set. Then, at level lev of the recursion, the ARMS procedure reorders the matrix in the above form and then factors it as follows,

$$P_{lev} A_{lev} P_{lev}^T = \begin{pmatrix} B_{lev} & F_{lev} \\ E_{lev} & C_{lev} \end{pmatrix} \approx \begin{pmatrix} L_{lev} & 0 \\ E_{lev} U_{lev}^{-1} & I \end{pmatrix} \begin{pmatrix} U_{lev} & L_{lev}^{-1} F_{lev} \\ 0 & A_{lev+1} \end{pmatrix}, \tag{4}$$

where P_{lev} and P_{lev}^T are the permutation and its transpose, respectively.

The idea then is to consider the matrix A_{lev+1} as the matrix of the next level and process it in turn by using the above strategy recursively.

ARMS(A_{lev}) factorization

```
1.  If $lev = last\_lev$ then
2.      Compute $A_{lev} \approx L_{lev} U_{lev}$
3.  Else:
4       Find an independent set permutation $P_{lev}$
5.      Apply permutation $A_{lev} := P_{lev}^T A_{lev} P$
6.      Compute block factorization
7.      Call ARMS($A_{lev+1}$)
8.  EndIf
```

The $last_{lev}$ variable in the above pseudocode is determined explicitly by the ARMS parameter *nlev*, which is the maximum number of levels for the recursion, or implicitly by the size of the Schur complement constructed. The size of the Schur complement A_{lev+1} is compared to the ARMS parameter *bsize*, which determines the minimum size of the Schur block. Hence if the size of A_{lev+1} is less than *bsize*, then $lev + 1 = last_{lev}$. Note that if $nlev = 0$ or $bsize \geq |A|$, then $last_{lev} = 0$ and the ARMS preconditioning operation is similar to an ILUT or ILUTP operation, depending on which approximation technique is used in step 2. In addition to these two ARMS parameters, ARMS also includes the usual ILU parameters, droptol and lfil, for dropping during the factorization at the intermediate levels, as well as the last level. Another key feature that adds to the robustness of this approach is that in step 2 of the factorization, different approximations may be used to solve the last system. In addition, Step 4 includes a simple heuristic to improve the quality of the resulting factors in the process: all rows that are judged poor from the point of view of diagonal dominance, are not considered as part of the independent set [22].

To improve robustness and performance on systems with poorly structured matrices, many preconditioners have been developed which make use of some form of permutation in their construction, see for instance [3, 6, 17]. This leads to ARMS with the diagonally-dominant PQ ordering which takes advantage of considering nonsymmetric permutations instead of the simple strategies based on independent sets. This technique is now described.

ARMS-ddPQ[1] is an extension of the standard ARMS that relies on nonsymmetric permutations to produce a better quality factorization. This is a two-sided approach where permutations P and Q are applied to the rows and columns respectively, to transform a matrix A into

$$PAQ^T = \begin{pmatrix} B & F \\ E & C \end{pmatrix}. \tag{5}$$

Unlike the standard ARMS, there is no particular structure and no assumptions for the B block. The permutation pair P,Q is selected such that the B block has the most diagonally dominant rows after the nonsymmetric permutation, and a few nonzero elements, so as to reduce fill-in.

At level *lev* of the above algorithm, the coefficient matrix is reordered as in (5), and the following block factorization is approximately computed (instead of (4)), [20]

$$P_{lev} A_{lev} Q_{lev}^T = \begin{pmatrix} B_{lev} & F_{lev} \\ E_{lev} & C_{lev} \end{pmatrix} \approx \begin{pmatrix} L_{lev} & 0 \\ E_{lev} U_{lev}^{-1} & I \end{pmatrix} \begin{pmatrix} U_{lev} & L_{lev}^{-1} F_{lev} \\ 0 & A_{lev+1} \end{pmatrix}. \tag{6}$$

[1] P and Q refer to the permutations used, and “dd” indicates that the ideal permutation would yield a diagonally dominant B block. [20]

The permutations P and Q are obtained in 3 steps:

1. Preselection Step: Filter out poor rows and sort the selected rows. This ensures that the selected rows best satisfy the diagonal dominance criterion.

2. Matching Step: Scan the candidate entries in the order given by the preselection, and accept them into a matching set M, or reject them. Here M is a set of n_M pairs (p_i, q_i), where $n_M \leq n$, the size of the matrix, and $1 \leq p_i, q_i \leq n$, for $i = 1, ..., n_M$ and $p_i \neq p_j$ for $i \neq j$, and $q_i \neq q_j$ for $i \neq j$. See [20, 15] for a few approaches to construct the matching set.

3. Matching Set Completion: The case where $n_M = n$ in the Matching step corresponds to a full permutation pair (P, Q). However, for $n_M \leq n$, the partial matching set M can be easily completed to a complete matching set (full pair of permutations (P, Q)) by a greedy approach.

Additional details on all the preconditioners described above can be found in a number of references, see, e.g., [19, 13, 12, 4, 22, 14, 23, 20, 15].

4 Use of complex shifts

4.1 Motivation

We begin this section with an experiment to illustrate the power of complex shifts in a very simple case. The experiment is programmed in matlab. We take the finite difference discretization of the operator $-\Delta$ on a 25×20 grid using centered differences. This results in a matrix of size $n = 500$. We then add a negative shift of -1 to the resulting matrix. The final matrix A has 30 negative eigenvalues (smallest is $\lambda_1 = -0.9631$ and largest $\lambda_{500} = 6.9631$). We then perform an ILU factorization of A and plot eigenvalues of $L^{-1}AU^{-1}$, calling the `LUINC` matlab function without pivoting and drop tolerance of 0.1. This yielded a preconditioner that is unstable. The condition number of the matrix LU was calculated to be 5.6e+17. This preconditioner is useless for any iterative method.

Next we shift the matrix by adding 0.25i to the diagonal of A and compute the resulting ILU preconditioner under the same conditions as above. As expected the condition number of LU is now better, and we find that $\kappa_1(LU) \approx 673.2$. The resulting L, U factors are used to precondition the original matrix A, not B so we plot the eigenvalues of $L^{-1}AU^{-1}$ where L, U are the incomplete LU factors of $B = A + 0.25 * i$. The result is shown in Figure 1. The new preconditioner now appears remarkably good. It yields a good clustering around one and few eigenvalues with negative real parts and none close to zero. The interesting observation here is that by making the problem complex, somewhat artificially, we avoid the very serious instability we initially encountered with the ILU factorization on the original (real) matrix.

The explanation for what happened can be seen by simply looking at the operator $L^{-1}AU^{-1}$, where now L, U are the *exact* L, U factors of B . The spectrum is shown on the right side of Figure 1.

Consider, more generally, A which has real eigenvalues and $B = A + \alpha i I_n$, where I_n is the nxn identity matrix, and let $B = LU$ be the LU factorization of B . The eigenvalues of the

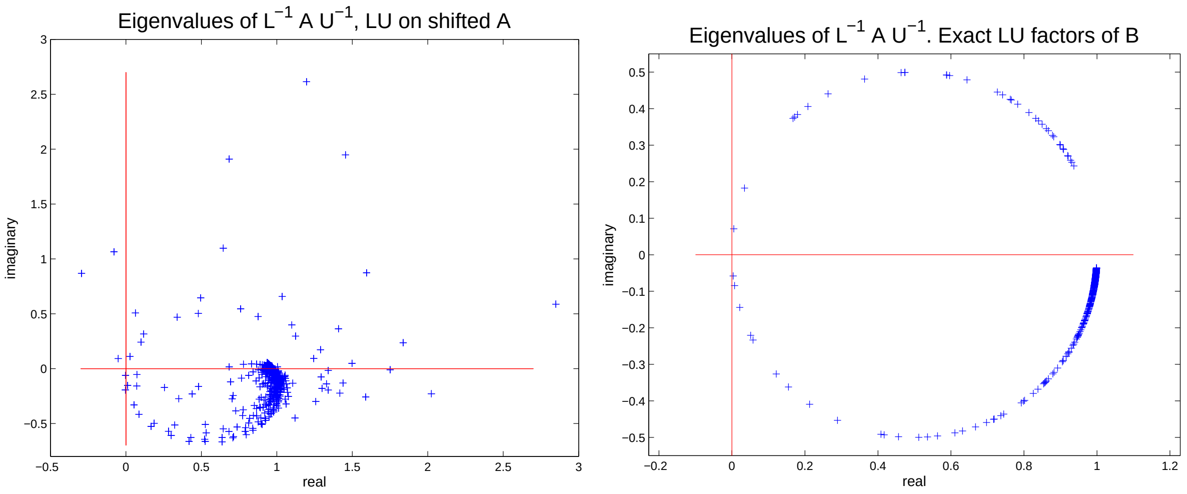


Figure 1: Spectrum of the preconditioned shifted Laplace operator. On the left side, the L and U factors are obtained from an incomplete LU factorization of $B = A + 0.25iI_n$. On the right side, the exact L, U factors of the same B are used.

matrix $L^{-1}AU^{-1}$ are simply

$$\mu_j = \frac{\lambda_j}{\lambda_j + i\alpha} \quad \rightarrow \quad \mu_j - \frac{1}{2} = \frac{1}{2}\,\frac{\lambda_j - i\alpha}{\lambda_j + i\alpha}\ . \tag{7}$$

As a result, $|\mu_j - 1/2| = 1/2$, so all eigenvalues are located in a circle centered at the point $1/2$ and with radius $1/2$. The large eigenvalues are mapped near the point 1 by this transformation. The good surprise is that all eigenvalues are on the right side of the complex plane. In addition, there is an accumulation point at one. One may wonder why most of the eigenvalues seem to be located on the lower side of the complex plane. This can be seen from another expression for μ_j which can be derived from (7), namely:

$$\mu_j = \frac{1}{2} + \frac{1}{2}\left[\frac{\lambda_j^2 - \alpha^2}{\lambda_j^2 + \alpha^2} - 2i\frac{\lambda_j\alpha}{\lambda_j^2 + \alpha^2}\right] = \frac{\lambda_j^2}{\lambda_j^2 + \alpha^2} - i\frac{\lambda_j\alpha}{\lambda_j^2 + \alpha^2}\ .$$

This indicates that for $\alpha > 0$ a (real) positive eigenvalue λ_j of A will be transformed into a complex eigenvalue with negative imaginary part and vice versa.

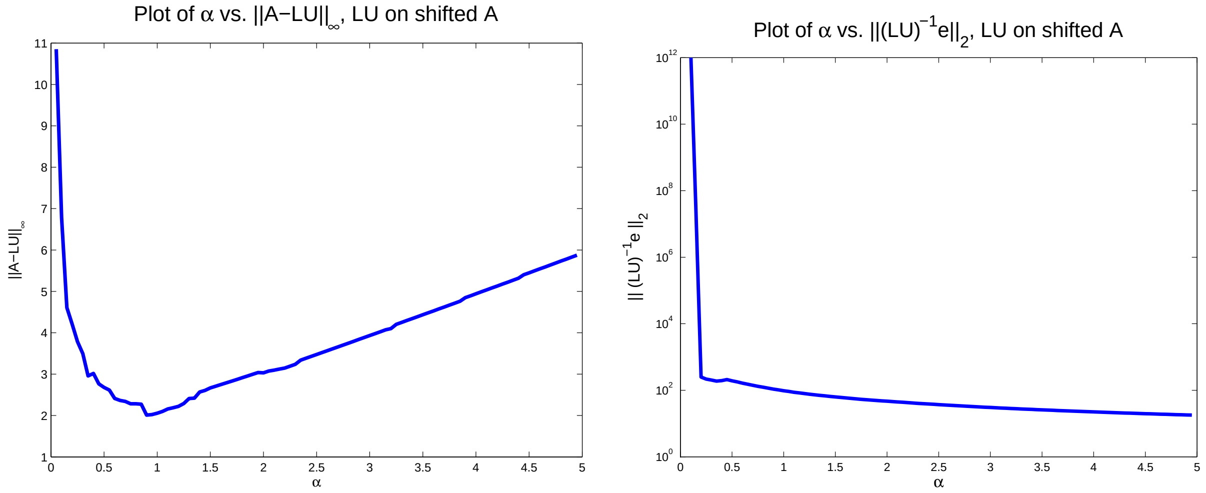
The above trivial example uses a constant shift and there was no specific effort made in its selection. When performing an ILU factorization such as ILUT, we need to modify the diagonal element of U by adding a complex shift. The main criterion we will use is to achieve a compromise between making α as large as possible to improve stability and making it not too large to maintain the accuracy of the ILU factorization.

4.2 Some sensitivity analysis

As mentioned earlier in the introduction, diagonal perturbations have been successfully used as a tool for safeguarding stability during an incomplete factorization. In order to better understand the effect of shifting on the factorization, and the quality of the resulting preconditioner, it is necessary to understand how it varies with certain quantities that determine the quality of the factorization.

Consider a small problem in acoustic scattering on a bounded obstacle governed by the Helmholtz equation, with wavenumber $k = 8\pi$. The resulting system matrix has size $n = 7380$, with 63900 non-zero entries. We solve the system using a preconditioned flexible GMRES [21], preconditioned with a shifted ILUT factorization with drop tolerance $\tau = 0.04$ and lfil set to be large. We use a constant shift, $i\alpha$ for each row of the factorization, and solve the system many times over varying α.

Figure 2 shows how α varies with $||A - LU||_\infty$ and $||(LU)^{-1}e||_2$, where e is a vector of all ones. The quantity $||A - LU||_\infty$ is shown to indicate how close is the product LU from A. The quantity $||(LU)^{-1}e||_2$, on the other hand, can be considered as a test for the stability of the factorization. Note that this is not a perfect measure as it can only provide a lower bound for the condition number of LU. An unstable factorization produces inverse factors that can be very large in norm, which renders the preconditioner useless. We see from figure 2(a) that the quantity $||A - LU||_\infty$ decreases rapidly to a minimum and then starts to increase again, with increasing α. This is easy to understand since by adding the diagonal shift, we make dropping smaller terms relatively easier, because the norm of the row has increased. As can be seen $||A - LU||_\infty$ decreases rapidly from $\alpha = 0$ (no compensation, $||A - LU||_\infty \approx 60$) to a minimum ($||A - LU||_\infty \approx 2$). However, as α increases, the perturbed system $B = A + i\alpha I_n$ becomes more and more different than A, and the LU factorization of B becomes a poor approximation to A. Moreover, as we will show later, increasing α causes more terms to be dropped during the factorization, which may result in LU factors that poorly approximate A. In figure 2(b), we see that the stability of the LU factorization generally improves with increasing α ($||(LU)^{-1}e||_2 \approx 10^{17}$ for $\alpha = 0.0$). This is to be expected since for larger perturbations, the matrix B becomes more diagonally dominant and yields a more stable factorization.



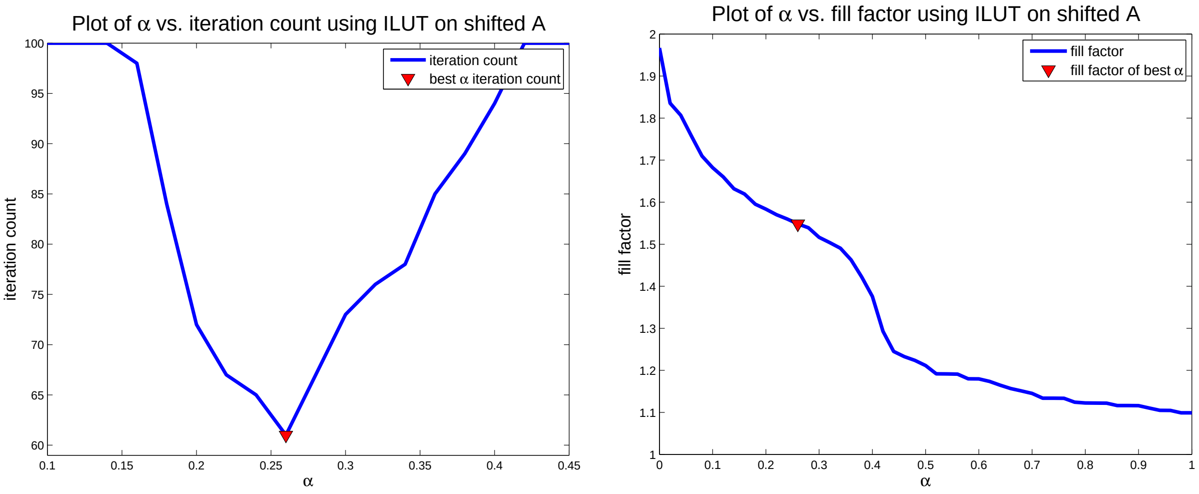
(a) **(b)**

Figure 2: An analysis of the effect of shifting on $||A - LU||_\infty$ (a); and $||(LU)^{-1}e||_2$ (b)

A good choice for α is one which balances the accuracy and stability of the factorization.

Figure 3 shows the effect of shifting on the iteration count for convergence, and the fill factor of the resulting factorization. We observe that the iteration count decreases as the quality of the factorization improves with increasing α to a minimum (61 iterations, $\alpha = 0.26$), and increases again after some optimal α. Here ILUT applied to the original matrix A failed to converge. In similar tests (not shown here), we reduced the drop tolerance parameter and observed a flatter profile for the minimum number of iterations, corresponding to more than one value for the optimal shift. This suggests that the choice of α is sensitive to the drop tolerance. We shall exploit this later in designing a scheme for choosing the shift α. In figure (3b), we see that the fill factor generally decreases with increasing α. This is because a large shift could result in small entries that are dropped during the elimination of a particular row of the factorization. As such, the factored row will contain less fill-in in the L and U parts. The figure shows that performing ILUT on the shifted matrix B, using the optimal (or best) α value ($\alpha = 0.26$) as the shift, results in a preconditioner that is about 25% cheaper in terms of memory costs, than the preconditioner derived from the original matrix.

These tests indicate that modifying the diagonal of the original matrix by adding a small complex perturbation, tends to improve the stability and accuracy of the resulting factorization, with the added benefit of reducing memory costs.



(a) **(b)**

Figure 3: An analysis of the effect of shifting on the iteration count for convergence (a); and the fill factor (b)

4.3 Choosing the complex shift

As mentioned in the introduction, several strategies have been proposed to modify the diagonal in order to improve the quality of the preconditioner. In [25], Van Gijzen et. al. proposed a "quasi" optimal value for the shift, as the value that minimized the upper bound of the GMRES residual norm. However, this optimal shift is derived with the assumption that an accurate factorization is performed during the preconditioning operation with the shifted-Laplace preconditioner, which may not be the case in practice. In [16], Made et. al. base their choice on the discretization of the problem, by relating the shift to the mesh size parameter.

We present two different algebraic strategies to select the complex shift. These strategies are based entirely on the premise that we wish to maximize the diagonal elements of the matrix to improve stability, without compromising the accuracy of the resulting preconditioner. Note that in these techniques, the shift is chosen locally with respect to each row of the matrix A, and hence is not constant (or global) as in the examples above in section 4.

Let $a_{kk} = \eta + i\beta$ denote the diagonal entry of the k-th row of the original matrix A. We will perturb this term by $i\alpha$ and seek to increase its squared modulus by at least the quantity γ^2, say. We therefore write:

$$\begin{aligned} |\eta + i(\beta+\alpha)|^2 &\geq |\eta + i\beta|^2 + \gamma^2 \\ \eta^2 + \beta^2 + 2\alpha\beta + \alpha^2 &\geq \eta^2 + \beta^2 + \gamma^2 \\ \alpha^2 + 2\alpha\beta - \gamma^2 &\geq 0 \end{aligned}$$

This is satisfied when $\alpha \geq -\beta + \sqrt{\beta^2+\gamma^2}$ or $\alpha \leq -\beta - \sqrt{\beta^2+\gamma^2}$. Increasing the square of the modulus of a_{kk} by γ^2 with the smallest perturbation is achieved by the following choice for α:

$$\alpha = \begin{cases} -\beta + \sqrt{\beta^2+\gamma^2} & \text{if} \quad \beta \geq 0 \\ \\ -\beta - \sqrt{\beta^2+\gamma^2} & \text{if} \quad \beta < 0 \end{cases} \tag{8}$$

Now, the question remains as to how to select the parameter γ. We propose two different schemes for selecting γ.

The first is based on improving the diagonal dominance of the rows of the matrix A. The motivation for this comes from the results of the analysis in section 4.2. We observed that even after reaching its minimum, the quantity $||A - LU||_\infty$ only grows slowly with increasing α. Moreover, only a small perturbation is required to reach the minimum. Thus by focusing on improving stability by improving diagonal dominance, we can obtain a factorization that is stable, and close to A in norm. Note that α need not be too large (for stability), as only a small shift is necessary to significantly improve stability, as depicted in figure 2.

Recall that row k is (strictly) diagonally dominant if

$$|a_{kk}| > \sum_{k \neq j} |a_{kj}|, \text{ for all } j.$$

Our strategy is based on selecting γ to be a weighted difference between $\sum_{k\neq j} |a_{kj}|$ and $|a_{kk}|$. The motivation for this is that the disparity between the two terms gives an indication of how far the row is from being diagonally dominant. Weighting is done to ensure that γ is not too large to affect the accuracy of the resulting preconditioner.

We refer to the quantity $\sigma = \sum_{k\neq j} |a_{kj}| - |a_{kk}|$ as *the diagonal dominance gap.* Note that the diagonal dominance gap is negative for strictly diagonal rows, in which case we need not employ any shifting. Consider the simple situation where the diagonal entry a_{kk} is perturbed so as to increase its modulus by $\rho\sigma$. Note that this is slightly different from the above requirement where the square of the modulus is augmented by γ^2. For some weight ρ, the gap σ_B after the shift is

$$\begin{aligned}\sigma_B &= \sum_{k\neq j}|a_{kj}| - \left[|a_{kk}| + \rho\left(\sum_{k\neq j}|a_{kj}| - |a_{kk}|\right)\right] = \left(\sum_{k\neq j}|a_{kj}| - |a_{kk}|\right)(1-\rho) \\ &= \sigma(1-\rho).\end{aligned}$$

With this viewpoint, the diagonal dominance gap will be reduced by a factor of $1-\rho$ and when $\rho = 1$ we obtain a row that is weakly diagonally dominant since $\sigma_B = 0$. The parameter ρ plays an important role in the size of the resulting shift. Note that ρ can be viewed as a scaling factor for the diagonal dominance gap and it is simpler if this scaling remained consistent across the rows of the matrix. That is, for each row k, we select γ as the same fraction of σ. In our work, we choose ρ as $\frac{n}{nnz}$, where n is the size of the matrix, and nnz is the number of non-zero entries in the matrix. We formally define γ as:

$$\gamma = \left(\sum_{k\neq j}|a_{kj}| - |a_{kk}|\right)\frac{n}{nnz}\ . \tag{9}$$

This is simply the diagonal dominance gap of row k scaled by the average number of nonzero elements per row in the matrix. We shall refer to this strategy for choosing α as the dd-based (diagonally dominant based) scheme.

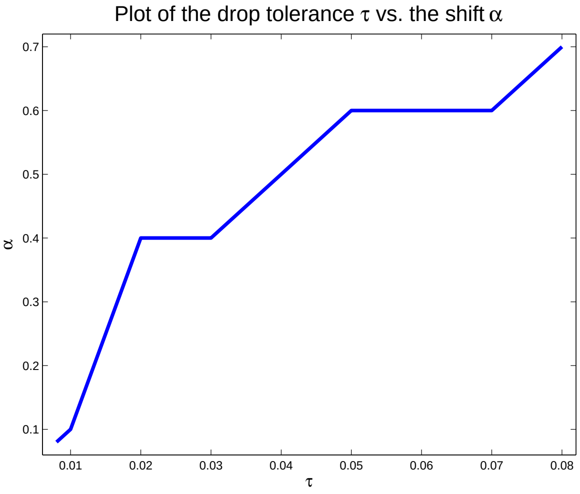


Figure 4: The effect of the drop tolerance τ, on the shift α. Values of τ are chosen so that the resulting fill factor is ≥ 1

The second heuristic is based on two ideas. The first is the assumption that each element that is dropped during the factorization is of the order of the drop tolerance τ. The second comes from the observation that α varies with τ, as was seen in figure 4. Standard ILUT yields a more accurate factorization (smaller $\|A - LU\|$) each time the drop tolerance is reduced (although this factorization may be unstable). Suppose it is also a relatively stable factorization. Then there need not be a large perturbation that could compromise accuracy. In the case where the factorization is unstable, again only a small perturbation may be necessary to remove unstable terms in the L and U factors to obtain better stability, as shown in section 4.2. Now suppose that the problem is such that a relatively large drop tolerance is needed to stabilize the factorization of the original matrix A. Then, again referring to section 4.2, a large shift could further improve stability without significantly affecting accuracy. The idea here is to define γ in terms of the drop tolerance τ. A heuristic which works well is to weight τ by an original norm of the row. For this scheme, we simply set γ as

$$\gamma_k = \tau \; \|a_{k:}\|_1, \tag{10}$$

where $||a_{k:}||_1$ denotes simply the 1-norm of the k-th row of A. We shall refer to this strategy as the τ-based scheme.

Figure 5 shows the eigenvalue spectrum of the original and preconditioned matrices for the problem described in section 4.2, using the different strategies. Here the eigenvalues were approximated from those of the Hessenberg matrix of size 300 × 300 of an underlying Arnoldi process. For the preconditioned matrices, we used ILUT as the preconditioner and adjusted τ so that the fill factor is the same (≈ 3.4) for the different strategies.

From the figure, we observe that the preconditioned matrices with the complex shift ((c) and (d)) have a better eigenvalue distribution compared to the preconditioned matrix without the shift (figure (5b)). Figure (5b) shows that the spectrum of the unshifted preconditioned matrix contains some eigenvalues with negative real parts. Moreover, although the majority of the eigenvalues seem to be clustered close to 1, there are a few extreme eigenvalues with relatively large real parts that lie in the range. The same cannot be said for figures (5c) and (5d), where all the eigenvalues are clustered very close to 1. The iterative solution to the problem required 55 iterations to converge for preconditioning with unshifted A, as opposed to 32 and 24 iterations for the dd-based scheme and τ-based scheme respectively.

5 Numerical Studies and Discussion

We compare the above mentioned preconditioners to solve the acoustic wave diffraction problem described in section 2, with respect to the following physical setting. We model a plane wave propagating along the x-axis, and incident on a bounded obstacle in the form of a disk of radius 0.5m. We implement a second order Bayliss-Turkel boundary condition [2] at the artificial boundary, at a distance 1.0m from the obstacle. We discretize the system using an isoparametric discretization over quad elements as shown in figure 6. The analytic solution to this problem is known, and can be found in [5, 26, 24].

In all test cases, we keep the ILUT fill level parameter to be large, so that dropping is controlled by the drop tolerance τ. For the test cases with the ARMS preconditioning, we use the ddPQ version of ARMS as it produced similar results as with the symmetric ARMS for simpler problems, and was more robust for harder problems. We also allowed a maximum of 4 intermediate levels for the ARMS factorization. We used restarted GMRES, with a restart dimension of 60, as the accelerator and set the convergence tolerance to 1.0x10^{-8}.

In the first test case, we compare the effect of shifting for the ARMS and ILUT preconditioners on a problem of size $n = 29241$, with wave number $k = 8\pi$. This corresponds to a mesh resolution of $\frac{\lambda}{h} = 20$, where h is the element mesh size. The fill factor for each solve is

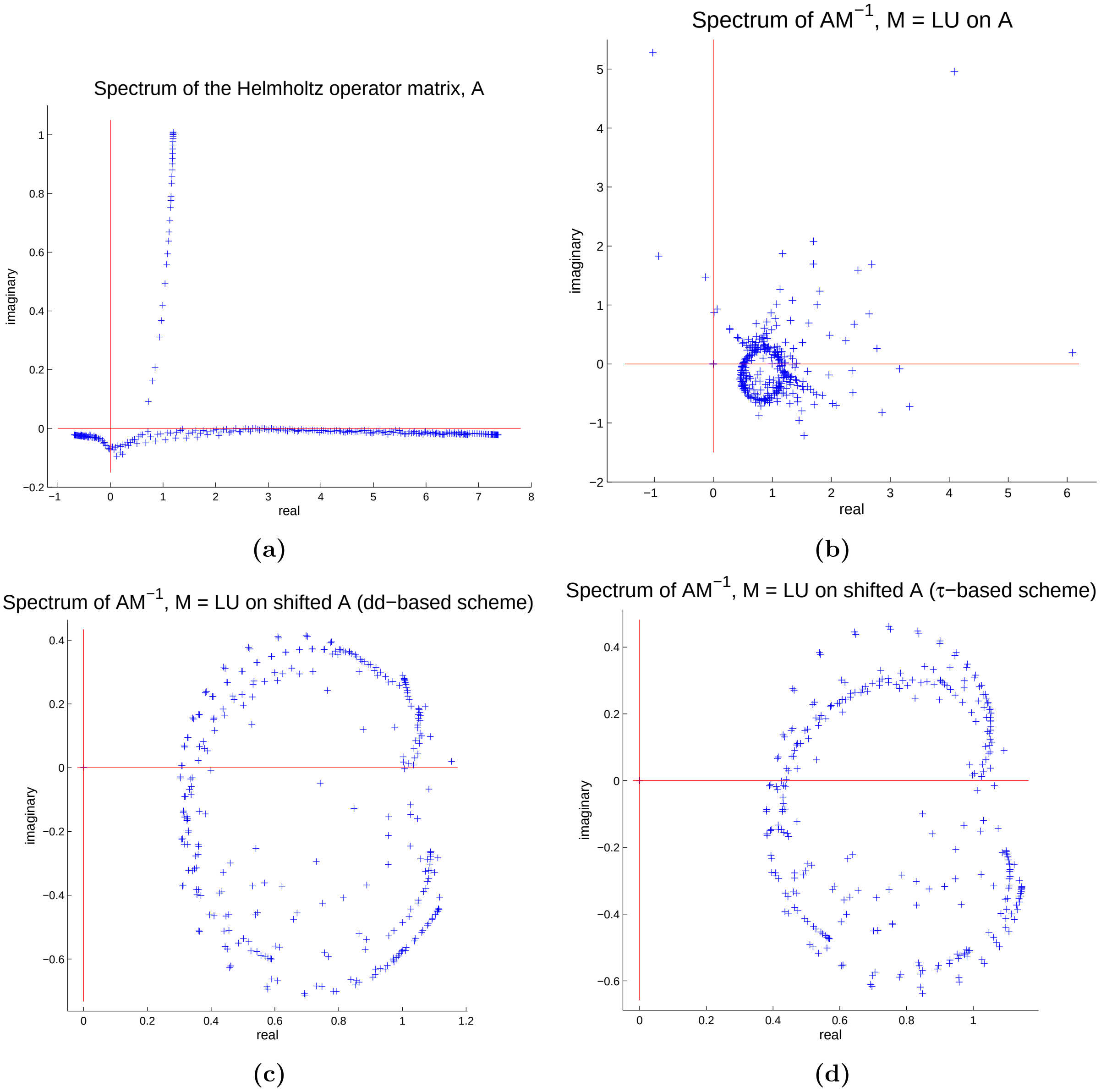


Figure 5: Approximate eigenvalues of the original matrix and the preconditioned matrices from the Helmholtz problem described in section 4.2: (a) spectrum of the original matrix; (b) spectrum of the preconditioned matrix, LU on original matrix A; (c) spectrum of the preconditioned matrix, LU on shifted matrix B (dd-based scheme); and (d) spectrum of the preconditioned matrix, LU on shifted matrix B (τ-based scheme)

fixed at ≈ 3.0 to allow for a fair comparison, and we allowed a maximum of 500 iterations for each test case. Figure 7 shows the convergence profiles for the numerical solution of the above problem with the ARMS and ILUT preconditioners, and their respective shifted variants. In both cases, we observe that the performance of the shifted variants is superior to the standard unshifted preconditioning. For the ILUT example, standard (unshifted) ILUT failed to converge for the fill factor allowed. In fact the factorization produced by this scheme under the

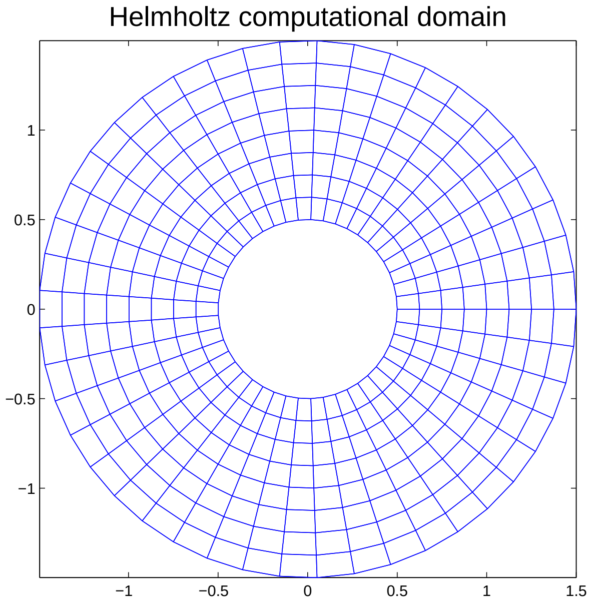


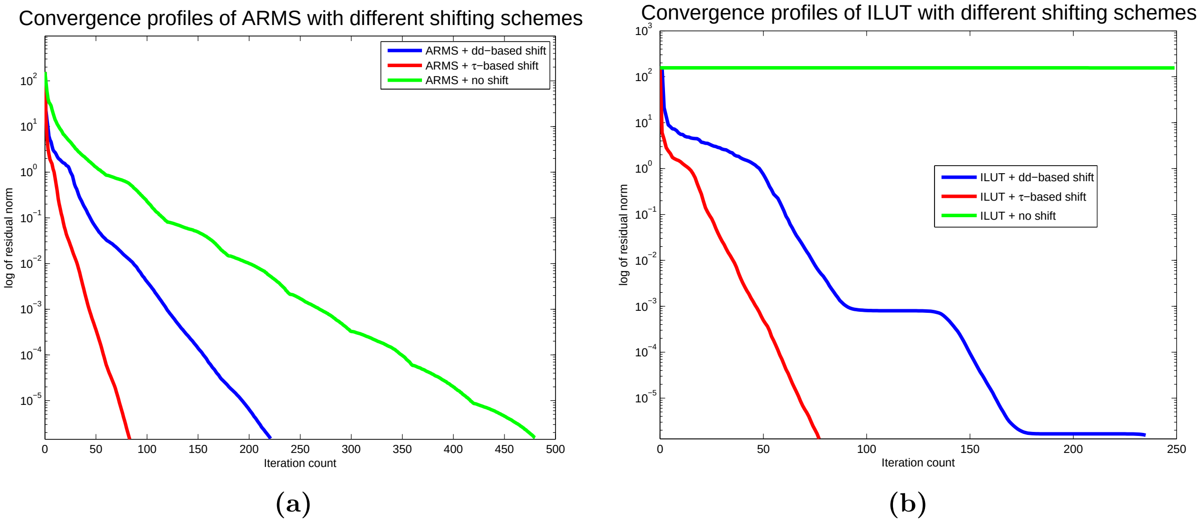
Figure 6: An example of the discretized domain mesh for the Helmholtz problem.

conditions was very unstable - $||(LU)^{-1}e||_2 \approx 1.3e+24$. Under the same conditions, we see that the shifted ILUT preconditioners handled the problem well. Shifting with the dd-based scheme converged although the factorization was not very stable either ($||(LU)^{-1}e||_2 \approx 5.3e + 14$). However, the result with the τ-based scheme was more impressive. It produced the smallest indicator of stable factorization ($||(LU)^{-1}e||_2 \approx 2.4e + 03$), and also yielded the best convergence. The results are similar in the tests with the ARMS preconditioner. Preconditioning with ARMS-ddPQ generally produced a more stable factorization, with ($||(LU)^{-1}e||_2 \approx 2.1e + 04$) for the unshifted preconditioner; ($||(LU)^{-1}e||_2 \approx 3.9e + 02$) for the dd-based scheme; and ($||(LU)^{-1}e||_2 \approx 2.2e + 03$) for the τ-based scheme.

Next we investigate the effect of the wave number k, on the different preconditioners. In particular, we are interested in whether or not shifting can help solve the Helmholtz problem at high wave numbers. The solution to the Helmholtz problem at high wave numbers is a highly researched topic. Work in [10] showed that the solution to this problem with standard ILUT (no shifting) was not very effective. This is because at high wave numbers, the Helmholtz problem is very indefinite, which could lead to an unstable factorization. Sometimes a relatively high drop tolerance (and hence less fill-in) may be required to drop unstable terms during the factorization. However, this is usually not enough to guarantee convergence as it may also compromise the accuracy of the factorization.

For this example, we solve the Helmholtz problem described ealier in this section on a 121 by 361 mesh. The discretized system has size $n = 43,681$, with $nnz = 387,000$ non-zero elements. We solve the problem for the wave numbers $k = 4\pi$, 8π, 16π, and 24π.

Table 1 shows results of the above problem with the ILUT preconditioner. Standard ILUT (no shift) stagnated for high values of k. At low values of k, it performed well, even doing better than with the dd-based scheme for the case with $k = 4\pi$. However, as k increases and



(a) **(b)**

Figure 7: Convergence Profiles for the ARMS and ILUT preconditioners, with their shifted variants: (a) ILUT and its shifted variants; (b) ARMS and its shifted variants

Preconditioner	k	$\frac{\lambda}{h}$	**No. iters**	**Fill Factor**	$\|\|(LU)^{-1}e\|\|_2$
ILUT (no shift)	4π	60	134	2.32	$3.65e+03$
	8π	30	263	2.25	1.23e+04
	16π	15	$-$	-	-
	24π	10	$-$	-	-
ILUT (dd-based)	4π	60	267	2.24	$2.29e+03$
	8π	30	255	2.23	4.73e+03
	16π	15	101	3.14	6.60e+02
	24π	10	100	3.92	2.89e+02
ILUT (τ-based)	4π	60	132	2.31	$2.98e+03$
	8π	30	195	2.19	4.12e+03
	16π	15	75	3.11	7.46e+02
	24π	10	86	3.85	2.73e+02

Table 1: Comparison of the different schemes for the ILUT preconditioner on application problem with different wave numbers.

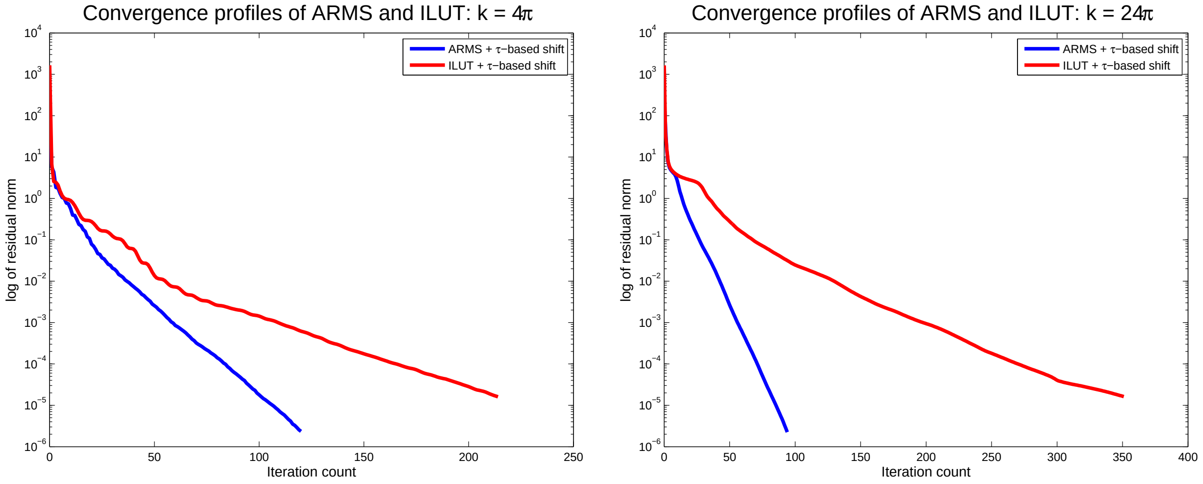
the system becomes more indefinite, the performance of the unshifted ILUT deteriorates, and we see the shifted schemes performing better. Once again the τ-based scheme outperforms the rest in all the different values of k.

Table 2 shows the results with the ARMS-ddPQ preconditioner. Here, standard ARMS without shifting stagnates for $k = 24\pi$. Once again we see the interesting situation where it performs better compared to the dd-based shifted scheme for relatively low values of k. But as k increases, the dd-based shifted scheme starts to do better. This is because at low values of k, the system is less indefinite, and standard unshifted ARMS can produce factors that are fairly stable and accurate. However, depending on the choice of the shift, the shifted scheme can yield a more stable factorization that is less accurate, as shown in figure 2 in section 4.2. The table shows that the choice of α used by the dd-based scheme seems to produce stable factors (as indicated by lower values for $||(LU)^{-1}e||_2$) for all the different values of k. This enables it to perform well at high wave numbers where the system is very indefinite. The τ-based scheme,

Preconditioner	k	$\frac{\lambda}{h}$	**No. iters**	**Fill Factor**	$\|\|(LU)^{-1}e\|\|_2$
ARMS (no shift)	4π	60	120	3.50	$7.48e+03$
	8π	30	169	4.03	1.66e+04
	16π	15	282	4.50	2.44e+03
	24π	10	$-$	-	-
ARMS (dd-based)	4π	60	411	3.83	$5.12e+02$
	8π	30	311	4.37	5.67e+02
	16π	15	187	4.71	3.92e+02
	24π	10	185	3.00	2.54e+02
ARMS (τ-based)	4π	60	106	3.45	$7.56e+03$
	8π	30	79	3.84	6.41e+03
	16π	15	39	3.95	1.26e+03
	24π	10	94	3.02	4.71e+02

Table 2: Comparison of the different schemes for the ARMS preconditioner on application problem with different wave numbers.

however, does not suffer at lower values of k, and converges the best for all choices of the wave number.



(a) **(b)**

Figure 8: Convergence Profiles for the τ-based shifted ARMS and ILUT preconditioners: (a) comparison for $k = 4\pi$; (b) comparison for $k = 24\pi$

At relatively low wave numbers, the ARMS-ddPQ and standard ILUT preconditioning techniques both give good results. However, as the wave number increases, standard ILUT begins to struggle. Figure 8 shows the convergence profiles of ARMS-ddPQ and ILUT, shifted by the τ-based scheme. The fill factor is fixed at approximately 3.0, and we plot results for $k = 4\pi$ and $k = 24\pi$. We see from the plot that the gap in performance between the two widens significantly for $k = 24\pi$.

In all our tests, shifting with the τ-based scheme yielded the best results for both ARMS and ILUT. It gave a stable and accurate factorization for the moderately to highly indefinite problems we investigated. The dd-based scheme typically produced the most stable factorization. However, as discussed in section 4.2, focusing on improving stability alone could adversely affect the accuracy of the resulting factorization. Since the accuracy of the standard ILUT technique is controlled by the drop tolerance, and since the best shift typically depends on the drop tolerance (as shown in figure 4), the τ-based scheme can improve stability while maintaining an accurate factorization. For the most indefinite problems discussed here, ARMS-ddPQ with the τ-based shifting strategy gave the best results in terms of both iteration count and memory costs.

6 Conclusion

Several authors have previously observed that adding purely imaginary shifts to the Laplace operator is a simple yet effective strategy for improving solution methods for the Helmholtz equation. In this paper we have adapted this technique to the algebraic context of ILU-type preconditioners. Our observation is that perturbing the diagonal entries of an indefinite system by a small complex perturbation can significantly improve the quality of the incomplete LU factorization of the matrix. Two different heuristics for selecting the complex shift were considered and tested with two incomplete LU factorization-based preconditioning techniques - ARMS and ILUT. Both strategies resulted in improved and more stable factors. The tests in this paper suggest that the drop tolerance used for dropping small terms during the factorization should be taken into account in the selection of the shift. Choosing the shift as the product of the drop tolerance and some original row norm (the 1-norm for instance) yielded good results overall, though much remains to be done to determine a truly optimal strategy.

Acknowledgements

The numerical tests described in this paper would not have been possible without a finite element code for acoustic waves provided to us by Kechroud and Soulaimani, see [10].

References

[1] T. Airaksinen, E. Heikkola, A. Pennanen, J. Toivanen, An algebraic multigrid based shifted-laplacian preconditioner for the helmholtz equation, Journal of Computational Physics 226 (1) (2007) 1196 – 1210.

[2] A. Bayliss, M. Gunzburger, E. Turkel, Boundary conditions for the numerical solution of elliptic equations in exterior regions, SIAM Journal on Applied Mathematics 42 (2) (1982) 430–451.
URL `http://link.aip.org/link/?SMM/42/430/1`

[3] M. Benzi, J. C. Haws, M. Tuma, Preconditioning highly indefinite and nonsymmetric matrices, SIAM Journal on Scientific Computing 22 (4) (2001) 1333–1353.
URL `citeseer.ist.psu.edu/benzi99preconditioning.html`

[4] M. Bollhöfer, A robust ILU based on monitoring the growth of the inverse factors, NLAA 338 (1) (2001) 201–218.

[5] J. J. Bowman, T. B. A. Senior, P. L. E. Uslenghi, Electromagnetic and acoustic scattering by simple shapes (Revised edition), Hemisphere Publishing Corp., New York, 1987.

[6] I. S. Duff, J. Koster, On algorithms for permuting large entries to the diagonal of a sparse matrix, SIAM Journal on Matrix Analysis and Applications 22 (4) (2001) 973–996.
URL citeseer.ist.psu.edu/duff99algorithms.html

[7] H. C. Elman, O. G. Ernst, D. P. O'Leary, A multigrid method enhanced by Krylov subspace iteration for discrete helmholtz equations, SIAM J. Sci. Comput 23 (2001) 1291–1315.

[8] Y. A. Erlangga, C. W. Oosterlee, C. Vuik, Comparison of multigrid and incomplete lu shifted-Laplace preconditioners for the inhomogeneous helmholtz equation, Appl. Numer. Math. 56 (2006) 648–666.

[9] Y. A. Erlangga, C. W. Oosterlee, C. Vuik, A novel multigrid based preconditioner for heterogeneous helmholtz problems, SIAM Journal on Scientific Computing 27 (2006) 1471–1492.

[10] R. Kechroud, A. Soulaimani, Y. Saad, S. Gowda, Preconditioning techniques for the solution of the Helmholtz equation by the finite element method, Math. Comput. Simul. 65 (4-5) (2004) 303–321.

[11] D. S. Kershaw, The incomplete cholesky-conjugate gradient method for the iterative solution of systems of linear equations, J. Comp. Phys. 26 (1) (1978) 43–65.

[12] N. Li, Y. Saad, Crout versions of the ILU factorization with pivoting for sparse symmetric matrices, Electronic Transactions on Numerical Analysis 20 (2006) 75–85.

[13] N. Li, Y. Saad, E. Chow, Crout versions of ILU for general sparse matrices, SIAM Journal on Scientific Computing 25 (2) (2003) 716–728.

[14] Z. Li, Y. Saad, M. Sosonkina, pARMS: a parallel version of the algebraic recursive multilevel solver, Numerical Linear Algebra with Applications 10 (2003) 485–509.

[15] S. MacLachlan, Y. Saad, Greedy coarsening strategies for non-symmetric problems, SIAM Journal on Scientific Computing 29 (5) (2007) 2115–2143.

[16] M. Magolu Monga Made, R. Beauwens, G. Warzee, Preconditioning of discrete helmholtz operators perturbed by a diagonal complex matrix, Comm. in Numer. Meth. in Engin. 16 (11) (2000) 801–817.

[17] M. Olschowka, A. Neumaier, A new pivoting strategy for Gaussian elimination, Linear Algebra and its Applications 240 (1–3) (1996) 131–151.
URL citeseer.ist.psu.edu/olschowka96new.html

[18] Y. Saad, ILUT: a dual threshold incomplete ILU factorization, Numerical Linear Algebra with Applications 1 (1994) 387–402.

[19] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd edition, SIAM, Philadelpha, PA, 2003.

[20] Y. Saad, Multilevel ILU with reorderings for diagonal dominance, SIAM Journal on Scientific Computing 27 (3) (2005) 1032–1057.
URL `http://link.aip.org/link/?SCE/27/1032/1`

[21] Y. Saad, M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (1986) 856–869.

[22] Y. Saad, B. Suchomel, ARMS: An algebraic recursive multilevel solver for general sparse linear systems, Numerical Linear Algebra with Applications 9.

[23] M. Sosonkina, Y. Saad, X. Cai, Using the parallel algebraic recursive multilevel solver in modern physical applications, Future Generation Computer Systems 20 (2004) 489–500.

[24] L. L. Thompson, P. M. Pinsky, A galerkin least squares finite element method for the two-dimensional helmholtz equation, Int. J. Numer. Meth. Eng 38 (3) (1995) 371–397.

[25] M. B. van Gijzen, Y. A. Erlangga, C. Vuik, Spectral analysis of the discrete helmholtz operator preconditioned with a shifted laplacian, SIAM Journal on Scientific Computing 29 (2007) 1942–1958.

[26] A. Zebic, Equation de Helmholtz: Étude numérique de quelques préconditionnements pour la methode GMRES, Tech. Rep. INRIA-RR-1802, INRIA, Unite de recherche de Rocquencourt, Rocquencourt, FRANCE (1992).