DELFT UNIVERSITY OF TECHNOLOGY

REPORT 09-11

Fast Iterative Solution Methods For The Helmholtz Equation

A.H. Sheikh, C. Vuik D. Lahaye

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2009

Fast Iterative Solution Methods For The Helmholtz Equation

A.H. Sheikh, C. Vuik, D Lahaye

November 9, 2009

Abstract

This report is oriented towards the step by step iterative solution of the Helmholtz equation. Discretization is done using the Finite Difference Method. Before solving iteratively, Krylov subspace methods are discussed, particularly the GMRES method which is used for experiments for Problems 2 in this report. Further *ILU* and *shifted Laplace preconditioners* [1] and [2] are reviewed and incorporated within GMRES. Deflation is used and its effectiveness is discussed for the Helmholtz equation with two right-hand side vectors as typically appears in adjoint based optimization problems.

1 Introduction

This report is related to the iterative solution of an elliptic partial differential equation, namely the *Helmholtz equation.* This equation often arises in the study of physical problems involving partial differential equations (PDE) in both space and time. (In this report we only consider the time-independent case.) Many problems related to steady state oscillations (mechanical, acoustical, thermal, electromagnetic) are modelled by the two-dimensional Helmholtz equation. Helmholtz equations have applications in many fields of science and technology, i.e in electromagnetics, aeronautics, acoustics, optics and in geo-physics.

The Helmholtz equation reads

$$-\Delta u(x,y) - (1-\alpha\iota)k^2(x,y)u(x,y) = g(x,y) \tag{1}$$

where $u(x,y)$ is the physical variable (quantity), $0 \leq \alpha \ll 1$ the *fraction of damping*, k the *wave number* and ι the imaginary unit i.e $\iota = \sqrt{-1}$. The equation is called homogeneous if the source function $g(x,y) = 0$ and inhomogeneous otherwise. A relation for the wavenumber k is

$$k(x) = \frac{2\pi}{\lambda} = \frac{\omega}{c(x)},$$

where $\omega = 2\pi f$ with f the *wave frequency*, $\lambda = \frac{c(x)}{f}$ the wavelength and $c(x)$ the speed of sound. The Helmholtz equation is basically derived from equations of Newton's law of motion and Hook's law [3].

Two test problems are discussed in this report. The first one has Dirichlet conditions and leads to symmetric positive definite systems. the second one has Sommerfeld radiation conditions and leads to nonsymmetric indefinite systems. Details are given in Section 2. A finite difference scheme is used for discretization as discussed in Section 3. Since the two problems have different boundary conditions, they need different boundary discretization schemes. In Section 4, Krylov methods and GMRES in particular are discussed and later properties of the linear systems of both problems are given. Section 5 contains few numerical results and further results are given in Section 6 and 7 for preconditioning and deflation respectively. As preconditioners, ILU and the shifted Laplace preconditioner [1,2] are used.

An overview of the literature review is given in the references.

2 Problem Description

Both Helmholtz problems we consider in this report are inhomogeneous, undamped (i.e. $\alpha = 0$), defined on the unit square $\Omega := (0,1) \times (0,1)$ and have a constant wavenumber.

2.1 Problem 1

In Problem 1 we choose negative constant wavenumber $k^2 = -5$ such that discretization of problem leads to symmetric and positive definite (SPD) system. The source function is defined as

$$g(x,y) = -6xy^2(x^2+2y^2) + 5x^3y^4 \tag{2}$$

Dirichlet conditions are imposed on this problem such that

$$u(x,y) = x^3y^4 \tag{3}$$

is the analytic solution of Problem 1.

2.2 Problem 2

In Problem 2 we choose the source function

$$g(x,y) = \delta(x_1 - \frac{1}{2}, x_2 - \frac{1}{2}) \tag{4}$$

with $x_1, x_2 \in (0,1)$ where Dirac delta function

$$\begin{aligned} \delta(x_1,x_2) &= +\infty \quad \text{if} \quad x_1 = 0, x_2 = 0 \\ &= 0 \qquad \text{if} \quad x_1 \neq 0, x_2 \neq 0, \end{aligned}$$

and is also constrained to satisfy

$$\int\int \delta^2(x_1,\, x_2)\, dx\, dy = 1.$$

This means that the waves propagates from the center of the domain outwards. Here the Sommerfeld radiation conditions of first order are imposed meaning that

$$\frac{\partial u}{\partial n} - \imath k u = 0. \tag{5}$$

Due to the complex term in the boundary conditions imposed on Problem 2, one ends up with a complex-valued linear system. The exact solution is

$$u(x) = \frac{\imath}{4} H_0^{(1)}(k|x|), \tag{6}$$

where $H_0^{(1)}$ is an *Hankel function.*
Numerical experiments for both problems are done with variation of the wave number k. Problem 1 will act as an auxiliary problem in solving Problem 2.

3 Discretization

The problems under discussion can be discretized by many methods including Finite Element Method (FEM) and the Finite Difference Method (FDM). In this report the latter is used. The procedure to solve a partial differential equation numerically is to obtain a discrete analogue of the given continuous equation and boundary conditions and then to solve it. Discretization is performed on a *square grid*

$$G_h = \left\{ (x_i, y_j) \mid x_i = ih, y_j = jh, h = \frac{1}{N}, 0 \leq i,j \leq N \right\}$$

with mesh-size $h = \frac{1}{N}$ for the both problems.

In FDM, the differential terms are approximated by finite differences using Taylor's polynomials. The differential terms in Equation 1 are approximated by finite differences as

$$u_{xx} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + O(h^2)$$

and

$$u_{yy} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} + O(h^2).$$

The error in this approximation is of $O(h^2)$.

Using above differences for differential terms on the square grid G_h, the discrete analogue of Equation 1 is

$$-(\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2}) + k^2 u_{i,j} = g_{i,j}.$$

Rewriting above equation results in

$$\frac{1}{h^2}\left\{-u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} + 4u_{i,j} - h^2k^2u_{i,j}\right\} = g_{i,j}. \tag{7}$$

3.1 Discretization of Problem 1

In Problem 1, the wave number k^2 is a fixed negative constant i.e $k^2 = -5$ and Dirichlet boundary conditions are imposed. The boundary data is evaluated in the boundary points and these values are added to the right-hand side vector.
Due to Dirichlet boundary conditions, only internal nodes of the grid are unknowns in the system. The stencil for those internal points is

$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & k^2+5h^2 & -1 \\ & -1 & \end{bmatrix} u_{i,j} = g_{i,j}$$

Due to the elimination of boundary conditions, the structure of the stencil for nodes next to the boundary is different from that of the internal nodes. The stencil for the point which is connected to the upper and the left boundary for instance, is given by

$$\frac{1}{h^2}\begin{bmatrix} & 0 & \\ 0 & k^2+5h^2 & -1 \\ & -1 & \end{bmatrix} u_{i,j} = g_{i,j}$$

The unknowns are lexicographically ordered, and conversion of two indices into one is according to formula $f_k = f_{i+(j-1)(N-1)}$. Finally, the discretization leads to the linear system

$$Au = g$$

where

$$A = \frac{1}{h^2}\begin{bmatrix} k^2+5h^2 & -1 & ... & -1 & & & & & \\ -1 & k^2+5h^2 & -1 & ... & -1 & & & & \\ ... & -1 & k^2+5h^2 & 0 & & -1 & & & \\ -1 & ... & 0 & k^2+5h^2 & -1 & & -1 & & \\ & -1 & & -1 & k^2+5h^2 & -1 & & -1 & \\ & & -1 & & -1 & k^2+5h^2 & 0 & & -1 \\ & & & -1 & & 0 & ... & -1 & ... \\ & & & & -1 & & -1 & ... & -1 \\ & & & & & -1 & ... & -1 & k^2+5h^2 \end{bmatrix} \tag{8}$$

and

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ ... \\ ... \\ u_{N-1} \\ n_N \\ ... \\ ... \\ u_{(N-1)^2} \end{bmatrix}$$

$g = [g_k]$ where $1 \leq k \leq (N-1)^2$.

Boundary elimination is such that the values of the boundary function at points on four different edges are added to right-hand side vector g as

- Lower(South) :$f_i + (\frac{1}{h^2})gs_{(1,i)}$ for $1 \leq i \leq (N-1)$
- Right(East): $f_{(N-1)i} + (\frac{1}{h^2})ge_{(1,i)}$ for $1 \leq i \leq (N-1)$
- Upper(North): $f_{(N-2)(N-1)+i} + (\frac{1}{h^2})gn_{(1,i)}$ for $1 \leq i \leq (N-1)$
- Left(West): $f_{1+(N-1)(i-1)} + (\frac{1}{h^2})gw_{(1,i)}$ for $1 \leq i \leq (N-1)$.

Note that the boundary function "g" was already stored on *four different edges* respectively as above. Order of discretization error is shown in Figure 2.

3.2 Discretization of Problem 2

In Problem 2, the Sommerfeld radiation condition of first order given in Equation 5 are imposed.
For discretization of this condition, a point is supposed to lie opposite every node on each boundary with same mesh size (Figure 1), which is called a *ghost point* [4]. For e.g. the right boundary, ghost point is $u_{N+1,j}$ where $1 \leq j \leq N$ and from discretized (by central difference) radiation boundary conditions at any point (i,j)

$$\frac{\partial u}{\partial n} - \imath k u = u_{i+1,j} - u_{i-1,j} - 2h\imath k u_{i,j} = 0,$$

the value of u at a ghost point is replaced by value of u at the first internal point to the boundary. The stencil for internal points is

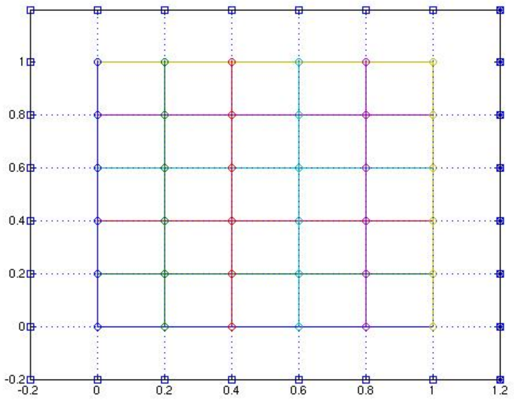

Figure 1: Grid for mesh size h = 1/5: circles show nodes for unknown and squares for (introduced) ghost points

$$\frac{1}{h^2}\left[\begin{array}{ccc} & -1 & \\ -1 & 4-k^2h^2 & -1 \\ & -1 & \end{array}\right] u_{i,j} = g_{i,j}$$

and after boundary eliminating with above vertex centered strategy, the stencil for points on boundaries (for instance on left boundary) and corner (for instance on lower-left corner) are given by

$$\frac{1}{h^2}\left[\begin{array}{ccc} & -1 & \\ 0 & 4-k^2h^2+2\imath kh & -2 \\ & -1 & \end{array}\right] u_{i,j} = g_{i,j}$$

$$\frac{1}{h^2}\left[\begin{array}{ccc} & -2 & \\ 0 & 4-k^2h^2+4\imath kh & -2 \\ & 0 & \end{array}\right] u_{i,j} = g_{i,j},$$

respectively. The resulting coefficient matrix A of the linear system obtained by this discretization is of the form

$$A = \frac{1}{h^2}\begin{bmatrix} p+4q & -2 & ... & -2 & ... & & & & \\ -1 & p+2q & -1 & ... & -2 & ... & & & \\ ... & -2 & p+4q & 0 & ... & -2 & ... & & \\ -1 & ... & 0 & p+2q & -2 & ... & -1 & ... & \\ ... & -1 & ... & -1 & p & -1 & ... & -1 & ... \\ & ... & -1 & ... & -2 & p+2q & 0 & ... & -1 \\ & & ... & -2 & ... & 0 & p+4q & -2 & ... \\ & & & ... & -2 & ... & -1 & p+2q & -1 \\ & & & & ... & -2 & ... & -2 & p+4q \end{bmatrix} \tag{9}$$

where $p = 4 - k^2h^2$ and $q = \imath kh$. For sake of simplicity, the matrix above is such that it is discretized on mesh of size $N = 3$.
Now the linear systems obtained from discretization of Problem 1 and Problem 2 are discussed in the next section, but before touching the linear systems, some information about iterative solvers, particularly Krylov subspace methods are discussed.

4 Linear System and Solvers

In this section, first Krylov subspace methods are discussed. Afterwards we describe the linear systems of our problems. Krylov subspace methods are considered as one of the important classes of numerical methods for *Sparse Linear Systems of Equations* and *Large Sparse Matrix Eigenvalue Problems.* Here, systems of equations are treated only as our concern, but many of the numerical methods for large eigenvalue problems are based on similar ideas as the ones treated below.

4.1 Introduction to Iterative Solvers (Krylov Space Methods)

For any linear system $Au = g$ with invertible matrix A, a *basic iterative method* is of the form

$$u^{n+1} = Qu^n + s$$

for all n, where Q is the *iteration matrix* and s a vector. This is equivalent to the original system, as it is obtained by splitting A. Variation of types of splitting lead to different *stationary* (basic iterative) methods. Using simple splitting of $A = B + (A - B)$, we get the following recursion which is equivalent to the given system

$$u^{n+1} = u^n + B^{-1}(g - Au^n) = (I - B^{-1}A)u^n - B^{-1}g. \tag{10}$$

Comparison tells that the iteration matrix $Q = (I - B^{-1}A)$ and $s = B^{-1}g$.

For the most problems, since exact solution $u^* = A^{-1}g$ is not available, we can not compute the error

$$e^n = u^* - u^n.$$

Thus for checking the convergence, one usually use the *residual* defined as:

$$r^n = g - Au^n. \tag{11}$$

Also an other relation for the residual is $r^n = Ae^n$. Hence once we have residual, we can analyze error by this relation.
Assuming $B = I$, then $Q = I - A$ and then we have

$$r^n = g - Au^n = (I - A)u^n + g - u^n = u^{n+1} - u^n.$$

Thus the iteration can be rewritten as:

$$u^{n+1} = u^n + r^n. \tag{12}$$

Recursion in equation 12 is

$$\begin{aligned} u^1 &= u^o + r^o \\ u^2 &= u^1 + r^1 \\ &= u^o + 2r^o - Ar^o \\ &\vdots \end{aligned}$$

The recursion for the residual can also be obtained from equation 12

$$\begin{aligned} r^{n+1} &= g - Au^{n+1} \\ &= g - A(u^n + r^n) \\ &= r^n - Ar^n \\ r^{n+1} &= Qr^n \end{aligned}$$

Combining above recursions leads to

$$r^{n+1} = P_n(A)r^o \in \mathit{Span}\left\{r^o, Ar^o, A^2r^o, ..., A^nr^o\right\},$$

where $P_n(A)r^o$ is an nth degree polynomial in A. Now Equation 10 takes the form

$$u^n = u^o + r^0 + .. + r^{n-1} = u^o + P_{n-1}(A)r^o$$

which assures the existence of an approximation to solution in the space

$$u^o + \mathit{Span}\left\{r^o, Ar^o, A^2r^o, ..., A^nr^o\right\}.$$

This space

$$K^n(A; r^o) = \mathit{Span}\left\{r^o, Ar^o, A^2r^o, ..., A^{n-1}r^o\right\}$$

is called the *Krylov subspace.*

The basics of Krylov subspace methods are to construct iterations initiating with an initial approximation u^o (usually taken the zero vector) and corresponding residual $r^o = f - Au^o$ untill one gets an accurate approximation to the exact solution u^*.

Conjugate Gradient (CG) [5], MINRES [6], *Generalized Minimal Residual Algorithm* (GMRES) [7], BICG, BICGSTAB, CGNR are some Krylov methods. CG is a standard Krylov method, however CG is limited in application to only symmetric and definite problems. On contrast, GMRES has the property that it can be used for indefinite and non-symmetric problems as well. This algorithm is developed by Saad and Schultz [7]. Basically GMRES minimizes the residual over the Krylov subspace as

$$\|r^n\| = \|g - Au^n\| = \min_{z \in K^n(A; r^o)} \|r^o - Az\|, \tag{13}$$

where $u^n = u^o + z^n$ with $z^n \in K^n\,(A; r^o)$.

4.2 GMRES: Algorithm

```
1.  Start: $r^o = g - Au^o$ with $u^o$ initial guess, $\beta =\|r^o\|$ and $v_1 = r^o/\beta$
2.  for  $j = 1, 2...n$ do
3.      $w_j = Av_j$
4.      for  $i = i, 2...j$ do
5.          $h_{i,j} = (w_j, v_i)$
6.          $w_j = w_j - h_{i,j}v_i$
7.     end do
8.     $h_{j+1,i} =\|w_j\|$
9.     $v_{j+1} = w_j h_{j+1,i}$
10. end do
11. $H_n = [h_{i,j}]$ Hessenberg matrix of dimension $(n + 1) \times n$
12. Computing of minimizer $y_n$ over $\|\beta e_1 - H_n y\|$
13. Set solution: $u^n = u^o + V_n y_n$
```

Here e_1 is unit vector $\in \mathbb{R}^{n+1}$.

At n^{th} iteration, for computed solution is u^n, corresponding residual

$$\|r^n\| =\|g - Au^n\| =\|u^o + z^n\|$$

As however $z^n \in K^n(A; r^o)$ and $z^n = V_n y$, where V_n is *orthonormal* basis of Krylov space computed according to first 9 lines of above algorithm, the residual can be written as

$$r^n = g - Au^n = g - A(u^o + V_n y). \tag{14}$$

GMRES approximates the solution u^n over the Krylov space, while minimizing residual in Equation 13. This approximation is obtained as in line 13 of the algorithm and with $V_n y$, the residual is

$$\begin{aligned} r^n &= r^o - AV_n y \\ &= \beta v_1 - V_{n+1} H_n y \\ &= V_{n+1}(\beta e_1 - H_n y) \end{aligned}$$

As V_n is an orthonormal matrix, its 2-norm is 1 and we have

$$J(y) =\|\beta e_1 - H_n y_n\| = \min_{y \in \mathbb{R}^n} \|\beta e_1 - H_n y\| \tag{15}$$

The problem reduces to finding out the solution y_n of the above least squares problem. To solve the least squares problem, the Hessenberg matrix H_n is transformed into a $Q_n R_n$ factorization using plane rotations [7]. For convenience we suppose H_n to be a 2×2 Hessenberg matrix. The rotation matrix Q_2 is then

$$Q_2 = \begin{vmatrix} c & -s \\ s & c \end{vmatrix}$$

where $c = \frac{h_{1,1}}{\sqrt{h_{1,1}^2+h_{2,1}^2}}$ and $s = \frac{h_{2,1}}{\sqrt{h_{1,1}^2+h_{2,1}^2}}$ with property $Q_2^T Q_2 = I$ [7].
Since our linear system is complex valued, hence elements of Q_2 are also complex, and therefore Q_2 must satisfy $\bar{Q}^T Q = I$ and therefore Q is developed as

$$Q_2 = \begin{vmatrix} \bar{c} & -\bar{s} \\ s & c \end{vmatrix}$$

with $c = \frac{h_{1,1}}{\sqrt{h_{1,1}\bar{h}_{1,1}+h_{2,1}\bar{h}_{2,1}}}$ and $s = \frac{h_{2,1}}{\sqrt{h_{1,1}\bar{h}_{1,1}+h_{2,1}\bar{h}_{2,1}}}$ [8].

Now some properties of linear systems of both problems are given.

4.3 Linear system

4.3.1 Properties of Linear System of Problem 1

The coefficient matrix in the Problem 1 with Dirichlet condition is sparse with five diagonals. The matrix has also zero elements in off-diagonals in the rows corresponding to (left and right) boundaries. The matrix is a real, symmetric, positive definite M-matrix.

4.3.2 Properties of Linear System of Problem 2

The coefficient matrix in the Problem 2 is large, sparse, complex-valued due to inclusion of Sommerfeld radiation condition and for the bigger value of wave number, it is indefinite, as discussed in spectral analysis of the system. Further more the matrix is non symmetric and non-hermitian.

5 Numerical Experiments

5.1 Problem 1

To show the order of the discretization, first the linear system obtained from the discretization of Problem 1 is solved in Matlab by the backslash operator with different values of N and the error obtained with the analytical solution for different step sizes is given in Table 1. This data is also plotted in Figure 2. Figure 2 indicates quadratic convergence for Problem 1.

Stepsize “h“	**Abs: Error-norm**
$\frac{1}{2^2}$	0.001650
$\frac{1}{2^3}$	0.000452
$\frac{1}{2^4}$	0.000116
$\frac{1}{2^5}$	0.000029
$\frac{1}{2^6}$	0.000007

Table 1: Increasing step size h decreases error with order $O(h^2)$

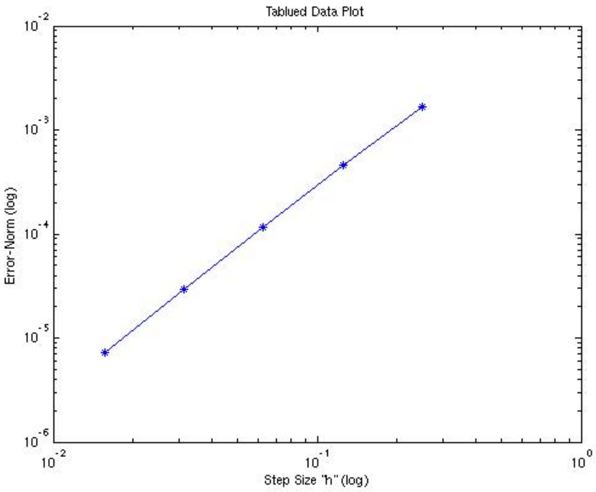

Figure 2: Stepsize-Norm of the Error

5.2 Problem 2

For all the experiments, $u^0 = 0$ is used as initial guess. As stopping criteria for the iterative algorithms, the following condition is used

$$\frac{\|g - Au^n\|}{\|g\|} \leq 10^{-7} \tag{16}$$

Also from here and onwards, the mesh size h is such that for a wave number k, it satisfies $kh = 0.625$ (equivalent to 10 grid points per wave length) in all numerical experiments, unless mentioned.

For Problem 2, a first result is shown in Figure 3. In this figure, the real part of the solution for $k = 50$, computed by using GMRES is compared with the analytical solution given in Equation 6.

Table 2 shows the performance of the GMRES algorithm with respect to a variation of k. Increasing the wave number k severely affects the number of iterations and the actual residual. The later is due to the structure of the right hand side vector. The effect of decreasing mesh size for a fixed wave number for e.g. $k = 20$ is shown in Figure 4. This

k	**Dim of A**	**Iterations**	**Norm of Actual Residual**
10	289	36	$8.19468\ 10^{-6}$
20	1089	82	$7.13048\ 10^{-5}$
30	2401	143	$1.57232\ 10^{-4}$
40	4225	231	$3.6748\ 10^{-4}$
50	6561	341	$5.77987\ 10^{-4}$

Table 2: An observation:Number of iterations by GMRES with different values of wave number k along with the Actual Residual.

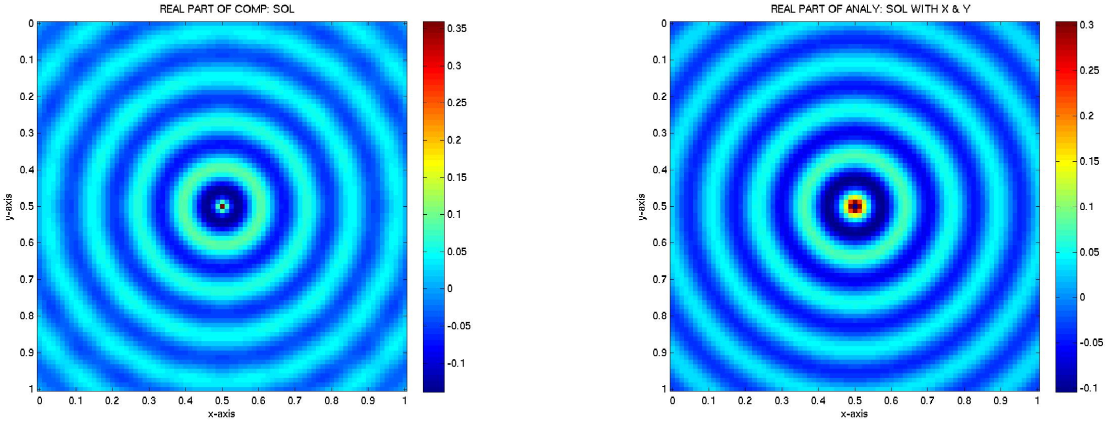

Figure 3: (a) Real part of Numerical Solution by GMRES at $k = 50$(b)Real part of Analytical solution at $k = 50$ and $\alpha = 0$

figure clearly shows that decreasing the mesh size is costly in terms of iterations.
In Figure 6, the convergence history of GMRES for wavenumber $k = 30$ is given and is compared with the convergence history of preconditioned GMRES (see Section 6.1).
In the next section, we discuss the preconditioning of our linear systems with various preconditioners.

6 Preconditioning

In our problems, the GMRES (and other Krylov subspace solvers) are slow to converge. In general, the convergence rate of Krylov subspace methods depends upon the spectrum, it is therefore necessary for all Krylov subspace methods to have favorable spectrum in order to converge fast. Therefore preconditioning is introduced.
Simply said, preconditioning means transforming a linear system by multiplying the preconditioner M(on left, right and split) into one favourable for any iterative solver, preserving the solution. In the extreme case, A itself is the best choice to choose as preconditioner, but it is impracticable, since using A as a preconditioner is as expensive as solving the original system. So the preconditioner must resemble the coefficient matrix A but also be easier to solve than A. Many preconditioners are developed and used for various problems, those were obtained by coefficient matrix of linear system or by a discrete operator of a related problem. After choosing a preconditioner M for any system $Au = g$, whose invertible is cheap to compute, then the transformed system is

$$M^{-1}Au = M^{-1}g.$$

This is called *left preconditioning.* Also *right* preconditioning

$$AM^{-1}\bar{u} = g,$$

where $\bar{u} = Mu$ and *split preconditioning* if $M = M_1M_2$ then

$$M_1AM_2M_2^{-1}u = M_1b.$$

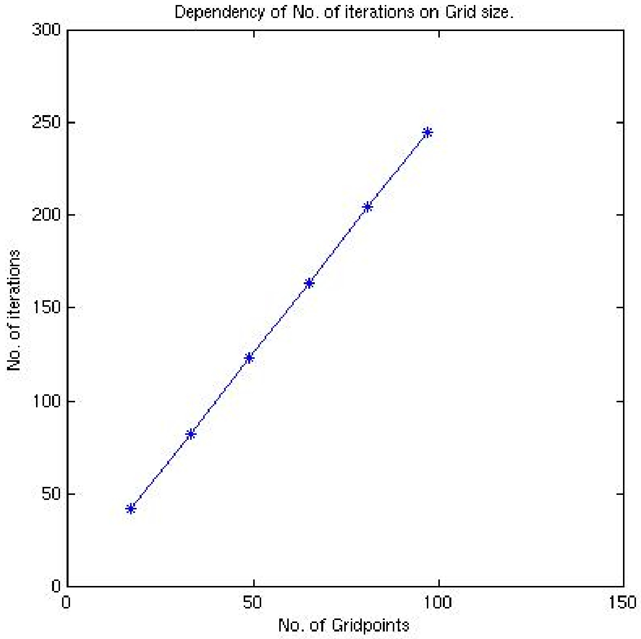

Figure 4: Dependence of number of GMRES iterations on grid size for a problem with $k = 20$.

6.1 ILU Preconditioner

One simple class of preconditioners is obtained by an *Incomplete LU* factorization of A, where L and U are a lower and upper triangular matrix respectively. There are many ways to obtain approximate ILU factorizations of A, e.g. Zero fill-in ILU and ILU with some tolerance. ILU obtained by restricting the structure of L and U to equal that of A leads to preconditioner, known as $ILU(0)$ which is easy to compute but not so effective for Krylov space methods. A more accurate ILU factorization of A is obtained by allowing more fill-in the ILU-factorization. Dropping the elements less than some given value in ILU gives rise to $ILU(tolerance)$.

Some experiments are done using GMRES with preconditioners $ILU(0)$ and $ILU(tolerance)$ with tolerance (0.01). Table 3 and 4 show the number of iterations with actual residual as it seems to be decreased with decreasing mesh size and also increasing wave number k. Results in Table 3 and 4 indicate that $ILU(0.01)$ works better than $ILU(0)$ but at the cost of big storage, that is due to more *fill-in* is allowed, which is indicated by the number of non zero elements in factorizations L and U.

We observe that the convergence depends upon the mesh size as shown in Figure 5. This figure shows that the rate of convergence increases when mesh size is decreased for a fixed wave number $k = 30$. One can observe from Table 3 that with for larger wave number $ILU(0.01)$ is more effective regarding the number of iterations. $ILU(0.01)$ is however not acceptable since the amount of fill-in is quite large. One can note from Table 4 that the ratio of the nonzero number of entries in L and U to nonzero entries of A is increasing

k	Dim of A	Iterations	nz(A)	nz(L and U)	Actual Residual
10	289	21	1377	833	7.19781 10^{-6}
20	1089	43	5313	3201	8.87075 10^{-5}
30	2401	71	11809	7105	2.20553 10^{-4}
40	4201	99	20865	12545	3.91858 10^{-4}
50	6561	120	32481	19521	6.37758 10^{-4}

Table 3: Number of iterations by GMRES with preconditioner $ILU(0)$ with different values of wave length k.

k	Dim of A	Its.	nz(A)	nz(L)	nz(U)	nz(L+U)/nz(A)	Actual Residual
10	289	8	1377	2531	2500	3.6536	9.53795 10^{-6}
20	1089	13	5313	11327	11208	4.2415	5.80282 10^{-5}
30	2401	21	11809	26665	26195	4.4762	2.02485 10^{-4}
40	4225	32	20865	48533	47688	4.6116	3.81552 10^{-4}
50	6561	49	32481	77094	75816	4.7077	4.38258 10^{-4}

Table 4: Number of iterations by GMRES with preconditioner $ILU(0.01)$ with different values of wave length k .

gradually and hence $ILU(0.01)$ is expensive to apply. Also in both versions of ILU, storage problems can occur, where the problem with ILU(0.01) can be severe. Actual residual is decreasing with respect to increasing wave number (See Table 3 and 4).

The convergence history for GMRES with $ILU(0)$ and $ILU(0.01)$ as preconditioners is given in Figure 6 and compared with that of GMRES without preconditioners. Some more effective preconditioner is needed. In next section, an other preconditioner called *"Shifted Laplace Preconditioner"* is discussed

6.2 Shifted Laplace Preconditioner

Another class of preconditioners for the Helmholtz equation is obtained by discretizing the *Laplace operator* with the same boundary conditions of the problem and subsequently adding some zeroth order term [2]. These preconditioners are called *shifted Laplace preconditioners.* This is developed by a discretization of the operator

$$M(\beta_1, \beta_2) = -\Delta - (\beta_1 - \iota\beta_2)k^2, \beta_1, \beta_2 \in \mathbb{R}$$

where β_1 and β_2 are real and imaginary shifts respectively. This class starts with a simple Laplace operator $M = \Delta$, which was used as preconditioner [9]. Later an additional real term *Shift* was added in the Laplace operator, making this preconditioner resembling more the Helmholtz operator but with an opposite sign as investigated in [10]. Later Laplace operator with imaginary shift was introduced in [3] and found to be more effective for the Helmholtz equation.
The optimal choice of real and imaginary shifts with restriction to be SPD (with positive

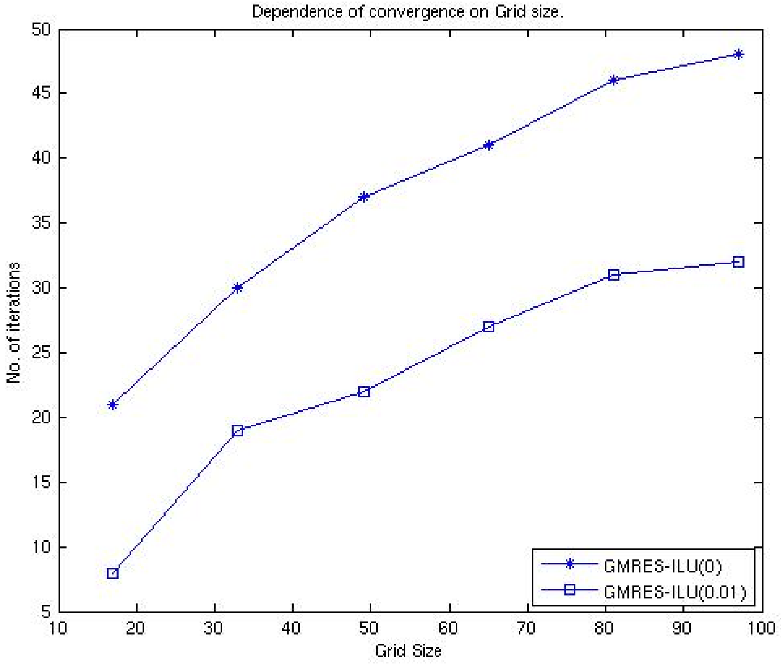

Figure 5: Dependency of number of iterations upon size of mesh for $k = 30$.

real parts of eigenvalues of preconditioner) is investigated in detail in [3] in the context of the condition number for Conjugate Gradient, first for a real shift and then generalized to complex shifts. $\beta_1 = -1$ for (only) real shifted Laplace preconditioner and $\beta_1 = 0$ and $\beta_2 = 1$ for shifted Laplace preconditioner are concluded as optimal choice. (For details, see [3]).

Although the above analysis for optimality of shifts is carried out in context of CG, but is also favorable for GMRES in terms of clustering of eigenvalues.

For the notation, $M(0,0)$ is simply discretized Laplace preconditioner without any shift, $M(-1,0)$ is preconditioning matrix with real shift 0, and $M(0,1)$ is complex shifted Laplace preconditioner.

Table 5 shows the numbers of GMRES iterations using three different shifted Laplace preconditioners. An observation is that for very small wave number k, $M(0,0)$ do work well, but for larger wave numbers, this is no more effective. For large wave number $M(0,1)$ is found satisfactory somehow. An additional result is taken into account for wave number $k = 100$ with mesh size $h = \frac{1}{160}$ $(kh = 0.625)$, the number of iterations taken by GMRES with $ILU(0.01)$ is 189 and the same with Shifted laplace preconditioner $M(0,1)$ is 83, indicating clearly that $M(0,1)$ is the best choice upto some hindrances to be discussed later.

The eigenvalue spectrum of the Helmholtz operator preconditioned with $M(0,1)$ is given

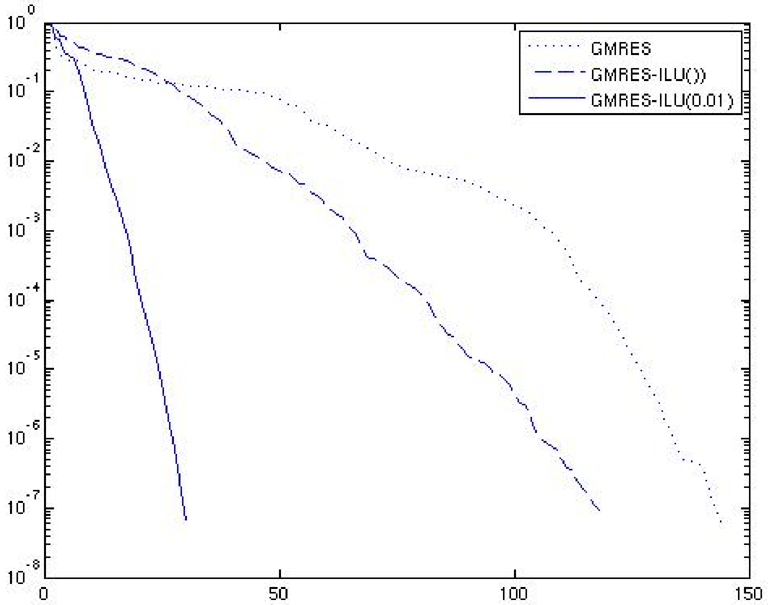

Figure 6: Compare of convergence history for GMRES without preconditioner and with ILU(0) and ILU(0.01) preconditioner for $k = 30$.

in Figure 7. Figure 7-a shows the spectrum for fixed wavenumber $k = 10$ with two different mesh sizes. The differences are small. Figure 7-b shows the spectrum with two different wavenumbers $k = 10$ and $k = 40$ with mesh size satisfying $kh = 0.625$. One observes that the eigenvalues are rushing to zero with increase in wavenumber. This is the drawback or hindrance of $M(0, 1)$ noted earlier. Some eigenvalues near to zero affect the convergence of GMRES badly. This is dealt in next section using *Deflation*.

Finally, the convergence history of GMRES with Laplace preconditioner $M(0, 0)$ and shifted Laplace preconditioners $M(-1, 0)$ and $M(0, 1)$ is presented in Figure 8. Also CPU time (in seconds) for direct solver is compared with the CPU times for GMRES and GMRES with different preconditioners in Table 6 and 7. Table 6 shows total time (problem construction time and solving time) where as the Table 7 shows the exclusive time for solver. The CPU time for GMRES preconditioned with both $ILU(0)$ and shifted Laplace preconditioners include the time of preconditioner construction and problem construction. GMRES with shifted Laplace preconditioner $M(0, 1)$ takes more time because a direct solver is used for the solution of preconditioner systems.

k	**Dim of A**	$M_h(0,0)$ Iterations	$M_h(-1,0)$ Iterations	$M_h(0,1)$ Iterations
10	289	9	12	10
20	1089	19	22	19
30	2401	37	38	30
40	4225	62	58	40
50	6561	96	84	51

Table 5: Number of iterations by GMRES with shifted Laplace preconditioners $M_h(0,0)$, $M_h(-1,0)$ and $M_h(0,1)$.

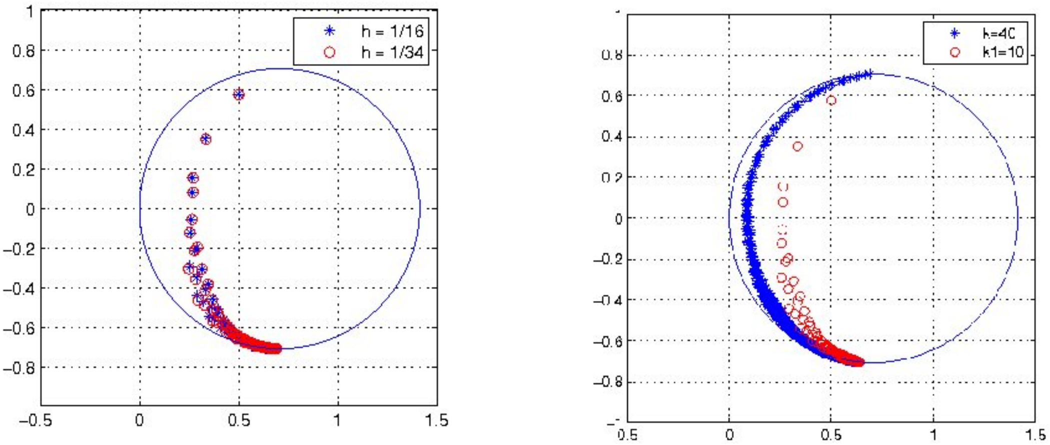

Figure 7: (a) Spectrum of Helmholtz operator preconditioned with shifted Laplace preconditioner $M(0,1)$ for different mesh sizes for $k = 10$ (b)Spectrum for different wave numbers $k = 10$ and $k = 40$

k	Direct Solver	GMRES	GMRES with ILU(0)	GMRES with $M(0,1)$
	Time in Seconds			
10	0.06	0.11	0.16	0.17
20	0.73	0.95	1.40	1.70
30	3.50	4.37	5.86	7.60
40	10.50	14.10	16.81	22.52
50	25.10	25.25	24.87	50.02

Table 6: Total time Problem construction time and time taken by different solvers.

k	Direct Solver	GMRES	GMRES with ILU(0)	GMRES with $M(0,1)$
	Time in Seconds			
10	0.0019	0.024	0.066	0.029
20	0.007	0.23	0.58	0.24
30	0.010	1.06	2.40	0.81
40	0.018	4.43	6.66	1.85
50	0.067	13.82	13.76	4.20

Table 7: Time taken by different solvers.

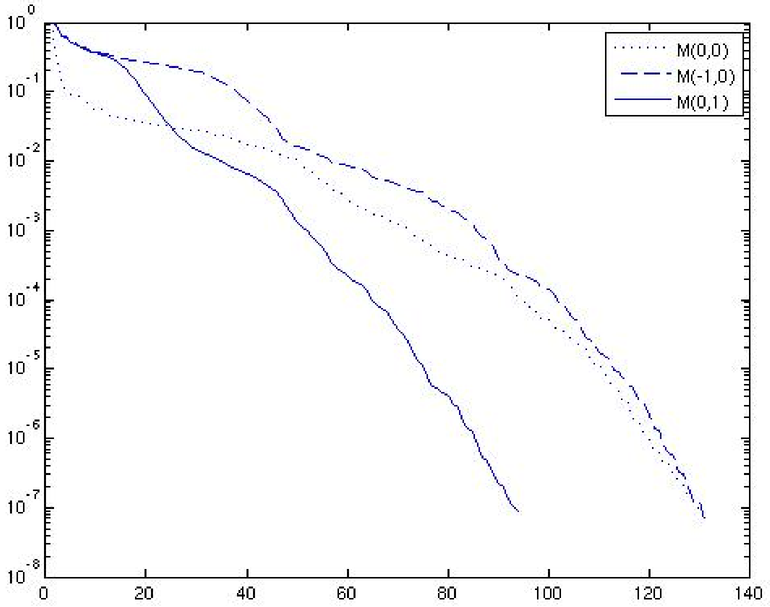

Figure 8: Comparison of the convergence history for GMRES with shifted Laplace Preconditioners $M_h(\beta_1, \beta_2)$ with different shifts for $k = 30$.

7 Deflation

The iterative solution of a linear system is typically adversely affected by a few unfavorable eigenvalues of the coefficient matrix. Deflation is a technique dealing with those undesired eigenvalues. Deflation for SPD systems is used in [11] and [12] with Conjugate Gradient to improve the condition number. Various ideas such as subdomain deflation are used in [12]. Later this idea was extended to non-symmetric systems in [13]. The basic idea is to deflate the smallest eigenvalues to zero by choosing eigenvectors or approximate Ritz vectors corresponding to those smallest eigenvalues as deflation vectors [11] and [12]. In [13], instead of deflating the small eigenvalues to zero, a deflating preconditioner is discussed which deflates the smallest eigenvalues to the maximum eigenvalue (in absolute value for a complex eigenvalues).

Defining deflation for any matrix $Z \in \mathbb{R}^{n\times k}$ of deflating vectors, deflation preconditioner is the projection defined as

$$P = I - AQ \qquad \text{with} \qquad Q = ZE^{-1}Z^T \qquad E = Z^TAZ$$

where E is called the Galerkin or coarse matrix and Z the matrix, whose columns span the deflation subspace, is chosen such that E is nonsingular. for A SPD, it is sufficient thta $Rank(Z) = k$. Further properties of deflation space for an arbitrary Z are elaborated in detail in [11] and [13].
By the splitting,

$$u = (I - P^T)u + P^Tu$$

one obtains

$$(I - P^T)u = (AQ)^Tu = Q^TA^Tu = QAu = Qb$$

where A is supposed to be SPD. We only need to compute P^Tu and for u we need to solve $PA\bar{u} = Pg$ by the Conjugate Gradient method and then solution is $u = Qb + P^T\bar{u}$.
In fact we aim to treat with unfavorable(smallest) eigenvalues of wellposed Helmholtz equation (Problem 2), which are causing slow convergence in iterative method. Here in this report, however we restrict to the SPD case and use CG and residual vectors as deflating vectors. The experiments are done with Problem 1 with different right-hand side vectors. In Problem 1, if there is an other source function along with g as typically appears in *Adjoint-based Optimized Methods* i.e

$$\bar{g} = 5x^3 - 5x^2y - 5xy^2 - 8x - 8y$$

then we have a linear system with two right-hand side vectors, $Au = g$ and $Au = \bar{g}$. The idea is to extract residuals from linear system $Au = \bar{g}$ and set those as deflation vectors i.e Z and then solve the linear system $Au = g$ by using those residuals Z as basis of deflation subspace.
In Figure 9-a, the convergence history of PCG and deflated PCG for $k = 10$ and $N = 20$ are shown. $ILU(0)$ is used as preconditioner. One observes that number of iterations is reduced after applying deflation to the system preconditioned with $ILU(0)$. It is also assured the that condition number of matrix PA is less than that of A i.e

$$\kappa(PA) = \frac{\max{(\lambda_{PA})}}{\min{(\lambda_{PA})}} = 21.972 \leq \frac{\max{(\lambda_A)}}{\max{(\lambda_A)}} = 27.400,$$

where $\min(\lambda_{PA})$ is the smallest non-zero eigenvalue of PA.
Figure 9-b shows the convergence history for wavenumber $k = 20$ and $N = 36$. One observes that the convergence of PCG and deflated PCG coincide and the effect on the condition number is very small. This is because we observe more clustered spectrum with increasing wave number.
For zero wavenumber $k = 0$, the spectrum of system is more scattered than with some nonzero wavenumber. In Figure 10-a, the convergence history of deflated CG is compared with the convergence history of CG for $k = 0$. We observe a good effect of deflation for $k = 0$, as compare to $k = 10$ and $k = 20$. Further we observe deflation also works for the preconditioned scheme. Using $ILU(0)$ as preconditioner,the convergence history of PCG and deflated PCG for wavenumber $k = 0$ is shown in Figure 10-b. Further the condition number for $k = 0$ and $N = 20$

$$\kappa(PM^{-1}A) = 64.300 \ll \kappa(M^{-1}A) = 161.44$$

is also reduced remarkably.

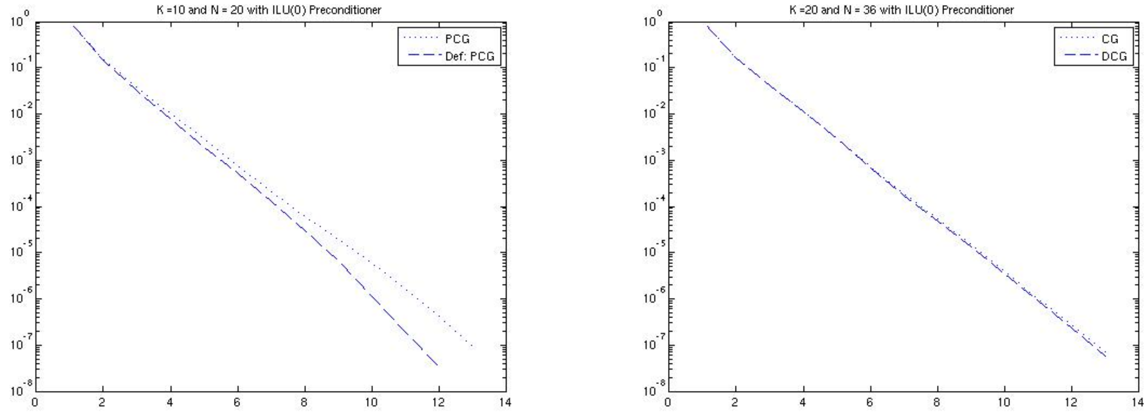

Figure 9: (a) Convergence history for PCG and deflated PCG for $N = 20$ and $k = 10$ (b) Convergence history of PCG and deflated PCG for $N = 36$ and $k = 20$. ILU is used as Preconditioner.

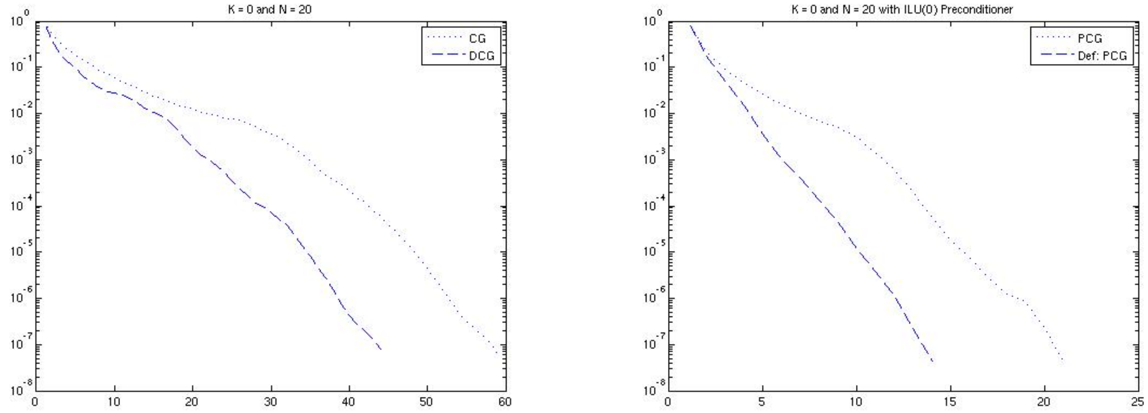

Figure 10: (a) Convergence History for CG and deflated CG for $k = 0$ and $N = 20$ (b) Convergence history of PCG and deflated PCG for $k = 0$ and $N = 20$. ILU is used as Preconditioner

8 Conclusion

For linear systems obtained from a discretization of the Helmholtz equation by the finite difference method, experiments are done with GMRES, GMRES preconditioned by ILU and shifted Laplace preconditioner. The comparison of preconditioners for the Helmholtz equation is carried out. For a small wavenumber k, ILU preconditioners with GMRES work well, but they are no more of use for larger k. For larger k, shifted Laplace preconditioner performs better than ILU. The preconditioned coefficient matrix however still has some eigenvalues near zero, causing GMRES to converge slowly. This problem appears to be more serious for increasing k, but can be fixed by deflation, by taking approximations to eigenvectors as deflation vectors. Approximated eigenvectors as deflation vectors for Problem 1 along with some preconditioned iterative solver will lead to an efficient iterative scheme for Problem 1.

References

[1] Y.A. Erlangga, C. Vuik, and C.W. Oosterlee. On a class of preconditioners for solving the helmholtz equation. *Appl. Numer. Math.*, 50(3-4):409–425, 2004.

[2] Y.A. Erlangga, C. Vuik, and C. Oosterlee. On a class of preconditionesr for solving the helmholtz equation. report 03-01,. Technical report, DIAM Delft University of Technology, Delft the Netherlands, 2003.

[3] Y.A.Erlangga. *A robust and effecient iterative method for numerical solution of Helmholtz equation*. PhD thesis, DIAM TU Delft, 2005.

[4] C. Vuik, P.van Beek, F. Vermolen, and J. van Kan. *Numerical Methods for Ordinary differential equations*. VSSD, 2007.

[5] M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, December 1952.

[6] C.C.Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Num.Analysis.*, 12:617–624, 1975.

[7] Y. Saad and M.H Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[8] J.J. Dongarra, J.Du Croz, S. Hammarling, and R.J. Hanson. An extended set of fortran basic linear algebra subprograms. *ACM Trans. Math. Softw.*, 14(1):1–17, 1988.

[9] C.I. Goldstein A. Bayliss and E. Turkel. An iterative method for the helmholtz equation. *Journal of Computational Physics*, 49:443 – 457, 1983.

[10] A.L. Laird. Preconditioned iterative solution of 2d helmholtz equation. Technical report, St. Hugh's college, 2001.

[11] J. M. Tang. *Two Level Preconditioned conjugate Gradient Methods with Applications to Bubbly Flow Problems*. PhD thesis, DIAM, TU Delft, 2008.

[12] J. Frank and C. Vuik. On the construction of deflation-based preconditioners. *SIAM J. Sci. Comp.*, 23:442–462, 2001.

[13] Y.A. Erlangga and R. Nabben. Multilevel projection-based nested krylov iteration for boundary value problems. *SIAM J. Sci. Comput.*, 30(3):1572–1595, 2008.

A Eigenvalues and Eigenvalues

An eigenvalue problem is finding a pair (u,λ) of eigenfunctions and eigenvalues respectively. For Problem (2)

$$-\Delta u + k^2 u = g$$

the corresponding eigenvalue problem is

$$-\Delta u + k^2 u = \lambda u$$

First we find eigenvalues and functions of Laplace operator $-\Delta$ then it is straightforward to find eigenvalues for $-\Delta + k^2 I$. Hence first we solve

$$-\Delta u = \lambda u \tag{A-1}$$

where λ is an eigenvalue and u is the corresponding eigenfunction. By separation of variables and substituting

$$u(x, y) = X(x)Y(y)$$

in equation $(A1)$, we get

$$\frac{-X^{''}(x)}{X(x)} = \frac{Y^{''}(y)}{Y(y)} + \lambda$$

Here two different function of two different variables are equal to each other. This implies that this can only be true if both are constant functions i.e $\frac{-X^{''}(x)}{X(x)} = \mu$ and $\frac{Y^{''}(y)}{Y(y)} + \lambda = \mu$ or $\frac{Y^{''}(y)}{Y(y)} = -\mu_1$ where $\lambda = \mu + \mu_1$. Solving these two ODEs with boundary conditions, we have

$$X(x) = A\sin(\sqrt{\mu}x) + B\cos(\sqrt{\mu}x)$$

and

$$Y(y) = \bar{A}\sin(\sqrt{\mu}x) + \bar{B}\cos(\sqrt{\mu}x)$$

For the case $\lambda \leq 0$, it only gives the trivial solution i-e $u(x, y) = 0$ and since we are interested in nontrivial solutions, these values are skipped. Using boundary conditions and collecting results, it reduces to

$$u(x, y) = \sin(\mu_1 x)\sin(\mu_2 y) \tag{A-2}$$

where $\mu_1 = (l_1\pi)^2$ and $\mu_2 = (l_2\pi)^2$ with l_1, $l_2 \in \mathbb{Z}$. Now the eigenvalues are $\lambda = \mu_1 + \mu_2 = (l_1\pi)^2 + (l_2\pi)^2$ and above u are eigenfunction. Now in discrete phenomenon, first for notation, u at node (i, j) on the grid of discretization is noted as $u_{i,j} = u(x_i, y_j)$.
$-\Delta u = \lambda u$ is then written as

$$\bar{A}u = \lambda u$$

From Section 3, the discretization on grid of size $h = \frac{1}{N}$ of $-\Delta$ leads to

$$-u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} + 4u_{i,j}$$

and substituting eigenfunctions u from equation $(A2)$, and after simplifying using trigonometric identities we have

$$-\Delta u = [4 - 2\cos(l_1\pi h) - 2\cos(l_2\pi h)]\sin(l_1\pi ih)\sin(l_2\pi jh)$$

from where, it is concluded that

$$4 - 2\cos(l_1\pi h) - 2\cos(l_2\pi h)$$

are $(N-1)^2$ eigenvalues for variation of l_1 and l_2 between 1 and $(N-1)$ and their corresponding eigenvectors are

$$u(x,y) = \sin(l_1\pi i h)\sin(l_2\pi j h)$$

For eigenvalues of $-\Delta u - k^2 u$, the discrete formulation of eigenvalue problem will be $(\bar{A} + (-k)I)u = \lambda u$ where I is identity matrix and $(-k)I$ is a diagonal matrix with eigenvalues as its diagonal entries and afterwards, it is straightforward that eigenvalues of $-\Delta - k^2 I$ are

$$4 - 2\cos(l_1\pi h) - 2\cos(l_2\pi h) - k^2$$

.