CgWaveHoltz: A Composite Grid Helmholtz Solver using the WaveHoltz Algorithm

D.A. Appelö[a], W. D. Henshaw[b,2,*]

[a] *University of Colorado, Boulder.*
[b] *Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA*

Abstract

Here are notes on CgWaveHoltz Helmholtz solver that is based on solving the Helmholtz equation using the wave equation solver CgWave.

Keywords: Maxwell

Contents

*Department of Mathematical Sciences, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA.
Email address: henshw@rpi.edu (W. D. Henshaw)

[1]This work was performed under DOE contracts from the ASCR Applied Math Program.
[2]Research supported by the National Science Foundation under grant DMS-1519934.
[3]Research supported by a U.S. Presidential Early Career Award for Scientists and Engineers.

1. Introduction

Here are some notes on the CgWaveHoltz solver based on WaveHoltz scheme [1] of Appelö et.al. The WaveHoltz algorithm solves the time-harmonic wave equation (Helmholtz equation) using a time-dependent wave equation solver. CgWaveHoltz uses the CgWave wave equation solver.

Things to do:

1. Compatibility boundary conditions for direct Helmholtz solver.
2. Sixth and eight-order accurate direct Helmholtz solver.

2. WaveHoltz algorithm

The Helmholtz solution $u(\mathbf{x},t)$ satisfies,

$$-\omega^2 u = c^2\Delta u + f(\mathbf{x}), \qquad \text{for } \mathbf{x}\in\Omega, \tag{1a}$$
$$Bu = g(\mathbf{x}) \qquad \text{for } \mathbf{x}\in\partial\Omega, \tag{1b}$$

Here $Bw = g$ denotes some appropriate boundary conditions.

Let $w(\mathbf{x},t)$ be a solution to the wave equation in second-order form with a time-harmonic forcing,

$$\partial_t^2 w = c^2\Delta w + f(\mathbf{x})\, e^{i\omega t}, \qquad \text{for } \mathbf{x}\in\Omega,\ t>0, \tag{2a}$$
$$Bw = g(\mathbf{x})\, e^{i\omega t} \qquad \text{for } \mathbf{x}\in\partial\Omega,\ t>0, \tag{2b}$$
$$w(x,0) = v_0(\mathbf{x}), \quad \partial_t w(\mathbf{x},0) = v_1(\mathbf{x}), \qquad \text{for } \mathbf{x}\in\partial\Omega. \tag{2c}$$

and initial conditions for u and u_t given by $v_0(\mathbf{x})$ and $v_1(\mathbf{x})$.

The WaveHoltz algorithm [1] defines an affine operator Π that acts on the initial conditions $\mathbf{v} = [v_0,\, v_1]^T$ by solving the wave equation (2) with initial conditions $\mathbf{v}$ over a time period $T = 2\pi/\omega$ and then forming the time integral

$$\Pi\mathbf{v} = \frac{2}{T}\int_0^T \left(\cos(\omega t) - \frac{1}{4}\right)\begin{bmatrix} w(\mathbf{x},t) \\ w_t(\mathbf{x},t)\end{bmatrix} dt. \tag{3}$$

Note that w implicitly depends on v and f, $w = w(\mathbf{x},t;v,f)$. Note that Π is an affine operator of the form

$$\Pi v = Lv + b = (I - A)v + b, \tag{4}$$

where L and A are a linear operators. The solution to the Helmholtz problem (1) is found as the solution to the fixed point iteration

$$\mathbf{v}^n = \Pi\mathbf{v}^{n-1}, \tag{5}$$

where $\mathbf{v}^0 = 0$ (or some other initial guess).

3. Using Krylov methods to accelerate the WaveHoltz algorithm

The fixed point iteration (5) can be accelerated with a Krylov method such as GMRES. A Krylov method solves a linear system

$$Ax = b. \tag{6}$$

At the most basic level Krylov methods only require a function to evaluate b and a function to compute A times a vector. For the fixed point iteration (5), b is defined as the application of Π to the vector $\mathbf{v}^0 = \mathbf{0}$ (zero initial conditions),

$$b \stackrel{\text{def}}{=} \Pi\,\mathbf{0}, \tag{7}$$

while the application of A to an iterate $\mathbf{v}^n$ is defined as

$$A\mathbf{v}^n = \mathbf{v}^n - \Pi\mathbf{v}^n + b. \tag{8}$$

Formula (8) looks a bit funny at first glance but this is the correct definition.

We use the Krylov solvers from PETSc [2] to solve the fixed point iteration (5). PETSc has a "matrix-free" option that allows one to provide function to compute a matrix-vector product. PETSc has many Krylov solvers suh as CG, GMRES, bi-CG-stab, etc.

4. Augmented GMRES (AUGMRES)

The augmented GMRES algorithm can be used to accelerate the convergence of the standard GMRES scheme by adding addition vectors to the usual Krylov space. These additional vectors can be accurate or approximate eigenvectors. The approach can be considered a a pre-conditioner. Unlike the deflation method described in Section 7 this approach does not require a discrete inner product. In addition the discrete WaveHoltz solution will converge to the discretized direct Helmholtz solution.

Algorithm 1 Augmented GMRES algorithm, from Baglama and Reichel [3].

function $\mathbf{x}_j$ =AUGMENTEDGMRES(A,$\mathbf{b}$,W,p, j)
  Input $W \in \mathbb{R}^{n\times p}$: matrix of p augmented vectors
  Input p, j : number of augmented vectors and number of GMRES iterations
  $\mathbf{v}_{p+1} = \mathbf{b}$ ▷ Will hold normalized $(I - V_p V_p^T)\mathbf{b}$
  if $p > 0$ **then**
    // $V_{1:p}$ holds columns $1:p$ of V.
    $AW = V_{1:p}\, H_{1:p,1:p}$ ▷ Compute QR factorization of AW.
    for $i = 1:2$ **do** ▷ Project twice if re-orthogonalization needed
      $\mathbf{v}_{p+1} = \mathbf{v}_{p+1} - V_p(V_p^T\mathbf{v}_{p+1})$ ▷ $\mathbf{v}_{p+1} = (I - V_pV_p^T)\mathbf{b}$
    end for
  end if
  $\mathbf{v}_{p+1} = \mathbf{v}_{p+1}/\|\mathbf{v}_{p+1}\|$ ▷ $\mathbf{v}_{p+1}$ is column $p+1$ of V

  for $k = p+1, p+2, \ldots, p+j$ **do** ▷ Arnoldi iterations
    $\mathbf{v}_{k+1} = A\mathbf{v}_k$
    for $i = 1:k$ **do** ▷ Make $\mathbf{v}_{k+1}$ orthogonal to $\mathbf{v}_i$
      $h_{ik} = \mathbf{v}_i^T\mathbf{v}_{k+1}$; $\mathbf{v}_{k+1} = \mathbf{v}_{k+1} - h_{ik}\mathbf{v}_i$
    end for
    for $i = 1:k$ **do** ▷ Re-orthogonalization (if needed)
      $\alpha = \mathbf{v}_i^T\mathbf{v}_{k+1}$; $\mathbf{v}_{k+1} = \mathbf{v}_{k+1} - \alpha\mathbf{v}_i$; $h_{ik} = h_{ik} + \alpha$
    end for
    $h_{k+1,k} = \|\mathbf{v}_{k+1}\|$; $\mathbf{v}_{k+1} = \mathbf{v}_{k+1}/\|\mathbf{v}_{k+1}\|$ ▷ Column $k+1$ of V
  end for
  $\mathbf{y} = \arg\min_{\mathbf{y}\in\mathbb{R}^{p+j}} \|V_{p+j+1}^T\mathbf{b} - H\mathbf{y}\|$ ▷ Solve a least squares problem
  $\mathbf{x}_j = \left[W \mid V_{p+1:p+j}\right]\mathbf{y}$ ▷ Approximate solution
end function

The augmented GMRES algorithm we give here is from "Augmented GMRES-type methods", by James Baglama and Lothar Reichel [3]. Suppose we which to solve the linear system,

$$A\mathbf{x} = \mathbf{b}, \qquad A \in \mathbb{R}^{n\times n}, \quad \mathbf{x}, \mathbf{b} \in \mathbb{R}^n. \tag{9}$$

The standard GMRES method finds an approximate solution, $\mathbf{x}_j$ in the Krylov space generated by b and A

$$\mathbf{x}_j \in \mathcal{K}_j(A, b), \qquad \mathcal{K}_j(A, b) \stackrel{\text{def}}{=} \text{span}(\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{j-1}\mathbf{b}), \tag{10}$$

that minimizes the 2-norm of the residual,

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}. \tag{11}$$

Let the columns of $W \in \mathbb{R}^{n\times p}$

$$W = \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \ldots & \mathbf{w}_p \\ | & | & & | \end{bmatrix}, \tag{12}$$

be the augmented vectors that span the space $\mathcal{W}$. The augmented GMRES method will find a solution $\mathbf{x}_j$ in the space

$$\mathbf{x}_j \in \mathcal{W} \cup \mathcal{K}_j(A, b) = \text{span}(\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_p,\ b, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{j-1}\mathbf{b}), \tag{13}$$

that minimizes the two-norm of the residual.

The augmented-GMRES algorithm is given in 4. The method starts by forming the reduced QR factorization of AW

$$AW = V_{1:p}\, H_{1:p,1:p}, \qquad V_{1:p} \in \mathbb{R}^{n\times p}, \quad H_{1:p,1:p} \in \mathbb{R}^{p\times p}, \tag{14}$$

$$V_{1:p} = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_p \\ | & | & & | \end{bmatrix}, \qquad H_{1:p,1:p} = \begin{bmatrix} h_{11} & h_{12} & \ldots & h_{1p} \\ & h_{22} & \ldots & h_{2p} \\ & & \ddots & \vdots \\ & & & h_{pp} \end{bmatrix}, \tag{15}$$

Here the notation $V_{1:p}$ indicates that this matrix holds columns 1 to p of matrix V. Let $\mathbf{b}'$ be the projected value of $\mathbf{b}$ that removes the components of $\mathbf{b}$ along the augmented vectors $\mathbf{w}_i$,

$$\mathbf{b}' = [I - V_{1:p}V_{1:p}^T]\,\mathbf{b} \tag{16}$$

Column $\mathbf{v}_{p+1}$ of V is the $\mathbf{b}'$ normalized by its length (if this length is zero then we solve for $\mathbf{x}$ from ...)

$$\mathbf{v}_{p+1} = \mathbf{b}'/\|\mathbf{b}'\|. \tag{17}$$

The Arnolid algorithm is now used to generate j additional columns $\mathbf{v}_{p+1}, \ldots, \mathbf{v}_{p+j+1}$, and the remaining entries in the rectangular matrix $H \in \mathbb{R}^{p+j+1\times p+j}$

$$H = \begin{bmatrix} h_{11} & h_{1,2} & \ldots & h_{1,p} & h_{1,p+1} & h_{1,p+2} & \ldots & h_{1,p+j+1} \\ & h_{2,2} & \ldots & h_{2,p} & h_{2,p+1} & h_{2,p+2} & \ldots & h_{2,p+j+1} \\ & & \vdots & & \vdots & \vdots & & \vdots \\ & & & & \vdots & \vdots & & \vdots \\ & & & h_{pp} & \vdots & \vdots & & \vdots \\ & & & & h_{p+1,p+1} & h_{p+1p+2} & \ldots & h_{p+1,p+j+1} \\ & & & & h_{p+2,p+1} & h_{p+2,p+2} & \ldots & h_{p+2,p+j+1} \\ & & & & & h_{p+2,p+3} & \ldots & h_{p+3,p+j+1} \\ & & & & & & \ddots & \vdots \\ & & & & & & & h_{p+j+1,p+j} \end{bmatrix} \tag{18}$$

such that

$$A\left[W \mid V_{p+1:p+j}\right] = \left[AW \mid AV_{p+1:p+j}\right] = \left[V_{1:p}\, H_{1:p,1:p} \mid AV_{p+1:p+j}\right] \tag{19}$$

$$= V_{1:p+j+1}H \tag{20}$$

The GMRES solution will be a linear combination of the columns of W and the columns of $V_{p+1:p+j}$ (note that we use W, not $V_{1:p}$)

$$\mathbf{x}_j = \left[W \mid V_{p+1:p+j}\right]\mathbf{y}. \tag{21}$$

Multiplying by A gives

$$A\mathbf{x}_j = \left[AW \mid AV_{p+1:p+j}\right]\mathbf{y} \tag{22}$$

$$= V_{1:p+j+1}H\mathbf{y} \tag{23}$$

The residual can be written as

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}_j = \mathbf{b} - V_{1:p+j+1} H \mathbf{y} \tag{24}$$

and $\mathbf{y}$ solves the least squares problem

$$\mathbf{y} = \underset{\mathbf{y}\in\mathbb{R}^{p+j}}{\arg\min} \|\mathbf{b} - V_{1:p+j+1} H \mathbf{y}\| \tag{25}$$

or multiplying through by $V_{1:p+j+1}^T$

$$\mathbf{y} = \underset{\mathbf{y}\in\mathbb{R}^{p+j}}{\arg\min} \|V_{1:p+j+1}^T \mathbf{b} - H \mathbf{y}\| \tag{26}$$

In standard GMRES $V^T\mathbf{b} = \|\mathbf{b}\|\mathbf{e}_1$, but this is not true for AUGMRES.

Algorithm 2 gives the AUGMRES algorithm that uses Givens rotations to solve the least squares problem. By incrementally updating the QR factorization and right-hand-side $\mathbf{g}$ with Givens rotations, the 2-norm residual can be found without having to solve for $\mathbf{x}$. This 2-norm residual is used in a stopping criteria.

4.1. Augmented GMRES results

To test the augmented GMRES algorithm we consider solving the boundary value problem

$$-\partial_x^2 u = f(x), \qquad x \in (a, b), \tag{27}$$

$$u(a) = g_a, \quad u(b) = g_b \tag{28}$$

discretized with

$$-D_{+x}D_{-x}U_i = f(x_i), \qquad i = 1, 2, \ldots, N-1, \tag{29a}$$

$$\frac{1}{\Delta x^2} U_0 = \frac{1}{\Delta x^2} g_a, \quad \frac{1}{\Delta x^2} U_N = \frac{1}{\Delta x^2} g_b \tag{29b}$$

where $\Delta x = (b-a)/N$ and $x_i = a + i\Delta x$. The discrete problem (29) leads to one of two matrix problems,

$$A\mathbf{U} = \mathbf{F}, \qquad \text{boundary conditions eliminated,} \tag{30}$$

$$\tilde{A}\tilde{\mathbf{U}} = \tilde{\mathbf{F}}, \qquad \text{boundary conditions included.} \tag{31}$$

We wish to determine whether the convergence depends on whether the boundary conditions are eliminated from the matrix. It turns out that, if the boundary conditions are included then one should scale the boundary conditions as in (29b). This causes the two *spurious* eigenvalues of $\tilde{A}$ (since we have two additional equations in $\tilde{A}$) to be large (they are actually both $\lambda_{bc} = 1/\Delta x^2$). Without scaling the spurious eigenvalues are $\lambda_{bc} = 1$ and these become the smallest eigenvalues: this causes problems when deflating the eigenvectors corresponding to the smallest eigenvalues.

Figure 1 shows convergence results for standard GMRES and augmented GMRES. Eigenvectors corresponding to the smallest (in magnitude) eigenvalues are deflated. The average convergence rate in the figure is computed as the n-th root of the first residual divided by the residual at step n.

If the matrix A is diagonalizable, $A = Z\Lambda Z^{-1}$, with real positive eigenvalues, then the residual at step n has the bound (Morgan 1995 [4])

$$\frac{\|r_n\|}{\|b\|} \le 2\|Z\|\|Z^{-1}\| \left(1 - \frac{2}{1+\sqrt{\kappa}}\right)^n \tag{32}$$

where κ (which is not the normal condition number if A is not symmetric) is

$$\kappa \overset{\text{def}}{=} \frac{\lambda_{\max}}{\lambda_1}, \tag{33}$$

Algorithm 2 Augmented GMRES algorithm with Givens Rotations.

function $\mathbf{x}_j$ =AUGMENTEDGMRES(A, $\mathbf{b}$, $\mathbf{x}_0$, W, p, maxit, tol)
  Input A,$\mathbf{b}$,$\mathbf{x}_0$: matrix, right-hand-side, and initial guess
  Input $W \in \mathbb{R}^{n\times p}$: matrix of p augmented vectors
  Input p, maxit : number of augmented vectors and max-number of GMRES iterations
  Input tol: convergence tolerance, check for 2-norm residual less than tol $\|\mathbf{b}\|$
  $\mathbf{b}_0 = \mathbf{b} - A\mathbf{x}_0$ ▷ Adjusted $\mathbf{b}$ for initial guess $\mathbf{x}_0$
  $\mathbf{v}_{p+1} = \mathbf{b}_0$ ▷ Will hold normalized $(I - V_p V_p^T)\mathbf{b}_0$
  if $p > 0$ **then**
    // $V_{1:p}$ holds columns $1:p$ of V.
    $AW = V_{1:p}\, H_{1:p,1:p}$ ▷ Compute QR factorization of AW.
    for $i = 1:2$ **do** ▷ Project twice if re-orthogonalization needed
      $\mathbf{v}_{p+1} = \mathbf{v}_{p+1} - V_p(V_p^T\mathbf{v}_{p+1})$ ▷ $\mathbf{v}_{p+1} = (I - V_pV_p^T)\mathbf{b}$
    end for
  end if
  $\mathbf{v}_{p+1} = \mathbf{v}_{p+1}/\|\mathbf{v}_{p+1}\|$ ▷ $\mathbf{v}_{p+1}$ is column $p+1$ of V
  $Q = I_{1:m,1:p+1}$ ▷ Holds product of Givens rotations
  for $k = 1:p+1$ **do**
    $g(k) = V(:,k)^T\mathbf{b}_0$ ▷ $\mathbf{g}$ is the RHS for the least squares solve
  end for

  for $k = p+1, p+2, \ldots, p+\text{maxit}$ **do** ▷ Arnoldi iterations
    $\mathbf{v}_{k+1} = A\mathbf{v}_k$
    for $i = 1:k$ **do** ▷ Make $\mathbf{v}_{k+1}$ orthogonal to $\mathbf{v}_i$
      $h_{ik} = \mathbf{v}_i^T\mathbf{v}_{k+1}$; $\mathbf{v}_{k+1} = \mathbf{v}_{k+1} - h_{ik}\mathbf{v}_i$
    end for
    for $i = 1:k$ **do** ▷ Re-orthogonalization (if needed)
      $\alpha = \mathbf{v}_i^T\mathbf{v}_{k+1}$; $\mathbf{v}_{k+1} = \mathbf{v}_{k+1} - \alpha\mathbf{v}_i$; $h_{ik} = h_{ik} + \alpha$
    end for
    $h_{k+1,k} = \|\mathbf{v}_{k+1}\|$; $\mathbf{v}_{k+1} = \mathbf{v}_{k+1}/\|\mathbf{v}_{k+1}\|$ ▷ Column $k+1$ of V

    // Apply Givens to H (part of solving final least squares problem):
    $H(1:k,k) = Q(1:k,1:k)\,H(1:k,k)$ ▷ Apply previous rotations to new column
    $\rho = H(k,k)$; $H(k,k) = \sqrt{\rho^2 + H(k+1,k)^2}$;
    $c = \rho/H(k,k)$; $s = H(k+1,k)/H(k,k)$; $H(k+1,k) = 0$;
    // Apply Givens rotation to Q
    $Q(k+1,:) = -s\,Q(k,:)$; $Q(k,:) = c\,Q(k,:)$; $Q(k+1,k+1) = c$; $Q(k,k+1) = s$;
    // Apply Givens rotation to RHS g
    $g(k+1) = -s\,(k,1)$; $g(k) = c\,g(k)$; ▷ $g(k+1)$ holds the current 2-norm residual
    if $|g(k+1)| < \text{tol}\,\|\mathbf{b}\|$ **then** break; **end if**

  end for
  // Solve $\mathbf{y} = \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|V_{1:k+1}^T\mathbf{b} - \bar{H}\mathbf{y}\|$
  $H_{1:k,1:k}\,\mathbf{y} = \mathbf{g}_{1:k}$ ▷ Solve the triangular square system
  $\mathbf{x} = \mathbf{x}_0 + \left[W \mid V_{p+1:k}\right]\mathbf{y}$ ▷ Approximate solution
end function

assuming the eigenvalues are ordered from smallest to largest. When the k smallest eigenvectors are deflated, the bound in (32) holds but with κ replaced by the *effective condition number*,

$$\kappa_e \stackrel{\text{def}}{=} \frac{\lambda_{\max}}{\lambda_{1+k}}. \tag{34}$$

For the model problem (with $a = 0$, $b = 1$) the discrete eigenvalues and eigenvectors of A are

$$\lambda_j = 4\frac{\sin^2(j\pi\Delta x/2)}{\Delta x^2}, \qquad j = 1, 2, \ldots, N-1 \tag{35}$$

$$\phi_{j,i} = \sin(j\pi x_i) \tag{36}$$

Then

$$\lambda_1 \approx \pi^2, \tag{37}$$

$$\lambda_{\max} \approx \frac{4}{\Delta x^2}. \tag{38}$$

and $\kappa \approx 4/(\pi^2\Delta x^2)$. The expected convergence rate for GMRES is thus

$$\text{ACR} \approx 1 - \frac{2}{1+\sqrt{\kappa}} \approx 1 - \frac{2}{\sqrt{\kappa}} = 1 - \pi\Delta x \tag{39}$$

If we augment with k eigenvectors then the smallest eigenvalue for the effective κ_e is

$$\lambda_{k+1} \approx (k+1)^2\pi^2 \tag{40}$$

and check me

$$\text{ACR}_{\text{aug}} \approx 1 - (k+1)\pi \tag{41}$$

and the convergence rate will be approximately $k+1$ times faster.

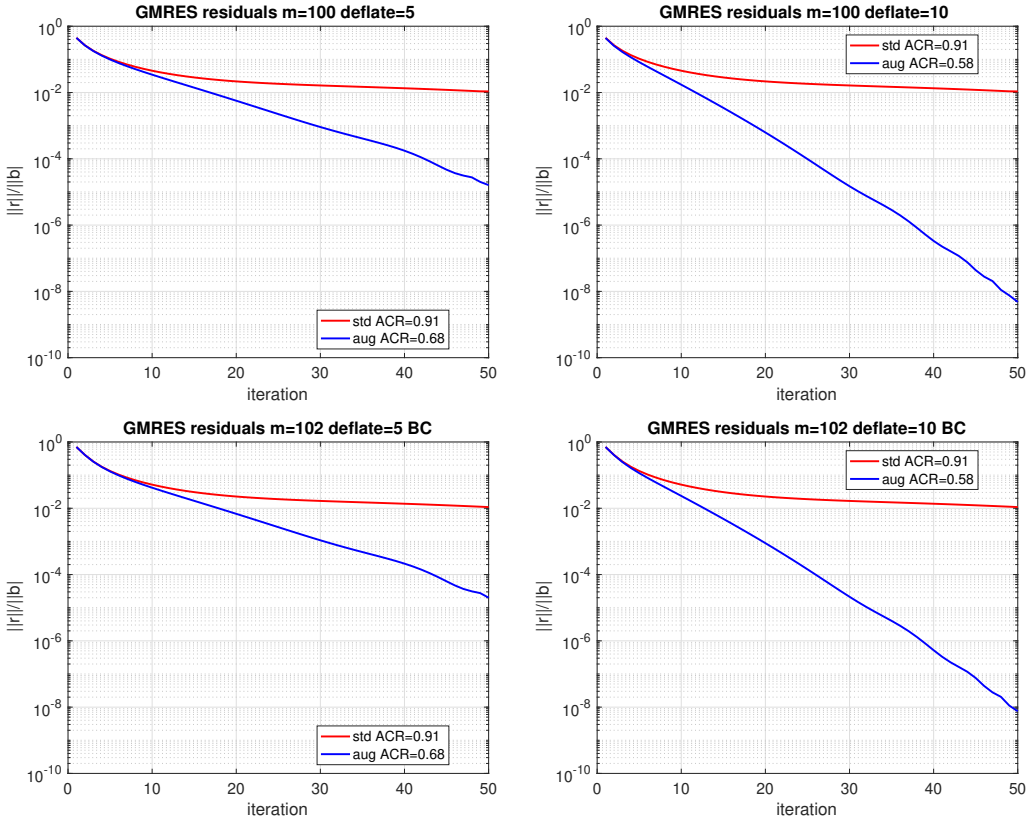

Figure 1: Augmented GMRES: model problem. Top row: boundary conditions eliminated. Bottom row: scaled boundary conditions included in the matrix. Left column: deflate 5 eigenvectors. Right column: deflate 10 eigenvectors.

5. Deflation

6. Deflation by directly removing and adding components along given eigenvectors

See the overHoltz paper for a direct algorithm to deflate the solution using known eigenpairs.

7. Deflation in CgWaveHoltz using augmented GMRES (AGMRES)

The augmented GMRES algorithm described in Section 4 can be used to perform deflation. This approach to deflation has several advantages. One advantage is that no discrete inner product is needed. A second advantage is that the eigenvectors used for deflation can be approximate. A third advantage is that the WaveHoltz algorithm will converge to the discrete solution of the direct Helmholtz solution (DHS).

Figure 2 compares results (Matlab code waveHoltz.m) using augmented GMRES (AGMRES) to direct removal of the eigenvectors. Deflation with AGMRES is seen to converge in a similar manner to direct deflation with GMRES. Note that for AGMRES the first residual in the plot corresponds to the first real Arnoldi step (after the 4 eigenvectors have been added to the basis). This explains why the first residual for AGMRES is smaller than that for GMRES.

Figure 3 shows results from CgWaveHoltz. The AGMRES results are closely following GMRES+Deflation. For the disk, we see that AGMRES converges to the DHS solution:

```
solveHelmholtz: max-res=9.763e-09 (Augmented-GMRES, numIts=31, CR= 0.52 deflate=13)
solveHelmholtz: rel-max-diff=3.43e-10, |v|_max= 7.51e-01 (between WaveHoltz and Direct Helmholtz solution, all frequencies)
```

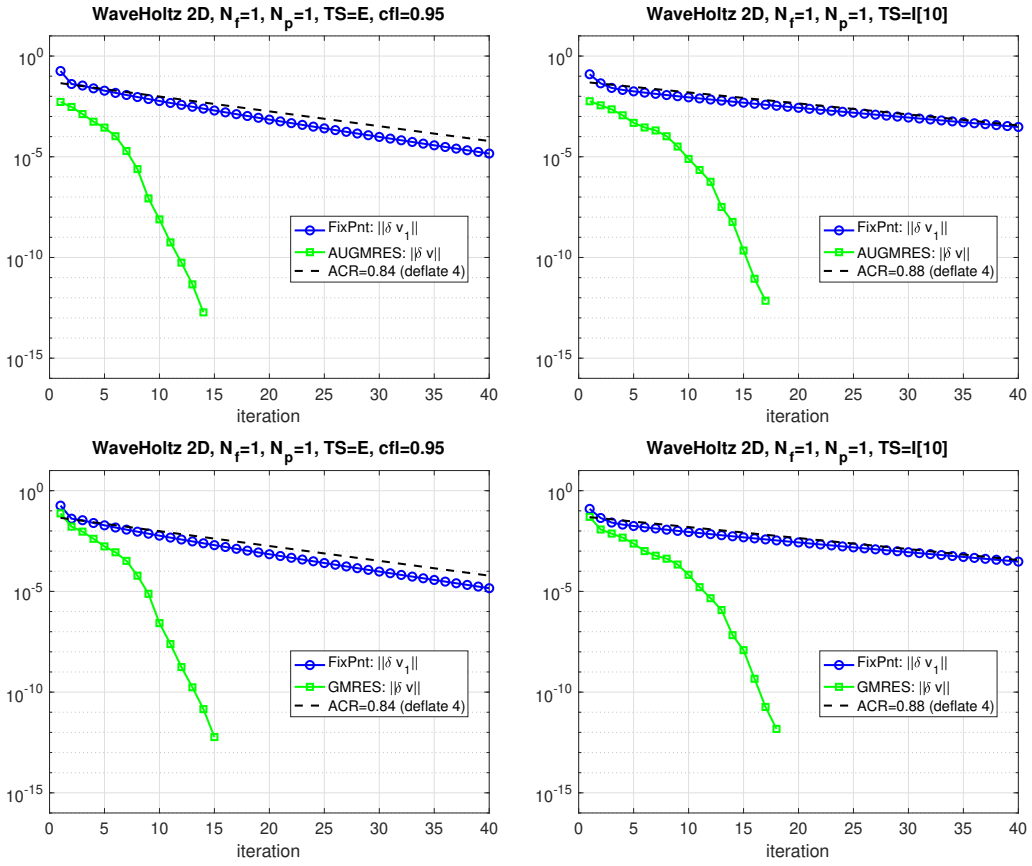

Figure 2: Deflation with augmented GMRES (top) compared to direct deflation (bottom). Left column: explicit time-stepping. Right column implicit time-stepping.

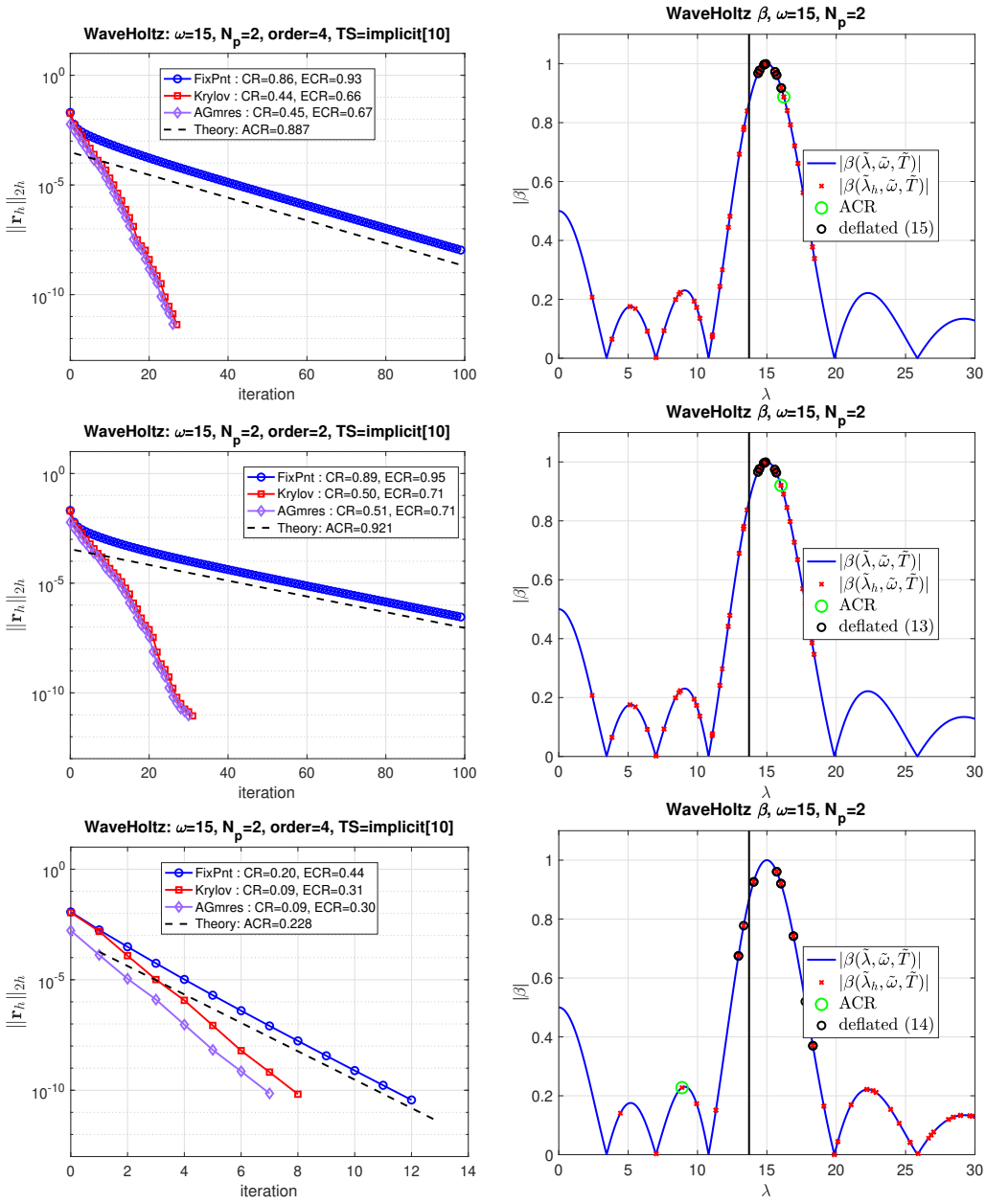

Figure 3: Comparing GMRES to AGMRES. Top: disk G8 order 4. Middle: disk G8 order 2. Bottom square G64 order 4.

8. Numerical Results

Here we present some numerical results.

8.1. Exact trigonometric Helmholtz solution

An exact solution to the Helmholtz equation on a square domain $[0,1]^2$ is given by the trigonometric function

$$u_e(\mathbf{x},t) \stackrel{\text{def}}{=} \sin(k_x x)\,\sin(k_y y)\,\cos(\omega t),$$

where the forcing function is

$$f(\mathbf{x},t) = \Big(-\omega^2 + c^2(k_x^2+k_y^2)\Big) u_e(\mathbf{x},t)$$

This solution is also used on other domains in which case we allow for in-homogeneous BCs.

FD22s Errors and convergence of WaveHoltz solutions on a square.

grid	N	u	r
square8	1	5.5e-2	
square16	2	1.3e-2	4.04
square32	4	3.4e-3	4.01
square64	8	8.4e-4	3.99
rate		2.00	

Table 1: CgWaveHoltz, boxHelmholtz, max norm, solver=krylov, order=2, $t = 0.5$, cfl=0.9, ad4=1, omega=2, tol=1e-05 Sat May 22 06:17:07 2021

FD44s Errors and convergence of WaveHoltz solutions on a square.

grid	N	u	r
square8	1	3.4e-3	
square16	2	3.4e-4	10.12
square32	4	1.5e-5	22.97
square64	8	5.1e-7	28.96
square128	16	3.8e-8	13.34
square256	32	2.7e-9	14.03
square512	64	1.9e-10	14.52
rate		4.10	

Table 2: CgWaveHoltz, boxHelmholtz, max norm, solver=krylov, order=4, $t = 0.5$, cfl=0.9, ad4=1, omega=2, tol=1e-8 Sat May 22 06:36:05 2021

******** OLD STUFF *********

Here are some initial results for $k_x = 2\pi$, $k_y = 2\pi$ and $\omega = 2$.
Notes on output:

- line with "TIME INTEGRAL v" gives the true solution error.
- lines with "Maximum residual = 9.319e-05" gives the residual in the Helmholtz equation.
- lines with "`it=3:  max(|v-vOld|)=8.47e-06 (tol=0.0001)`" give the residual in the WaveHoltz iteration $\|v^{n+1} - \mathbf{v}^n\|_\infty$

Square : fixed-point iteration. SECOND-ORDER - looks OK

```
RESIDUAL IS SMOOTH

../bin/cgwh -grid=square64.order2 -cmd=boxHelmholtz.cmd -omega=2 -tol=1e-4
../bin/cgwh -grid=square32.order2 -cmd=boxHelmholtz.cmd -omega=2 -tol=1e-3

G64
cgWave: t=5.030e-01 (65 steps) maxErr= 4.55e-04
cgWave: t=1.006e+00 (130 steps) maxErr= 3.53e-04
cgWave: t=1.509e+00 (195 steps) maxErr= 8.45e-04
cgWave: t=2.012e+00 (260 steps) maxErr= 5.42e-04
cgWave: t=2.515e+00 (325 steps) maxErr= 2.72e-04
cgWave: t=3.018e+00 (390 steps) maxErr= 8.21e-04
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 8.45e-04
cgWave: t=3.142e+00 (406 steps) maxErr= 8.54e-04
it=3:  max(|v-vOld|)=8.47e-06 (tol=0.0001)
#################### DONE CgWaveHoltz: CALL cgWave : number of WaveHoltz iteration =3 ##########################
Maximum residual = 9.319e-05


G32
advWave: sosupParameter=  0.10E+01
cgWave: t=4.952e-01 (32 steps) maxErr= 2.00e-03
cgWave: t=9.905e-01 (64 steps) maxErr= 9.78e-04
cgWave: t=1.486e+00 (96 steps) maxErr= 3.71e-03
cgWave: t=1.981e+00 (128 steps) maxErr= 2.45e-03
cgWave: t=2.476e+00 (160 steps) maxErr= 1.27e-03
cgWave: t=2.971e+00 (192 steps) maxErr= 3.05e-03
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 3.38e-03
cgWave: t=3.142e+00 (203 steps) maxErr= 3.81e-03
it=2:  max(|v-vOld|)=4.51e-04 (tol=0.001)
#################### DONE CgWaveHoltz: CALL cgWave : number of WaveHoltz iteration =2 ##########################
Maximum residual = 1.058e-03
```

Square : fixed-point iteration. FOURTH-ORDER – not bad, errors may be skewed by poor BC's

```
LOOKS BETTER -- residual largest near the boundary -- need to fix ghost
COMPARE ERRORS IN TIME INTEGRAL ratio= 30

../bin/cgwh -grid=square64.order4 -cmd=boxHelmholtz.cmd -omega=2 -tol=1e-4
../bin/cgwh -grid=square32.order4 -cmd=boxHelmholtz.cmd -omega=2 -tol=1e-3

G64
cgWave: t=5.030e-01 (65 steps) maxErr= 2.20e-06
cgWave: t=1.006e+00 (130 steps) maxErr= 7.10e-06
cgWave: t=1.509e+00 (195 steps) maxErr= 5.93e-06
cgWave: t=2.012e+00 (260 steps) maxErr= 4.95e-06
cgWave: t=2.515e+00 (325 steps) maxErr= 7.88e-06
cgWave: t=3.018e+00 (390 steps) maxErr= 1.40e-06
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 5.34e-07
cgWave: t=3.142e+00 (406 steps) maxErr= 8.17e-06
it=3:  max(|v-vOld|)=8.09e-06 (tol=0.0001)
#################### DONE CgWaveHoltz: CALL cgWave : number of WaveHoltz iteration =3 ##########################
```

```
Maximum residual = 7.956e-04


G32
cgWave: t=9.905e-01 (64 steps) maxErr= 3.30e-04
cgWave: t=1.486e+00 (96 steps) maxErr= 3.31e-04
cgWave: t=1.981e+00 (128 steps) maxErr= 1.31e-04
cgWave: t=2.476e+00 (160 steps) maxErr= 4.08e-04
cgWave: t=2.971e+00 (192 steps) maxErr= 1.18e-04
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 1.63e-05
cgWave: t=3.142e+00 (203 steps) maxErr= 3.85e-04
it=2:  max(|v-vOld|)=4.00e-04 (tol=0.001)
#################### DONE CgWaveHoltz: CALL cgWave : number of WaveHoltz iteration =2 ##########################
Maximum residual = 5.925e-03
```

CIC grid : fixed-point iteration. SECOND-ORDER - looks OK

```
G4  - ORDER=2
cgWave: t=5.045e-01 (44 steps) maxErr= 2.23e-03 (max(|u|)= 5.35e-01)
cgWave: t=1.009e+00 (88 steps) maxErr= 2.95e-03 (max(|u|)= 4.35e-01)
cgWave: t=1.513e+00 (132 steps) maxErr= 4.82e-03 (max(|u|)= 9.98e-01)
cgWave: t=2.018e+00 (176 steps) maxErr= 1.64e-03 (max(|u|)= 6.28e-01)
cgWave: t=2.522e+00 (220 steps) maxErr= 3.43e-03 (max(|u|)= 3.30e-01)
cgWave: t=3.027e+00 (264 steps) maxErr= 4.38e-03 (max(|u|)= 9.78e-01)
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 4.58e-03
cgWave: t=3.142e+00 (274 steps) maxErr= 3.90e-03 (max(|u|)= 1.00e+00)
it=3:  max(|v-vOld|)=4.52e-04 (tol=0.001)
#################### DONE CgWaveHoltz: CALL cgWave : number of WaveHoltz iteration =3 ##########################
Maximum residual = 4.049e-03

G2 - ORDER=2
cgWave: t=5.094e-01 (24 steps) maxErr= 9.28e-03 (max(|u|)= 5.33e-01)
cgWave: t=1.019e+00 (48 steps) maxErr= 7.21e-03 (max(|u|)= 4.57e-01)
cgWave: t=1.528e+00 (72 steps) maxErr= 1.72e-02 (max(|u|)= 1.01e+00)
cgWave: t=2.038e+00 (96 steps) maxErr= 1.14e-02 (max(|u|)= 6.05e-01)
cgWave: t=2.547e+00 (120 steps) maxErr= 6.96e-03 (max(|u|)= 3.80e-01)
cgWave: t=3.057e+00 (144 steps) maxErr= 1.74e-02 (max(|u|)= 1.00e+00)
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 1.74e-02
cgWave: t=3.142e+00 (148 steps) maxErr= 1.80e-02 (max(|u|)= 1.02e+00)
it=5:  max(|v-vOld|)=9.19e-04 (tol=0.001)
#################### DONE CgWaveHoltz: CALL cgWave : number of WaveHoltz iteration =5 ##########################
Maximum residual = 8.508e-03
```

CIC grid : Krylov iteration. FOURTH-ORDER - v error looks OK, check residual compared to order=2, residual is large near boundaries

```
../bin/cgwh boxHelmholtz.cmd -g=cice4.order4.ng3.hdf -omega=2 -solver=krylov -tol=1e-5 -imode=1
../bin/cgwh boxHelmholtz.cmd -g=cice2.order4.ng3.hdf -omega=2 -solver=krylov -tol=1e-5 -imode=1

G4 : ORDER=4
advWave: ADVANCE dim=2 order=4                grid=curvilinear... t=  0.11E-01
cgWave: t=4.966e-01 (46 steps) maxErr= 1.12e-05, ||u||= 5.46e-01
cgWave: t=9.932e-01 (92 steps) maxErr= 8.34e-06, ||u||= 4.04e-01
cgWave: t=1.490e+00 (138 steps) maxErr= 2.00e-05, ||u||= 9.87e-01
cgWave: t=1.986e+00 (184 steps) maxErr= 1.39e-05, ||u||= 6.74e-01
cgWave: t=2.483e+00 (230 steps) maxErr= 5.05e-06, ||u||= 2.51e-01
cgWave: t=2.980e+00 (276 steps) maxErr= 1.97e-05, ||u||= 9.48e-01
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 2.05e-05
cgWave: t=3.142e+00 (291 steps) maxErr= 2.07e-05, ||u||= 1.00e+00
it=-1:  max(|v-vOld|)=1.30e-06, tol=1e-05

 *************** DONE KYRLOV ITERATIONS -- numberOfIterations=8 *************

Maximum residual = 6.903e-03

G2  order=4
```

```
cgWave: t=4.950e-01 (26 steps) maxErr= 2.82e-04, ||u||= 5.49e-01
cgWave: t=9.901e-01 (52 steps) maxErr= 2.00e-04, ||u||= 3.98e-01
cgWave: t=1.485e+00 (78 steps) maxErr= 4.84e-04, ||u||= 9.86e-01
cgWave: t=1.980e+00 (104 steps) maxErr= 3.73e-04, ||u||= 6.83e-01
cgWave: t=2.475e+00 (130 steps) maxErr= 1.88e-04, ||u||= 2.36e-01
cgWave: t=2.970e+00 (156 steps) maxErr= 4.49e-04, ||u||= 9.42e-01
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 4.98e-04
cgWave: t=3.142e+00 (165 steps) maxErr= 4.97e-04, ||u||= 1.00e+00
it=-1:  max(|v-vOld|)=3.40e-06, tol=1e-05

 ################ DONE KYRLOV ITERATIONS -- numberOfIterations=8 #############
Maximum residual = 3.699e-02
```

CIC grid : Krylov iteration. SECOND-ORDER - looks OK, residual is smooth

```
  ../bin/cgwh boxHelmholtz.cmd -g=cice4.order2.hdf -omega=2 -solver=krylov -tol=1e-4
ORDER=2
advWave: ADVANCE dim=2 order=2                grid=curvilinear... t=  0.11E-01
cgWave: t=5.045e-01 (44 steps) maxErr= 1.46e-03, ||u||= 5.34e-01
cgWave: t=1.009e+00 (88 steps) maxErr= 1.16e-03, ||u||= 4.34e-01
cgWave: t=1.513e+00 (132 steps) maxErr= 2.72e-03, ||u||= 9.96e-01
cgWave: t=2.018e+00 (176 steps) maxErr= 1.78e-03, ||u||= 6.28e-01
cgWave: t=2.522e+00 (220 steps) maxErr= 9.08e-04, ||u||= 3.27e-01
cgWave: t=3.027e+00 (264 steps) maxErr= 2.66e-03, ||u||= 9.76e-01
cgWave: t=3.142e+00 TIME INTEGRAL v:  maxErr= 2.74e-03
cgWave: t=3.142e+00 (274 steps) maxErr= 2.74e-03, ||u||= 1.00e+00
plot: finish chosen...
it=-1:  max(|v-vOld|)=9.43e-05, tol=0.0001

 ********** DONE KYRLOV ITERATIONS -- numberOfIterations=6 **********

Maximum residual = 4.955e-04
```

8.2. Gaussian forcing

We consider solving the Helmholtz problem with a source term consisting of a sum of generalized Gaussians,

$$f(x,y,z,t) = \sum_m a_m \cos(\omega_m(t - t_{0,m}))\, e^{-\beta_m[(x-x_{0,m})^2+(y-y_{0,m})^2+(z-z_{0,m})^2]^{p_m}}. \tag{42}$$

8.2.1. Gaussian source in a square

Figure 4 shows results from CgWaveHoltz for a square. Source term: $\omega = 17.17$, $a = 100$, $(x_0, y_0) = (.7, .7)$, $\beta = 50$, $p = 1$.

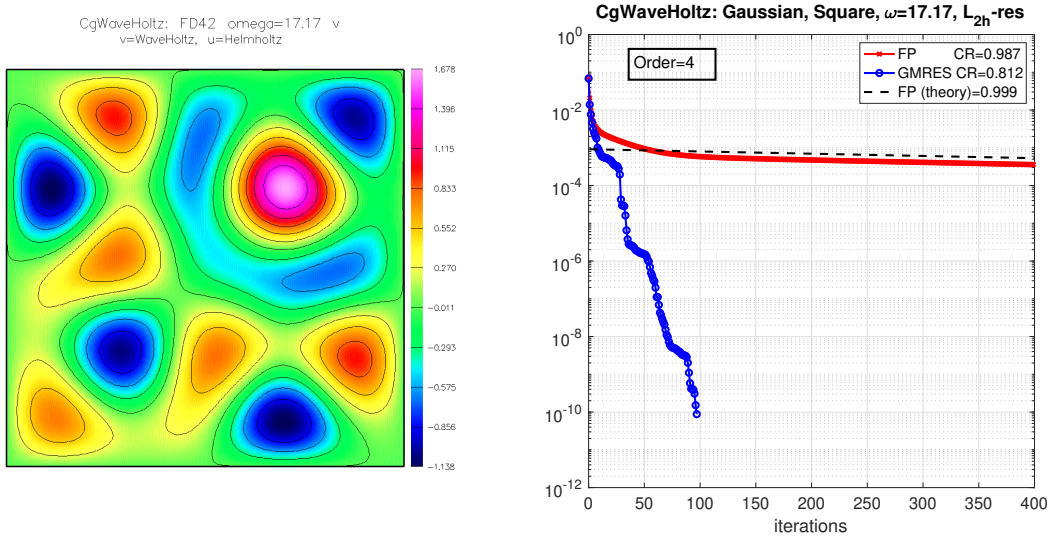

Figure 4: Gaussian source, $\omega = 17.17$, square64, order 4

Figure 5 shows results from CgWaveHoltz for a square. Source term: $\omega = 15$, $a = 100$, $(x_0, y_0) = (.7, .7)$, $\beta = 50$, $p = 1$. The FPI convergence rate matches the theory.

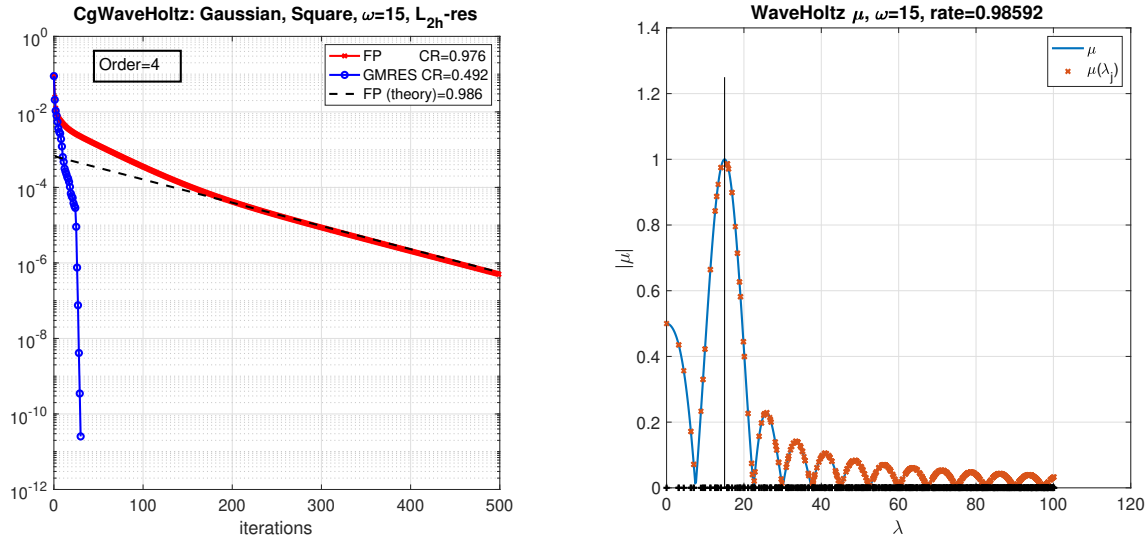

Figure 5: Gaussian source, $\omega = 15$, square128, order 4. Right: β function with distribution of eigenvalues λ_j

```
 ################ DONE KYRLOV ITERATIONS -- KSP residual=2.54e-11 (tol=1.00e-10) numberOfIterations=31 #############
CgWaveHoltz::residual: c=1, omega= 1.50000e+01, omegas= 1.49974e+01 (from symbol of D+D-), dt=4.273544e-03, adjustOmega=1
CgWaveHoltz::residual: max-res=4.010e-04 (using omega), max-res=5.004e-02 (using omega from discrete symbol)
CgWaveHoltz: omega=1.500e+01, max-res=4.010e-04, CR=0.492 CR-perPeriod=0.492, cpu= 1.58e+00(s)
CgWaveHoltz::Results saved to file gaussianWHSquare128O4Omega15Krylov.m
```

8.2.2. Gaussian source in an annulus

Figure 6 shows results from CgWaveHoltz for an Annulus. Eigenvalues for this problem were computed in `cg/ad/codes/annulusEigenvalues.maple`.

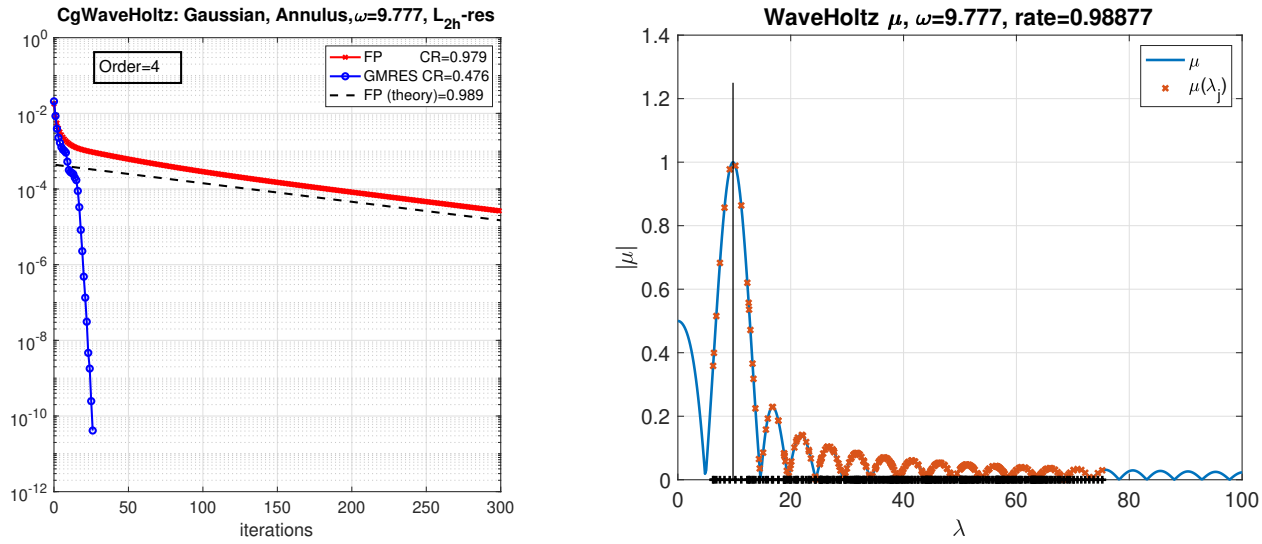

Figure 6: Gaussian source, $\omega = 9.777$, annulus8, order 4. Right: β function with distribution of eigenvalues λ_j

Figure 7 shows results from CgWaveHoltz for an Annulus for $\omega = 18.4$. It takes quite a iterations for the CR of the FPI to approach the theory; likely due to there being a few eigenvalues near the worst case. GMRES is also quite slow.

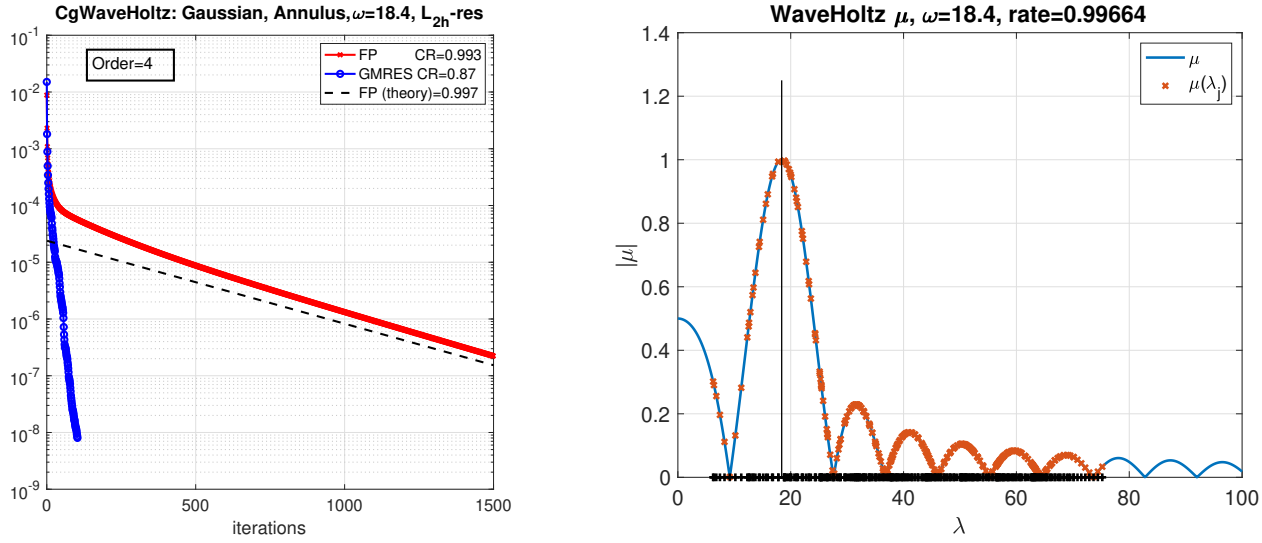

Figure 7: Gaussian source, $\omega = 18.4$, annulus8, order 4. Right: β function with distribution of eigenvalues λ_j

8.2.3. Gaussian source in a disk

Figure 8 shows results from CgWaveHoltz for an disk. Eigenvalues for this problem were computed in `cg/ad/codes/diskEigenvalues.maple`.

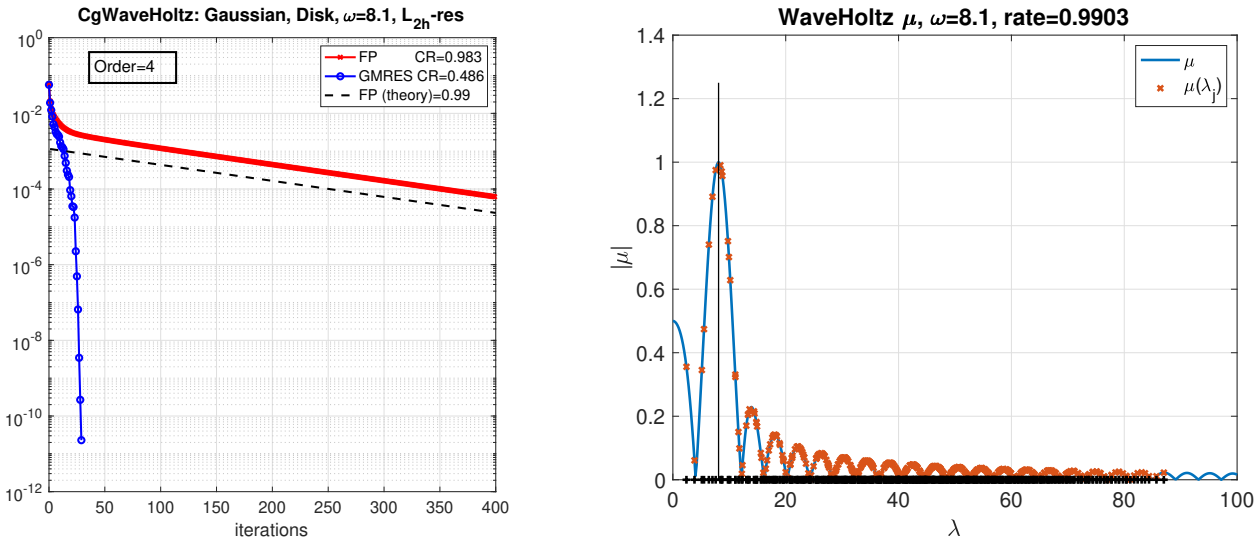

Figure 8: Gaussian source, $\omega = 8.1$, disk8, order 4. Right: β function with distribution of eigenvalues λ_j

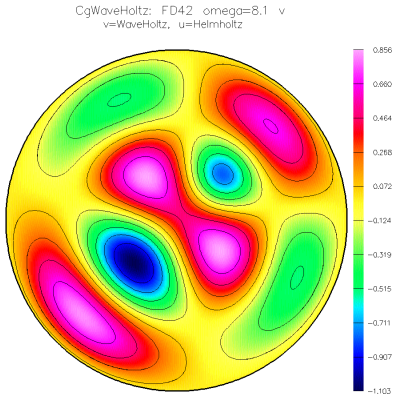

Figure 9: Gaussian source, $\omega = 8.1$, disk8, order 4.

Solution is close to the direct solver.

```
 ################ DONE KYRLOV ITERATIONS -- KSP residual=2.29e-11 (tol=1.00e-10) numberOfIterations=30 #############
CgWaveHoltz::residual: c=1, omega= 8.10000e+00, omegas= 8.09901e+00 (from symbol of D+D-), dt=6.687085e-03, adjustOmega=0
CgWaveHoltz::residual: max-res=1.776e-02 (using omega), max-res=9.750e-05 (using omega from discrete symbol)
CgWaveHoltz: omega=8.100e+00, max-res=1.776e-02, CR=0.486 CR-perPeriod=0.486, cpu= 3.41e+00(s)
CgWaveHoltz::Results saved to file gaussianWHDisk04Omega8p1Krylov.m
CgWaveHoltz: max-diff=3.25e-03 (between WaveHoltz and Direct Helmholtz solution)
```

8.2.4. Gaussian source in a box

Figure 10 shows results from CgWaveHoltz for a 3D box.

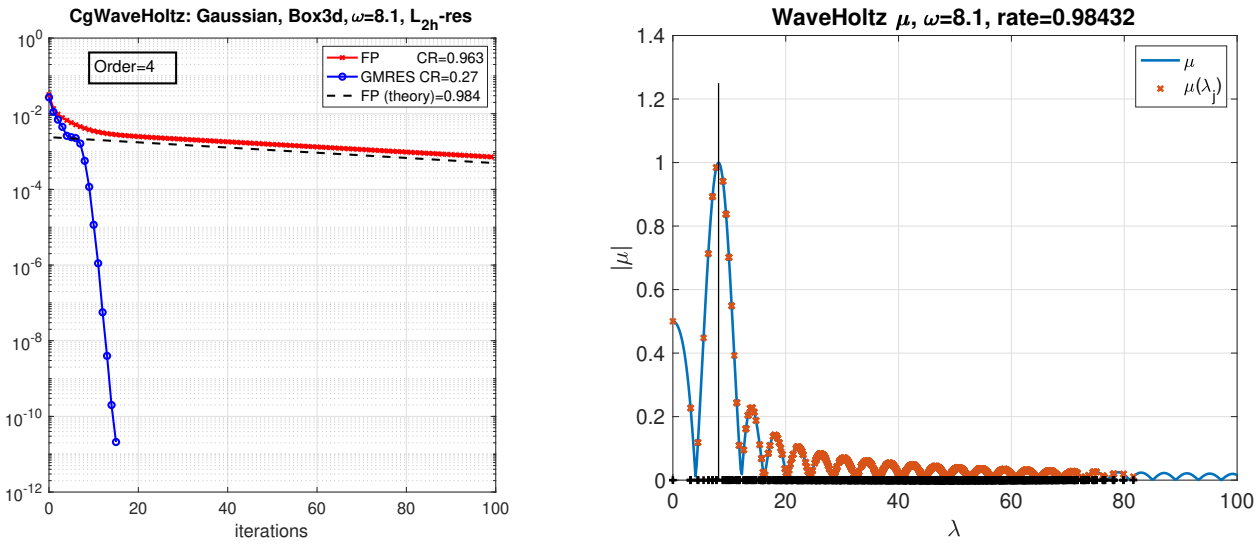

Figure 10: Gaussian source, $\omega = 8.1$, box4, order 4. Right: β function with distribution of eigenvalues λ_j

8.2.5. Gaussian source in a sphere

Figure 11 shows results from CgWaveHoltz for a 3D solid sphere.

FIX ME – EIGENVALUES FOR A SPHERE NEEDED

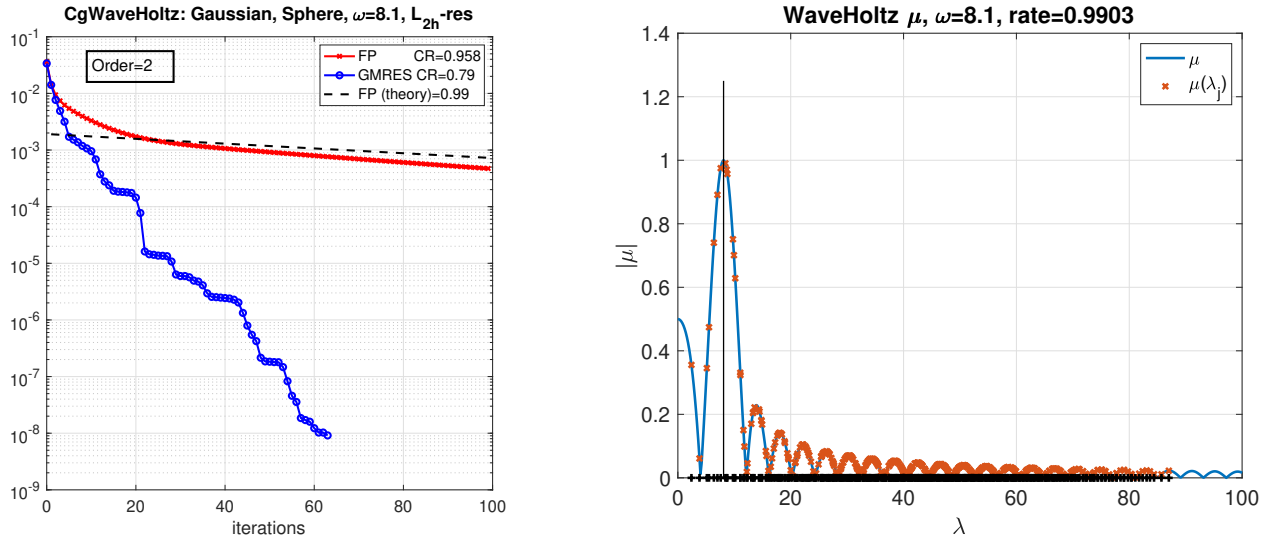

Figure 11: Gaussian source, $\omega = 8.1$, sphere4, order 2. Right: β function with distribution of eigenvalues λ_j

8.2.6. Gaussian source in a domain with three shapes

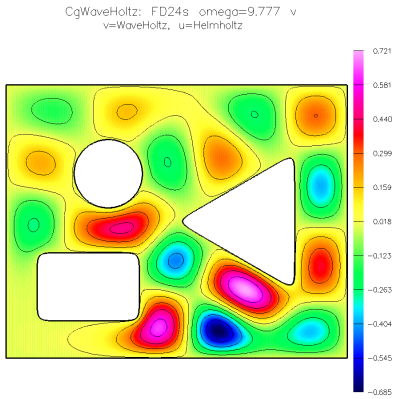

Figure 12: Gaussian pulse for shapes, $\omega = 9.777$, source at $(.3, -.75)$

```
SHAPES: omega=9.777
################ DONE KYRLOV ITERATIONS -- KSP residual=9.71e-09 (tol=1.00e-08) numberOfIterations=93 ##############
CgWaveHoltz::residual: c=1, omega= 9.77700e+00, omegas= 9.77691e+00 (from symbol of D+D-), dt=1.501518e-03, adjustOmega=0
CgWaveHoltz::residual: max-res=3.500e-02 (using omega), max-res=3.382e-02 (using omega from discrete symbol)
CgWaveHoltz: omega=9.777e+00, max-res=3.500e-02, CR=0.872 CR-perPeriod=0.872, cpu= 9.54e+01(s)
CgWaveHoltz::Results saved to file cgWaveHoltz.m
CgWaveHoltz: max-diff=1.16e-02 (between WaveHoltz and Direct Helmholtz solution)
```

******* OLD STUFF *******

- The Krylov method works much better than the fixed-point iteration.

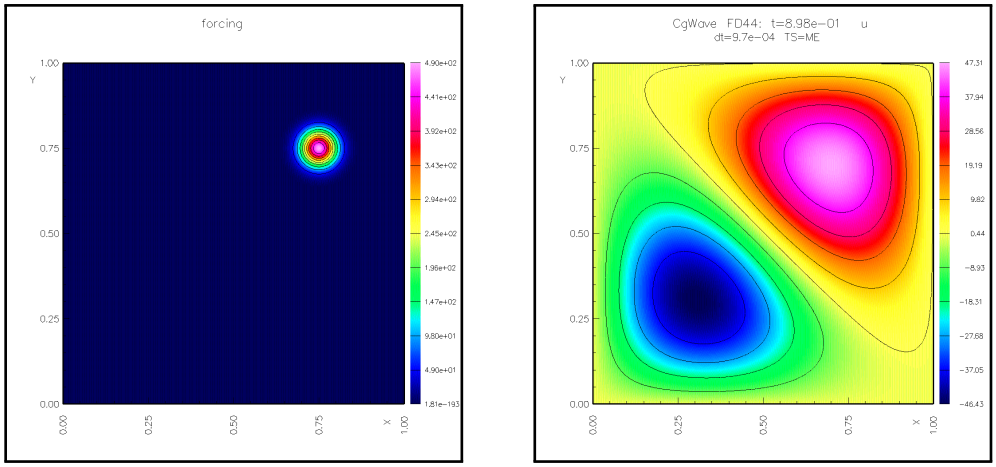

Figure 13: Gaussian forcing on a square. Left forcing. Right solution for $\omega = 7$.

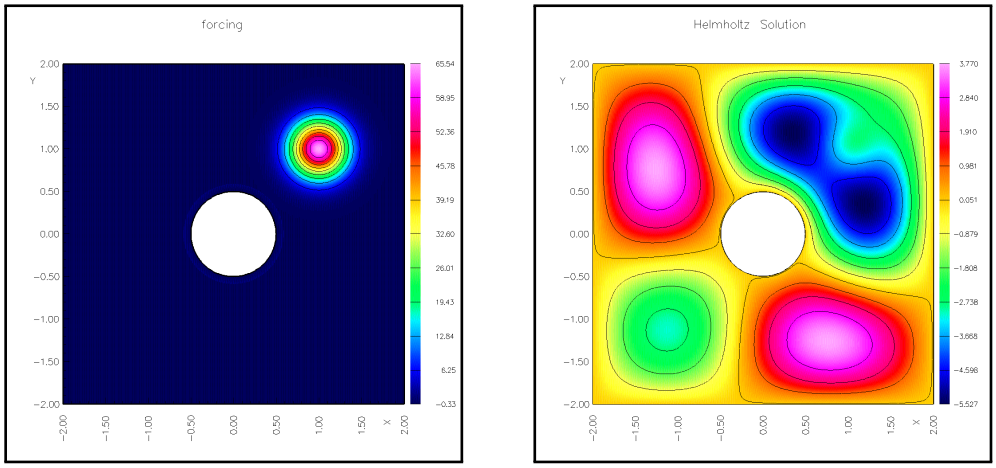

Figure 14: Gaussian forcing on the cic. Left forcing. Right solution for $\omega = 2.56$.

Square : Krylov iteration. SECOND-ORDER - looks OK

```
================  ORDER=2 -- looks OK

cgwh waveHoltz.cmd -g=square512.order2.hdf -x0=.75 -y0=.75 -omega=7 -solver=krylov -tol=1e-5 -tp=1 -ad4=0 -imode=1
cgwh waveHoltz.cmd -g=square256.order2.hdf -x0=.75 -y0=.75 -omega=7 -solver=krylov -tol=1e-4 -tp=1 -ad4=0 -imode=1
cgwh waveHoltz.cmd -g=square128.order2.hdf -x0=.75 -y0=.75 -omega=7 -solver=krylov -tol=1e-3 -tp=1 -ad4=0 -imode=1

 ################ DONE KYRLOV ITERATIONS -- numberOfIterations=11 #############
CgWaveHoltz::residual: c=1, omega=7

G512
 ################ DONE KYRLOV ITERATIONS -- KSP residual=3.35e-04 (tol=1.00e-05) numberOfIterations=12 #############
CgWaveHoltz::residual: c=1, omega=7
Maximum residual = 8.888e-03

G256
 ################ DONE KYRLOV ITERATIONS -- KSP residual=3.17e-03 (tol=1.00e-04) numberOfIterations=11 #############
CgWaveHoltz::residual: c=1, omega=7
```

```
Maximum residual = 3.792e-02

G128
 ################ DONE KYRLOV ITERATIONS -- KSP residual=1.64e-03 (tol=1.00e-03) numberOfIterations=11 #############
CgWaveHoltz::residual: c=1, omega=7
Maximum residual = 1.449e-01

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
G128 FIXED-POINT ITERATION
FIXED-POINT -omega=7 has trouble, may be close to a resonance!! TRY -omega=6.869 ...converges NOTE numPeriods
cgwh waveHoltz.cmd -g=square128.order2.hdf -x0=.75 -y0=.75 -omega=6.869 -solver=fixedPoint -tol=1e-3 -tp=1 -ad4=0 -imode=1 -maxl

it=75:  max(|v-vOld|)=9.70e-04 (tol=0.001)
#################### DONE CgWaveHoltz: CALL cgWave : number of WaveHoltz iteration =76 ########################
Maximum residual = 4.527e-02
```

Square : Krylov iteration. FOURTH-ORDER - not bad – RESIDUAL LARGE AT BOUNDARY probably degrades convergence

```
cgwh waveHoltz.cmd -g=square512.order4.hdf -x0=.75 -y0=.75 -omega=7 -solver=krylov -tol=1e-6 -tp=1 -ad4=0 -imode=1
cgwh waveHoltz.cmd -g=square256.order4.hdf -x0=.75 -y0=.75 -omega=7 -solver=krylov -tol=1e-5 -tp=1 -ad4=0 -imode=1
cgwh waveHoltz.cmd -g=square128.order4.hdf -x0=.75 -y0=.75 -omega=7 -solver=krylov -tol=1e-3 -tp=1 -ad4=0 -imode=1
cgwh waveHoltz.cmd -g=square64.order4.hdf -x0=.75 -y0=.75 -omega=7 -solver=krylov -tol=1e-3 -tp=1 -ad4=0 -imode=1

G512
################ DONE KYRLOV ITERATIONS -- KSP residual=1.54e-05 (tol=1.00e-06) numberOfIterations=13 #############
Maximum residual = 4.732e-05
################ DONE KYRLOV ITERATIONS -- KSP residual=7.44e-07 (tol=1.00e-07) numberOfIterations=14 #############
Maximum residual = 4.734e-05


G256
 ################ DONE KYRLOV ITERATIONS -- KSP residual=1.67e-04 (tol=1.00e-05) numberOfIterations=12 #############
Maximum residual = 5.578e-04

--- tol=1e-4 not enough
################ DONE KYRLOV ITERATIONS -- KSP residual=3.14e-03 (tol=1.00e-04) numberOfIterations=11 #############
Maximum residual = 4.488e-03


G128
################ DONE KYRLOV ITERATIONS -- KSP residual=1.57e-03 (tol=1.00e-03) numberOfIterations=11 #############
Maximum residual = 1.392e-02


G64
################ DONE KYRLOV ITERATIONS -- KSP residual=7.93e-04 (tol=1.00e-03) numberOfIterations=11 #############
CgWaveHoltz::residual: c=1, omega=7
Maximum residual = 1.531e-01
```

CIC : Krylov iteration. SECOND-ORDER – NOT BAD, some wiggles in RESIDUAL near interpolation points, worse for cice16.

```
cgwh waveHoltz.cmd -g=cice8.order2 -beta=10 -x0=1 -y0=1 -omega=2.56 -solver=krylov -tol=1e-5 -tp=1 -ad4=1 -imode=1
cgwh waveHoltz.cmd -g=cice4.order2 -beta=10 -x0=1 -y0=1 -omega=2.56 -solver=krylov -tol=1e-3 -tp=1 -ad4=1 -imode=1


G8
################ DONE KYRLOV ITERATIONS -- KSP residual=1.31e-03 (tol=1.00e-05) numberOfIterations=14 #############
omega=2.560e+00, Maximum residual = 7.921e-04

G4
```

```
 ################ DONE KYRLOV ITERATIONS -- KSP residual=5.26e-02 (tol=1.00e-03) numberOfIterations=12 ##############
omega=2.560e+00, Maximum residual = 3.118e-03

G2
cgwh waveHoltz.cmd -g=cice2.order2 -beta=10 -x0=1 -y0=1 -omega=2.56 -solver=krylov -tol=1e-3 -tp=1 -ad4=1 -imode=1

################ DONE KYRLOV ITERATIONS -- KSP residual=3.38e-02 (tol=1.00e-03) numberOfIterations=12 ##############
omega=2.560e+00, Maximum residual = 1.640e-02
```

9. EigenWave results

Here are some results for eigenWave. More results can be found in the EigenWave paper.

References

[1] D. Appelo, F. Garcia, O. Runborg, WaveHoltz: Iterative solution of the Helmholtz equation via the wave equation, SIAM J. Sci. Comput. 42 (4) (2020) A1950–A1983.

[2] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, The portable extensible toolkit for scientific computation, Tech. Rep. http://www.mcs.anl.gov/petsc/petsc.html, Argonne National Laboratory (1999).

[3] J. Baglama, L. Reichel, Augmented GMRES-type methods, Numerical Linear Algebra with Applications 14 (4) (2007) 337–350.
URL

[4] R. B. Morgan, A restarted gmres method augmented with eigenvectors, SIAM Journal on Matrix Analysis and Applications 16 (4) (1995) 1154–1171.
URL