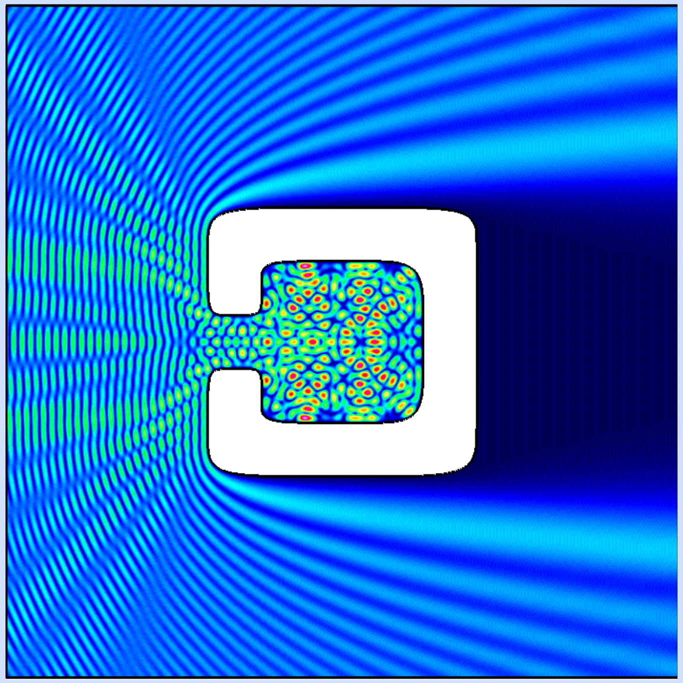
# An O(N) Helmholtz Solver from Time-Filtering the Wave Equation

(Or how to turn a time domain solver into a Helmholtz solver)




Bill Henshaw

Rensselaer Polytechnic Institute,
Troy New York USA

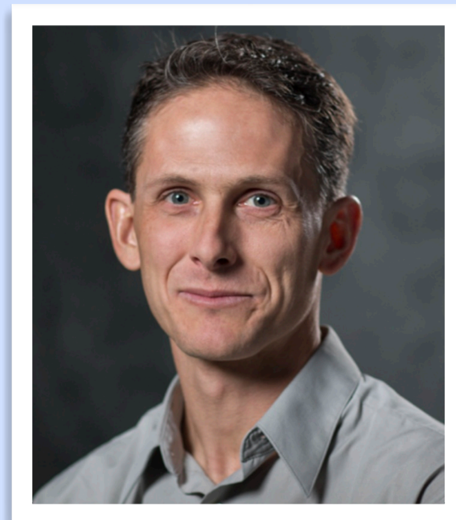
RPI


FEM@LLNL Seminar Series, November 4, 2025

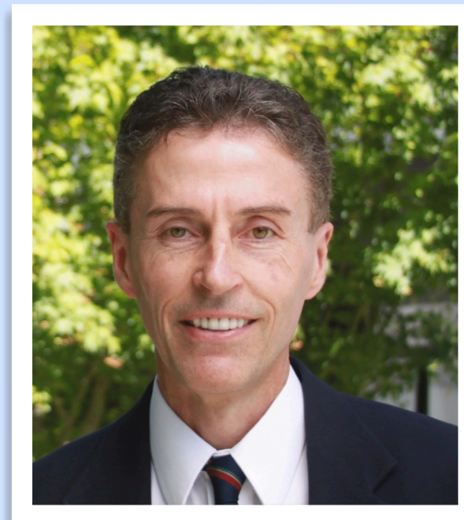
Lawrence Livermore National Laboratory

## Collaborators -

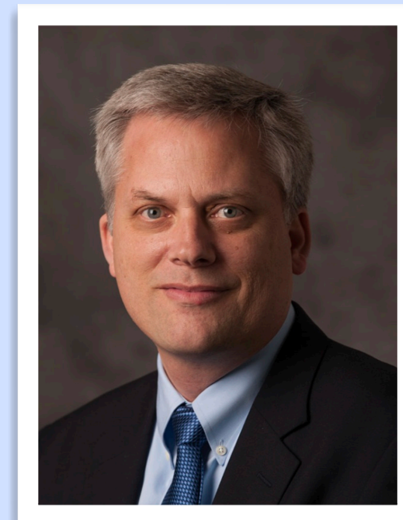
Rensselaer Polytechnic Institute



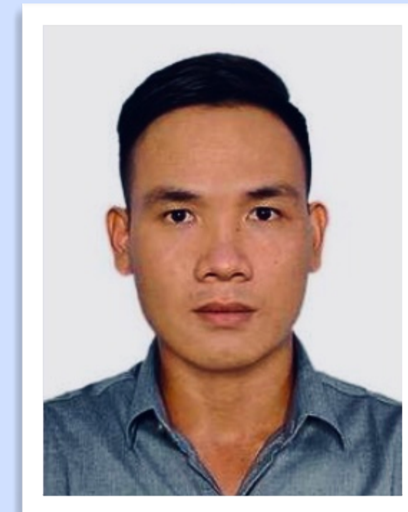
Jeff Banks



Bill Henshaw



Don Schwendeman



Ngan Le,
Grad Student

Daniel Appelö,
Virginia Tech

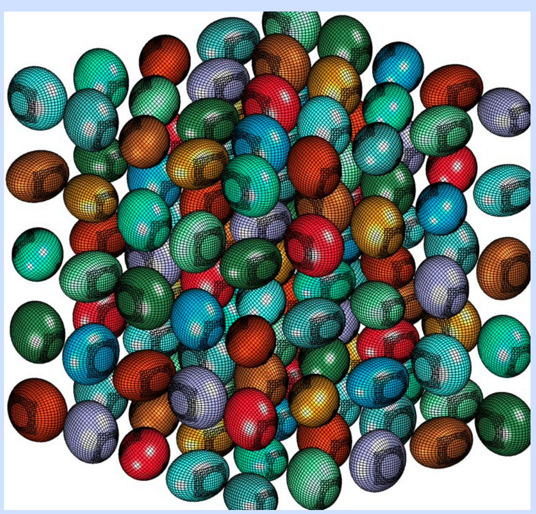
**With support from the NSF**

U.S. National
Science Foundation

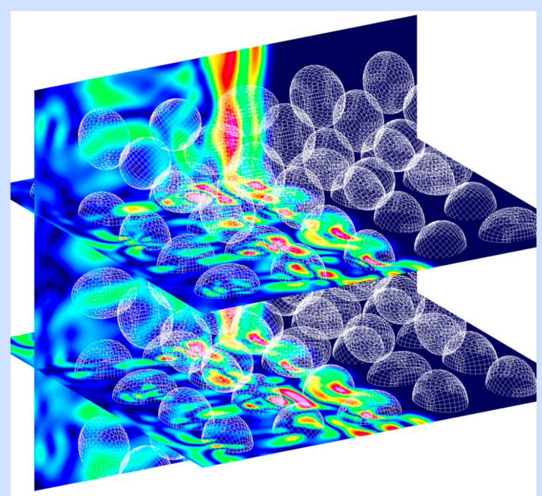
# Motivation ...

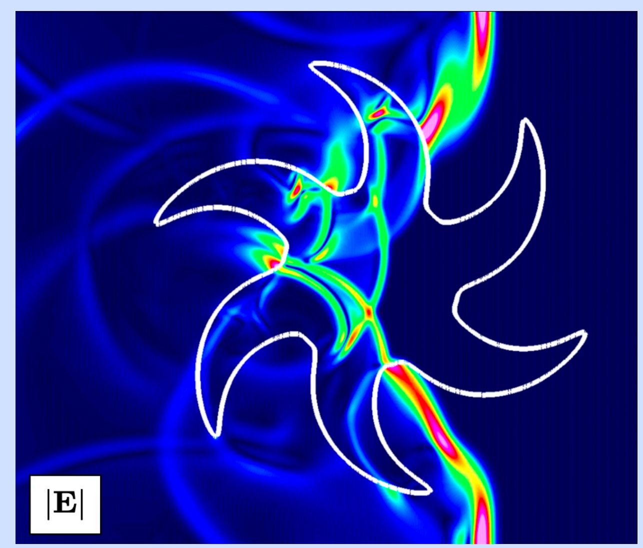
Helmholtz problems arise for wave propagation problems :
solid-mechanics, acoustics, electromagnetics

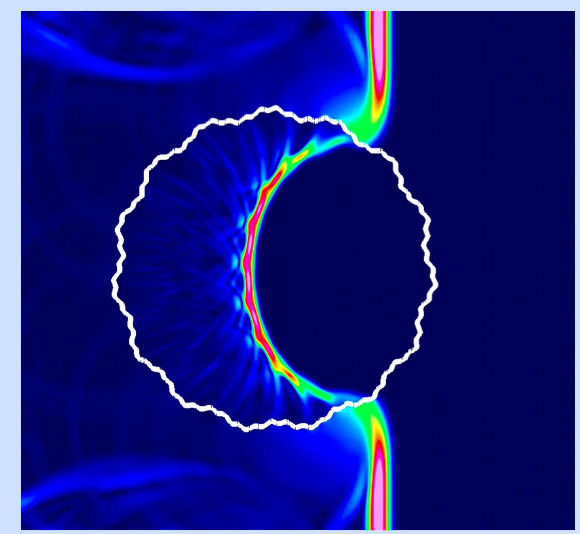
We have developed high-order accurate time domain solvers for the dispersive Maxwell's equations on overset grids:



Overset grid for solid ellipsoids





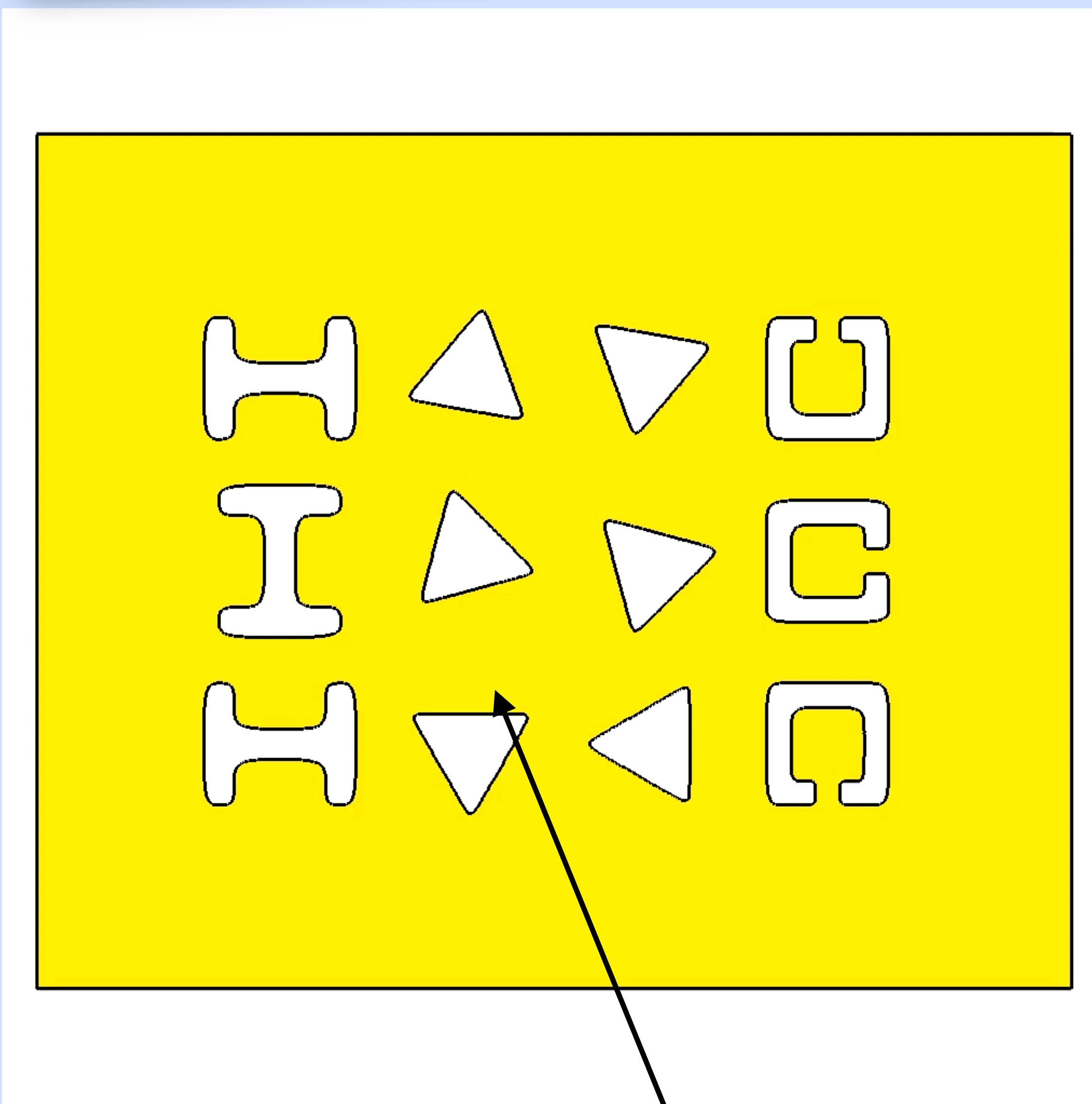



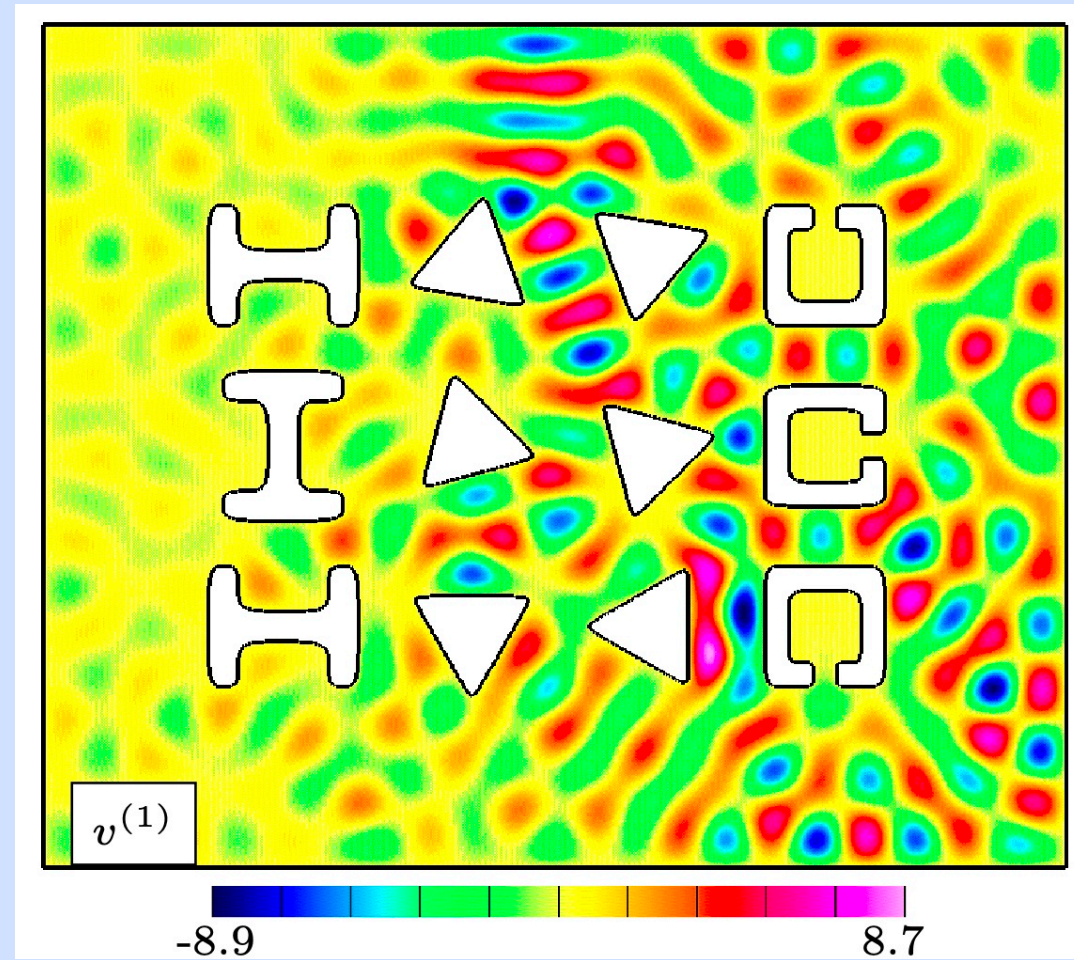
Scattering from various dispersive solids

**Question: Can we use existing time-domain solvers to solve the Helmholtz problem?**

See publications at OvertureFramework.org

Question: Given a time-domain solver for the wave equation how can we find the time-periodic solution?

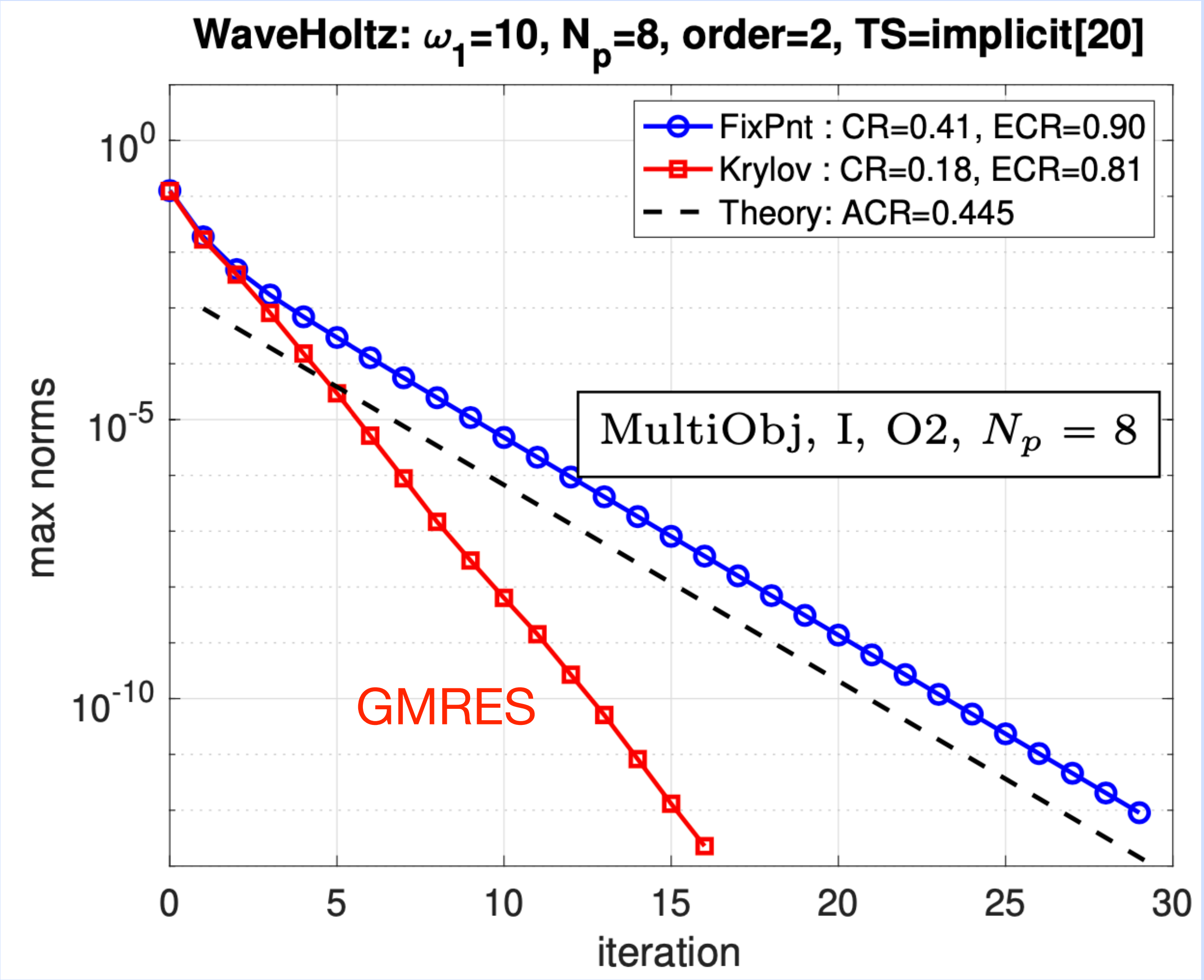





Time periodic solution

Wave equation with a time periodic Gaussian source:
Initial value problem (Dirichlet boundary conditions)

Answer: WaveHoltz can use a time domain solver to find time-periodic solutions




Convergence of WaveHoltz with deflation. Iteration = one wave solve

## Goal: Find time-periodic solutions to large scale PDE problems
## Solve the Helmholtz equation.

There are many ways to solve Helmholtz problems. For example:

1. Boundary integral, boundary element methods, fast multipole methods
2. Fast direct solvers
3. Domain decomposition with special transmission conditions, sweeping preconditioners, complex shifted Laplacian preconditioners, multigrid, ….

Possible issues with the current state of the art:

1. Poor performance for high frequencies.
2. Poor scaling (memory/CPU) as mesh is refined.
3. Pollution errors - low order methods require fine grids at high frequencies.
4. Poor parallel scaling or large initialization costs.
5. Limited to specialized geometries and or boundary conditions.

D.Lahaye, J.Tang, K.Vuik, Modern Solvers for Helmholtz Problems, Birkhauser, 2017

O.G. Ernst, M.J. Gander, Why it is Difficult to Solve Helmholtz Problems with Classical Iterative Methods, 2012.

# WaveHoltz Algorithm (closed domains, energy conserving case):

Goal: Solve the Helmholtz problem (e.g. with Dirichlet/Neumann BCs):

$$\begin{aligned} \mathcal{L}u + \omega^2 u &= f(\mathbf{x}), && \mathbf{x} \in \Omega, \\ \mathcal{B}u &= g(\mathbf{x}), && \mathbf{x} \in \partial\Omega, \end{aligned}$$

$$\mathcal{L} = c^2 \Delta$$

Given an initial guess $v^{(k)} \approx u(x)$ solve the forced wave equation:

$$\begin{aligned} \partial_t^2 w &= \mathcal{L}w - f(\mathbf{x})\cos(\omega t), && \mathbf{x} \in \Omega,\ t \in [0, \bar{T}] \\ \mathcal{B}w &= g(\mathbf{x})\cos(\omega t), && \mathbf{x} \in \partial\Omega, \\ w(\mathbf{x}, 0) &= v^{(k)}, && \mathbf{x} \in \Omega, \\ \partial_t w(\mathbf{x}, 0) &= 0, && \mathbf{x} \in \Omega. \end{aligned}$$

$$\bar{T} = N_p \frac{2\pi}{\omega}$$

Current iterate is the initial condition

Number of periods to integrate over

Compute $v^{(k+1)}$ from the time filter of w:

$$v^{(k+1)}(\mathbf{x}) = \frac{2}{\bar{T}} \int_0^{\bar{T}} \left( \cos(\omega t) - \frac{\alpha}{2} \right) w^{(k)}(\mathbf{x}, t; v^{(k)})\, dt,$$

New iterate is the time filtered solution to the wave equation

# WaveHoltz Algorithm (closed domains, energy conserving case):

**Algorithm 1** WaveHoltz Fixed-Point Iteration for Real Helmholtz Solutions.

1: **function** $[u_{\mathbf{i}}] = \textsc{WaveHoltz}(\omega,\, f)$ ▷ Solve the Helmholtz problem with frequency $\omega$ and forcing $f$

2: $v_{\mathbf{i}}^{(0)} = 0;$ ▷ Initial guess (e.g. zero) for $w(x,0)$.

3: **for** $k = 0, 1, \ldots$ **do** ▷ WaveHoltz iterations.

4: $w_{\mathbf{i}}^{0} = v_{\mathbf{i}}^{(k)}$ ▷ Initial condition for wave equation solve.

5: $\{w_{\mathbf{i}}^{n}\}_{n=0}^{N} = \textsc{solveWaveEquation}(\ v_{\mathbf{i}}^{(k)},\ \cos(\omega t^n)\, f_{\mathbf{i}},\ 0,\mathrm{T}\ )$ ▷ Solve for $w_{\mathbf{i}}^{n}$, over a period $T$.

6: $v_{\mathbf{i}}^{(k+1)} = \dfrac{2}{T} \displaystyle\sum_{n=0}^{N} \sigma_n \left( \cos(\omega t^n) - \frac{1}{4} \right) w_{\mathbf{i}}^{n},$ ▷ Filter step for $v_{\mathbf{i}}^{(k)}$ (quadrature weights $\sigma_n$).

7: **end for**

8: $u_{\mathbf{i}} = v_{\mathbf{i}}^{(k+1)}$ ▷ Real Helmholtz solution

9: **end function**

In practice, line 6 is accumulated during the time-stepping to avoid storing the solution at all time-steps.

## Some WaveHoltz References


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

Z. Peng, D. Appelö, EM-WaveHoltz: A flexible frequency-domain method built from time-domain solvers, IEEE Transactions on Antennas and Propagation 70(7) (2022) 5659--5671.

D.Appelö, F.~Garcia, A. Alvarez Loya, O. Runborg, El-WaveHoltz: A time-domain iterative solver for time-harmonic elastic waves, Computer Methods in Applied Mechanics and Engineering 401 (2022) 15603

D. Appelö, J.W. Banks, WDH, D.W. Schwendeman, An optimal O(N) Helmholtz solver for complex geometry using WaveHoltz and overset grids, arXiv:2504.03074 (2025).

# WaveHoltz transformed the problem to one that is easier to solve

The WaveHoltz algorithm defines an affine operator:

$$\boxed{v^{(k+1)}(\mathbf{x}) = \mathcal{A}v^{(k)} + b}$$

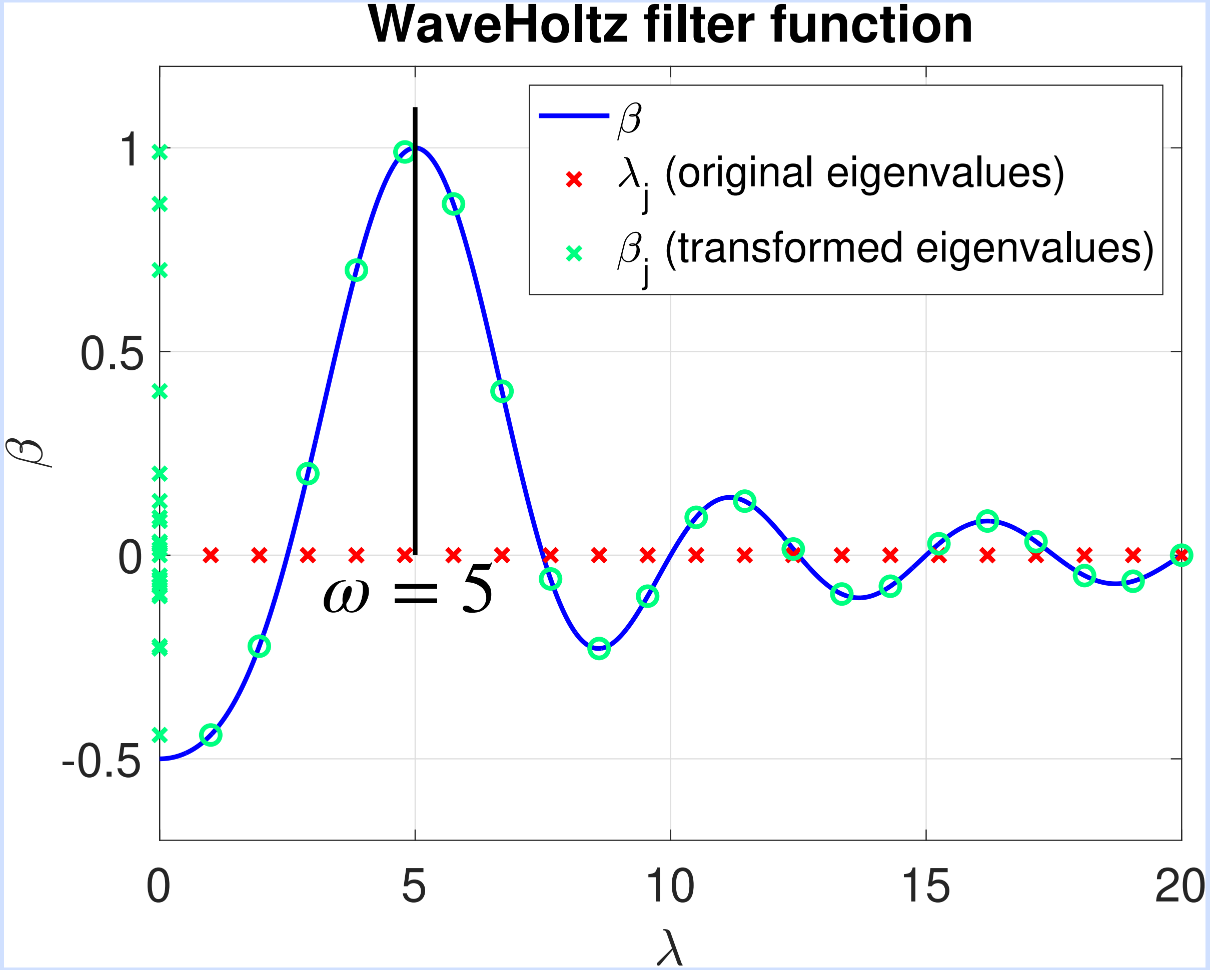
Where $\mathcal{A}$ has the same eigenfunctions as the Helmholtz (or $\Delta$) operator:

$$\boxed{\mathcal{L}\phi_j = -\lambda_j^2\,\phi_j} \quad \boxed{\mathcal{A}\phi_j = -\beta_j^2\,\phi_j}$$

But eigenvalues:

$$\boxed{\beta_j \overset{\text{def}}{=} \beta(\lambda_j;\omega)}$$

$$\boxed{\beta(\lambda;\omega) = \frac{2}{\bar{T}}\int_0^{\bar{T}} \left(\cos(\omega t) - \frac{1}{4}\right)\cos(\lambda t)\,dt}$$



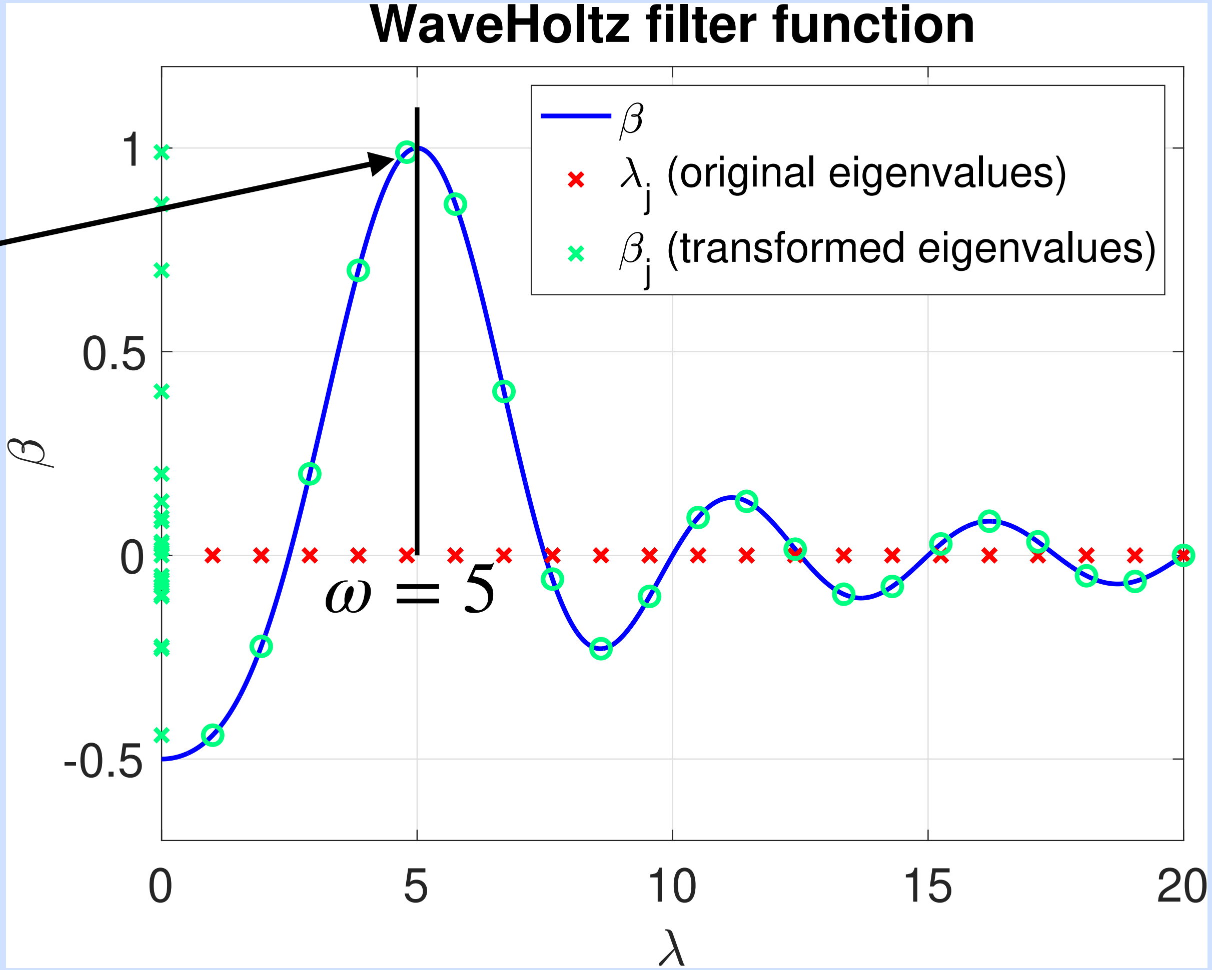

The transformed problem has eigenvalues $\beta_j \in [-.5, 1]$, largest near $\omega$, and more easily solved with traditional iterative solvers! No need to invert an indefinite matrix.

# The Fixed-Point Iteration will converge if the problem is not at resonance

**Theorem 1 (WaveHoltz FPI Convergence Rate).** *Assume $\lambda_m \neq \omega$ are the eigenvalues of $\mathcal{L}$ (with boundary conditions) so that $|\beta(\lambda_m)| < 1$ for all $\lambda_m$. The WaveHoltz fixed-point iteration has asymptotic convergence rate $\mu$ given by*

$$\mu = \max_{\lambda_m} |\beta(\lambda_m)|.$$

Asymptotic convergence rate $\mu$



## Enhancements to the Basic WaveHoltz Algorithm:

1. Acceleration by matrix-free Krylov methods such as GMRES
2. Implicit time-stepping with a large time-step
3. Deflation using precomputed eigenvalues and eigenvectors

# Time-stepping and time-corrections

Implicit/explicit time-stepping, $\beta^I = 1 - 2\alpha^I$ (for 2nd order accuracy)

$$D_{+t}D_{-t}W_{\mathbf{j}}^n = L_{ph}\left(\alpha^I W_{\mathbf{j}}^{n+1} + \beta^I W_{\mathbf{j}}^n + \alpha^I W_{\mathbf{j}}^{n-1}\right) - f(\mathbf{x_j})\cos(\tilde{\omega}t^n)\left(\beta^I + 2\alpha^I\cos(\tilde{\omega}\Delta t)\right)$$

Second-order time difference

$(W^{n+1} - 2W^n + W^{n-1})/\Delta t^2$

Order p difference operator, $L_{ph} \approx c^2\Delta$

$\alpha^I = 0$ : explicit

$\alpha^I = \frac{1}{2}$ : Trapezodial

$$\tilde{\omega} = \frac{1}{\Delta t}\cos^{-1}\left(\frac{1 - (\beta^I/2)(\omega\Delta t)^2}{1 + \alpha^I(\omega\Delta t)^2}\right)$$

Adjusted frequency

Time dispersion errors can be removed by adjusting the forcing

Note: the implicit time-stepping matrix $M = I - \Delta t^2 L_{ph}$ is definite

Amazingly only 10 implicit time-steps per period are needed for good convergence!

## Acceleration with Krylov space solvers

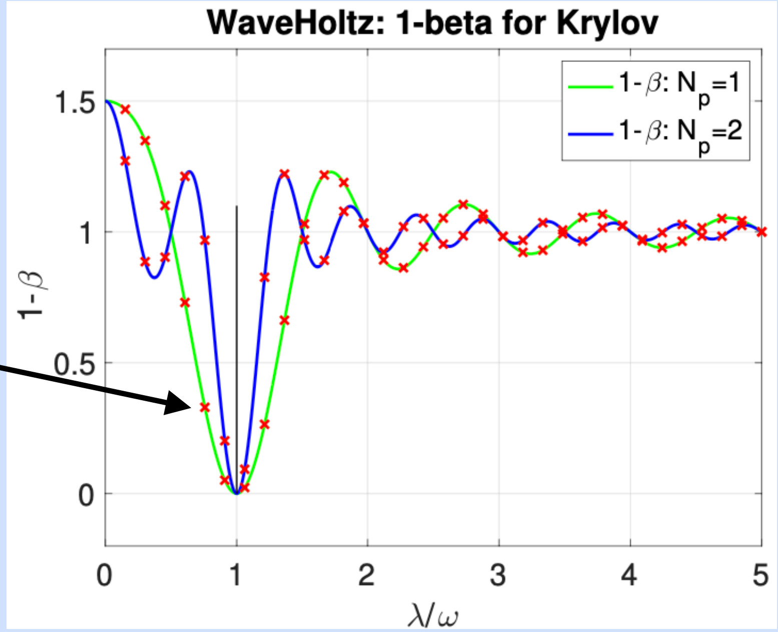
WaveHoltz can be accelerated with a matrix free Krylov solver.
The linear system to solve is ( $\mathcal{A}_h$ is the discrete form of $\mathcal{A}$ )

$$\boxed{(I - \mathcal{A}_h)u_h = b_h}$$

Applying $\mathcal{A}_h$ : Wave solve and filter with zero forcing

1. Conjugate Gradient can be used for symmetric discretizations
2. For overset grids we use GMRES or bi-CG-stab

Convergence depends on the eigenvalues 1-$\beta(\lambda_m)$



# “Direct” Eigenvector Deflation accelerates convergence

1. Uses precomputed eigenvalues $\lambda_m$ and eigenvectors $\phi_m$
2. Typically deflate eigenpairs with $\lambda_m \approx \omega$

**Algorithm 1** WaveHoltz Algorithm with Eigenfunction Deflation.

1: **function** WaveHoltz($\omega$,$f$,$N_p$)

2: // Final time is $\bar{T} = N_p T$ where $T = 2\pi/\omega$.

3: $k = 0$ ▷ WaveHoltz iteration counter.

4: $v^{(k)} = 0$ ▷ Initial guess for Helmholtz iterate (deflate if non-zero)

5: $f_d = f - \sum_{\phi_m \in \mathcal{D}} (f, \phi_m)_\Omega \, \phi_m$ ▷ Deflate forcing.

(1) Deflate forcing

6: **while** not converged **do** ▷ Start WaveHoltz iterations.

7: $w^{(k)}(\mathbf{x}, 0) = v^{(k)}(\mathbf{x})$ ▷ Initial condition for wave equation solve.

8: $w^{(k)}(\mathbf{x}, 0 : \bar{T})=$ solveWaveEquation($w^{(k)}(\mathbf{x}, 0)$,$f_d$) ▷ Solve for $\mathbf{w}(\mathbf{x}, t)$ (deflated forcing $f_d$).

9: $v^{(k+1)}(\mathbf{x}) = \frac{2}{\bar{T}} \int_0^{\bar{T}} \left(\cos(\omega t) - \frac{\alpha}{2}\right) w^{(k)}(\mathbf{x}, t) \, dt$ ▷ Time filter.

10: $v^{(k+1)} = v^{(k+1)} - \sum_{\phi_m \in \mathcal{D}} (v^{(k+1)}, \phi_m)_\Omega \, \phi_m$ ▷ Deflate iterate (skip with true eigenfunctions).

11: $k = k + 1$

12: **end while** ▷ End WaveHoltz iterations.

13: $v^{(k)} = v^{(k)} + \sum_{\phi_m \in \mathcal{D}} \frac{(f, \phi_m)_\Omega}{\omega^2 - \lambda_m^2} \phi_m$ ▷ Inflate.

(2) Inflate solution

14: $u = v^{(k)}$; ▷ Approximate Helmholtz solution.

15: **end function**

On overset grids line 10 is added since eigenvectors are only approximately orthogonal

## We have implemented WaveHoltz on Overset Grids:

1. CgWave: Uses fast and efficient high-order accurate finite differences for the wave equation (built on the Overture framework).
2. Matrix free geometric multigrid algorithms for overset grids can be used to solve the implicit time-stepping equations.
3. High-order and accurate treatment of boundaries using compatibility boundary conditions.

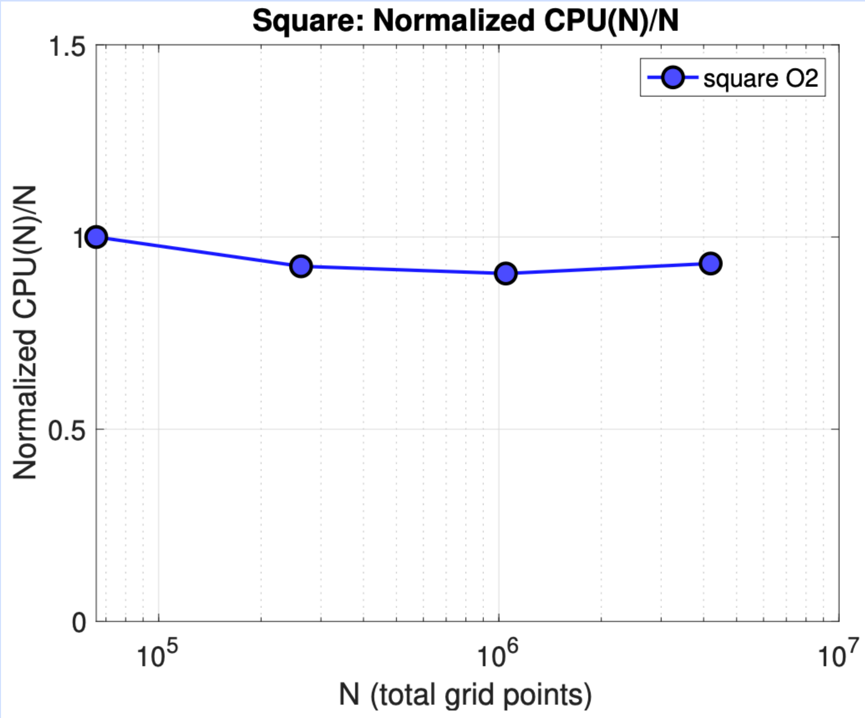
**This leads to an optimal O(N) algorithm at fixed frequency**
**N= total number of grid points**

A.M. Carson, J.W. Banks, WDH, D.W. Schwendeman, High-order accurate implicit-explicit time-stepping schemes for wave equations on overset grids, Journal of Computational Physics 520 (2025) 113513.

C.Liu, WDH, Multigrid with nonstandard coarse-level operators and coarsening factors, J. of Scientific Computing 94 (58) (2023) 1-27.

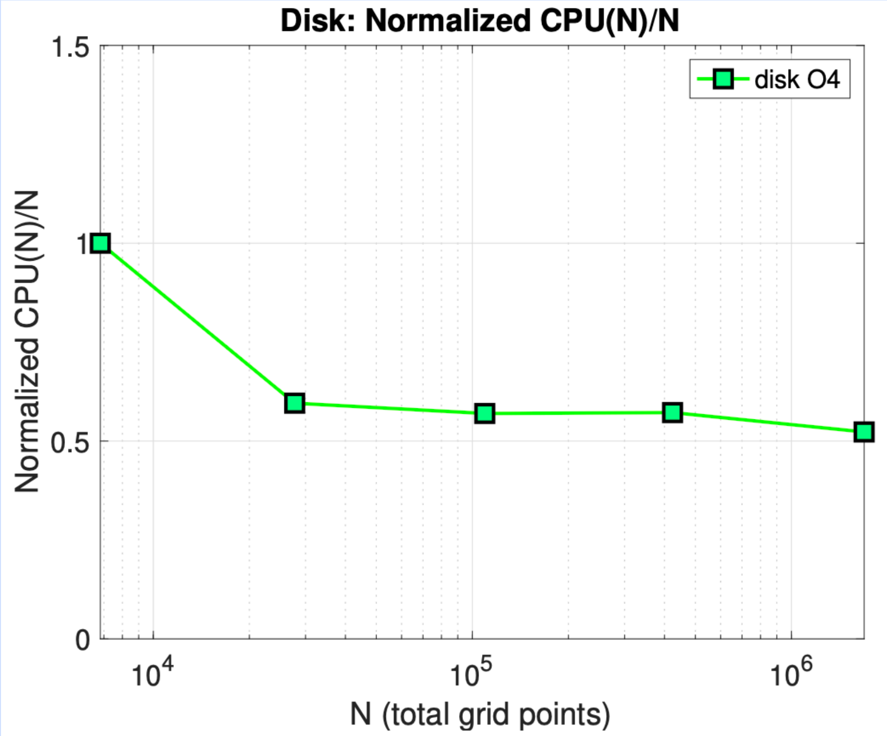
N.G. Al Hassanieh, J.W. Banks, WDH, D.W. Schwendeman, Local compatibility boundary conditions for high-order accurate finite-difference approximations of PDEs, SIAM J. Sci. Comput. 44 (2022) A3645-A3672.

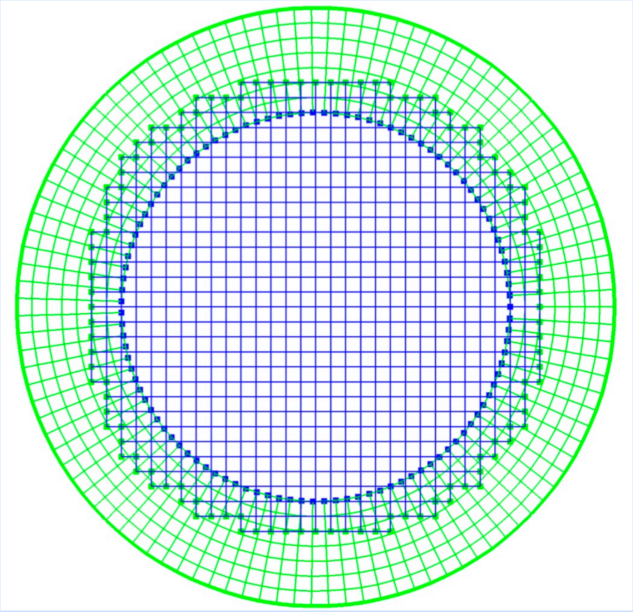
# WaveHoltz CPU-time/N is a nearly constant function of N



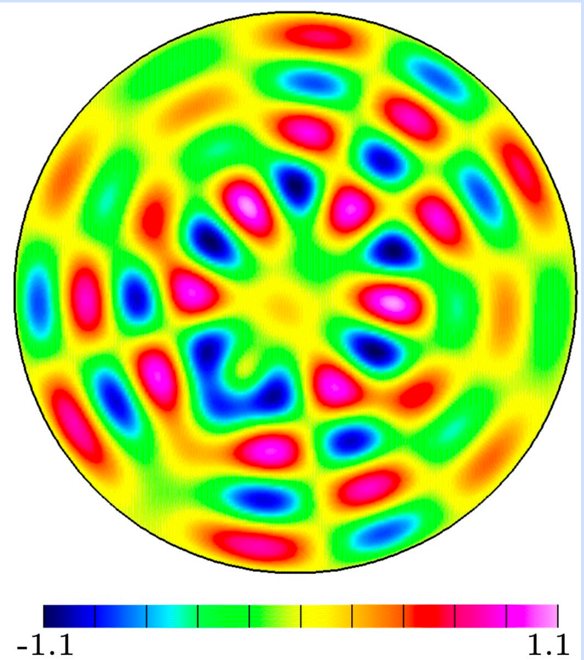

Computations on a square and disk demonstrate O(N) scaling for WaveHoltz

Using the Ogmg geometric multigrid solver for overset grids.



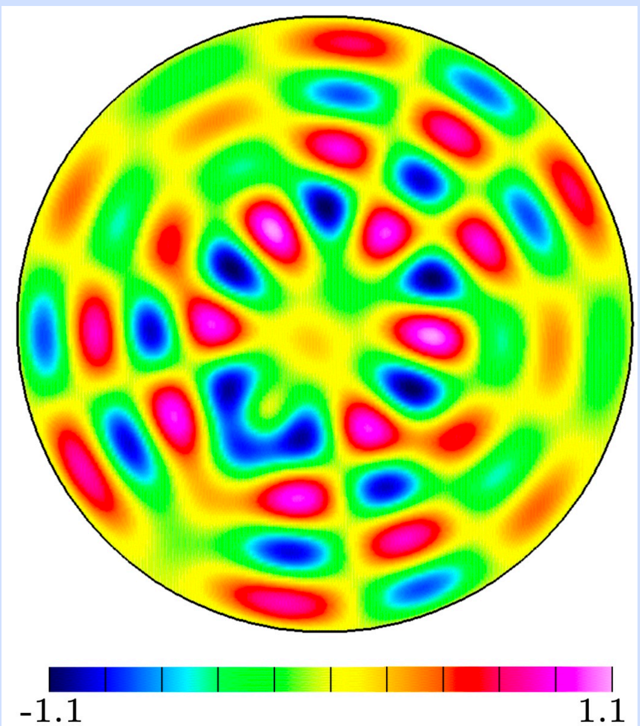
Overset grid

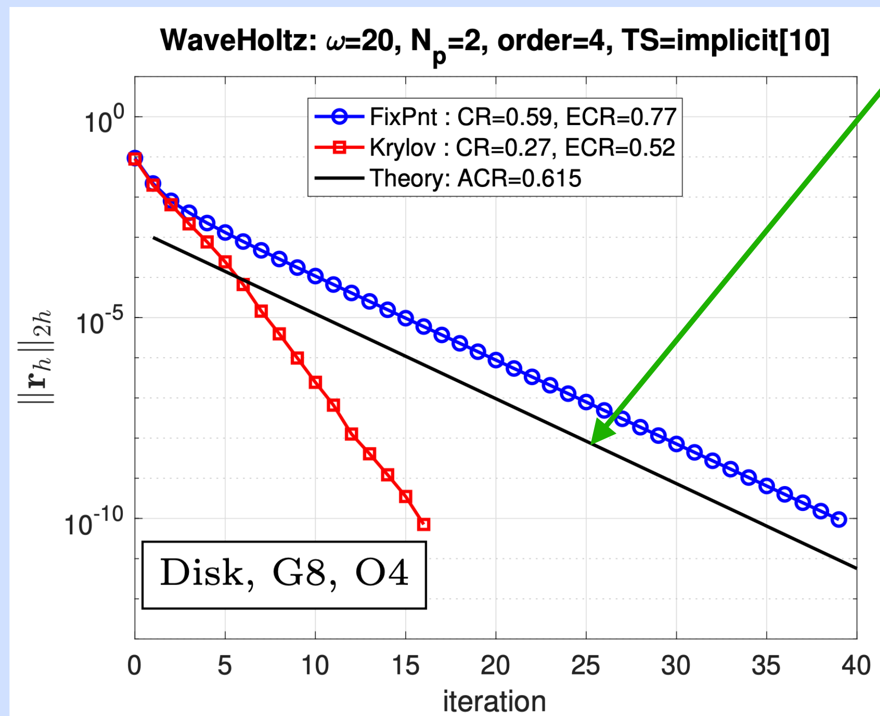


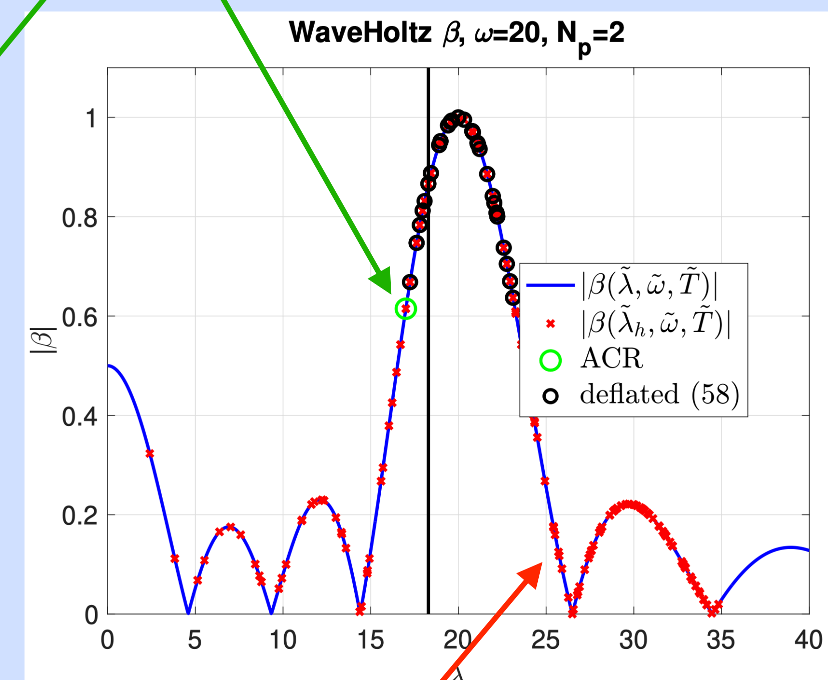
# WaveHoltz example: Gaussian source in a disk

$\omega = 20$, order=4, implicit time-stepping,
2 periods, 10 time-steps per period

Asymptotic convergence rate = .615



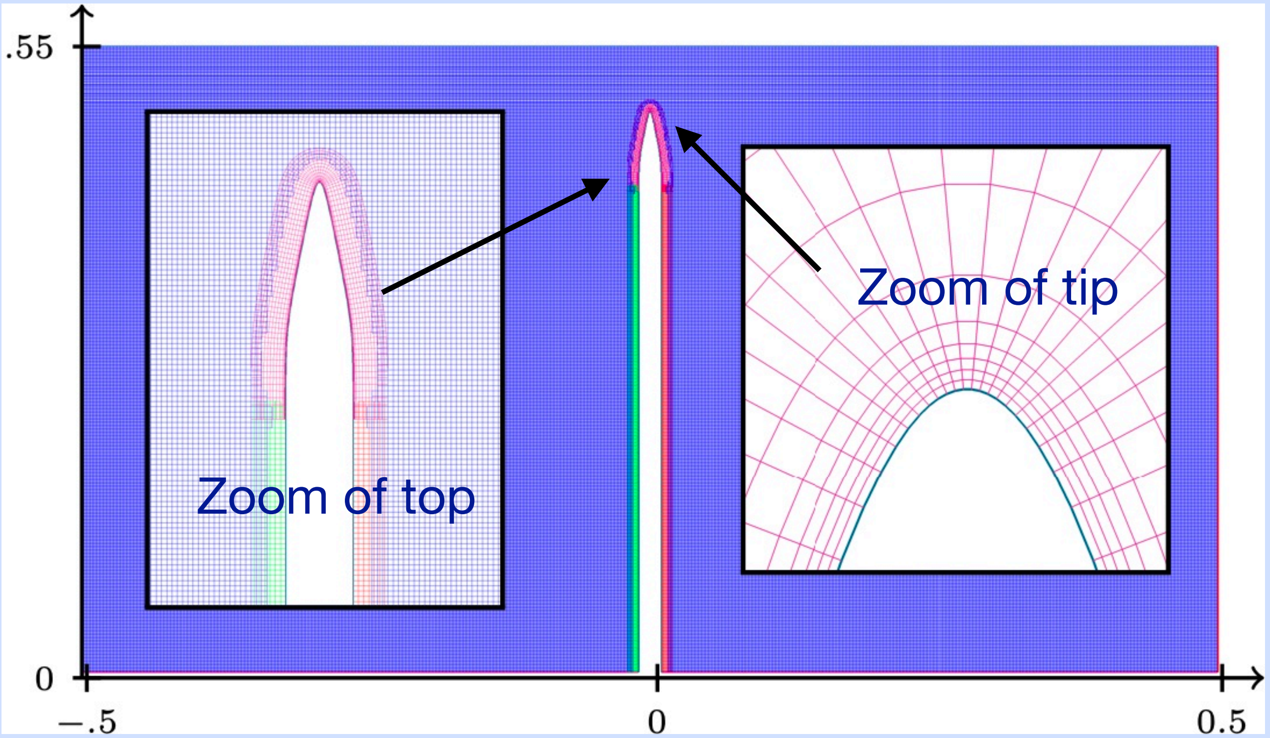




Convergence with deflation. FPI matches theory

Red x’s mark filter function evaluated at eigenvalues of $\Delta$

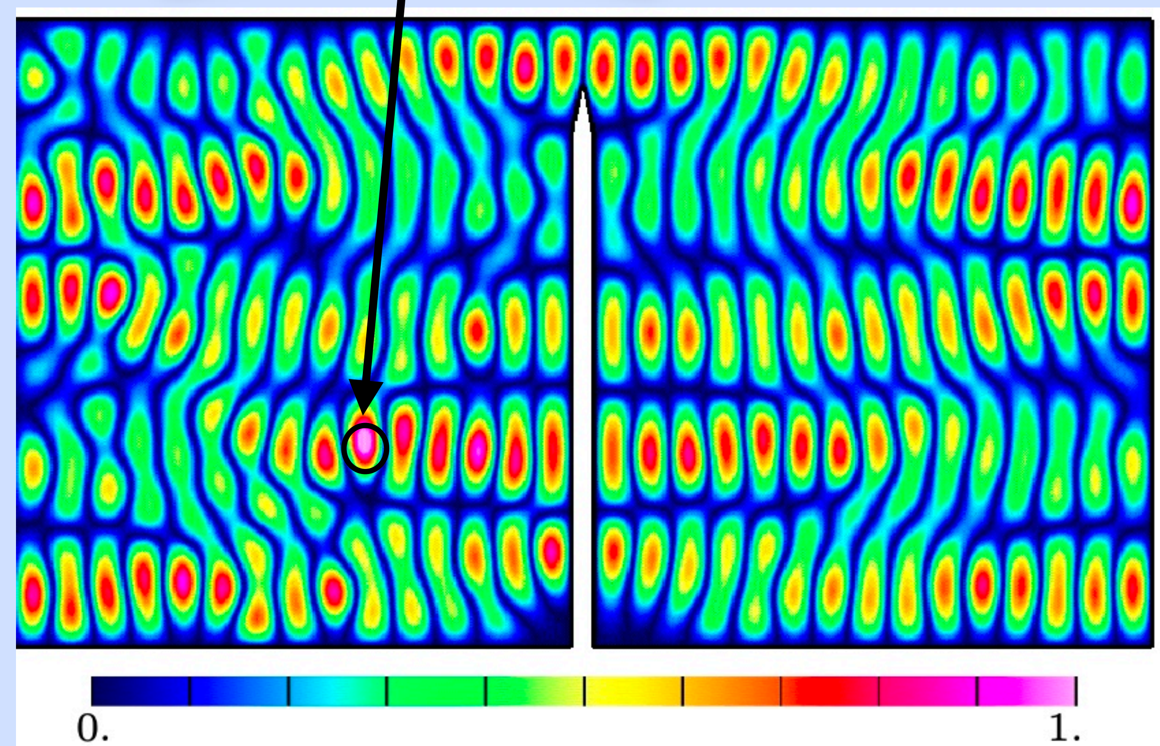
Overset grid for the disk, Dirichlet boundary conditions

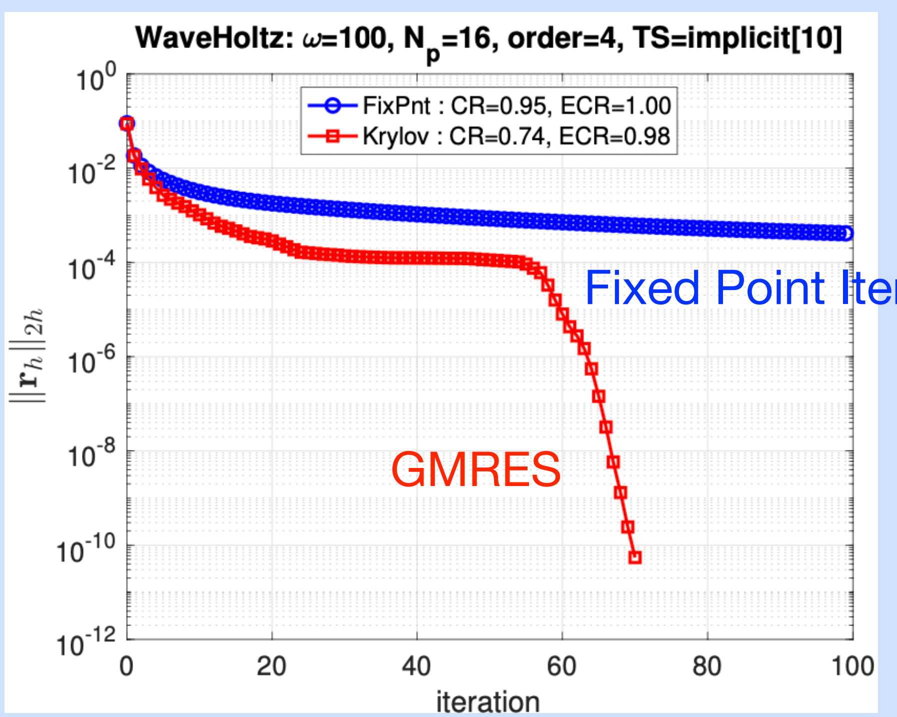
# Example: knife edge, $\omega = 100$



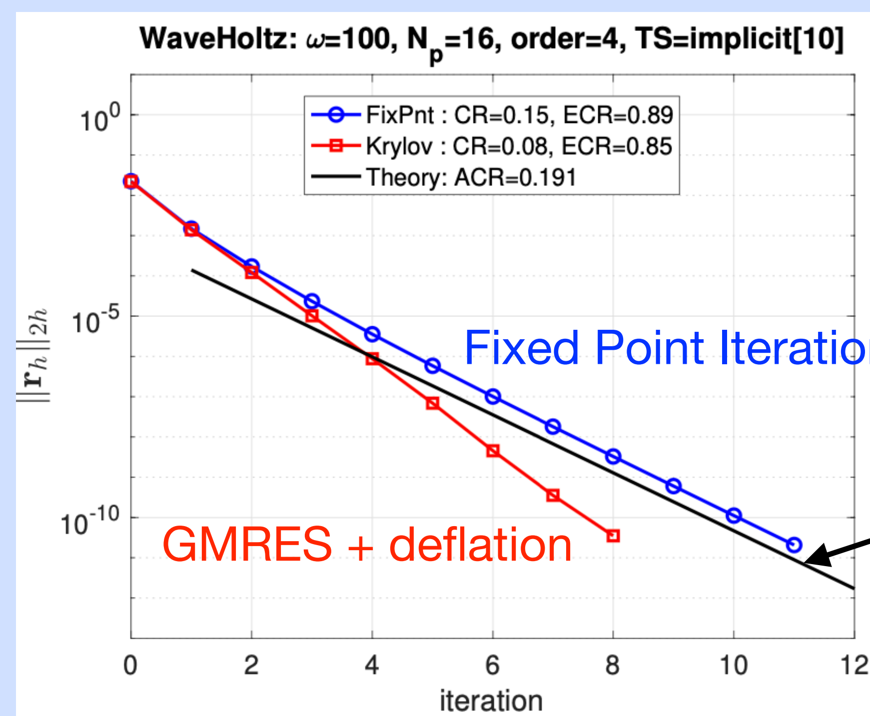

Small cells pose no problem with implicit method

Gaussian source




Convergence - no deflation

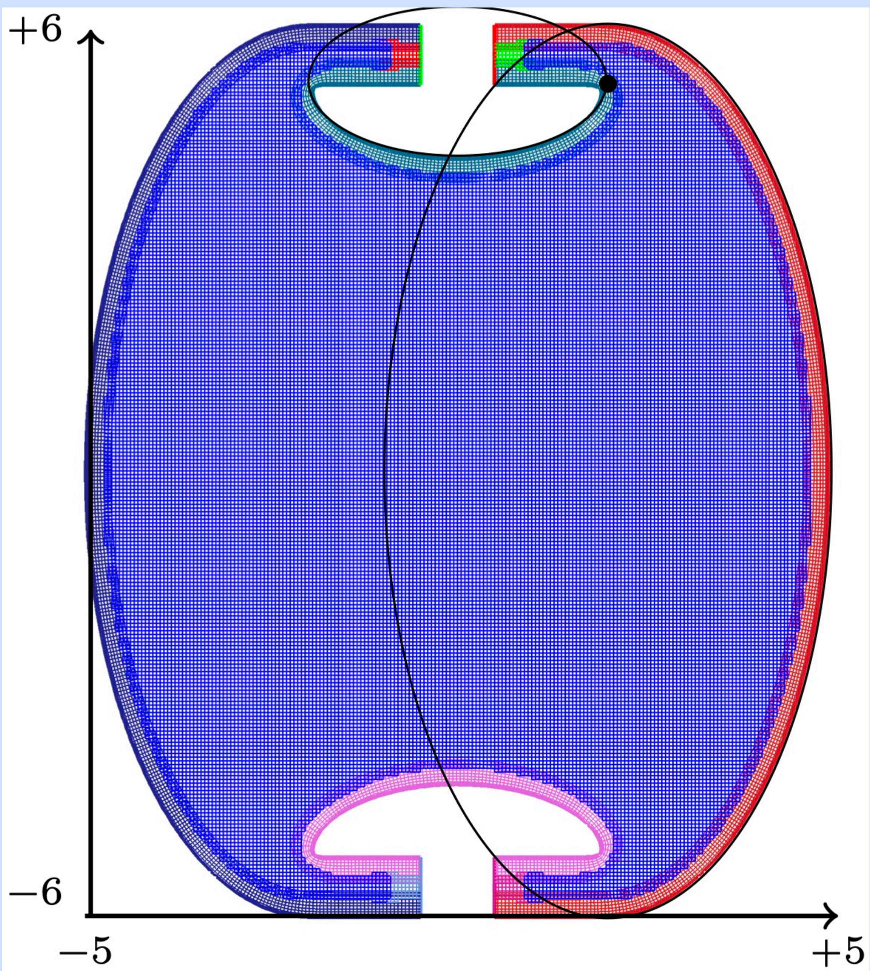


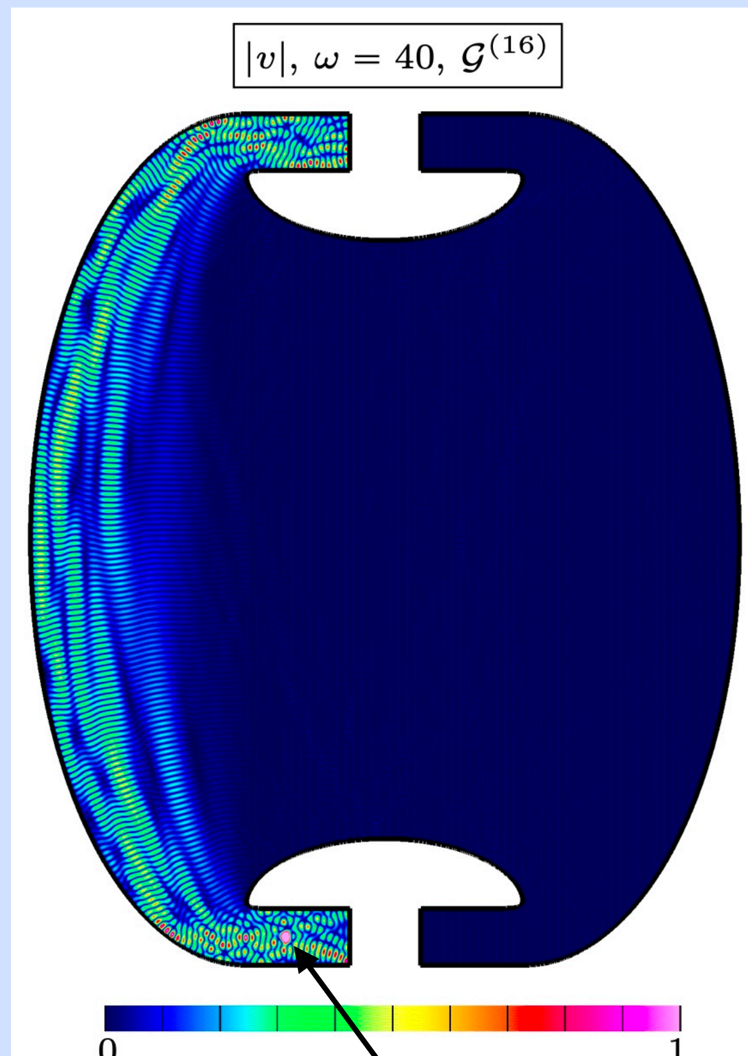

Convergence - with 64 deflated eigenmodes

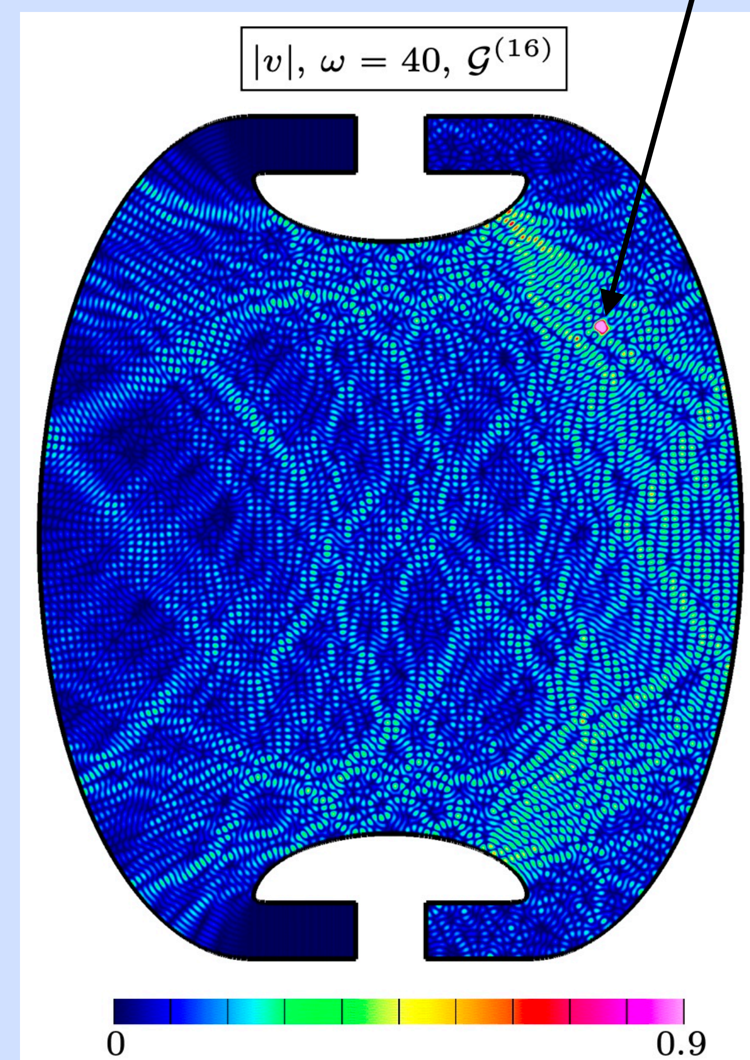
# Penrose un-illuminable room: high frequency example…

A room with reflective walls such that a light (or sound) source placed anywhere in the room will leave a dark (or quiet) region somewhere.

Source



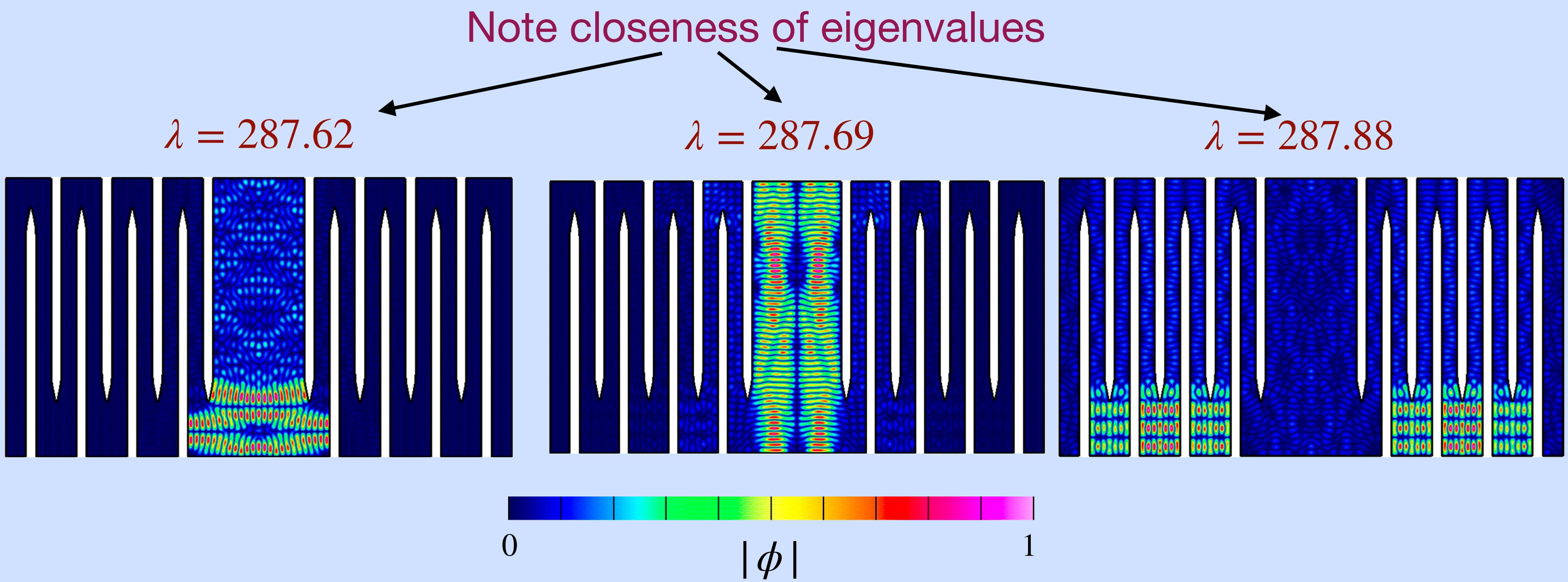

Overset grid
(coarse version)

Source

Fourth-order scheme

# High Frequency Helmholtz problems can be very hard

The relative spacing between eigenvalues of the Laplacian scales as $\lambda^{-d}$ in d-dimensions $\rightarrow$ we are almost always close to resonance




Selected eigenvectors of the Laplacian with Dirichlet BCs

$$\Delta\phi = -\lambda^2\phi$$

The Helmholtz solution can change dramatically as $\omega$ varies

## Indirect Deflation using Augmented Krylov Solvers

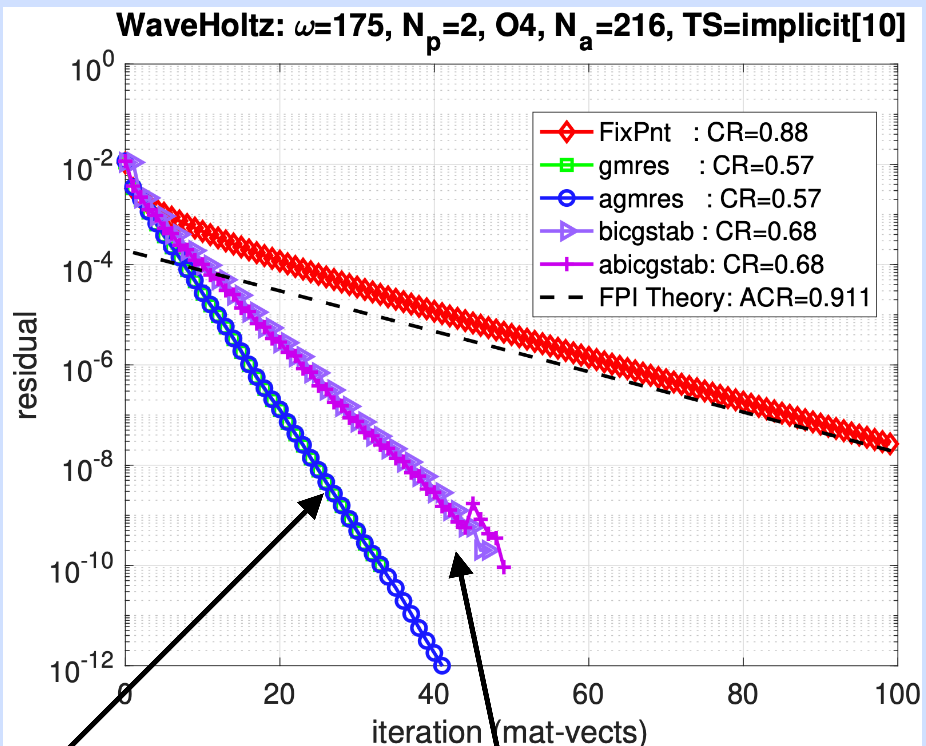
1. Augmented Krylov solvers add user provided vectors to the Krylov space
2. Also known as *recycled* Krylov solvers (i.e. recycle vectors from a previous Krylov solve into a new Krylov solve)
3. May use exact eigenvectors or <u>approximate</u> eigenvectors (e.g. from a coarse grid or lower-order accurate approximation)

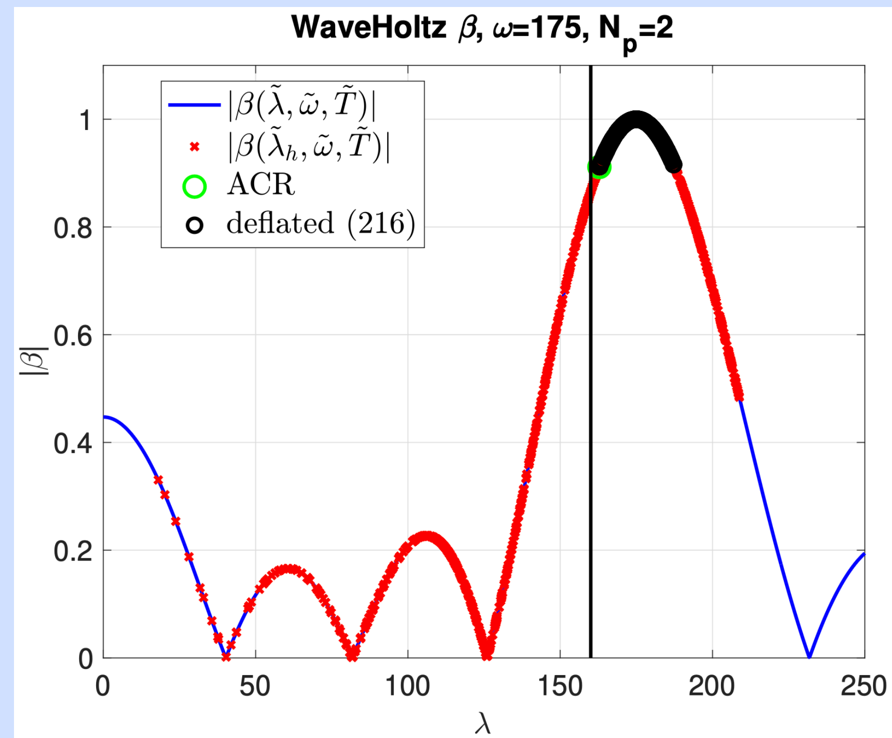

Y. Saad, M. Yeung, J. Erhel, F. Guyomarc'h, A deflated version of the conjugate gradient algorithm, SIAM Journal on Scientific Computing 21(5) (2000) 1909--1926.

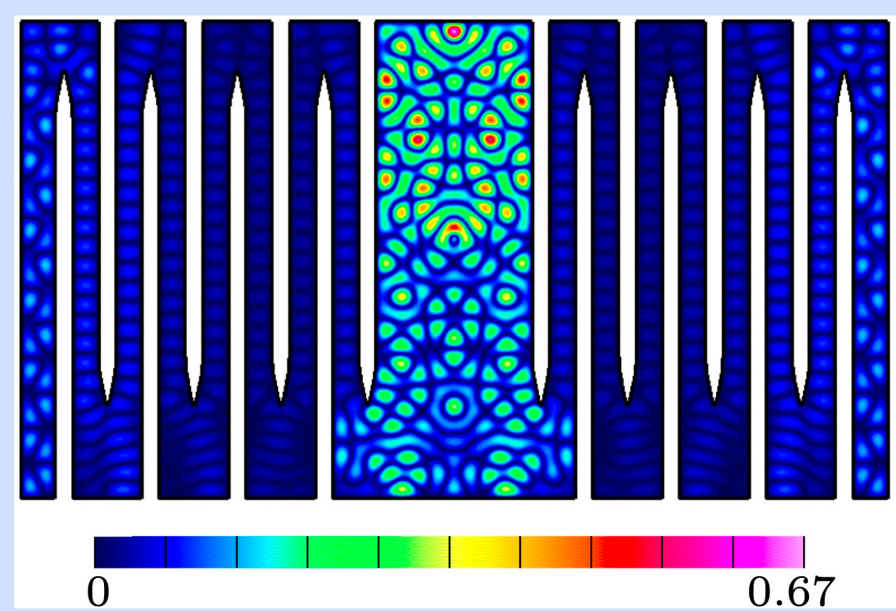
J. Baglama, L. Reichel, Augmented GMRES-type methods, Numerical Linear Algebra with Applications 14(4) (2007) 337-350.

A. Amritkar, E. de Sturler, K. Świrydowicz, D. Tafti, K. Ahuja, Recycling Krylov subspaces for CFD applications and a new hybrid recycling solver, Journal of Computational Physics 303 (2015) 222--237.

# Comparison: Direct Deflation versus Augmented Krylov Solvers




GMRES + direct deflation matches
Augmented GMRES

bi-CG-stab + direct deflation matches
augmented bi-CG-stab

# Open Domains with Radiation Boundary Conditions

For open domains the Helmholtz solution is complex.
We iterate on the real and imaginary parts of the solution.

$$
\begin{aligned}
\partial_t^2 w &= \mathcal{L}w - f(\mathbf{x})\cos(\omega t), && \mathbf{x} \in \Omega,\ t \in [0, \bar{T}],\\
\mathcal{B}w &= g(\mathbf{x})\cos(\omega t), && \mathbf{x} \in \partial\Omega,\\
w(\mathbf{x}, 0) &= v^{(k)}, && \mathbf{x} \in \Omega,\\
\partial_t w(\mathbf{x}, 0) &= \dot{v}^{(k)}, && \mathbf{x} \in \Omega.
\end{aligned}
$$

Iterate on two initial conditions

$$
v^{(k+1)}(\mathbf{x}) = \frac{2}{\bar{T}} \int_0^{\bar{T}} \left(\cos(\omega t) - \frac{\alpha}{2}\right) w^{(k)}(\mathbf{x}, t; v^{(k)})\, dt,
$$

$$
\dot{v}^{(k+1)}(\mathbf{x}) = \frac{2}{\bar{T}} \int_0^{\bar{T}} \left(\cos(\omega t) - \frac{\alpha}{2}\right) \partial_t w^{(k)}(\mathbf{x}, t; v^{(k)})\, dt,
$$

Filter step

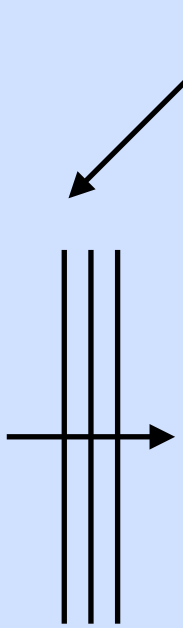
$$
u \approx v^{(k)} + \frac{i}{\omega}\dot{v}^{(k)}
$$

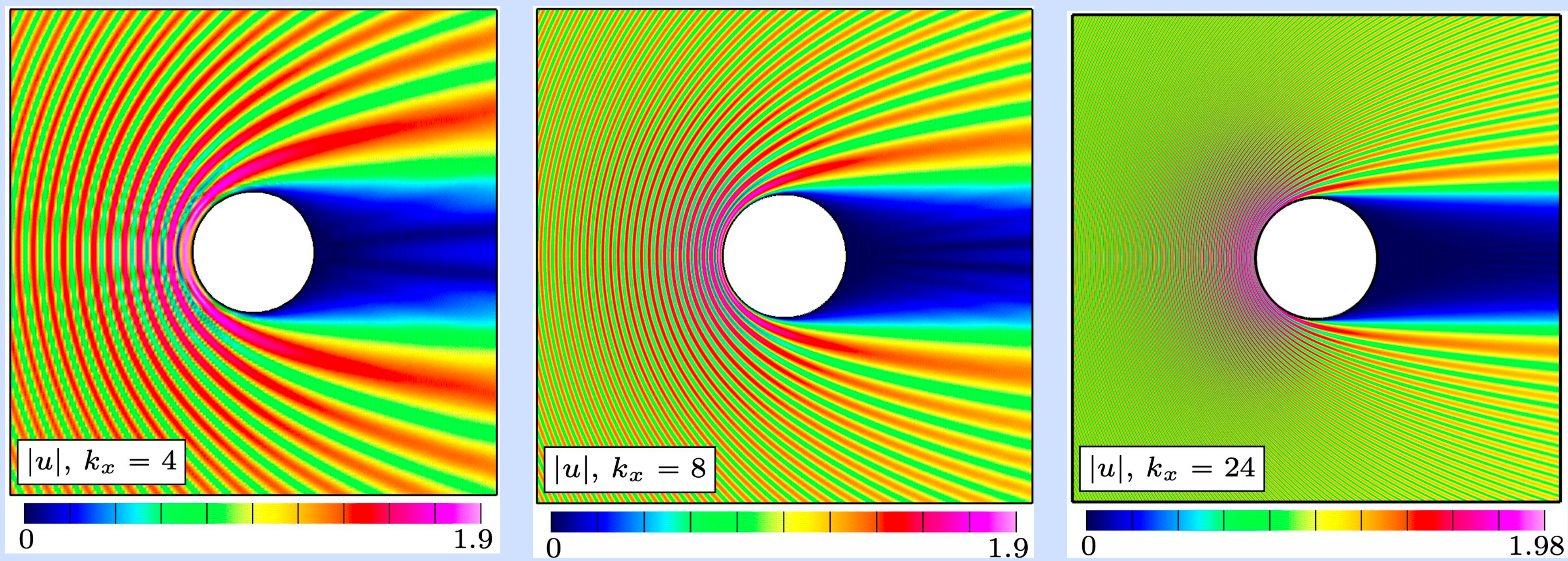
Helmholtz solution

Note: we still solve for a single (real valued) w(x,t)

# WaveHoltz for High Frequency Scattering from a Cylinder

Incident field: $w^I = a^I e^{i(k_x x - \omega t)}$ (we solve for the scattered field)





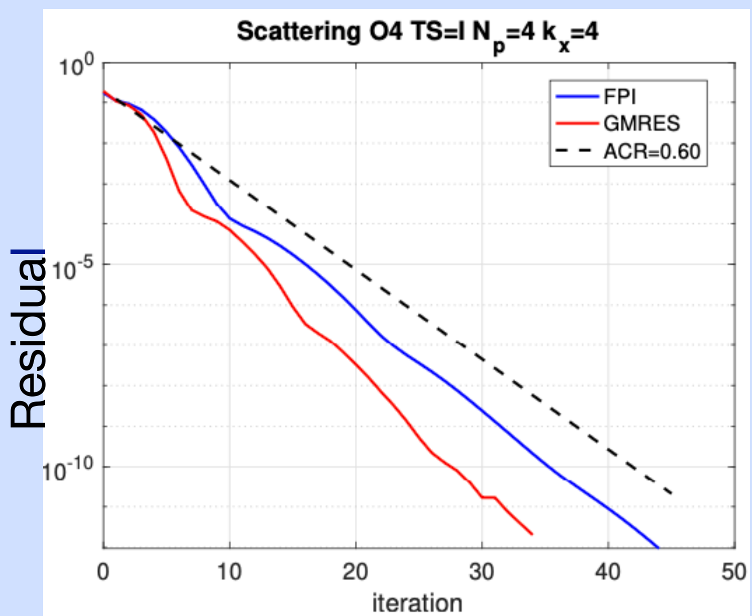

Absolute value of the (complex) total field.

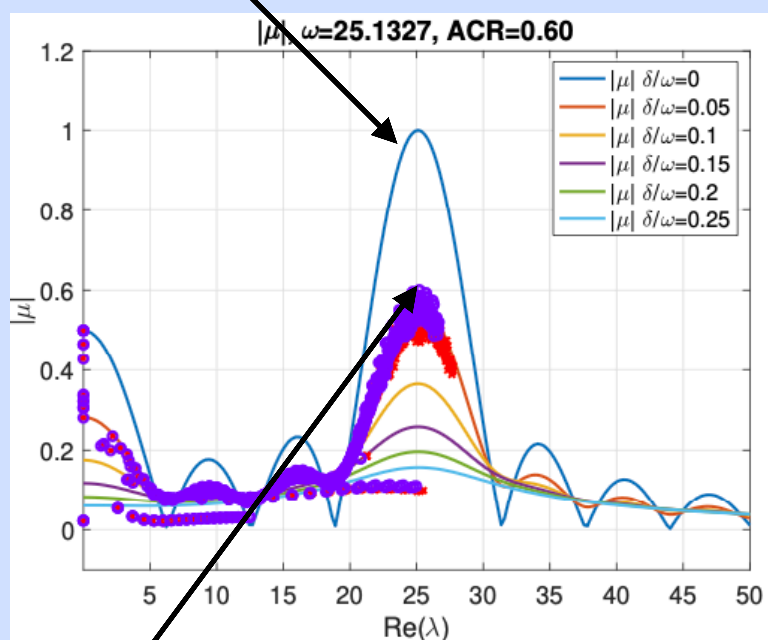
Engquist Majda radiation boundary conditions are used on the outer boundaries.

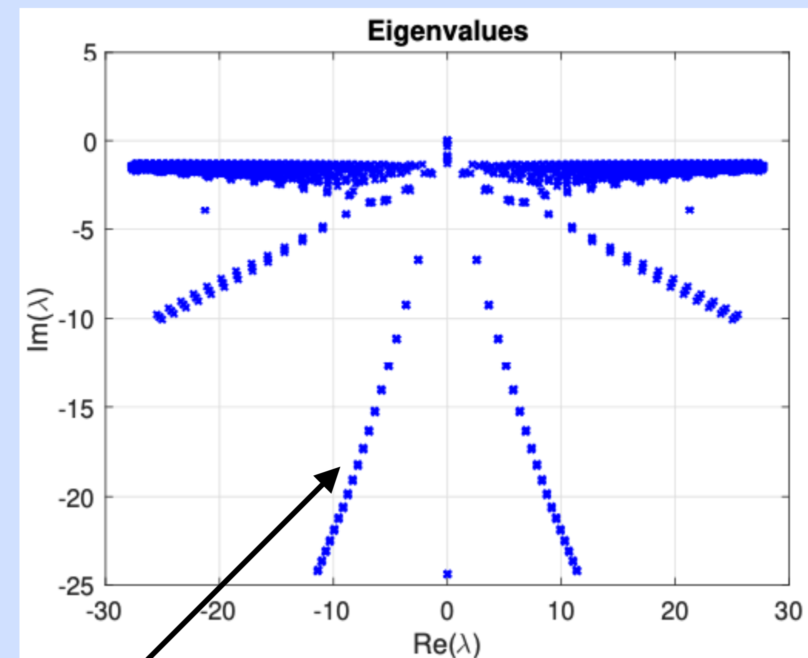
# WaveHoltz convergence for scattering from a cylinder

The radiation boundary conditions add damping (energy loss) to the problem which leads to faster convergence rates of the WaveHoltz algorithm

Convergence curve with no damping




Convergence

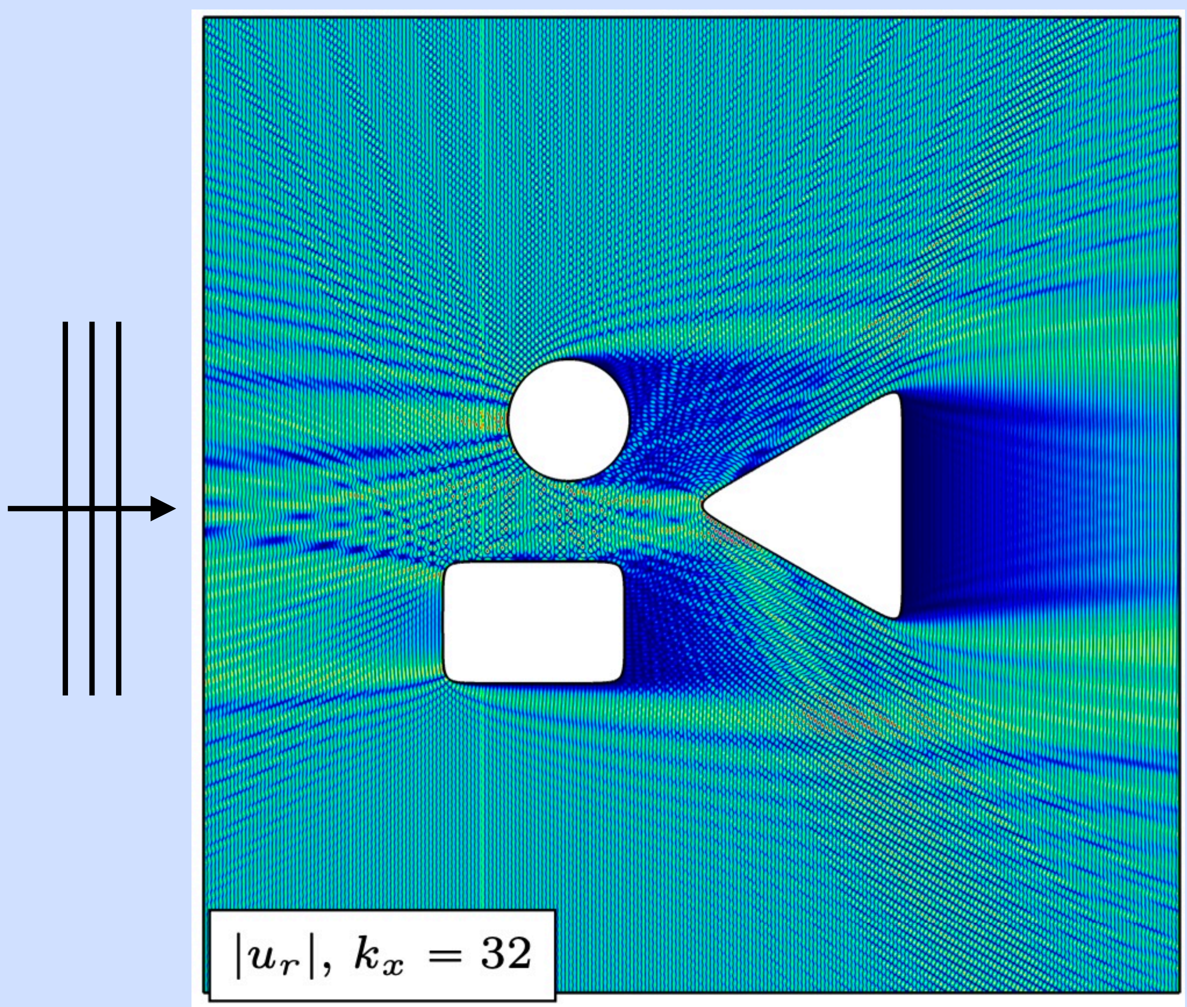
Estimated FPI convergence rate

Eigenvalues of Laplacian with Radiation BCs

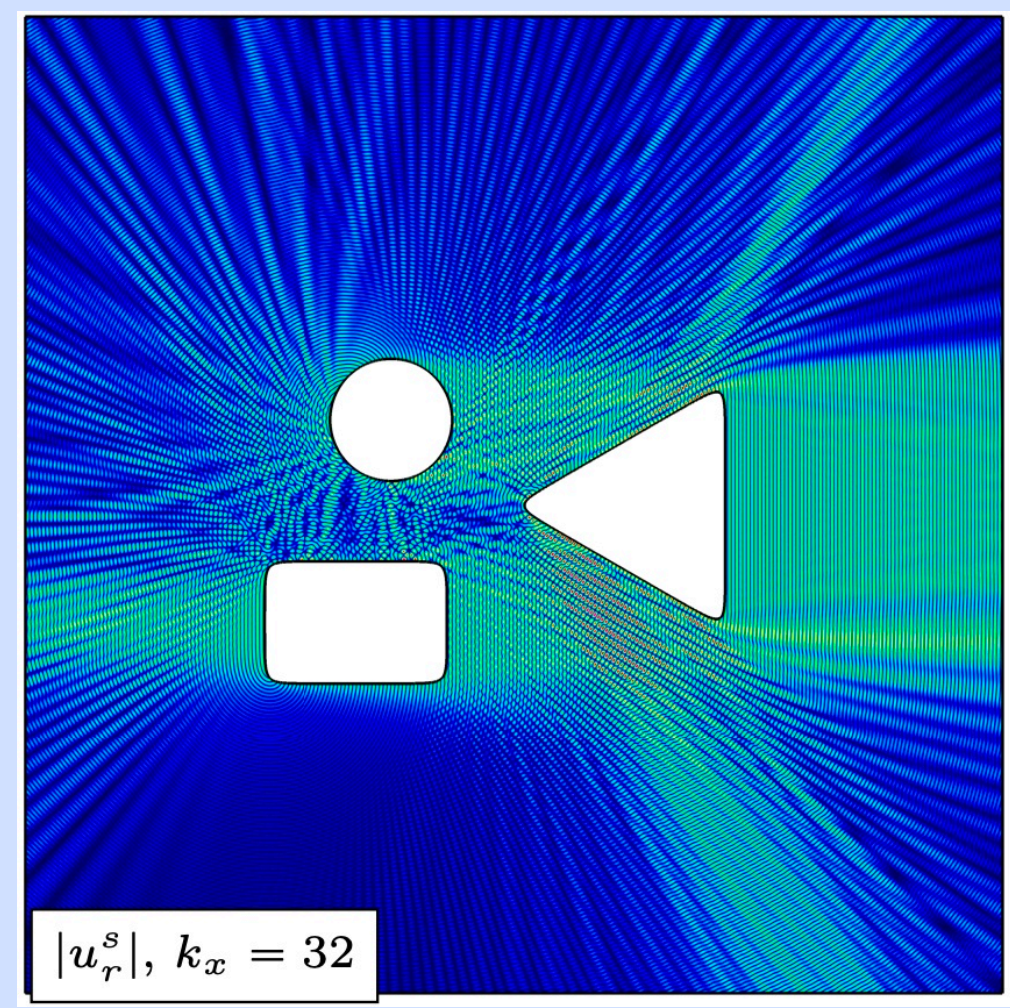
Negative imaginary part implies damping

# High Frequency Scattering from Three Shapes

Open domain problems are often easier to solve than closed domain (energy conserving) problems due to the implicit damping

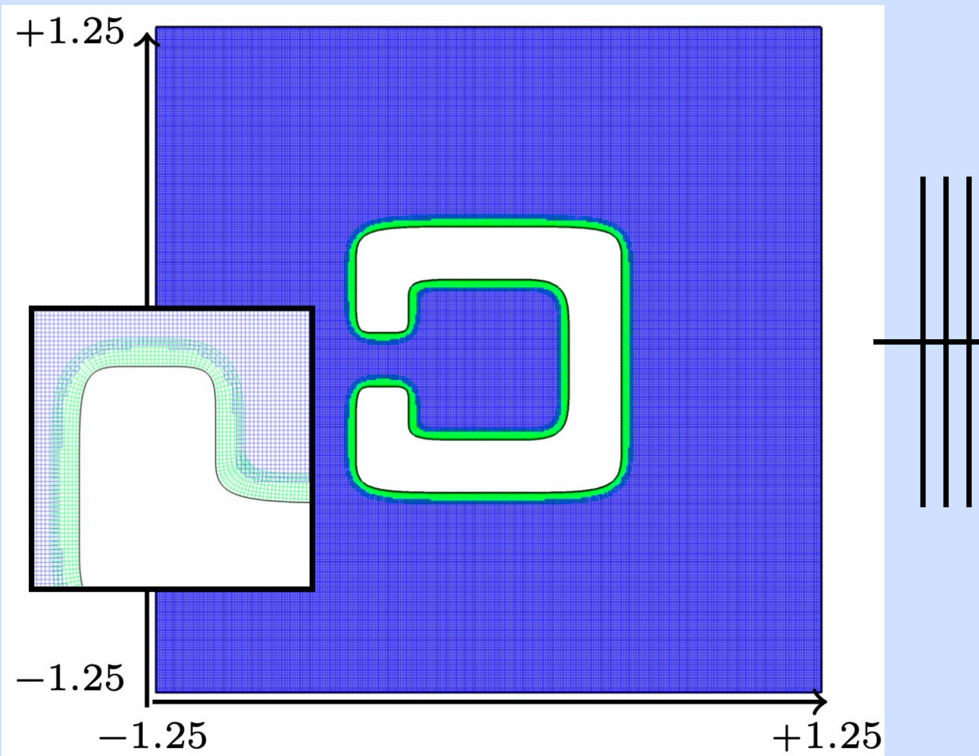


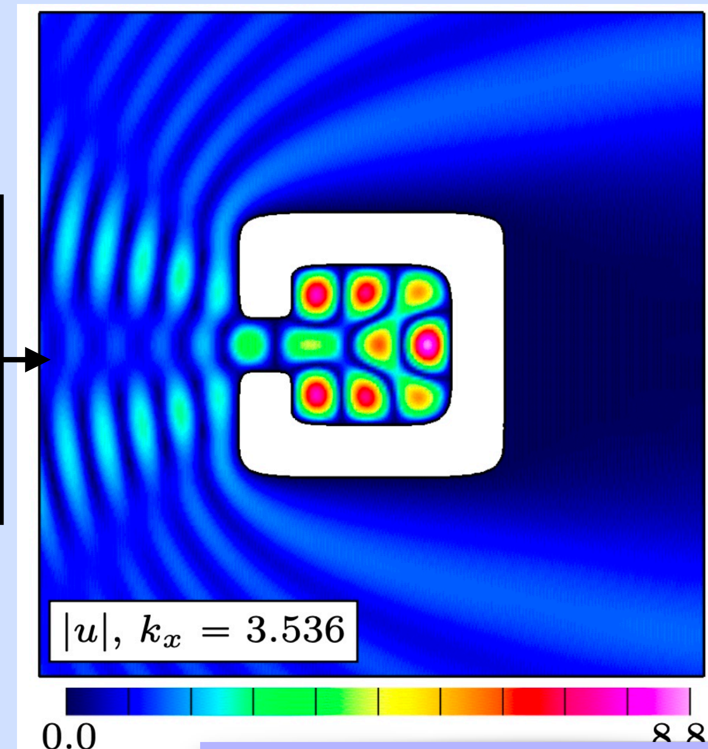

Real part of total field

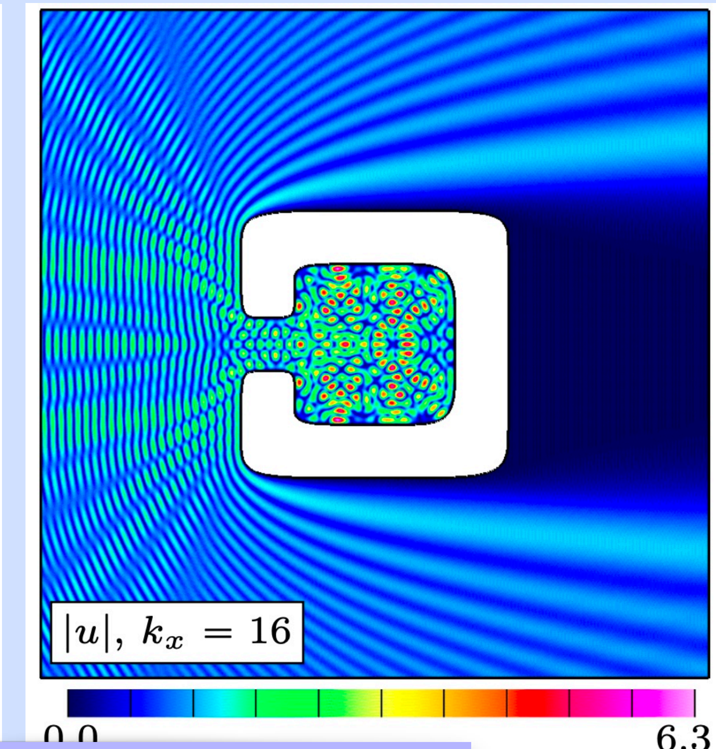



Real part of scattered field

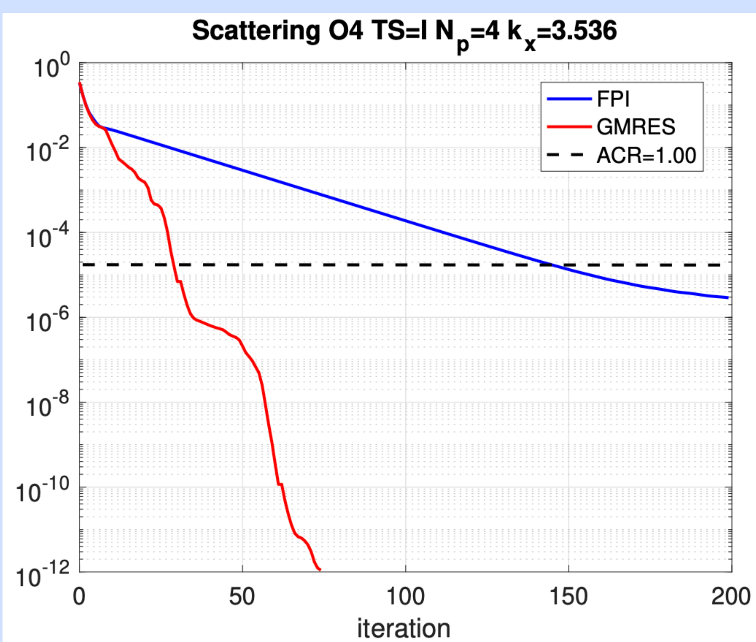
# But, scattering on open domains with resonators can be hard…

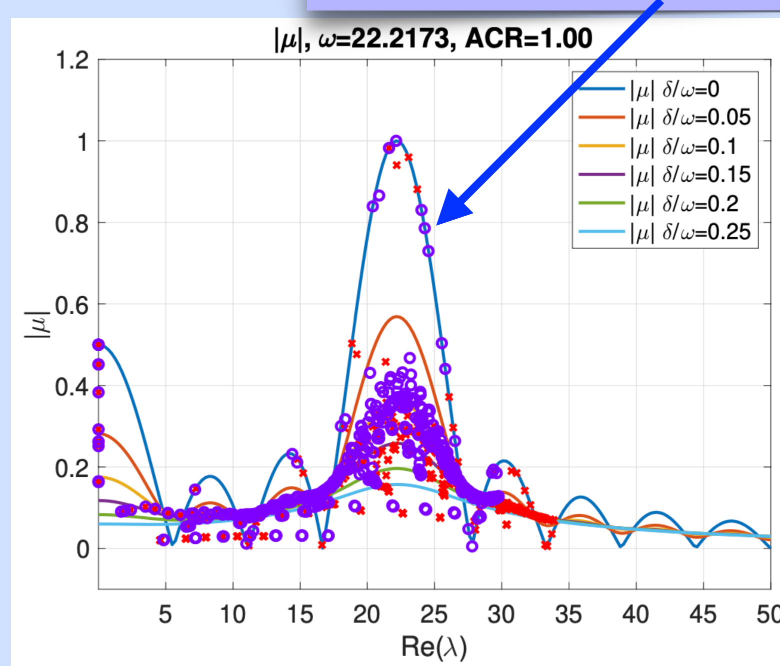


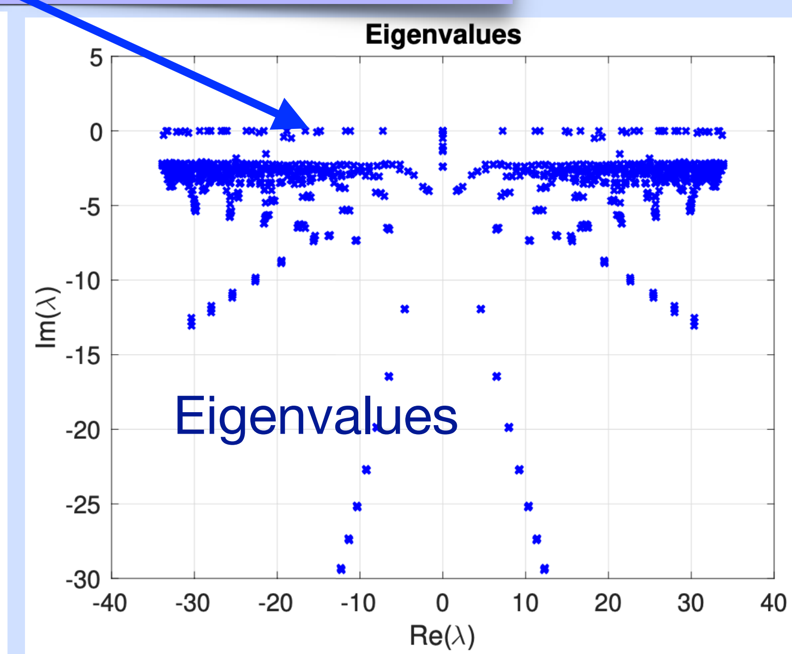




Cavity modes have little damping



# Pollution/Dispersion Errors

Observed: Obtaining an accurate Helmholtz solution requires a much finer grid than expected

Problem: How do we choose the mesh spacing?

# High Frequency problems suffer from Pollution (Dispersion) Errors

Kreiss and Oliger [1972]: Points-per-wave-length (PPW) rule of thumb for the advection equation $u_t - u_x = 0$ ( $p^{\text{th}}$-order accurate FD scheme)

$$PPW \approx 2\pi \left(2\pi a_{p/2}\right)^{1/p} \left[\frac{N_{\text{periods}}}{\epsilon}\right]^{1/p}$$

$N_{\text{periods}} = t_{\text{final}}/T_{\text{period}}$

$\epsilon =$ relative error tol

$p = 2,4,6,\ldots$

$$a_0 = 1,\ a_1 = \frac{1}{6},\ a_2 = \frac{1}{30},\ \alpha_\nu = \frac{\nu}{4\nu + 2}\alpha_{\nu-1}$$

Helmholtz problems: FEM analysis indicates (Babuška et al. 1995)

$$\text{error} \approx K_P\,(kL)\,(kh)^{P+1}$$

$K_p$ (unknown)

Pollution error

$P =$ polynomial degree

$L =$ domain length

$h =$ mesh spacing

$k = \dfrac{\omega}{c}$

# A new PPW rule of thumb for Helmholtz problems

Analysis of a model Helmholtz problem leads to an explicit rule-of-thumb for finite difference approximations or order p (even):

$$\boxed{\mathrm{PPW} \approx 2\pi \left(\pi\, b_{p/2}\right)^{1/p} \left[\frac{N_\Lambda}{\epsilon}\right]^{1/p}}$$

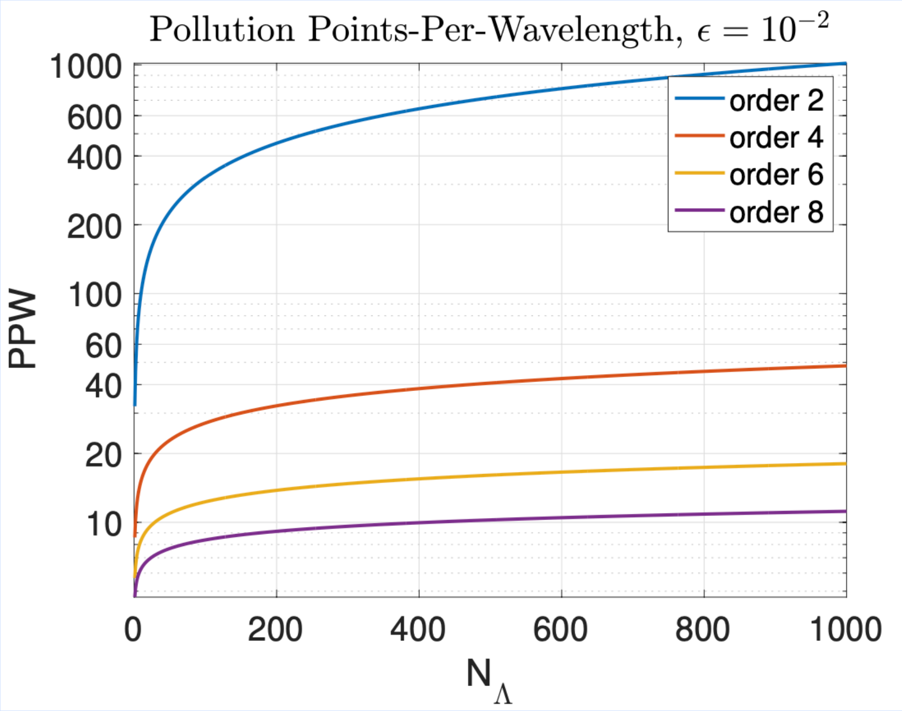
$N_\Lambda$ = domain length in wavelengths

$\epsilon$ = relative error tolerance

$p = 2,4,6,\ldots$

$$b_1 = \frac{1}{12}, b_2 = \frac{1}{90}, b_\mu = \frac{2(\mu!)^2}{(2\mu+2)!}$$

Similar to Kreiss-Oliger PPW for $u_t - u_x = 0$



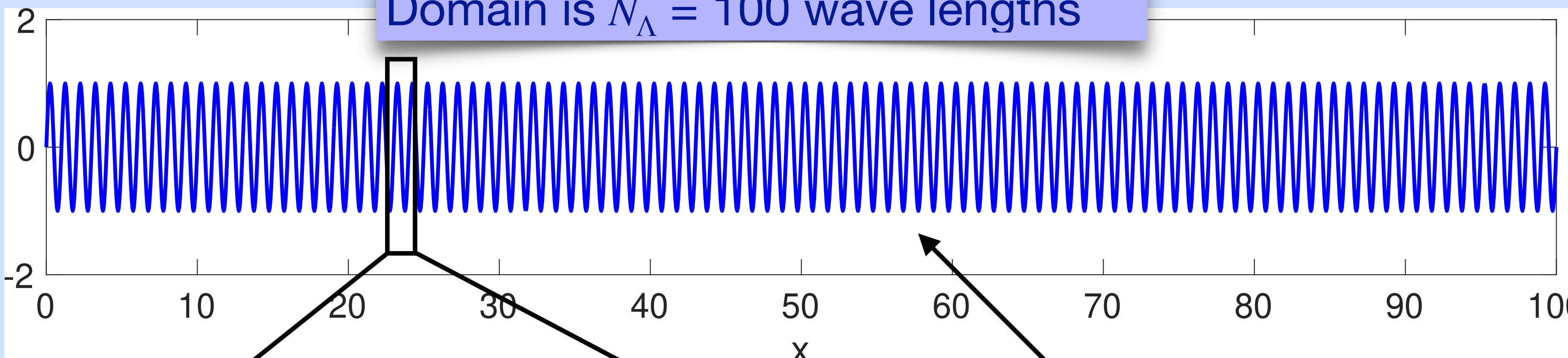

Pollution: PPW increases for

1. fixed frequency $\omega$, increasing L
2. increasing $\omega$, fixed L

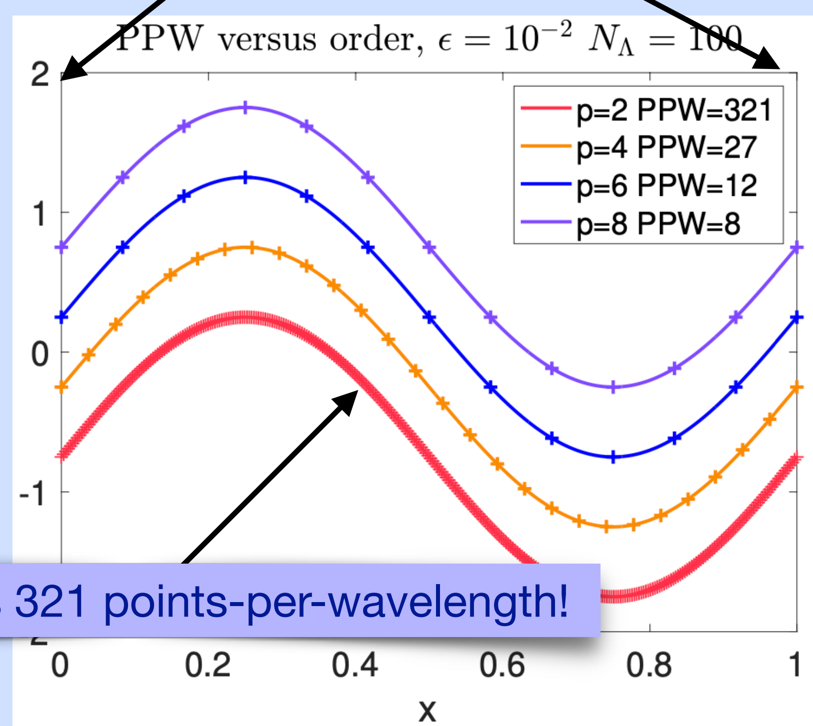
D. Appelo, J.W. Banks, WDH, D.W. Schwendeman, A rule of thumb for choosing points-per-wavelength for finite difference approximations of Helmholtz problems, submitted (2025).

# Illustrating Points-Per-Wavelength Requirements

Domain is $N_\Lambda$ = 100 wave lengths




Cartoon of a Helmholtz Solution




relative error $\epsilon = 10^{-2}$

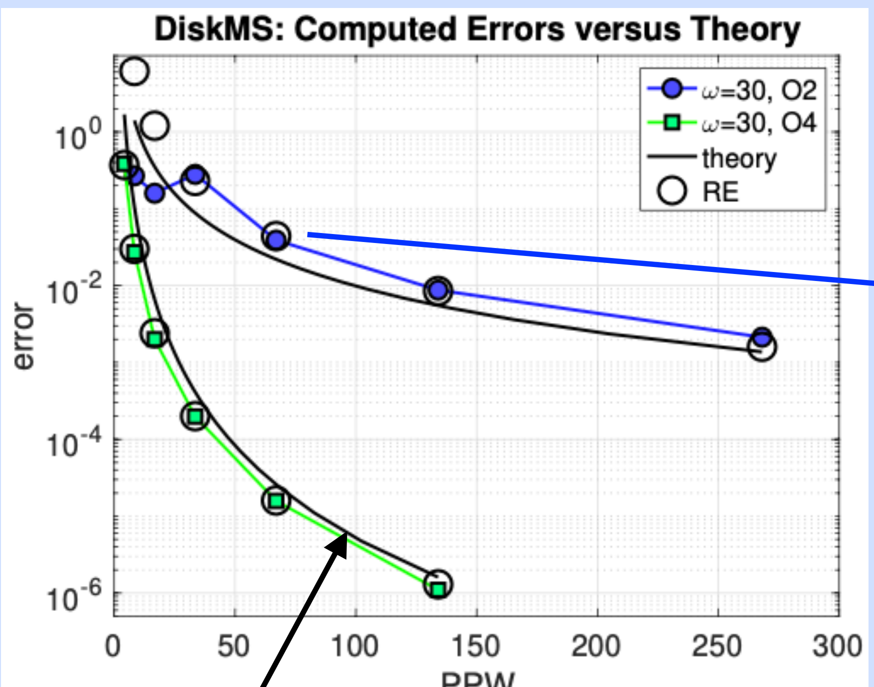
2nd order needs 321 points-per-wavelength!

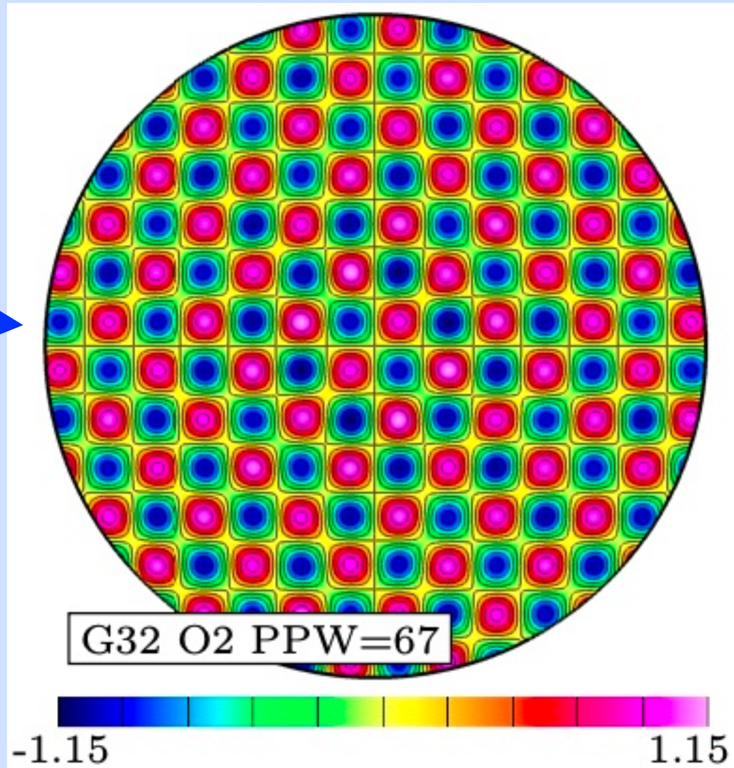
Take away: High-order accuracy is very useful for high-frequency problems

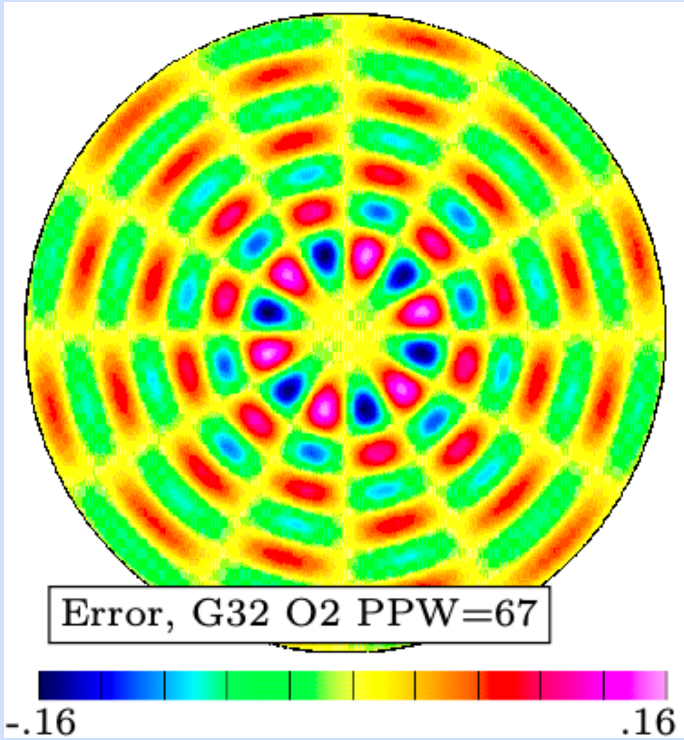
## The rule-of-thumb works reasonably well in practice

### One example: Helmholtz problem on a disk

Manufactured solution




Theory = error based on rule-thumb

RE = estimated errors from Richardson extrapolation for comparison to true errors

Order=2 is under-resolved at PPW=67

## EigenWave: Computing Eigenvalues and Eigenvectors using the WaveHoltz iteration

# EigenWave: Computing Eigenvalues and Eigenvectors

1. Choose a target frequency $\omega$ (target eigenvalue)
2. Apply Waveholtz algorithm with zero forcing
3. Power iteration gives eigenvector $\phi$ but transformed eigenvalue $\beta(\lambda)$
4. Use Rayleigh quotient to obtain eigenvalue $\lambda$ from $\phi$

**Algorithm 1** EigenWave algorithm - power iteration on $\mathcal{A}_h$ to compute one eigenpair $(\lambda, \phi)$.

**function** $[\lambda, \phi] = \text{EIGENWAVE}(\omega, v^{(0)}, N_p)$
 // Input: target frequency $\omega$, initial guess $v^{(0)}$ with norm one, number of periods $N_p$
 $T = 2\pi/\omega, \;\; T_f = N_p T$   $\triangleright$ Period and final time.
 **for** k=0,1,... **do**   $\triangleright$ Start EigenWave iterations.
  $w^{(k)}(\mathbf{x}, 0) = v^{(k)}(\mathbf{x})$   $\triangleright$ Initial condition for wave equation solve.
  $w^{(k)}(\mathbf{x}, t) = \text{SOLVEWAVEEQUATION}(w^{(k)}(\mathbf{x}, 0), T_f)$   $\triangleright$ Solve for $\mathbf{w}(\mathbf{x}, t)$ for $t \in [0, T_f]$.
  $v^{(k+1)}(\mathbf{x}) = \dfrac{2}{T_f} \displaystyle\int_0^{T_f} \left( \cos(\omega t) - \frac{1}{4} \right) w^{(k)}(\mathbf{x}, t; v^{(k)})\, dt$   $\triangleright$ $v^{(k+1)} = \mathcal{A}_h v^{(k)}$.
  $\beta^{(k+1)} = (v^{(k+1)}, v^{(k)})$   $\triangleright$ Rayleigh quotient estimate for eigenvalue of $\mathcal{A}_h$
  $v^{(k+1)} = v^{(k+1)} / \|v^{(k+1)}\|$   $\triangleright$ Normalize
  **if** $\|v^{(k+1)} - \text{sign}(\beta^{(k+1)})\, v^{(k)}\| <$ tolerance **then**   $\triangleright$ $\text{sign}(\beta^{(k+1)}) = \pm 1$
   break from loop
  **end if**
 **end for**   $\triangleright$ End EigenWave iterations.
 $\phi(\mathbf{x}) = v^{(k+1)}(\mathbf{x})$   $\triangleright$ Approximate eigenfunction.
 $\lambda = \sqrt{(\phi, -\mathcal{L}\phi)}$   $\triangleright$ Approximate eigenvalue of $\mathcal{L}$ from a Rayleigh quotient.
**end function**

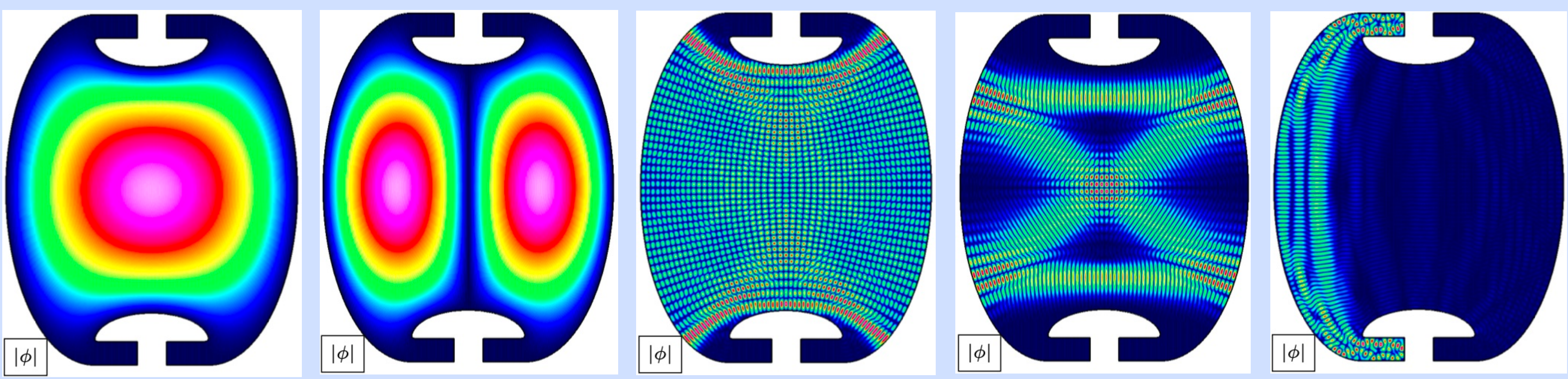
# EigenWave can be used with existing Eigenvalue software

1. Choose target $\omega$ to find eigenvalues anywhere in the spectrum.
2. No need to invert a shifted Laplacian $\Delta_h + \sigma I$ to find interior eigenvalues (this indefinite matrix is hard to invert by iteration)
3. Use existing high-quality Arnoldi based eigenvalue software in a matrix-free fashion (e.g. Implicitly Restarted Arnoldi Method, or Krylov-Schur)
4. O(N) algorithm as the mesh is refined

| EigenWave: double ellipse, ts=implicit, $\omega = 11$, $N_p = 6$, KrylovSchur | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| order | num eigs | wave solves | time-steps per period | wave-solves per eig | time-steps per-eig | max eig-err | max evect-err | max eig-res |
| 2 | 331 | 768 | 10 | 2.3 | 139 | 8.72e-14 | 7.29e-10 | 6.01e-10 |
| 4 | 324 | 768 | 10 | 2.4 | 142 | 6.95e-13 | 2.46e-10 | 1.71e-09 |

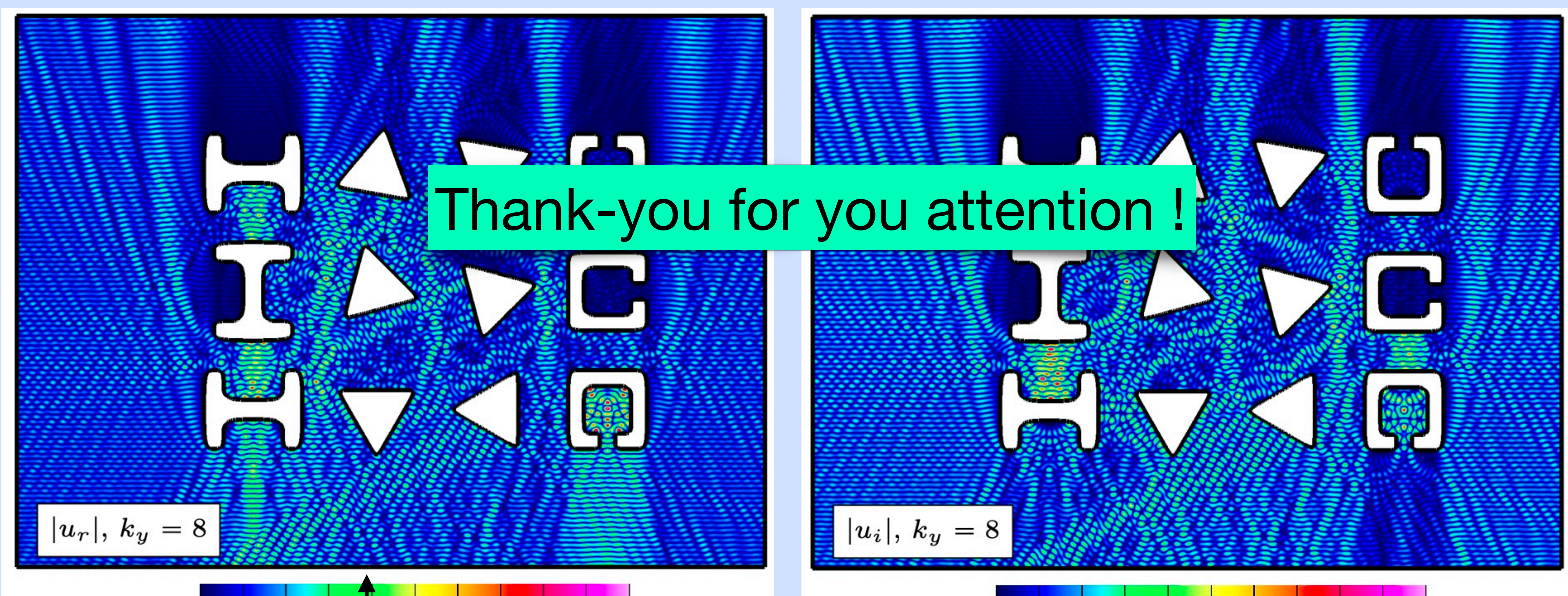
Only 2.4 wave-solves per eigenpair

Eigenwave results match direct computation



## Summary: WaveHoltz

1. The WaveHoltz algorithm can be used to solve Helmholtz problems by time-filtering solutions to the wave equation.
2. Implicit time-stepping, GMRES, and deflation can accelerate the convergence and leads to an O(N) algorithm at fixed frequency.
3. High-order accuracy to over-come pollution errors.
4. EigenWave: solve for eigenpairs using the WaveHoltz iteration.




Thank-you for you attention !