Accelerating Helmholtz solvers using an outer multigrid iteration

Chris Stolk

Univ. of Amsterdam

TUDelft Helmholtz Workshop, May 18, 2015

The Helmholtz equation

The Helmholtz equation reads

$$(-\Delta - k^2)u(x) = f(x).$$

with $k(x) = \frac{\omega}{c(x)}$, where ω = angular frequency and $c(x)$ = medium velocity.

Setup

- Rectangular domain in $\mathbb{R}^n$, $n = 2, 3$, with Dirichlet boundary conditions. Regular mesh discretizations.
- Damping layers for simulating an unbounded domain using the perfectly matched layer or simply an imaginary contribution to k. PML means that

$$\frac{\partial}{\partial x_j} \text{ is replaced by } \frac{1}{1 + i\omega^{-1}\sigma(x_j)} \frac{\partial}{\partial x_j}.$$

- High-frequency regime: domain size $\gg$ wavelength (e.g. ~ 100 wavelengths/domain)

Issues for today

Denote λ = wavelength,
h = grid spacing,
G = # gridpoints per wavelength (ppw) $= \frac{2\pi}{hk}$.

Issues for today:

- Numerical dispersion: For standard schemes we need large G or high order.
- Multigrid: Improved performance at coarse meshes downto $G = 3$ at the coarse level
- A hybrid domain decomposition + multigrid solver.

Numerical dispersion

In 1-D, propagating waves $u = e^{i\xi x}$ satisfy

$$(-\frac{d^2}{dx^2} - k^2)e^{i\xi x} = 0,$$

hence $\xi^2 - k^2 = 0$ or $\xi = \pm k = \pm\frac{\omega}{c}$ and hence $\lambda = \frac{2\pi}{|\xi|} = \frac{2\pi}{k}$.

Using second order finite differences, the homogeneous Helmholtz equation becomes

$$\frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} - k^2 u_i = 0.$$

Inserting $u_i = e^{i\xi x_i}$ leads to the equation

$$\frac{2 - 2\cos(h\xi)}{h^2} - k^2 = 0 \qquad ((*))$$

with solution

$$\xi_{\mathrm{FD2}} = \pm 2h^{-1}\arcsin(\frac{hk}{2})$$

Since $|\xi_{\mathrm{FD2}}| \neq k$ the numerical solution has wave length errors called numerical dispersion. The relative error in $\frac{\xi}{\omega}$ is called the phase slowness error. It will be denoted by $E(\nu, G)$, with ν the direction in S^{n-1}. The same can be done in 2-D and 3-D.

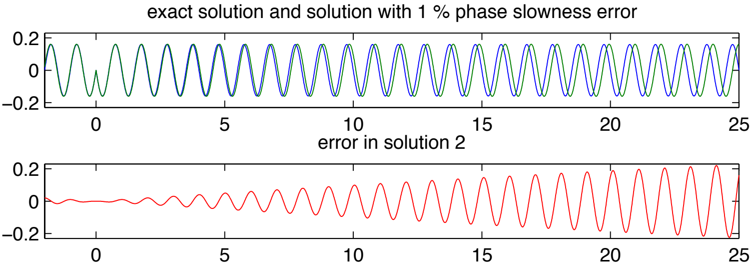
Effect of numerical dispersion

To show the effects of numerical dispersion, consider the equation

$$(-\frac{d^2}{dx^2} - k^2)e^{i\xi\cdot\mathbf{x}} = \delta$$

The exact solution is given by $u = \frac{-i}{2k}e^{ik|x|}$.

Exact solution vs. solution with 1 % phase slowness error:

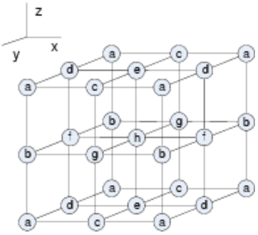


We see that phase error in solution $= \frac{\text{distance}}{\lambda} \cdot E$. We should require at least $E \lesssim 10^{-4}$.

Compact stencil discretizations

Idea: Discretize using $3 \times 3 \times 3$ cubic stencil

$$
\begin{aligned}
&(H_{\text{compact}} u)_{i,j,k} \stackrel{\text{def}}{=} \\
&A_0 u_{i,j,k} \\
&+ A_1 \big(u_{i-1,j,k} + u_{i+1,j,k} + u_{i,j-1,k} \\
&\qquad + u_{i,j+1,k} + u_{i,j,k-1} + u_{i,j,k+1} \big) \\
&+ A_2 \big(u_{i-1,j-1,k} + \ldots + u_{i,j+1,k+1} \big) \\
&+ A_3 \big(u_{i-1,j-1,k-1} + \ldots + u_{i+1,j+1,k+1} \big)
\end{aligned}
$$



with $A_0, \ldots, A_3$ chosen depending on G too minimize phase errors.

Many choices exist (Babuska et al., 1995; Jo, Shin and Suh, 1998; Operto et al., 2007; Chen et al., 2012; Turkel et al., 2013).

Interpolated optimized finite differences

- Consider all symmetric second order discretizations. These are described by five parameters α_j, $j = 1, \dots, 5$, with

$$A_0 = 6\alpha_4 - (kh)^2\alpha_1 \qquad A_2 = -\tfrac{1}{2}\alpha_5 + \tfrac{1}{2}(1 - \alpha_4 - \alpha_5) - (kh)^2\tfrac{1}{12}\alpha_3$$

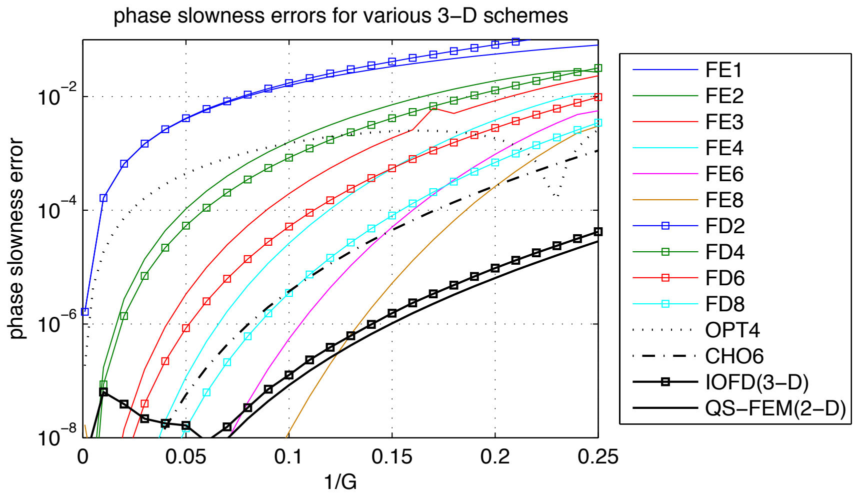
$$A_1 = -\alpha_4 + \alpha_5 - (kh)^2\tfrac{1}{6}\alpha_2 \qquad A_3 = -\tfrac{3}{4}(1 - \alpha_4 - \alpha_5) - (kh)^2\tfrac{1}{8}(1 - \alpha_1 - \alpha_2 - \alpha_3)$$

- Let the α_j slowly vary with $1/G$ using Hermite interpolation from control values
- Optimize the coefficients using nonlinear least squares with $0 \leq 1/G \leq 0.4$, i.e. downto 2.5 ppw.

$1/G$	α_1	$\frac{\partial\alpha_1}{\partial(1/G)}$	α_2	$\frac{\partial\alpha_2}{\partial(1/G)}$	α_3	$\frac{\partial\alpha_3}{\partial(1/G)}$	α_4	$\frac{\partial\alpha_4}{\partial(1/G)}$	α_5	$\frac{\partial\alpha_5}{\partial(1/G)}$
0.0000	0.517047	-0.128231	0.333081	0.002857	0.283241	-0.000089	0.694875	-0.032150	0.275886	0.003602
0.0125	0.523738	-0.038278	0.324029	0.014698	0.280697	-0.010244	0.706215	-0.107275	0.254147	0.003752
0.0250	0.530888	0.026484	0.313399	-0.058155	0.279935	-0.015825	0.708390	-0.066629	0.248576	0.016901
0.0500	0.537095	0.039560	0.303340	-0.063072	0.279560	-0.092408	0.708425	-0.094977	0.244634	-0.014794
0.1000	0.542482	0.090854	0.292077	-0.164698	0.278376	-0.146901	0.701350	-0.181811	0.244231	-0.005689
0.1500	0.546494	0.054652	0.280352	-0.260799	0.276818	0.040023	0.690703	-0.239478	0.243554	-0.027007
0.2000	0.549472	0.086849	0.266004	-0.399426	0.277537	0.090829	0.678083	-0.249610	0.241806	-0.063416
0.2500	0.550195	-0.047748	0.251441	-0.225406	0.278403	-0.007111	0.665015	-0.269271	0.238819	-0.060470
0.3000	0.549247	-0.003809	0.235504	-0.389836	0.278536	0.001697	0.653948	-0.162012	0.234406	-0.116501
0.3500	0.540024	-0.340977	0.225416	-0.096558	0.281206	0.188504	0.642841	-0.285104	0.229717	-0.102619
0.4000	0.521570	-0.406300	0.220498	-0.113976	0.287583	0.107225	0.630481	-0.205847	0.225579	-0.066426

(details on arXiv:1504.01609)

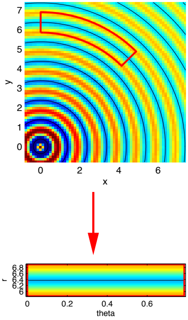
Comparison of phase slowness errors



QS-FEM (2-D) and IOFD (3-D) have small dispersion errors for $G \gtrsim 4$.
Numerical simulations support this.

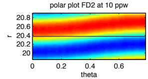
Simulations at constant k (2-D)

Polar plots

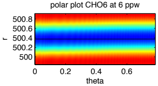


(spline interpolation)

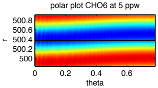
FD2, 10ppw, 20wl



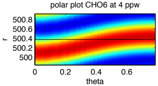
CHO6, 6ppw, 500wl



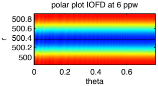
CHO6, 5ppw, 500wl



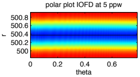
CHO6, 4ppw, 500wl



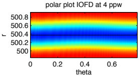
IOFD, 6ppw, 500wl



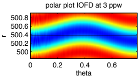
IOFD, 5ppw, 500wl



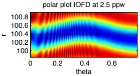
IOFD, 4ppw, 500wl



IOFD, 3ppw, 500wl



IOFD, 2.5ppw, 100wl



Multigrid

IOFD gives very small phase errors at coarse meshes. Can we use this speed up the solution at finer meshes?

- Multigrid with IOFD discretization used at the coarse level.
- Two-grid cycle
  - ν iterations of ω-Jacobi (or similar) (presmoothing)
  - compute error and restrict to coarse mesh
  - coarse grid correction: coarse level solve with error as r.h.s.
  - interpolate correction to fine mesh and add it to solution
  - ν iterations of ω-Jacobi (or similar) (postsmoothing)
- the coarse grid correction handles small wave numbers, the smoothing steps the large wave numbers
- Two-grid cycle can be iterated or applied as preconditioner for GMRES, and it can also be used recursively

Testing multigrid with IOFD

Tested multigrid with IOFD used at the coarse level (S., Ahmed and Bhowmik, SIAM J. Sci. Comput. 2014)

- IOFD method at the coarse level was modified to minimize the phase speed differences with the fine level discretization. Tested IOFD-IOFD, FD2-IOFD
- Varied multigrid parameters
- Convergence analysis in Fourier domain
- Tested convergence in numerical simulations
- Tested standard multigrid in the same way

Two-grid convergence factors

Convergence factors describe the error reduction.
Can be computed using Fourier analysis on $\mathbb{R}^n$ for $k =$ constant.
A small uniform damping must be added $\operatorname{Im} k = \alpha \operatorname{Re} k$, $\alpha = 0.0025$ or $\alpha = 0.01$.

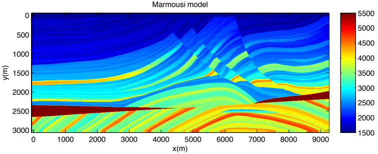
	standard FD2-Galerkin $\alpha = 0.01$, **10 ppw**			FD2-OPT $\alpha = 0.0025$, **3.5 ppw**		
	$\nu = 1$	2	3	$\nu = 2$	3	4
Jac0.6	> 1	> 1	> 1	> 1	> 1	0.557
Jac0.7	> 1	> 1	> 1	> 1	0.685	0.307
Jac0.8	> 1	> 1	> 1	> 1	0.362	0.209
Jac0.9	> 1	> 1	> 1	> 1	> 1	> 1
Jacobi	> 1	> 1	> 1	> 1	> 1	> 1

Multigrid with optimized FD works with 3.5 ppw at coarse level, and very small damping. Standard multigrid requires $\gtrsim 10$ ppw and increased damping.

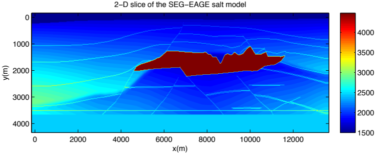
Two-grid iteration count for GMRES

Iterations for residual reduction by 10^{-6} with PML bdy conditions.

Marmousi velocity model



2-D slice of SEG-EAGE model

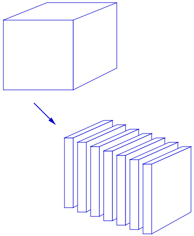


	constant 2400 × 2400		Marmousi 4600 × 750		salt model 2700 × 836	
ppw	freq	its	freq	its	freq	its
5	480	29	150	23	60	18
6	400	8	125	11	50	8
7	342.9	6	107.1	9	42.9	7
8	300	5	93.8	8	37.5	6
9	266.7	5	83.3	7	33.3	6
10	240	4	75	6	30	5

Conclusion: By using optimized FD as coarse level multigrid works well, even with quite coarse meshes (downto 3 ppw at the coarse level).

Multigrid with inexact coarse level solver

- Use double sweep domain decomposition (S., J. Comp. Phys. 2013) as coarse level solver
- Domain decomposition into thin layers, using PML-based interface conditions and the "double sweep" approach. Converges rapidly even with many subdomains.
- Cheaper than a direct solve and than direct domain decomposition.



Iterations for convergence 1e-6 in Marmousi

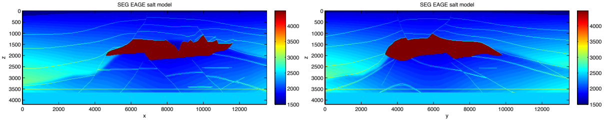
$N_x \times N_y$	h (m)	$\frac{\omega}{2\pi}$ (Hz)	Number of x-subdomains				
			3	10	30	100	300
600 × 212	16	12.5	4	5	6		
1175 × 400	8	25	5	6	6		
2325 × 775	4	50	6	6	6	7	
4625 × 1525	2	100	6	6	6	7	
9225 × 3025	1	200		6	6	6	7

Implementation

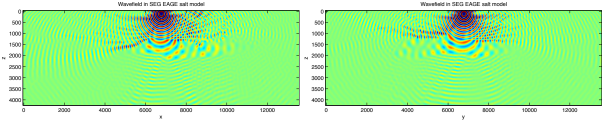
- 3-D implementation using C++ and MPI on Lisa @ Surfsara: use up to 256 cores on 16 nodes, 1 TB memory.
- Subdomain solves are sequential. We use MUMPS on 16 or 32 cores, and pipelining for further parallellization of the domain decomposition method. Scalability of this procedure is an issue.

Example: SEG-EAGE Salt model

Velocity: SEG-EAGE salt model, $670 \times 670 \times 210$ points, $h = 20$ m.



Solution for $f = 12.5$ Hz: *xz* and *yz* slices



frequency	6.25	7.87	9.91	12.5
size	338x338x106	426x426x132	536x536x166	676x676x210
# dof	$1.3 \cdot 10^7$	$2.5 \cdot 10^7$	$5.0 \cdot 10^7$	$1.0 \cdot 10^8$
cores	32	64	128	256
# of rhs.	1	2	4	8
iterations	12	12	13	15
time/rhs.	26	35	45	73

Fast compared to methods in the literature!

Conclusions

- An optimized, compact FD method with very small numerical dispersion was constructed
- In multigrid methods, good convergence with few points per wavelength can be obtained by using coarse level discretizations with accurate phase speeds. Downto 3 ppw at the coarse level.
- When used in combination with double sweep domain decompositions, this results in a very fast solver. Compared for example to a two-grid + shifted Laplacian method (Calandra et al., 2013) we gain roughly a factor 8 in speed.

Further questions

- Better parallellization of the subdomain solves
- Can Shifted-Laplacian methods be used as approximate coarse level solver?
- Non-rectangular domains