Fast 3D Helmholtz Solvers for Seismic Inversion in the Frequency Domain

Russell J. Hewett

Mathematics & CMDA, Virginia Tech

Theory and Experience in Solving Inverse Problems in Geophysics Workshop
Uppsala University

April 10, 2019

Collaborators

- Leonardo Zepeda-Nuñez, Lawrence Berkeley National Lab
- Matthias Taus, TU Wien
- Laurent Demanet, MIT
- Adrien Scheuer, Universitè Catholique de Louvain

FWI in the Frequency Domain

PDE constrained optimization in frequency domain

- $\min J(m) = \frac{1}{2}||d - \mathcal{F}(m)||_2^2$ s.t. $Lu = f$

Advantages:

- No need to invert source time series

$$\hat{f}(\omega) = \mathsf{FFT}(f(t))$$

- Only need specific frequency components

FWI in the Frequency Domain

PDE constrained optimization in frequency domain

- $\min J(m) = \frac{1}{2}||d - \mathcal{F}(m)||_2^2$ s.t. $Lu = f$

Advantages:

- Reduced memory and disk requirements in inverse problem

$$\delta m = -\langle q, \partial_{tt} u_0 \rangle_T = -\int_0^T q(x,t) \partial_{tt} u_0(x,t) dt$$

becomes

$$\delta m = -\left\langle q, -\omega^2 u_0 \right\rangle_\Omega = -\sum_\omega \hat{q}(x,\omega) -\omega^2 \hat{u}_0(x,\omega)$$

- Hybrid modeling: Use time-domain + DFT to achieve frequency domain update

FWI in the Frequency Domain

PDE constrained optimization in frequency domain

- $\min J(m) = \frac{1}{2}||d - \mathcal{F}(m)||_2^2$ s.t. $Lu = f$

Advantages:

- Multiple simultaneous right-hand sides
- With a factorization based method, only need to Helmholtz operator once per domain
- Compare to explicit time-stepping: matvec required for each time step for each source

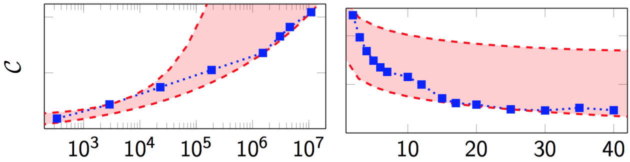
FWI in the Frequency Domain

PDE constrained optimization in frequency domain

- $\min J(m) = \frac{1}{2}||d - \mathcal{F}(m)||_2^2$ s.t. $Lu = f$

Advantages:

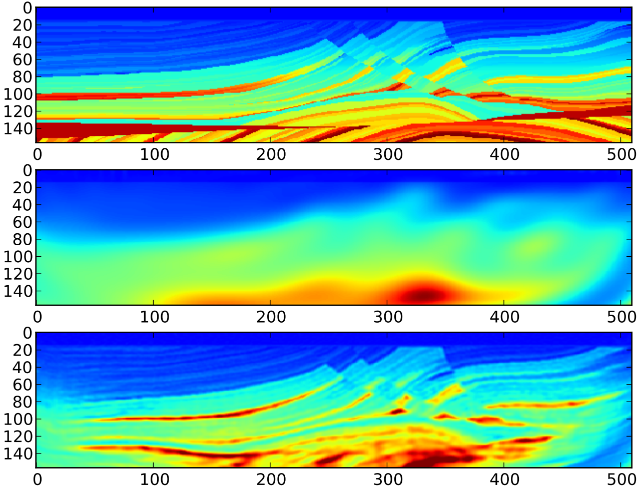
- Heirarchichal frequency "sweeping" $\Rightarrow$ Convergence guarantees



(a) N (b) ω

(E. Beretta, M.V. de Hoop, F. Faucher, O. Scherzer (SIMA 2016))

FWI in the Frequency Domain



Created with PySIT (www.pysit.org).

FWI in the Frequency Domain

PDE constrained optimization in frequency domain

- $\min J(m) = \frac{1}{2}||d - \mathcal{F}(m)||_2^2$ s.t. $Lu = f$

Challenges:

- Helmholtz in high frequency regime
- Helmholtz in 3D at high resolution
- Scalable Helmholtz in HPC environment

Motivation for Sweeping Solvers

Helmholtz at high frequency is **hard**

$$Hu = (-\omega^2 - \triangle)u = f + \mathrm{ABCs}$$

- Frequency ω grows with n
- Computational load N scales with n^d

Motivation for Sweeping Solvers

Helmholtz at high frequency is **hard**

$$Hu = (-\omega^2 - \triangle)u = f + \mathrm{ABCs}$$

- Frequency ω grows with n
- Computational load N scales with n^d

Classical dense direct methods in 3D

- memory-intensive
- hard to parallelize

Motivation for Sweeping Solvers

Helmholtz at high frequency is **hard**

$$Hu = (-\omega^2 - \triangle)u = f + \mathrm{ABCs}$$

- Frequency ω grows with n
- Computational load N scales with n^d

Classical dense direct methods in 3D

- memory-intensive
- hard to parallelize

Multigrid methods

- poor frequency scaling
- down-sampling oscillatory waves is hard

Motivation for Sweeping Solvers

Helmholtz at high frequency is **hard**

$$Hu = (-\omega^2 - \triangle)u = f + \mathrm{ABCs}$$

- Frequency ω grows with n
- Computational load N scales with n^d

Classical dense direct methods in 3D

- memory-intensive
- hard to parallelize

Multigrid methods

- poor frequency scaling
- down-sampling oscillatory waves is hard

Classical iterative schemes

- n_{iter} grows with ω

Sweeping Solvers and Domain Decomposition Methods

Sweeping Solvers/Preconditioners

- First $O(N)$ claim (Engquist and Ying, 2010)
- First $O(N)$ claim w/ domain decomposition (Stolk 2013)

Sweeping Solvers and Domain Decomposition Methods

Sweeping Solvers/Preconditioners

- First $O(N)$ claim (Engquist and Ying, 2010)
- First $O(N)$ claim w/ domain decomposition (Stolk 2013)

Other domain-decomposition methods (DDMs):

- Multifrontal w/ HSS compression (Xia, et al., 2013)
- Hierarchical Poincare-Steklov methods (Gillman, et al., 2014)
- Common challenges:
  - Hazy scalability
  - Issues with rough media

Sweeping Solvers and Domain Decomposition Methods

Sweeping Solvers/Preconditioners

- First $O(N)$ claim (Engquist and Ying, 2010)
- First $O(N)$ claim w/ domain decomposition (Stolk 2013)

Other domain-decomposition methods (DDMs):

- Multifrontal w/ HSS compression (Xia, et al., 2013)
- Hierarchical Poincare-Steklov methods (Gillman, et al., 2014)
- Common challenges:
  - Hazy scalability
  - Issues with rough media

Our approach: DDMs + sweeping w/ polarized traces

- Use direct methods distributed over tractable subproblems
- Glue with boundary integral formulations
- Embedded within iterative scheme

Take-home Messages

1. Polarized traces: **domain decomposition done right**
   - Maximizes leveraging legacy direct solvers in the subdomains
   - Maximal flexibility in parallel computation

Take-home Messages

1. Polarized traces: **domain decomposition done right**
   - Maximizes leveraging legacy direct solvers in the subdomains
   - Maximal flexibility in parallel computation

2. The number of iterations is
   - Weakly dependent on the frequency
   - Weakly dependent on the number L of subdomains
   - Robust to roughness in background model

Take-home Messages

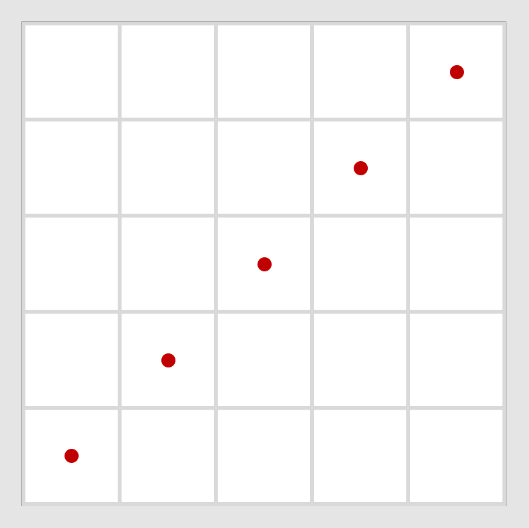
1. Polarized traces: **domain decomposition done right**
   - Maximizes leveraging legacy direct solvers in the subdomains
   - Maximal flexibility in parallel computation

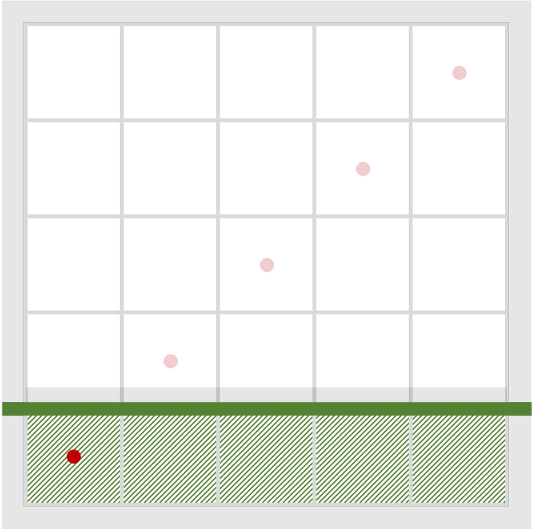
2. The number of iterations is
   - Weakly dependent on the frequency
   - Weakly dependent on the number L of subdomains
   - Robust to roughness in background model

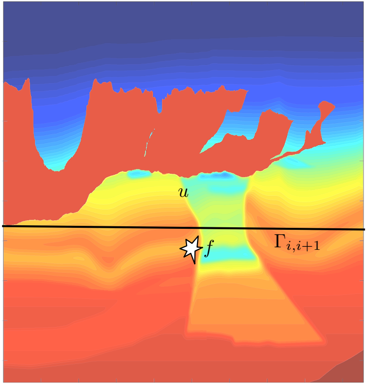
3. Precondition interdomain communication with **polarizing integral conditions**

Take-home Messages

1. Polarized traces: **domain decomposition done right**
   - Maximizes leveraging legacy direct solvers in the subdomains
   - Maximal flexibility in parallel computation
2. The number of iterations is
   - Weakly dependent on the frequency
   - Weakly dependent on the number L of subdomains
   - Robust to roughness in background model
3. Precondition interdomain communication with **polarizing integral conditions**
4. Can achieve sublinear complexity: better than $O(RN)$

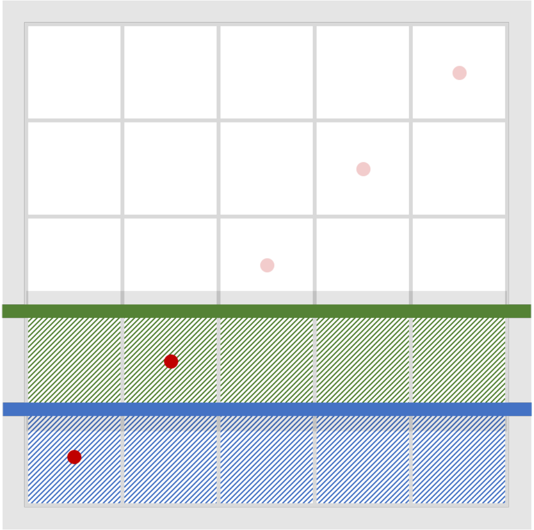


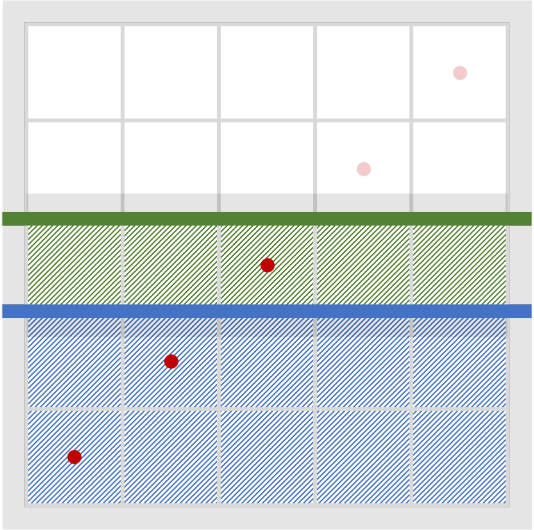


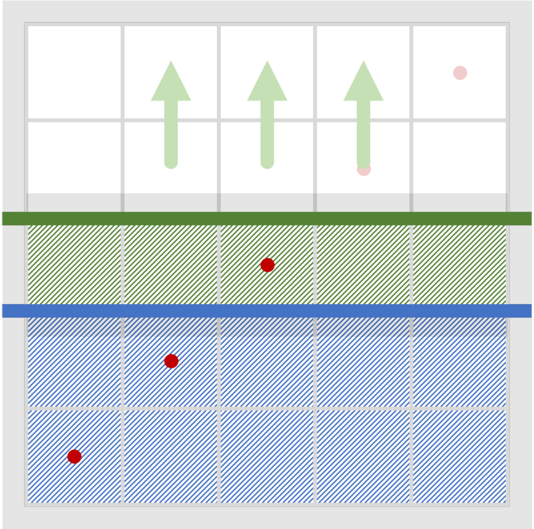


Polarization condition:

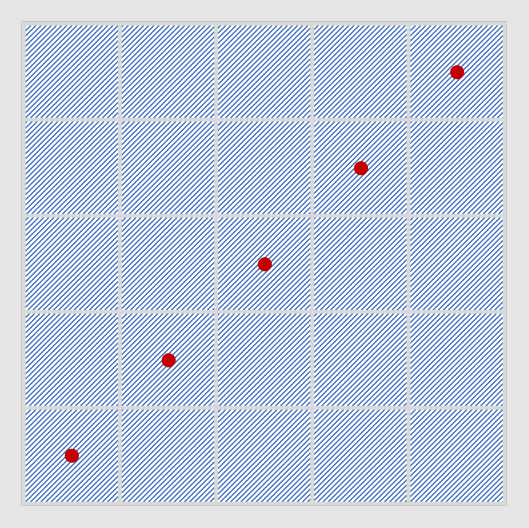
$$0 = -\int_{\Gamma} G(x,y)\partial_{n_y} u^{\uparrow}(y) ds_y + \int_{\Gamma} \partial_{n_y} G(x,y) u^{\uparrow}(y) ds_y$$

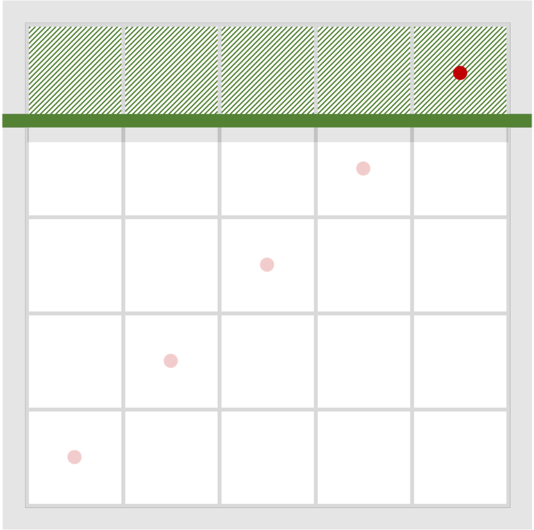


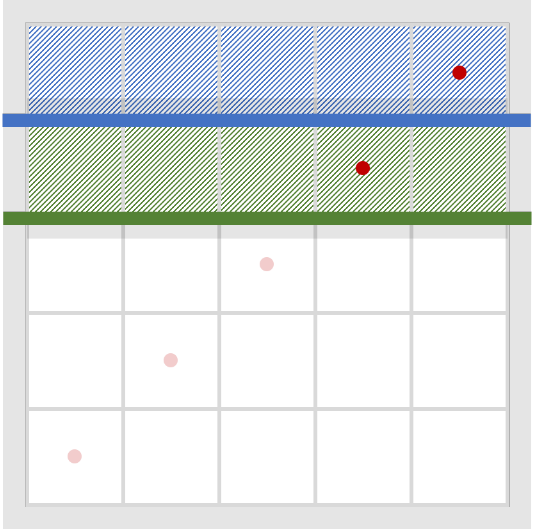


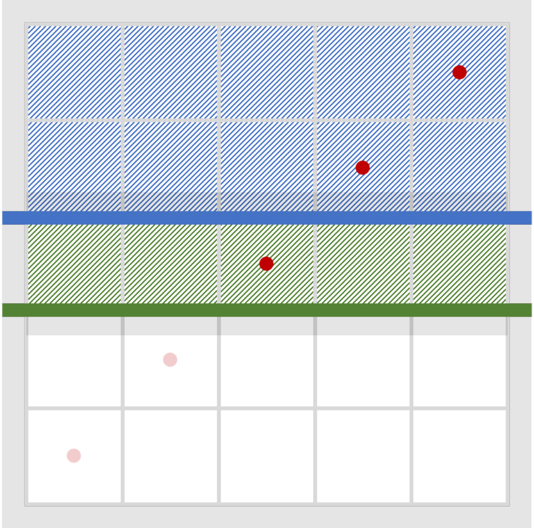


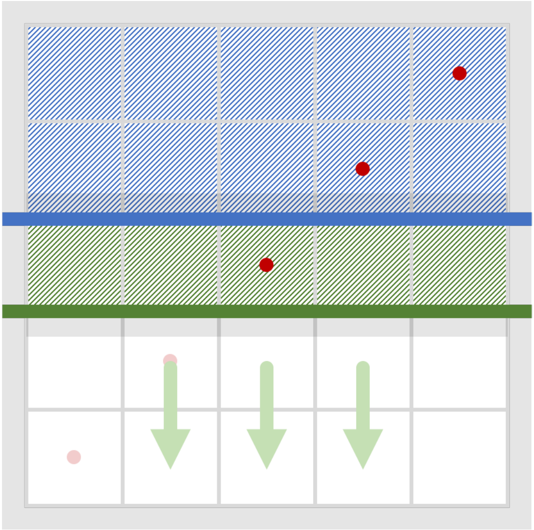
Method Of Polarized Traces

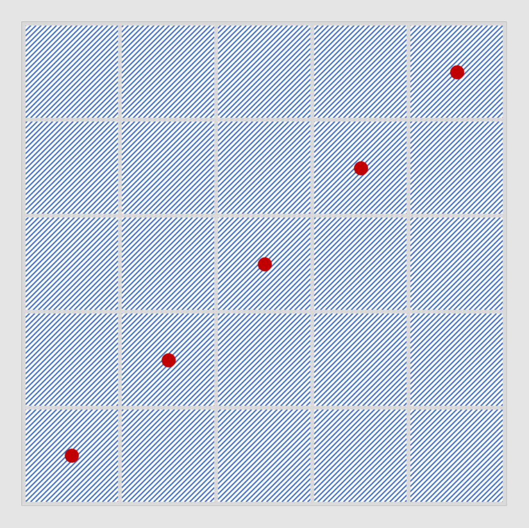






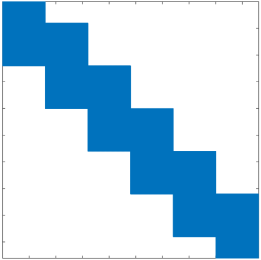






A System for Traces

- Assume local PDE is solved in the bulk
- Traces can be found by solving $\underline{\mathbf{M}}\ \underline{\mathbf{u}} = \underline{\mathbf{f}} = \begin{bmatrix} \mathbf{v}_n^1 \\ \mathbf{v}_1^2 \\ \mathbf{v}_n^2 \\ \vdots \\ \mathbf{v}_1^L \end{bmatrix}$
- $\underline{\mathbf{M}}$ is constructed from dense Green's function blocks. . .



- . . . non-trivial to invert

Polarization

- Annihilation relation:
  - If $\mathbf{u}^{\uparrow}$ is an up-going wavefield, then the annihilator relations are true on the lower-half plane, i.e.

$$\mathcal{G}_i^{\downarrow,\ell}(\mathbf{u}_{-}^{\uparrow}, \mathbf{u}_{1}^{\uparrow}) = 0, \qquad \text{for } i \geq 1.$$

  - If $\mathbf{u}^{\downarrow}$ is a down-going wavefield, then the annihilator relations are true on the upper-half plane, i.e.

$$\mathcal{G}_i^{\uparrow,\ell}(\mathbf{u}_{n}^{\downarrow}, \mathbf{u}_{+}^{\downarrow}) = 0, \qquad \text{for } i \leq n^{\ell}$$

Polarization

1. Seek to solve $\underline{\mathbf{M}}\,\underline{\mathbf{u}} = \underline{\mathbf{f}}$

Polarization

1. Seek to solve $\underline{\mathbf{M}}\,\underline{\mathbf{u}} = \underline{\mathbf{f}}$
2. Transform to underdetermined system

$$\begin{bmatrix} \underline{\mathbf{M}} & \underline{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{bmatrix} = -\underline{\mathbf{f}}$$

Polarization

1. Seek to solve $\underline{\mathbf{M}}\,\underline{\mathbf{u}} = \underline{\mathbf{f}}$
2. Transform to underdetermined system

$$\left[\begin{array}{cc} \underline{\mathbf{M}} & \underline{\mathbf{M}} \end{array}\right] \left[\begin{array}{c} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{array}\right] = -\underline{\mathbf{f}}$$

3. Constrain with annihilation relations

$$\left[\begin{array}{cc} \underline{\mathbf{M}} & \underline{\mathbf{M}} \\ \underline{\mathbf{A}}^{\downarrow} & \underline{\mathbf{A}}^{\uparrow} \end{array}\right] \left[\begin{array}{c} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{array}\right] = -\left[\begin{array}{c} \underline{\mathbf{f}} \\ \underline{\mathbf{0}} \end{array}\right]$$

Polarization

1. Seek to solve $\underline{\mathbf{M}}\,\underline{\mathbf{u}} = \underline{\mathbf{f}}$
2. Transform to underdetermined system

$$\begin{bmatrix} \underline{\mathbf{M}} & \underline{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{bmatrix} = -\underline{\mathbf{f}}$$

3. Constrain with annihilation relations

$$\begin{bmatrix} \underline{\mathbf{M}} & \underline{\mathbf{M}} \\ \underline{\mathbf{A}}^{\downarrow} & \underline{\mathbf{A}}^{\uparrow} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{bmatrix} = - \begin{bmatrix} \underline{\mathbf{f}} \\ \underline{\mathbf{0}} \end{bmatrix}$$

4. Additional minor transformations and permutations

$$\begin{bmatrix} \underline{\mathbf{D}}^{\downarrow} & \underline{\mathbf{U}} \\ \underline{\mathbf{L}} & \underline{\mathbf{D}}^{\uparrow} \end{bmatrix} \underline{\underline{\mathbf{u}}} = \underline{\underline{\mathbf{f}}}$$

Polarization

1. Seek to solve $\underline{\mathbf{M}}\,\underline{\mathbf{u}} = \underline{\mathbf{f}}$
2. Transform to underdetermined system

$$\begin{bmatrix} \underline{\mathbf{M}} & \underline{\mathbf{M}} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{bmatrix} = -\underline{\mathbf{f}}$$

3. Constrain with annihilation relations

$$\begin{bmatrix} \underline{\mathbf{M}} & \underline{\mathbf{M}} \\ \underline{\mathbf{A}}^{\downarrow} & \underline{\mathbf{A}}^{\uparrow} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^{\downarrow} \\ \underline{\mathbf{u}}^{\uparrow} \end{bmatrix} = -\begin{bmatrix} \underline{\mathbf{f}} \\ \underline{\mathbf{0}} \end{bmatrix}$$

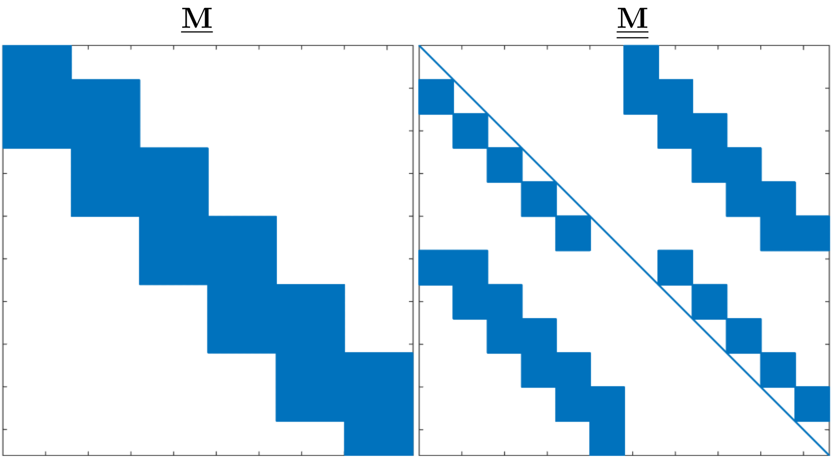
4. Additional minor transformations and permutations

$$\begin{bmatrix} \underline{\mathbf{D}}^{\downarrow} & \underline{\mathbf{U}} \\ \underline{\mathbf{L}} & \underline{\mathbf{D}}^{\uparrow} \end{bmatrix} \underline{\underline{\mathbf{u}}} = \underline{\underline{\mathbf{f}}}$$

5. Final system of equations

$$\underline{\underline{\mathbf{M}}}\,\underline{\underline{\mathbf{u}}} = \underline{\underline{\mathbf{f}}}$$

Structure of Polarized SIE matrix $\underline{\underline{\mathbf{M}}}$



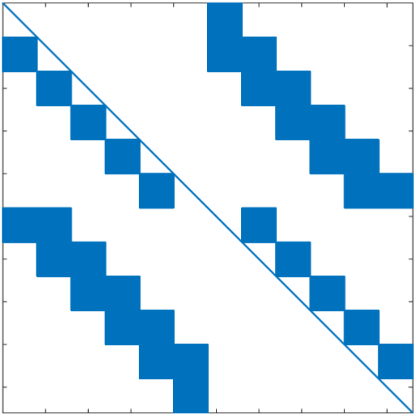
- ... $\underline{\underline{\mathbf{M}}}$ is far easier to invert

Solving for Traces

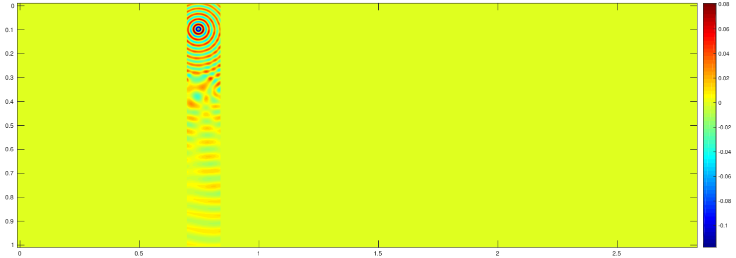
- $$\underline{\underline{\mathbf{M}}} = \begin{bmatrix} \underline{\mathbf{D}}^{\downarrow} & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{D}}^{\uparrow} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \underline{\mathbf{U}} \\ \underline{\mathbf{L}} & \mathbf{0} \end{bmatrix}$$

- Block diagonal, and block upper and lower triangular
  - Perfect for block Gauss-Seidel

- Embed Gauss-Seidel iteration into GMRES to achieve convergence independent of number of layers

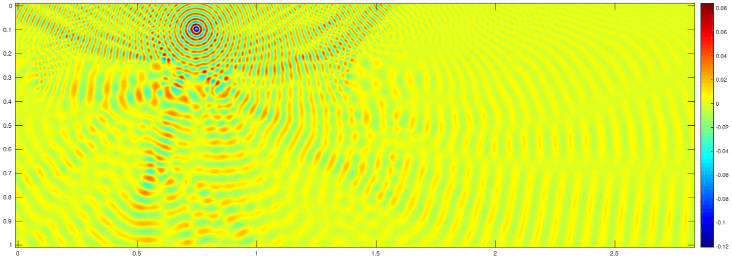


BP 2004 solution



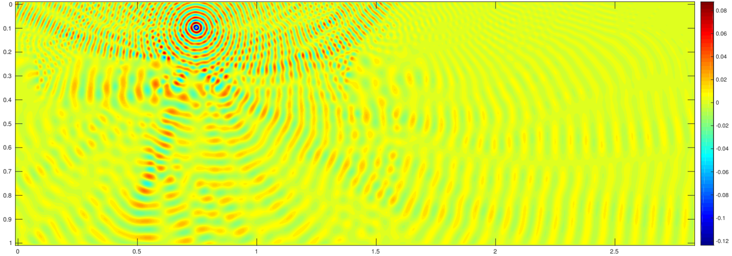
Iteration 0

BP 2004 solution



Iteration 1 (2 domain sweeps)

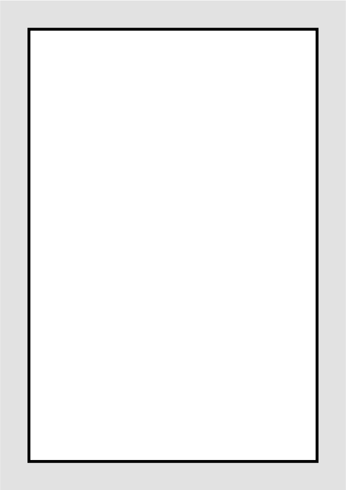
BP 2004 solution



Iteration 2 (4 domain sweeps)

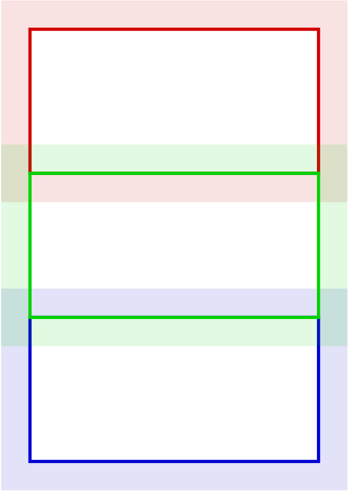
Sources of Parallelism

Domain



Sources of Parallelism: Layers

MPI: Parallelize over Layers

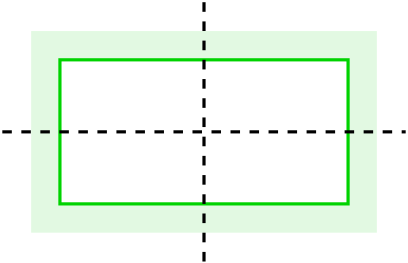


Sources of Parallelism: Layers

MPI: Parallelize within Layers

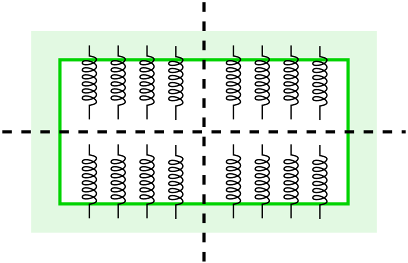
Sources of Parallelism: Local Solver

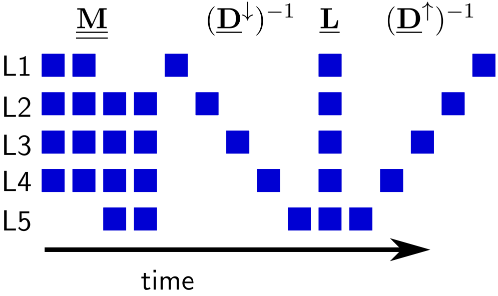
MPI: Multifrontal/Nested dissection



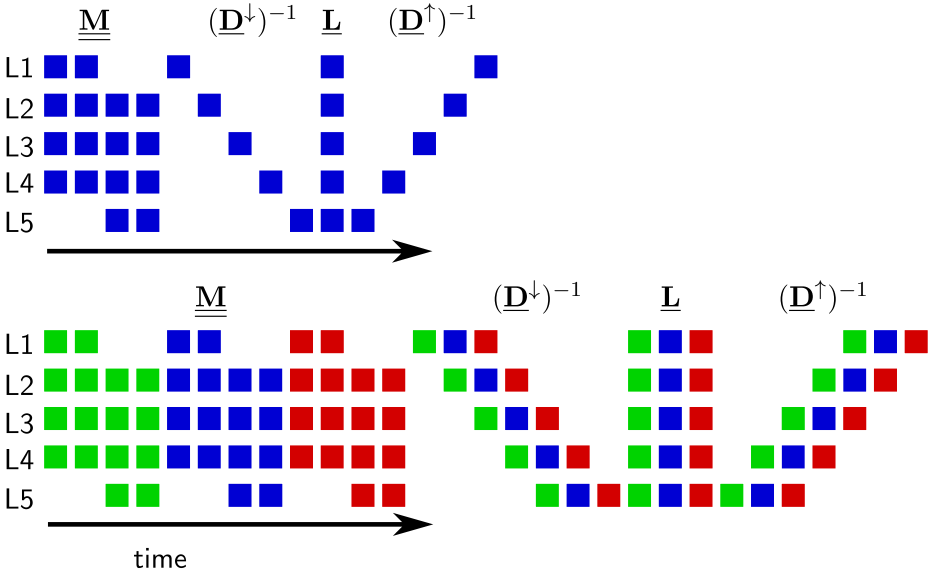
Sources of Parallelism: Local Solver

OpenMP: Parallelize within MPI tasks



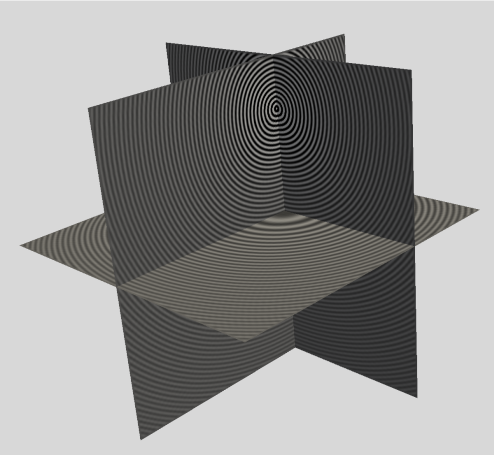

$\underline{\underline{\mathbf{M}}}$
$(\underline{\mathbf{D}}^{\downarrow})^{-1}$
$\underline{\mathbf{L}}$
$(\underline{\mathbf{D}}^{\uparrow})^{-1}$
L1
L2
L3
L4
L5
time

Pipelining: Parallelizing the Sequential Part



Performance of 3D Polarized Traces

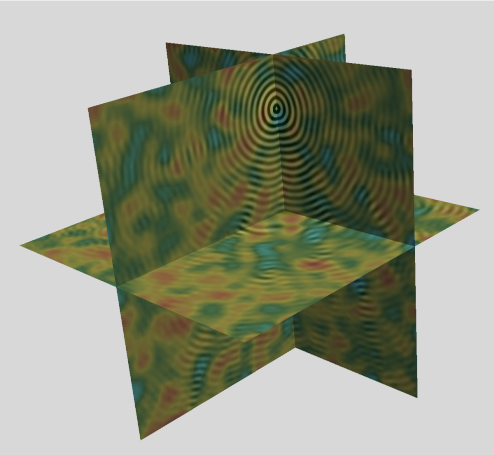
Homogeneous Problem



Homogeneous Problem

N	50^3	100^3	100^3	200^3	200^3	400^3	400^3	400^3
L	5	10	10	20	20	40	40	40
MPI Tasks	5	10	10	80	80	640	640	640
OMP Threads/Task	1	1	2	1	2	1	2	3
Total Cores	5	10	20	80	160	640	1280	1920
Total Nodes	1	1	2	5	10	80	80	128
Single rhs								
# GMRES Iterations	4	4	4	5	5	6	6	6
Initialization [s]	0.2	1.0	0.9	6.9	4.4	18.9	18.9	18.4
Factorization [s]	4.1	41.1	21.9	153.2	78.3	320.5	200.1	148.6
Online [s]	4.0	39.2	22.6	182.0	109.7	696.6	401.4	315.5
Avg. GMRES [s]	0.9	8.4	4.8	32.0	19.2	103.5	59.3	46.6
Pipelined rhs								
R (number of rhs)	5	10	10	20	20	40	40	40
Online [s]	15.8	189.4	106.2	1255.5	668.5	3994.2	2654.4	1878.1
Avg. GMRES [s]	3.4	40.6	22.7	223.8	118.6	599.9	401.0	283.0
Online/rhs [s]	3.2	18.9	10.6	62.8	33.4	99.9	66.4	47.0
Avg. GMRES/rhs [s]	0.7	4.1	2.3	11.2	5.9	15.0	10.0	7.1

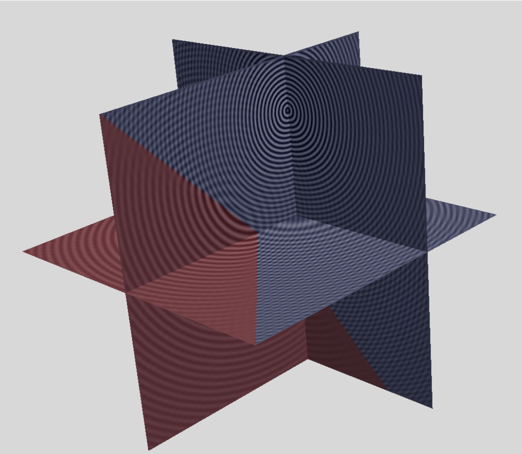
Smooth Heterogeneous Problem



Smooth Heterogeneous Problem

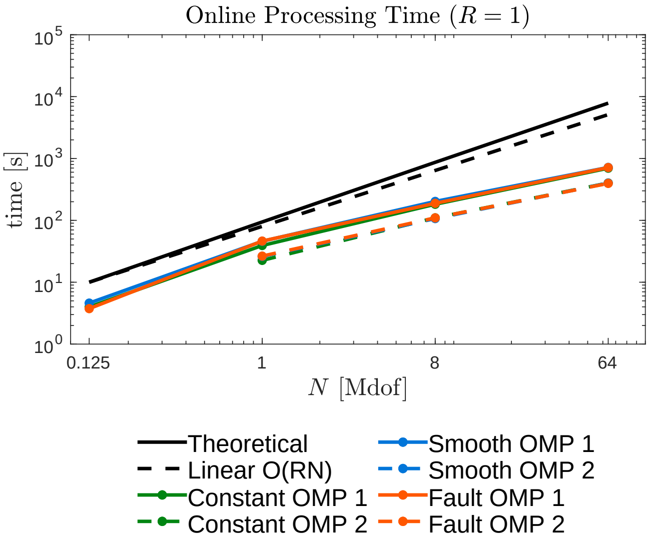
N	50^3	100^3	100^3	200^3	200^3	400^3	400^3	400^3
L	5	10	10	20	20	40	40	40
MPI Tasks	5	10	10	80	80	640	640	640
OMP Threads/Task	1	1	2	1	2	1	2	3
Total Cores	5	10	20	80	160	640	1280	1920
Total Nodes	1	1	2	5	10	80	80	128
Single rhs								
# GMRES Iterations	5	5	5	5	5	6	6	6
Initialization [s]	0.2	1.1	1.0	7.3	4.6	21.3	21.2	20.8
Factorization [s]	3.8	41.1	21.8	156.0	79.4	323.7	204.5	151.5
Online [s]	4.6	45.9	26.1	202.2	106.9	717.0	400.1	314.5
Avg GMRES [s]	0.8	8.1	4.6	35.5	18.7	106.4	59.2	46.5
Pipelined rhs								
R (number of rhs)	5	10	10	20	20	40	40	40
Online [s]	17.1	225.1	118.8	1260.9	650.2	4085.0	2714.8	1872.1
Avg GMRES [s]	3.0	39.8	20.9	223.6	115.6	613.3	409.2	281.9
Online/rhs [s]	3.4	22.5	11.9	63.0	32.5	102.1	67.9	46.8
Avg GMRES/rhs [s]	0.6	4.0	2.1	11.2	5.8	15.3	10.2	7.0

Fault Problem

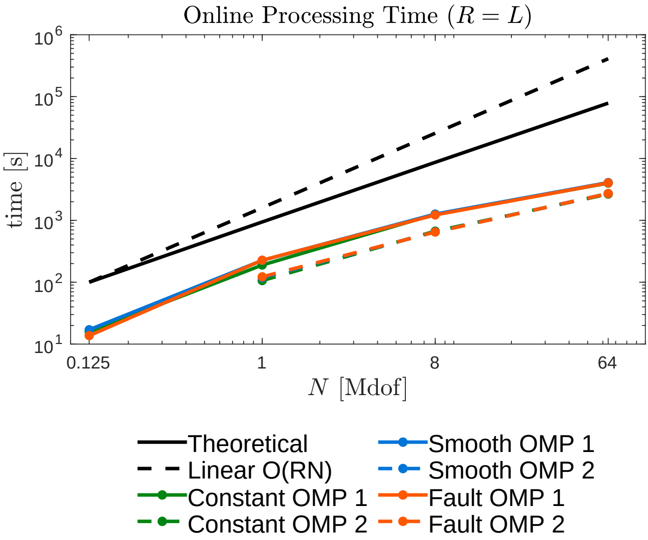


Fault Problem

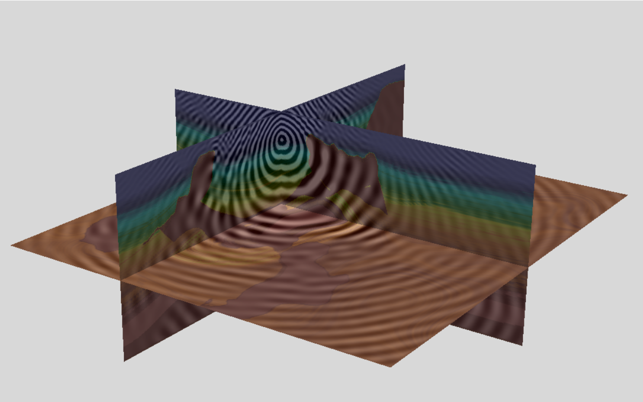
N	50^3	100^3	100^3	200^3	200^3	400^3	400^3	400^3
L	5	10	10	20	20	40	40	40
MPI Tasks	5	10	10	80	80	640	640	640
OMP Threads/Task	1	1	2	1	2	1	2	3
Total Cores	5	10	20	80	160	640	1280	1920
Total Nodes	1	1	2	5	10	80	80	128
Single rhs								
# GMRES Iterations	4	5	5	5	5	6	6	6
Initialization [s]	0.4	1.1	1.0	7.3	4.7	20.4	20.3	21.0
Factorization [s]	3.8	40.4	22.1	152.2	79.9	317.6	199.5	152.5
Online [s]	3.7	46.2	26.2	188.5	109.8	713.2	395.8	315.6
Avg GMRES [s]	0.8	8.1	4.6	33.0	19.2	106.2	58.7	46.5
Pipelined rhs								
R (number of rhs)	5	10	10	20	20	40	40	40
Online [s]	13.7	226.7	122.4	1222.7	647.1	4031.6	2710.6	1838.9
Avg GMRES [s]	2.9	40.1	21.6	216.5	114.7	605.0	409.9	276.3
Online/rhs [s]	2.7	22.7	12.2	61.1	32.4	100.8	67.7	46.0
Avg GMRES/rhs [s]	0.6	4.0	2.2	10.8	5.7	15.1	10.2	6.9


Online Processing Time ($R = 1$)
time [s]
10^5
10^4
10^3
10^2
10^1
10^0
0.125
1
8
64
N [Mdof]
Theoretical
Linear O(RN)
Constant OMP 1
Constant OMP 2
Smooth OMP 1
Smooth OMP 2
Fault OMP 1
Fault OMP 2

Performance of 3D Polarized Traces



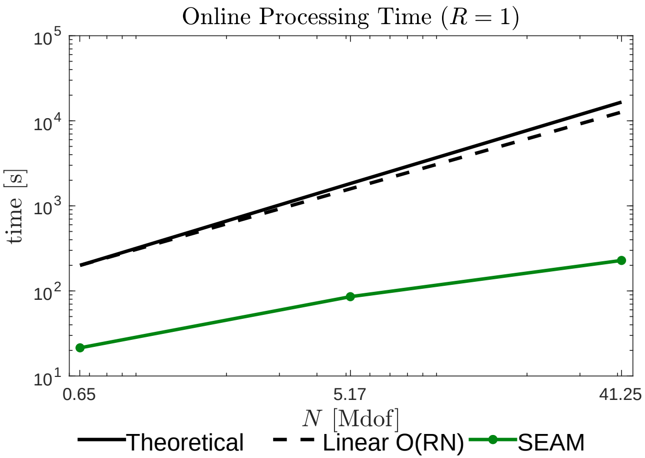
SEAM Problem

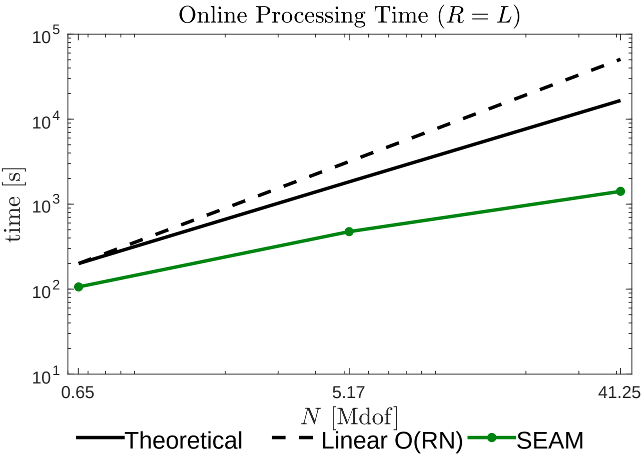


SEAM Problem

N	$6.51 \cdot 10^5$	$5.16 \cdot 10^6$	$4.12 \cdot 10^7$	$4.12 \cdot 10^7$
L	12	24	48	48
MPI Tasks	12	48	384	384
OpenMP Threads per Task	1	2	2	3
Total Cores	12	96	768	1152
Total Nodes	1	6	77	77
Single rhs				
# GMRES Iterations	4	5	6	6
Initialization [s]	0.6	2.3	10.4	10.7
Factorization [s]	15.2	46.5	111.4	97.9
Online [s]	21.4	85.6	269.8	228.4
Average GMRES [s]	4.6	14.9	40.0	33.7
Pipelined rhs				
R (number of rhs)	12	24	48	48
Online [s]	106.3	474.8	1527.1	1415.4
Average GMRES [s]	22.8	83.9	229.4	212.9
Online per rhs [s]	8.8	19.8	31.8	29.5
Average GMRES per rhs [s]	1.9	3.5	4.8	4.4

Performance of 3D Polarized Traces




Online Processing Time $(R = L)$
time [s]
10^5
10^4
10^3
10^2
10^1
0.65
5.17
41.25
N [Mdof]
Theoretical
Linear O(RN)
SEAM

Can We Do Better?

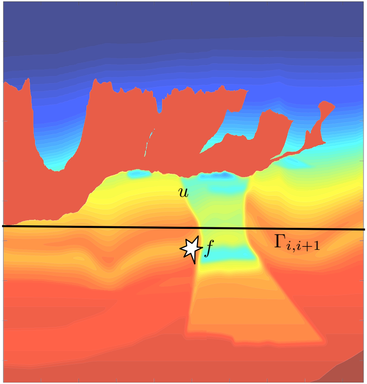
Pipelined Parallel Run-time complexity: $\mathcal{O}(\max(1, R/L)N \log N)$

Question: Can we *better* parallelize this preconditioner?

Problem: Serial nature of the sweeps

Problem: 2D memory growth due to planar slabs

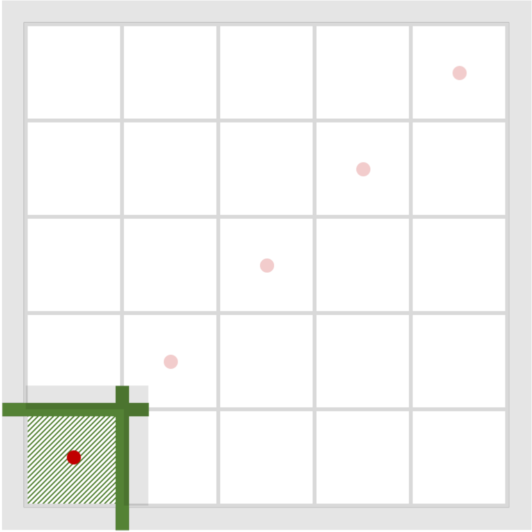
Problem: Interface “communication” volume



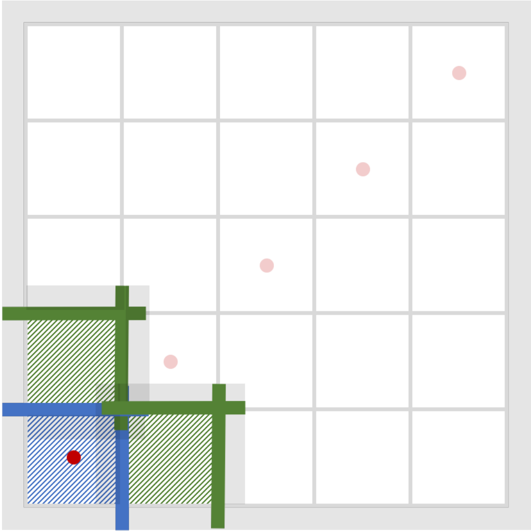
Polarization condition:

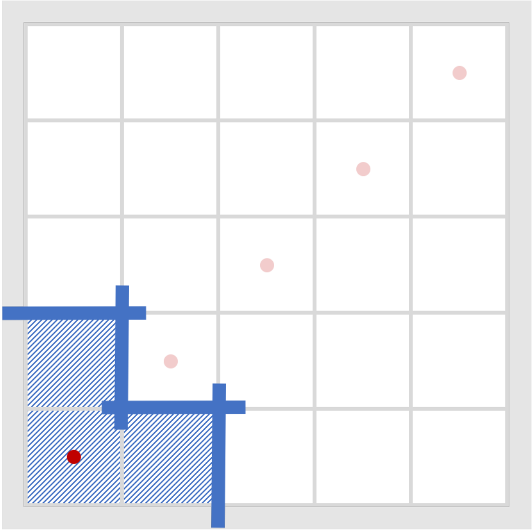
$$0 = -\int_\Gamma G(x,y)\partial_{n_y} u^\uparrow(y) ds_y + \int_\Gamma \partial_{n_y} G(x,y) u^\uparrow(y) ds_y$$

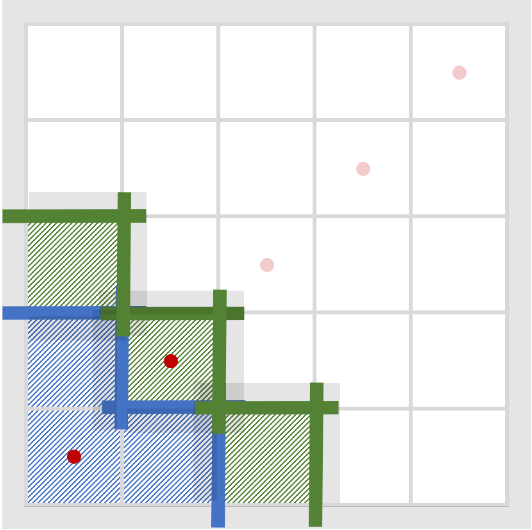
Solution: L-sweeps

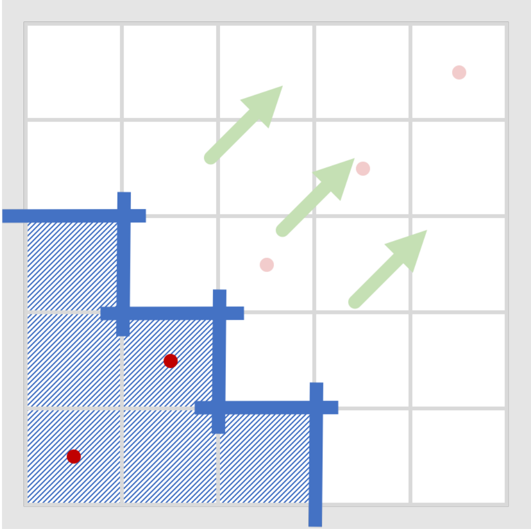


Solution: L-sweeps





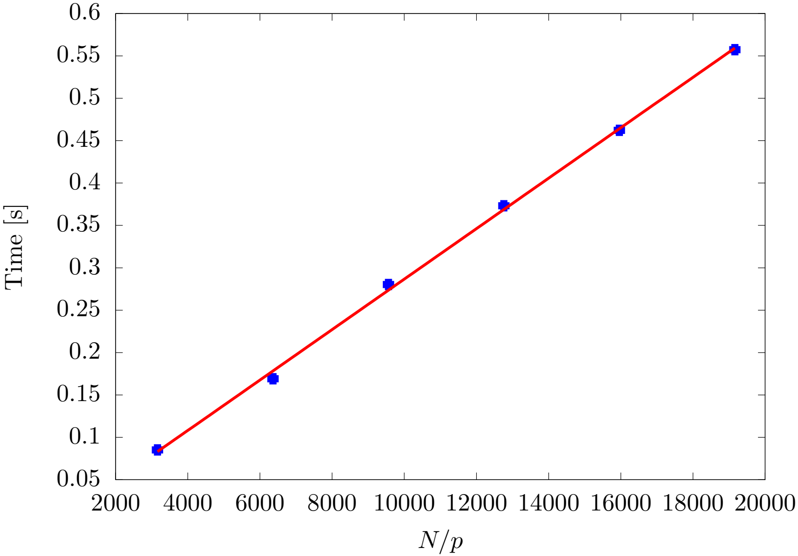




M O V I E! :)

Each propagation onto the next diagonal can is embarrassingly parallel on a cell-wise level!

$\Rightarrow O(N/p)$ complexity
(as long as $p = O(N^{1/d})$)


0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
Time [s]
2000
4000
6000
8000
10000
12000
14000
16000
18000
20000
N/p

4 points per wavelength

Wavelengths in domain	Number of cells	Wavelengths in PML: 1	1.5	2	2.5	3
16	2	5	3	3	3	3
32	4	7	5	5	5	5
64	8	7	6	6	6	6
128	16	9	6	7	7	7
256	32	12	9	7	7	7
512	64	17	11	8	9	8
1024	128	29	14	11	9	9

4 points per wavelength

Wavelengths in domain	Number of cells	Wavelengths in PML: 1	1.5	2	2.5	3
16	2	5	3	3	3	3
32	4	7	5	5	5	5
64	8	7	6	6	6	6
128	16	9	6	7	7	7
256	32	12	9	7	7	7
512	64	17	11	8	9	8
1024	128	29	14	11	9	9

Numerical Example: Iteration Count

4 points per wavelength

Wavelengths in domain	Number of cells	Wavelengths in PML: 1	1.5	2	2.5	3
16	2	5	3	3	3	3
32	4	7	5	5	5	5
64	8	7	6	6	6	6
128	16	9	6	7	7	7
256	32	12	9	7	7	7
512	64	17	11	8	9	8
1024	128	29	14	11	9	9

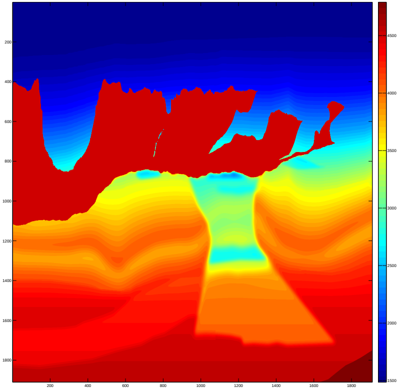
6 points per wavelength

Wavelengths in domain	Number of cells	Wavelengths in PML: 1	1.5	2	2.5	3
16	2	4	3	3	3	3
32	4	5	3	3	3	3
64	8	7	3	3	3	3
128	16	9	5	4	3	3
256	32	11	6	5	5	4
512	64	17	9	7	5	5
1024	128	32	11	8	7	6

8 points per wavelength

Wavelengths in domain	Number of cells	Wavelengths in PML: 1	1.5	2	2.5	3
16	2	5	3	3	3	3
32	4	5	3	3	3	3
64	8	7	3	3	3	3
128	16	8	5	3	3	3
256	32	11	6	5	3	3
512	64	19	8	6	5	4
1024	128	-	11	9	7	5

Numerical Example: BP Model Setup



- Second order finite difference discretization
- unit square

4 points per wavelength

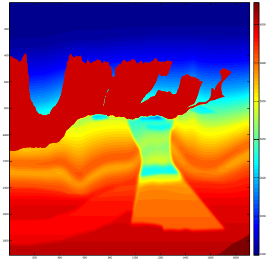
Wavelengths in domain	Number of cells	Wavelengths in PML: 1	1.5	2	2.5	3
16	2	9	7	6	6	6
32	4	12	7	7	7	7
64	8	14	9	10	10	10
128	16	16	12	12	12	12
256	32	25	25	23	22	23
512	64	30	26	26	26	26
1024	128	-	29	29	28	28

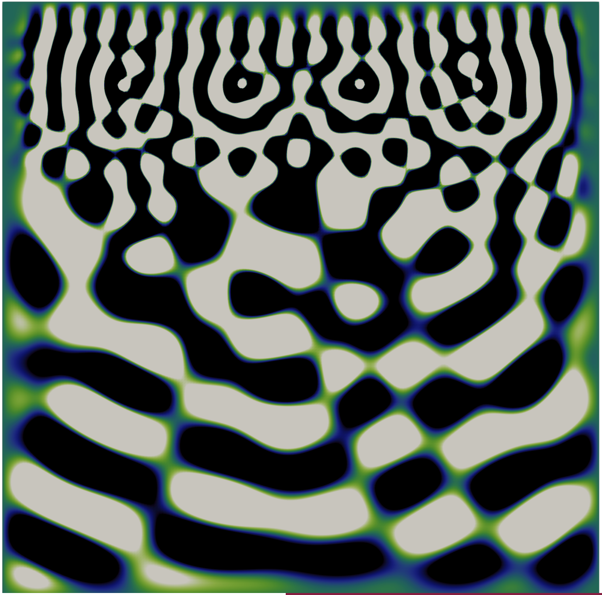
6 points per wavelength

Wavelengths in domain	Number of cells	Wavelengths in PML				
		1	1.5	2	2.5	3
16	2	9	6	6	6	6
32	4	11	7	7	7	7
64	8	13	9	8	8	8
128	16	16	11	11	11	11
256	32	24	18	18	19	18
512	64	-	25	25	24	24
1024	128	-	29	28	28	27

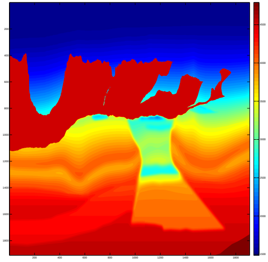
8 points per wavelength

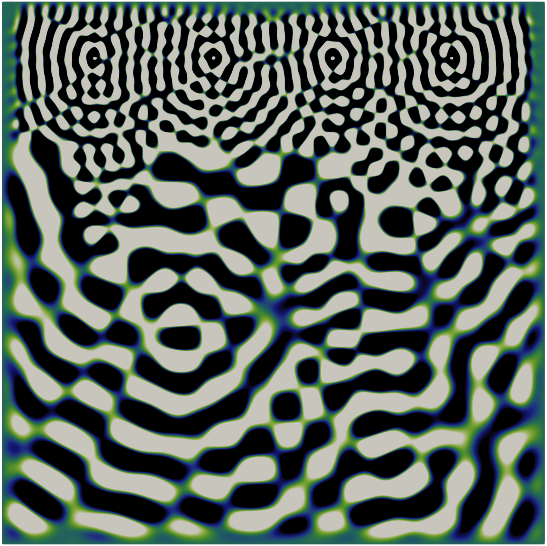
Wavelengths in domain	Number of cells	Wavelengths in PML 1	1.5	2	2.5	3
16	2	9	6	6	6	6
32	4	11	6	6	6	6
64	8	14	9	9	9	9
128	16	22	16	17	14	13
256	32	-	16	16	15	15
512	64	-	22	21	21	21
1024	128	-	-	26	26	26





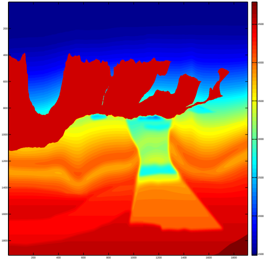
- max. 16 wavelengths in domain
- PML width: 1.25 wavelengths
- 2×2 domain decomposition

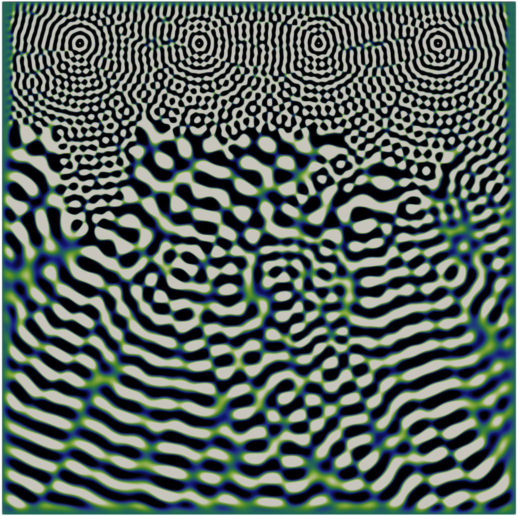




- max. 32 wavelengths in domain
- PML width: 1.5 wavelengths
- 4×4 domain decomposition

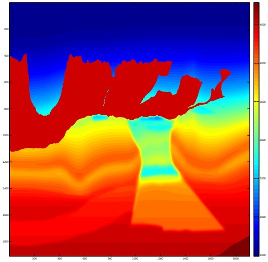
Numerical Example: High Frequency Solutions

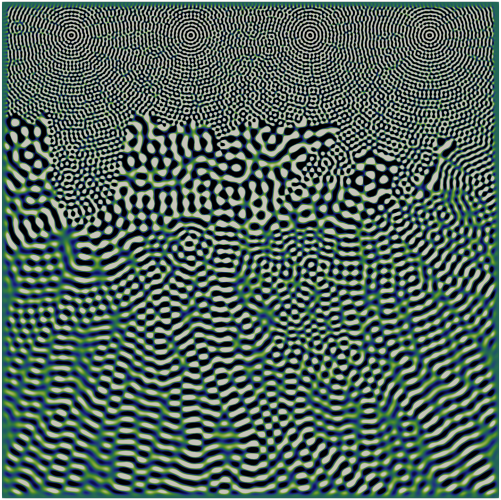




- max. 64 wavelengths in domain
- PML width: 1.75 wavelengths
- 8×8 domain decomposition

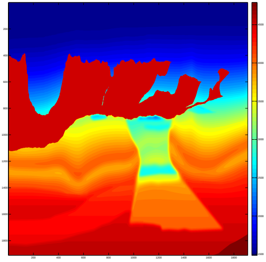
Numerical Example: High Frequency Solutions

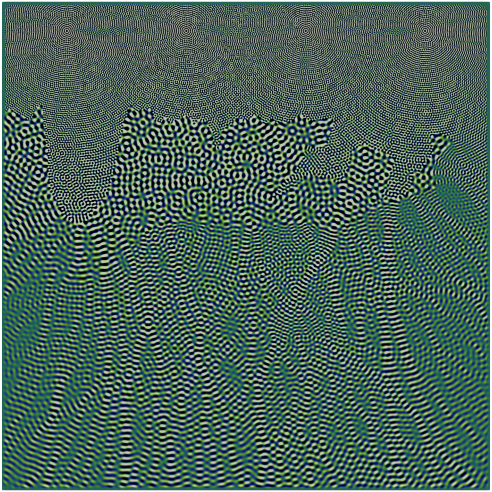




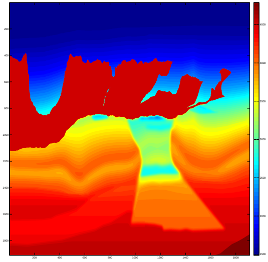
- max. 128 wavelengths in domain
- PML width: 2.00 wavelengths
- 16×16 domain decomposition

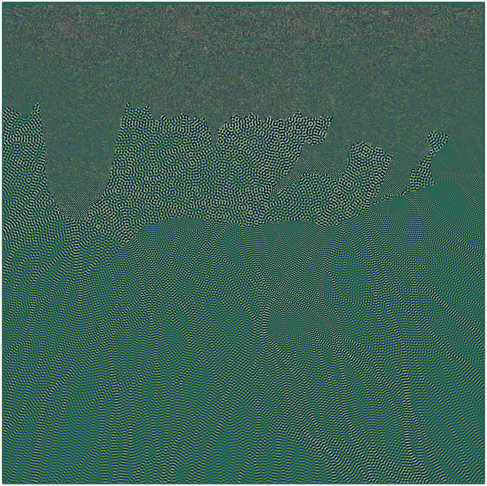
Numerical Example: High Frequency Solutions





- max. 256 wavelengths in domain
- PML width: 2.25 wavelengths
- 32×32 domain decomposition





- max. 512 wavelengths in domain
- PML width: 2.5 wavelengths
- 64×64 domain decomposition

Successful construction of a scalably parallelizable preconditioner for the high-frequency Helmholtz equation.

- $O(N/p)$ complexity as long as $p = O(N^{1/d})$
- Independent of the discretization
- Applicable to heterogeneous media

Successful construction of a scalably parallelizable preconditioner for the high-frequency Helmholtz equation.

- $O(N/p)$ complexity as long as $p = O(N^{1/d})$
- Independent of the discretization
- Applicable to heterogeneous media

Next steps:

- $O(N/p)$-scaling in 3D where $p = O(N^{2/3})$
- several right-hand sides ($O(1)$ scaling per right hand side?)

Where do we go from here?

Next steps:

- How large can we reasonably scale?
- One strategy: Treat the layer as the largest building block
- Problem: Dense solver fundamentally limits layer size and scaling
- Potential solution: Nesting
- Finite difference $\rightarrow$ discontinuous Galerkin

Where do we go from here?

Next steps:

- How large can we reasonably scale?
- One strategy: Treat the layer as the largest building block
- Problem: Dense solver fundamentally limits layer size and scaling
- Potential solution: Nesting
- Finite difference $\rightarrow$ discontinuous Galerkin

Open questions:

- How to incorporate free-surface BCs?
- How do we make the solver robust to system failure? (petascale computing)
- How well does it perform for more complex physics? (Elastics, Acoustic-elastic coupling, etc.)
- How best to leverage accelerators? (GPU, many-core, etc.)

Where do we go from here?

Next steps:

- How large can we reasonably scale?
- One strategy: Treat the layer as the largest building block
- Problem: Dense solver fundamentally limits layer size and scaling
- Potential solution: Nesting
- Finite difference $\rightarrow$ discontinuous Galerkin

Open questions:

- How to incorporate free-surface BCs?
- How do we make the solver robust to system failure? (petascale computing)
- How well does it perform for more complex physics? (Elastics, Acoustic-elastic coupling, etc.)
- How best to leverage accelerators? (GPU, many-core, etc.)

Application to inverse problem:

- Adjoint formulation?
- How exact do we need to solve?