Evolution of a Scalable 3D Helmholtz Solver with Geophysical Applications

Russell J. Hewett
Mathematics & CMDA, Virginia Tech

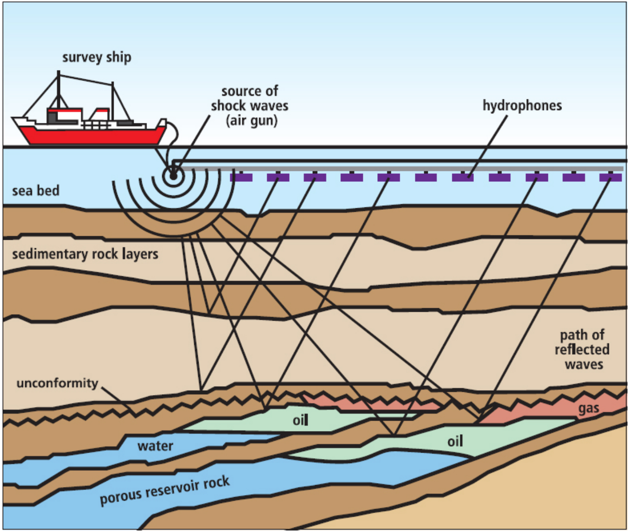
UMD Numerical Analysis Seminar

December 03, 2019

Collaborators

- Leonardo Zepeda-Nuñez, University of Wisconsin
- Matthias Taus, TU Wien
- Laurent Demanet, MIT
- Adrien Scheuer, Universitè Catholique de Louvain

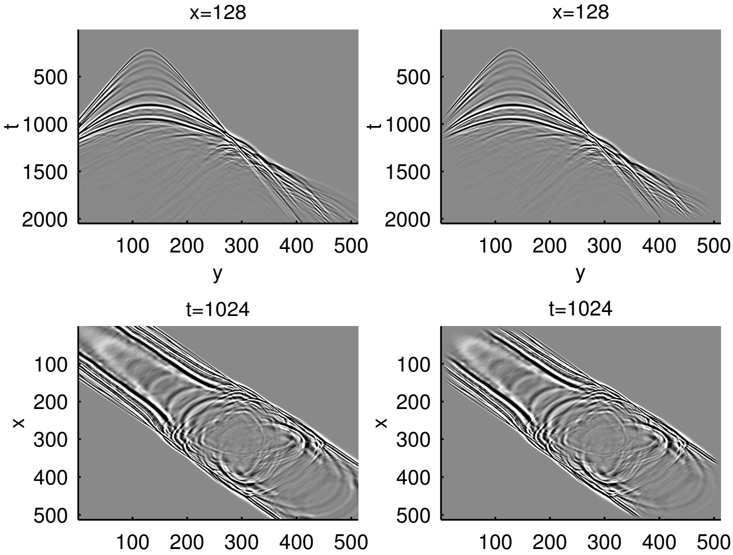
Full Waveform Inversion



Full Waveform Inversion

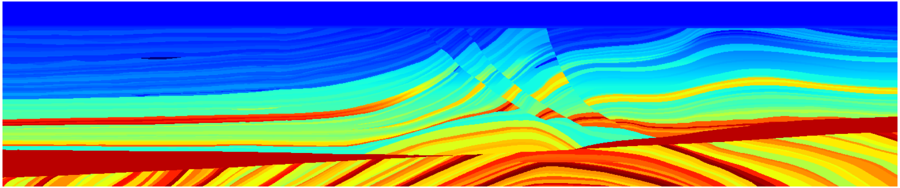
seismic source + receivers → data	d
Earth's physical parameters	m
model physics (wave propagation)	$\mathcal{F}(m)$
full waveform inversion	$\min J(m) = \frac{1}{2}\|\|d - \mathcal{F}(m)\|\|_2^2$
gradient optimization	$m^{(k+1)} = m^{(k)} + f(\nabla J[m^{(k)}])$

Full Waveform Inversion: Data

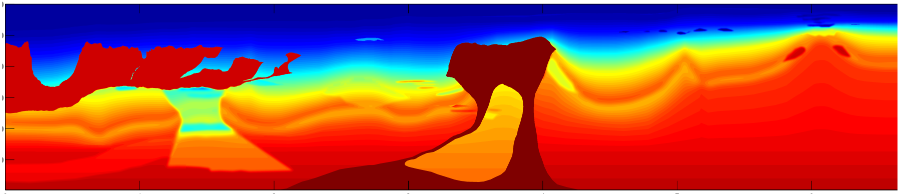


Full Waveform Inversion: Earth Models

Marmousi 2 Velocity

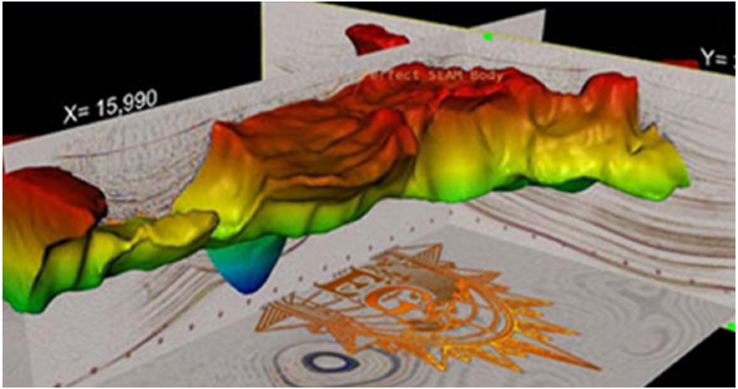


BP 2004 Velocity



Full Waveform Inversion: Earth Models

SEAM Phase I Velocity (Fehler; SEG)



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

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

- 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

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

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:

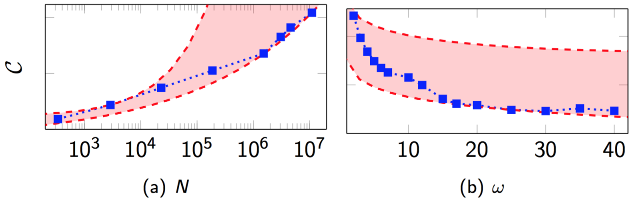
- Multiple simultaneous right-hand sides
- With a factorization based method, only need to solve Helmholtz operator once per domain
- Compare to explicit time-stepping: "matvec" required for each time step for each source

PDE constrained optimization in frequency domain

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

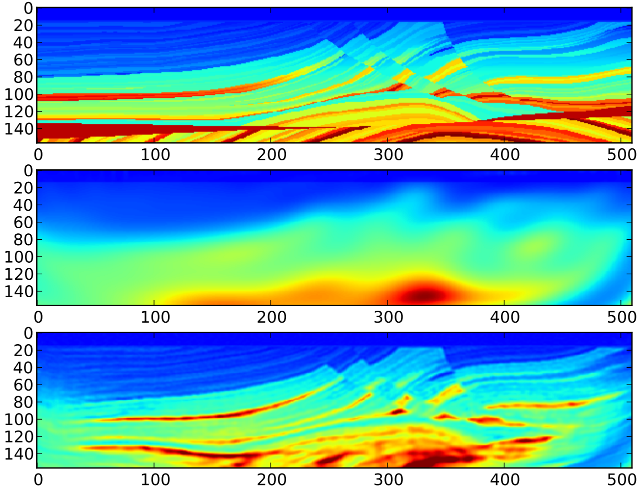
Advantages:

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



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

FWI in the Frequency Domain



FWI in the Frequency Domain

Challenges for Frequency Domain Inversion . . . it's all in the foward problem:

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

Take-home from this talk:

- With the right mix of tools, addressing all three is tractible
- Still need fast, parallelizable dense linear algebra
- **Sub-linear complexity is achieved in parallel environments**

Motivation for Sweeping Solvers

Helmholtz at high frequency is **hard**

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

- 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**

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

- 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)

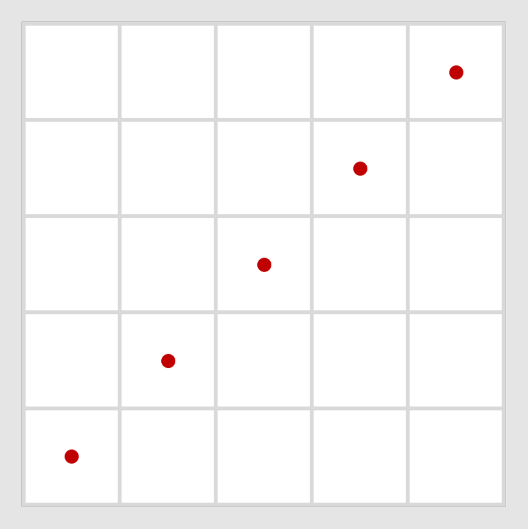
Other domain-decomposition methods (DDMs):

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

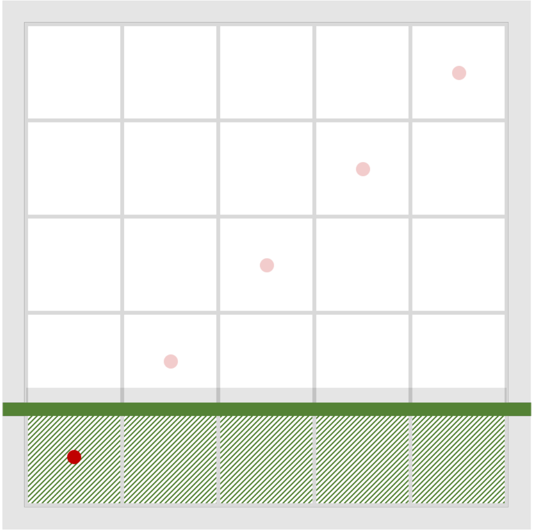
Our approach: DDMs + sweeping w/ polarized traces

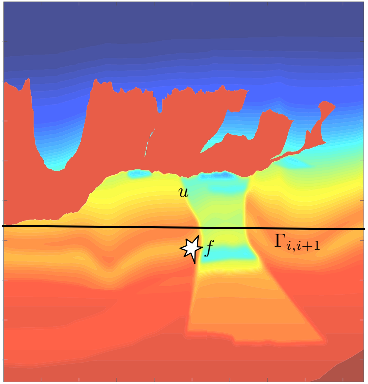
- Use direct methods distributed over tractable subproblems
- Glue with boundary integral formulations
- Embedded within iterative scheme

Method Of Polarized Traces & the Half-space Problem



Method Of Polarized Traces & the Half-space Problem

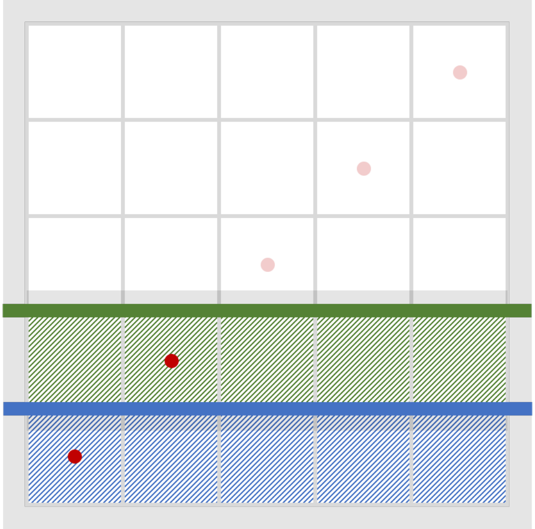




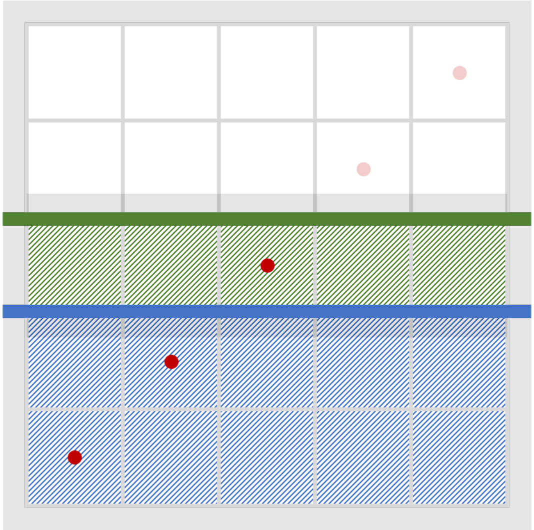
Polarization condition:

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

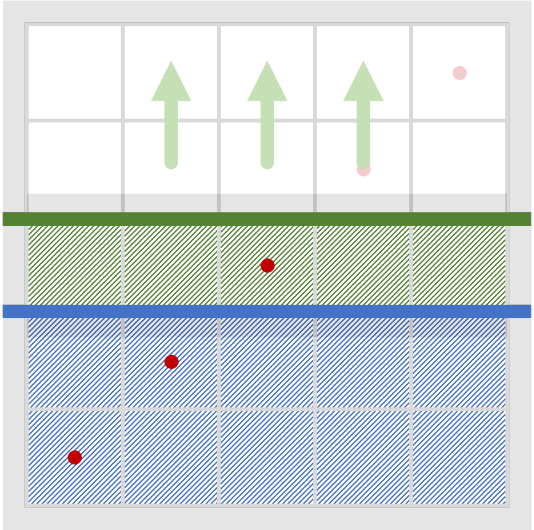
Method Of Polarized Traces & the Half-space Problem



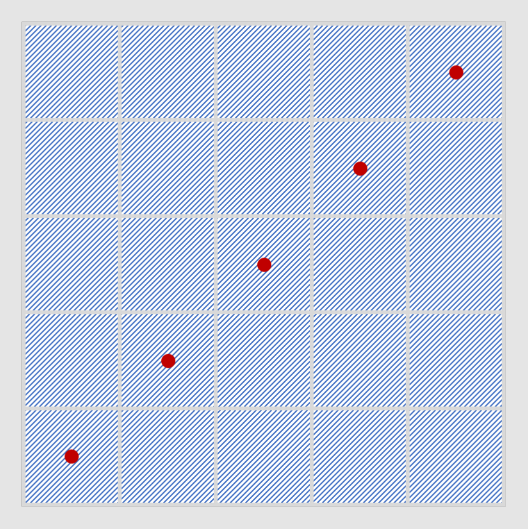
Method Of Polarized Traces & the Half-space Problem



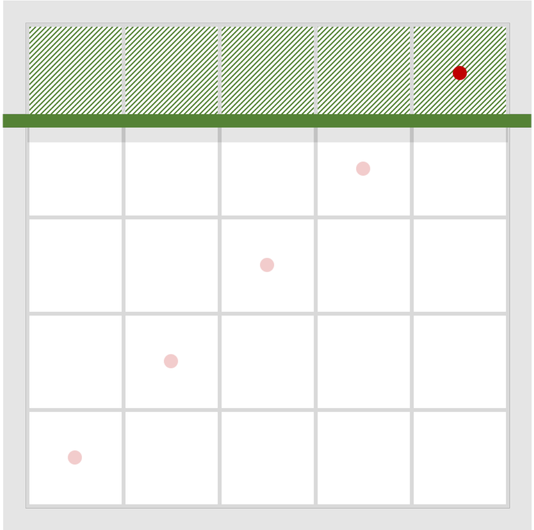
Method Of Polarized Traces & the Half-space Problem



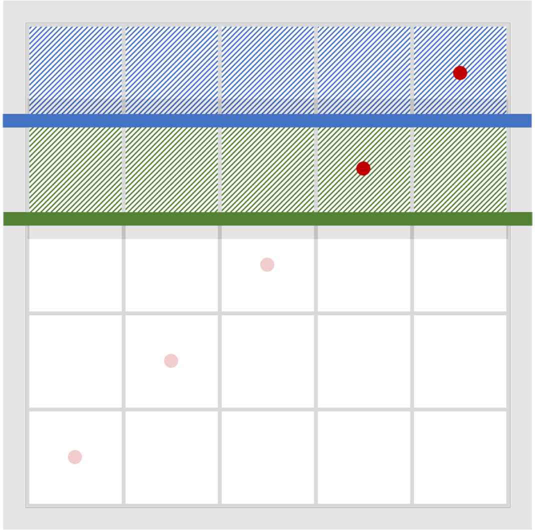
Method Of Polarized Traces & the Half-space Problem



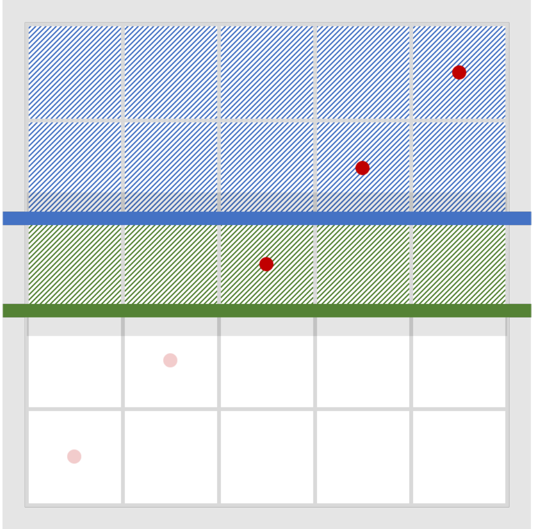
Method Of Polarized Traces & the Half-space Problem



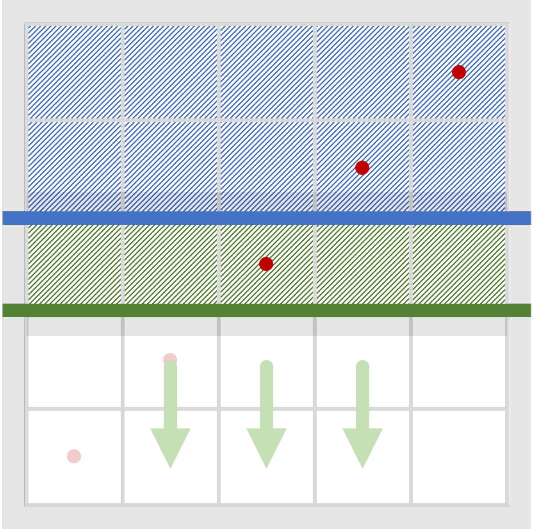
Method Of Polarized Traces & the Half-space Problem



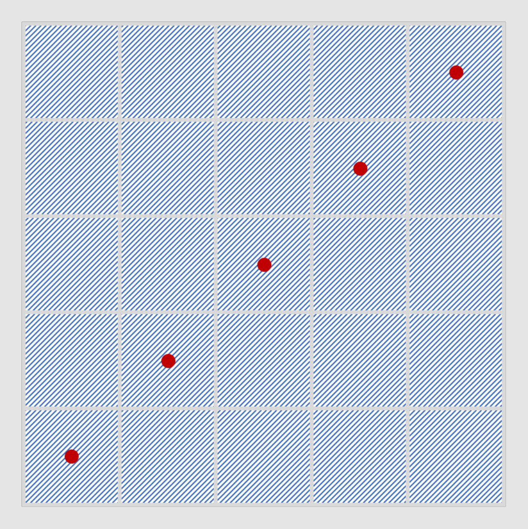
Method Of Polarized Traces & the Half-space Problem



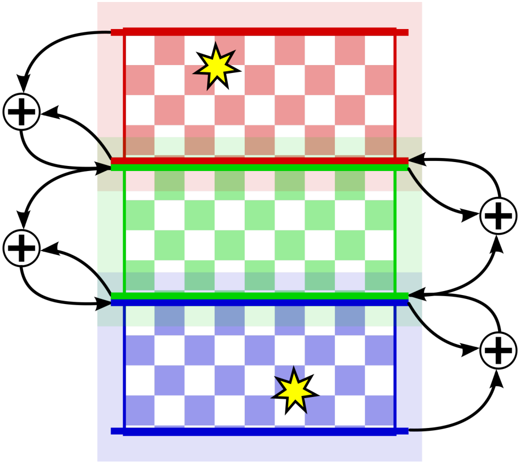
Method Of Polarized Traces & the Half-space Problem



Method Of Polarized Traces & the Half-space Problem



Sweeping Algorithm

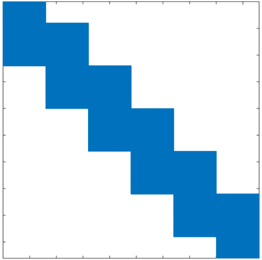


Zepeda-Nuñez, RJH, & Demanet, SEG, 2014

Zepeda-Nuñez & Demanet, JCP, 2016

Polarized Traces & A Sequential Bottleneck

- 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... but is non-trivial to invert



Polarized Traces & A Sequential Bottleneck

- 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.

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

Polarized Traces & A Sequential Bottleneck

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

Polarized Traces & A Sequential Bottleneck

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

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

Polarized Traces & A Sequential Bottleneck

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

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

Polarized Traces & A Sequential Bottleneck

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

Polarized Traces & A Sequential Bottleneck

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

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

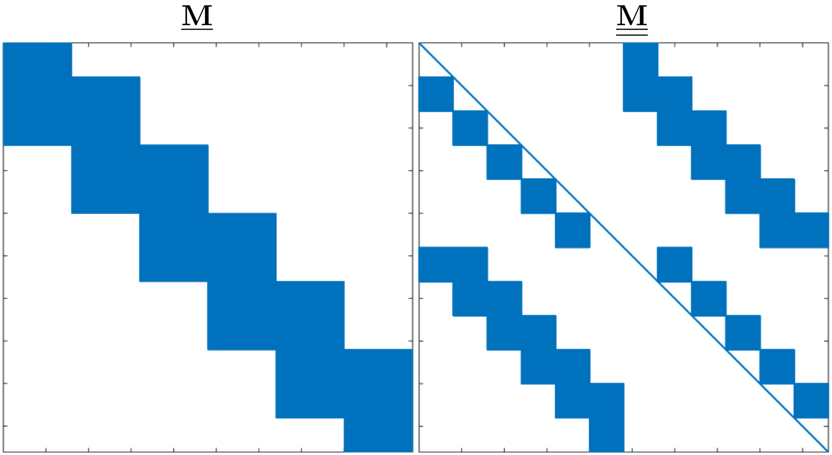
4. Additional minor transformations and permutations

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

5. Final system of equations

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

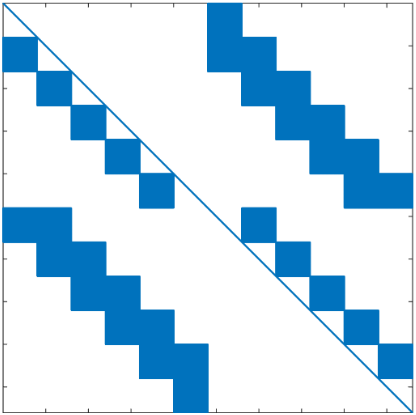
Polarized Traces & A Sequential Bottleneck



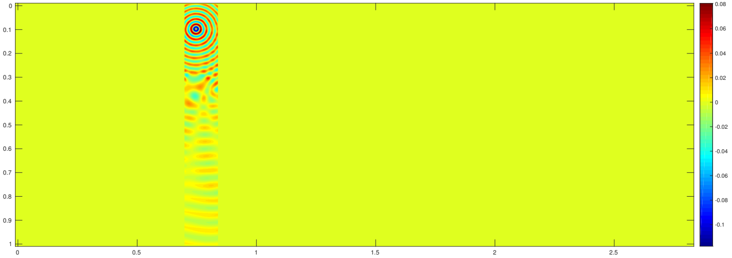
- ... $\underline{\underline{\mathbf{M}}}$ is far easier to invert

Polarized Traces & A Sequential Bottleneck

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

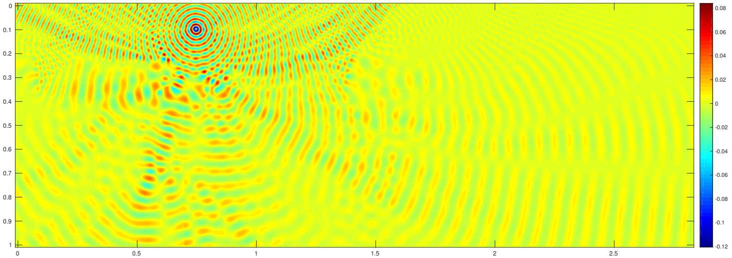


BP 2004 2D solution



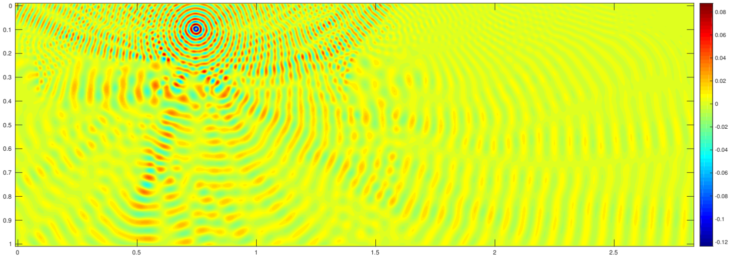
Iteration 0

BP 2004 2D solution



Iteration 1 (2 domain sweeps)

BP 2004 2D solution

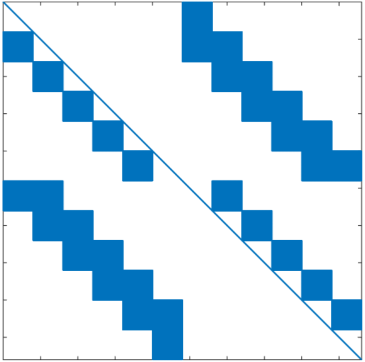


Iteration 2 (4 domain sweeps)

The State of Method So Far

- Explicit representation of $\underline{\underline{\mathbf{M}}}$
- Matrix blocks are concrete representations of the Green's function
- Green's function blocks are compressed using PLR format
- Green's functions can be computed off-line, in parallel

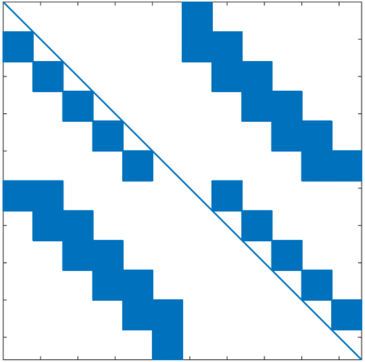
- Gauss-Seidel sweep is inherently sequential
- Sequential nature corresponds directly to the physics



Challenges in Moving to 3D

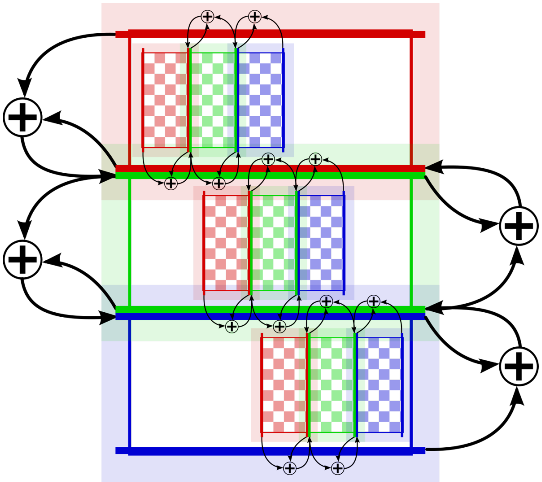
- Green's function blocks: n (in 2D) vs n^2 (in 3D)
- Explicit computation is impractical at scale
- PLR does not work as well for 2D Green's functions
- Sequential portion is still sequential

- **Solution:** Don't compute Green's functions. Solve local systems.
- **Problem:** Local systems are still computationally difficult.



Can We Apply Polarized Traces Recursively?

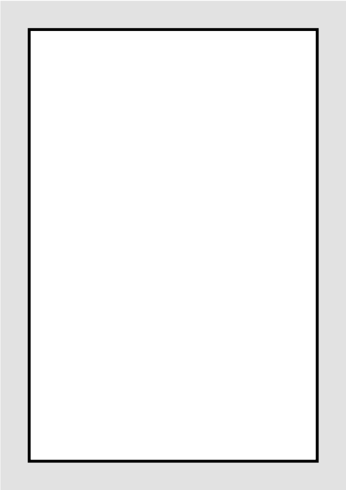
- **Solution:** Use polarized traces to solve local systems.
- **Problem:** Same major sequential bottleneck.



Zepeda-Nuñez & Demanet, SISC, 2018

Sources of Parallelism

Domain

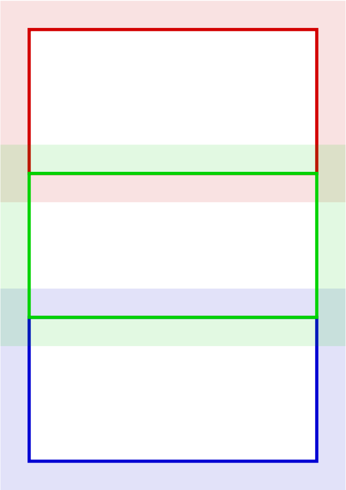


Zepeda-Nuñez, RJH, Demanet, & Scheuer, SEG, 2016

Zepeda-Nuñez, Scheuer, RJH, & Demanet, Geophysics, 2019

Sources of Parallelism: Layers

MPI: Parallelize over Layers



Zepeda-Nuñez, RJH, Demanet, & Scheuer, SEG, 2016

Zepeda-Nuñez, Scheuer, RJH, & Demanet, Geophysics, 2019

Sources of Parallelism: Layers

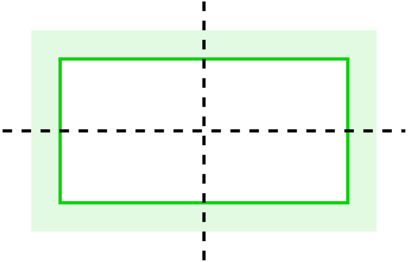
MPI: Parallelize within
Layers

Zepeda-Nuñez, RJH, Demanet, & Scheuer, SEG, 2016

Zepeda-Nuñez, Scheuer, RJH, & Demanet, Geophysics, 2019

Sources of Parallelism: Local Solver

MPI: Multifrontal/Nested dissection

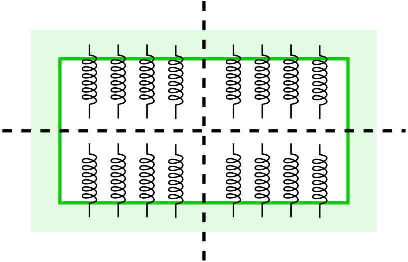


Zepeda-Nuñez, RJH, Demanet, & Scheuer, SEG, 2016

Zepeda-Nuñez, Scheuer, RJH, & Demanet, Geophysics, 2019

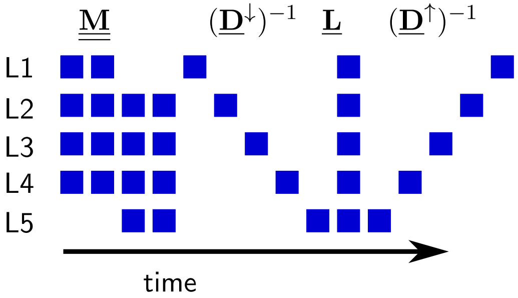
Sources of Parallelism: Local Solver

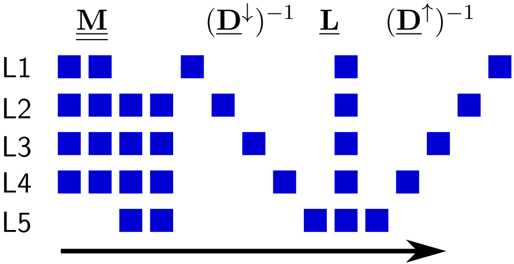
OpenMP: Parallelize within MPI tasks

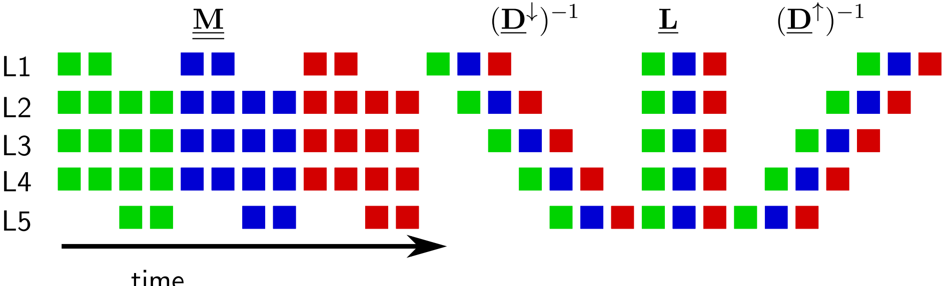


Zepeda-Nuñez, RJH, Demanet, & Scheuer, SEG, 2016

Zepeda-Nuñez, Scheuer, RJH, & Demanet, Geophysics, 2019

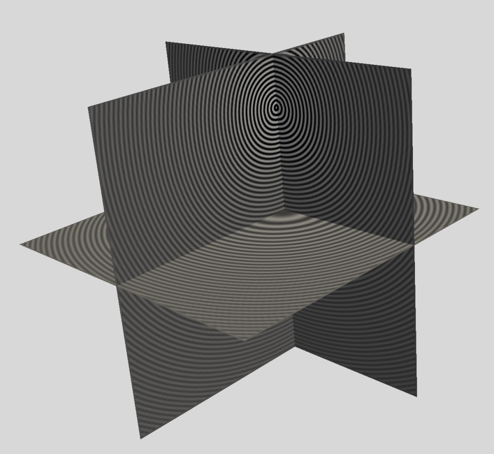

$\underline{\underline{\mathbf{M}}}$
$(\underline{\mathbf{D}}^{\downarrow})^{-1}$
$\underline{\mathbf{L}}$
$(\underline{\mathbf{D}}^{\uparrow})^{-1}$
L1
L2
L3
L4
L5
time


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


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

Performance of 3D Polarized Traces

Homogeneous Problem



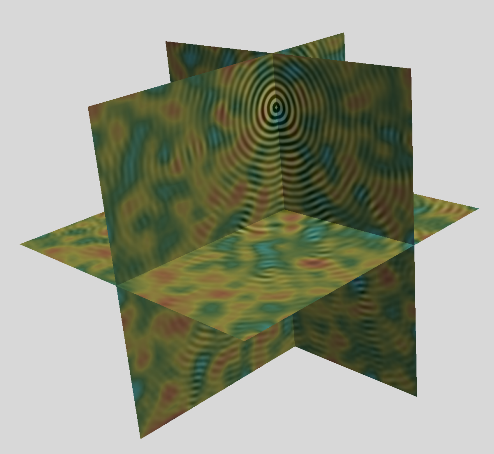
Performance of 3D Polarized Traces

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

Performance of 3D Polarized Traces

Smooth Heterogeneous Problem



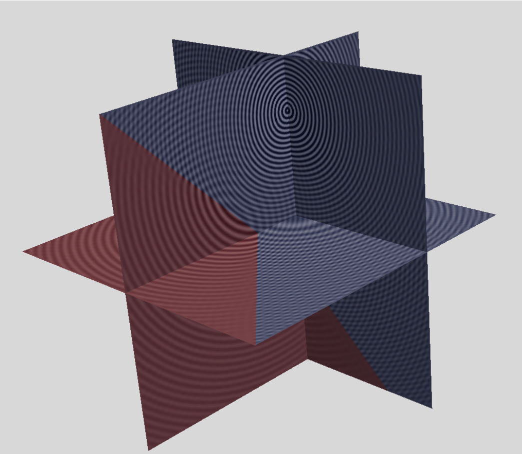
Performance of 3D Polarized Traces

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

Performance of 3D Polarized Traces

Fault Problem

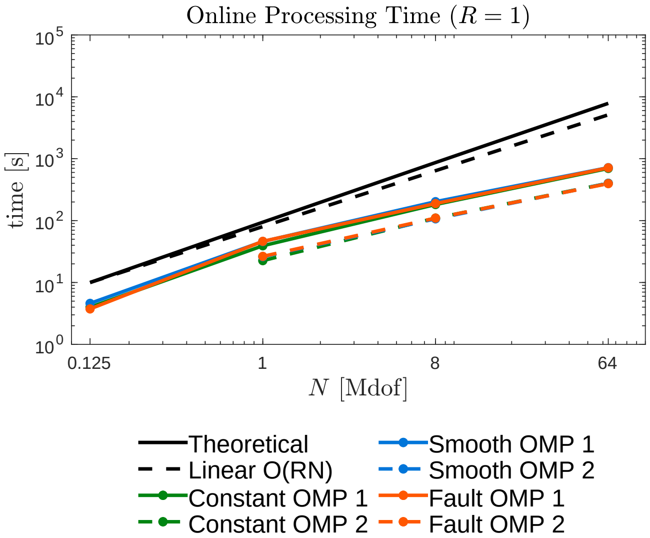


Performance of 3D Polarized Traces

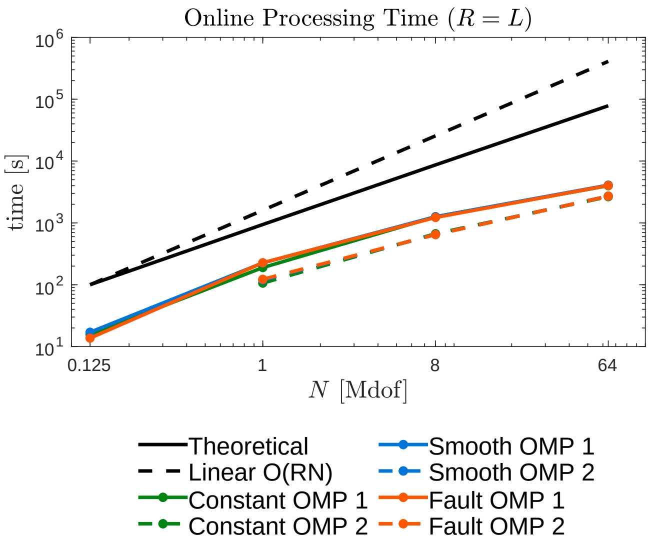
Fault Problem

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

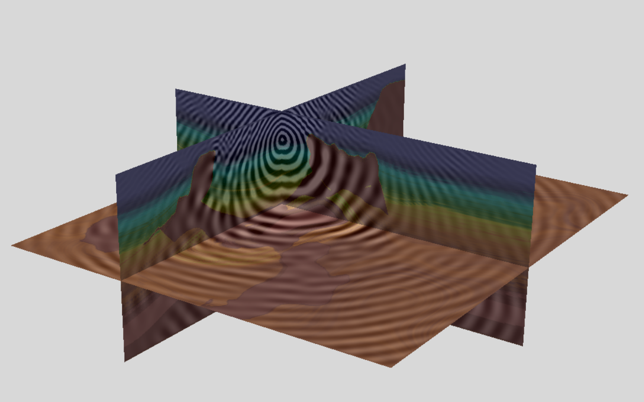
Performance of 3D Polarized Traces



Performance of 3D Polarized Traces



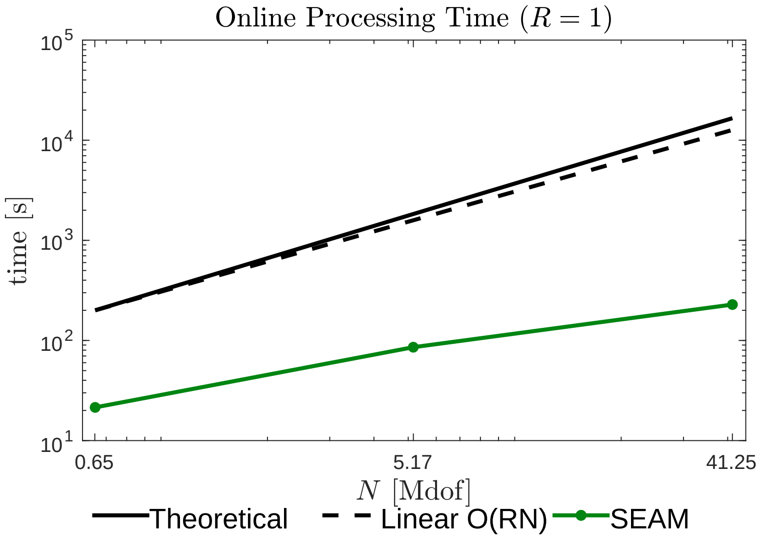
SEAM Problem



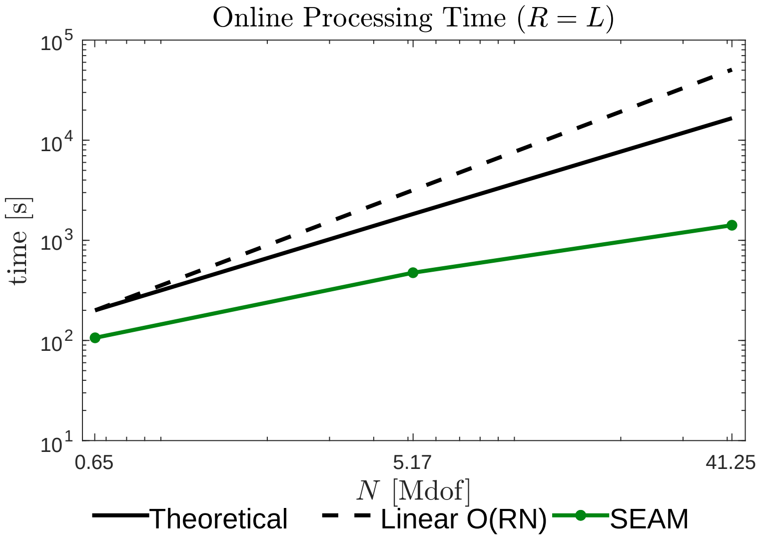
SEAM Problem

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



Performance of 3D Polarized Traces



Can We Do Better?

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

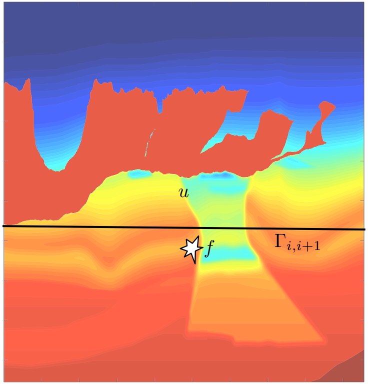
Question: Can we *better* parallelize this preconditioner?

Problem: Serial nature of the sweeps

Problem: 2D memory growth due to planar slabs

Problem: Interface "communication" volume

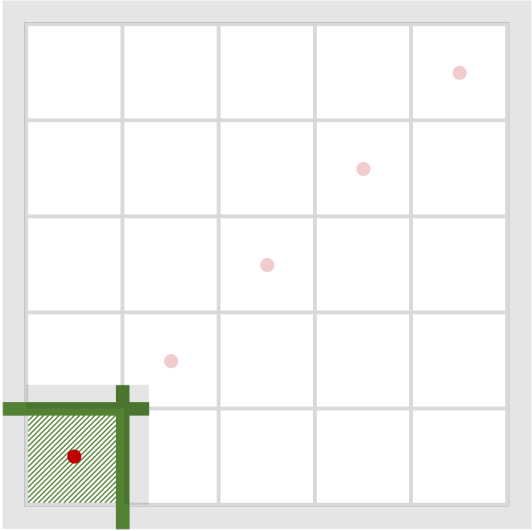
Half-space Problem



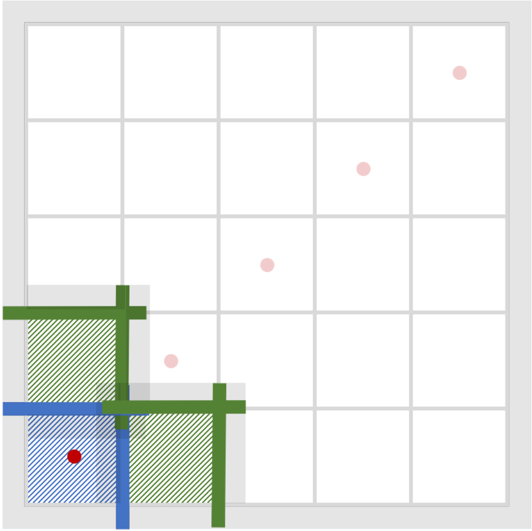
Polarization condition:

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

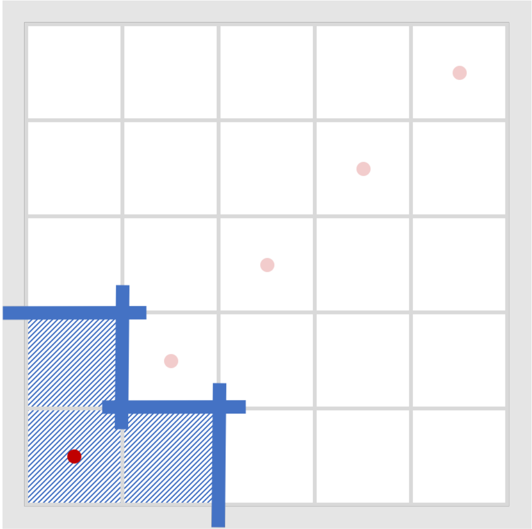
Solution: L-sweeps



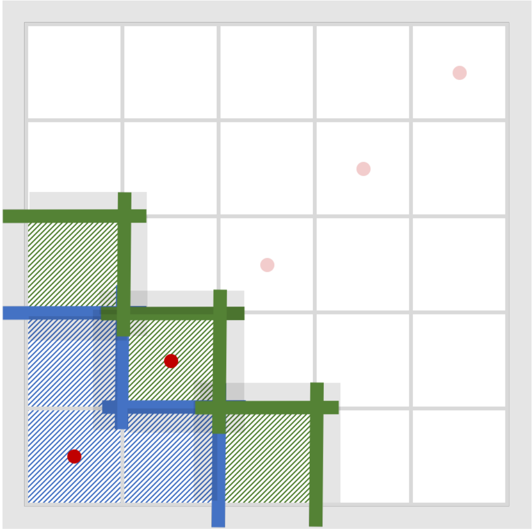
Solution: L-sweeps



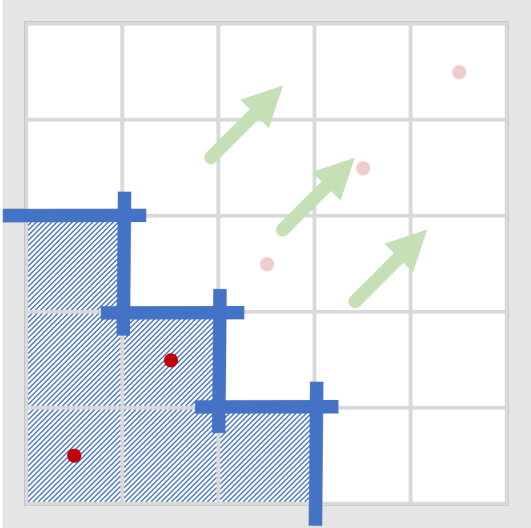
Solution: L-sweeps



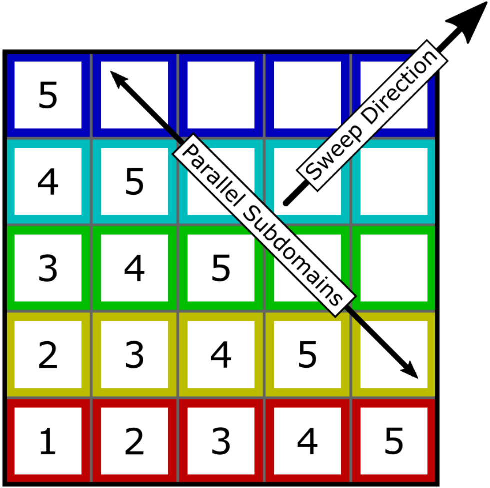
Solution: L-sweeps



Solution: L-sweeps



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



Parallelism Pattern

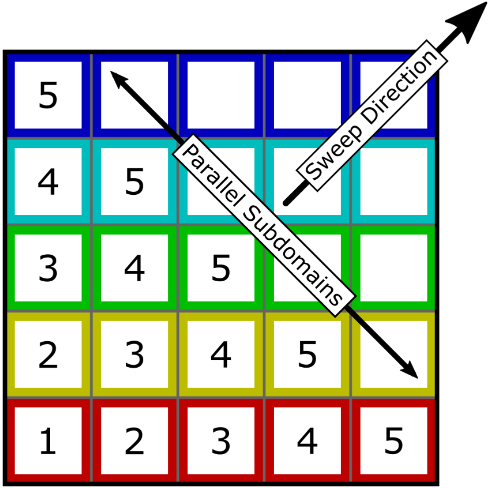
M O V I E! :)

Parallelism Pattern

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

$\Rightarrow O(N/p)$ complexity

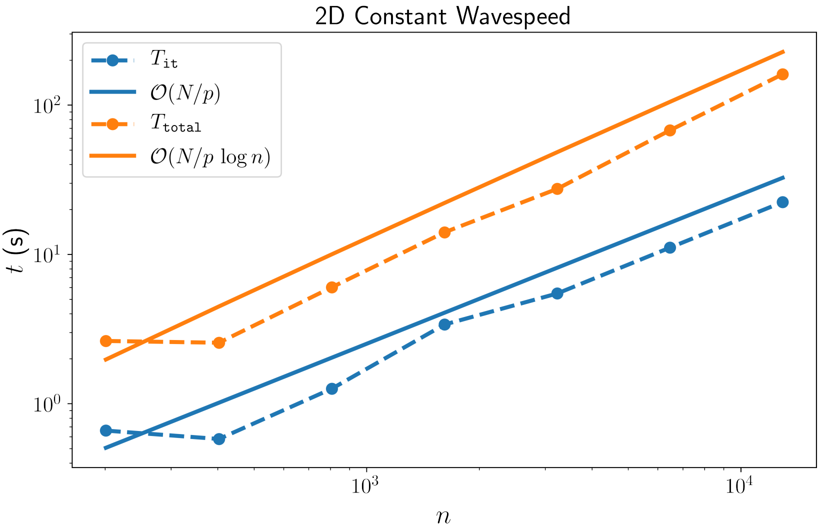
(as long as $p = O(N^{1/d})$)



Numerical Example: Homogeneous Velocity

N	$\omega/2\pi$	p	$T_{\mathtt{fact}}$	$N_{\mathtt{it}}$	$T_{\mathtt{it}}$	$T_{\mathtt{total}}$
202×202	20.1	2	1.09	2	0.66	2.63
404×404	40.3	4	1.00	3	0.58	2.56
808×808	80.7	8	1.41	3	1.26	6.02
1616×1616	161.5	16	2.80	2	3.39	14.05
3232×3232	323.1	32	4.41	3	5.47	27.47
6464×6464	646.3	64	8.34	4	11.09	67.74
12928×12928	1292.7	128	15.66	5	22.39	160.88

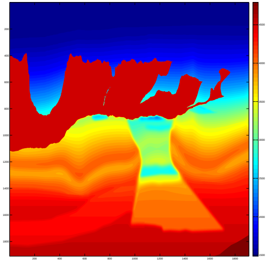
Numerical Example: Homogeneous Velocity



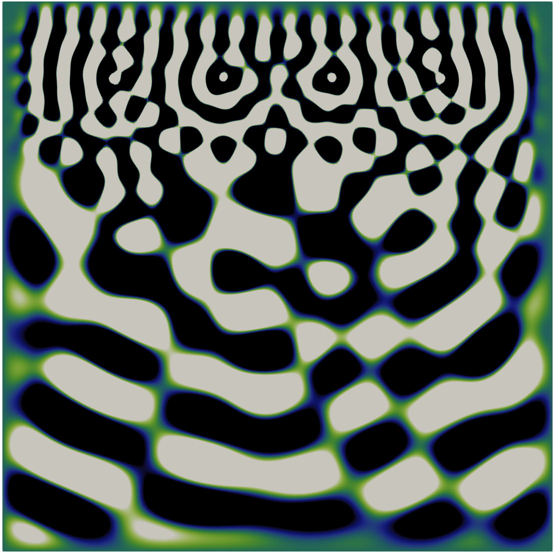
Numerical Example: BP Model

N (without PML)	$\omega/2\pi$	$q=r$	BG1	BG2	BG1 with salt	BG2 with salt	BP model
202×202	20.1	2	1	4	7	6	7
404×404	40.3	4	2	4	9	9	9
808×808	80.7	8	4	6	12	12	12
1616×1616	161.4	16	5	6	15	15	15
3232×3232	323.1	32	6	7	17	17	16
6464×6464	646.3	64	7	7	19	19	19
12928×12928	1292.7	128	8	8	21	21	20

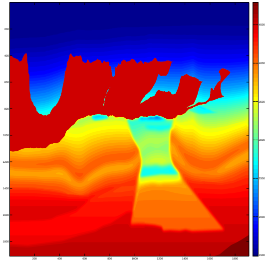
Numerical Example: BP Model



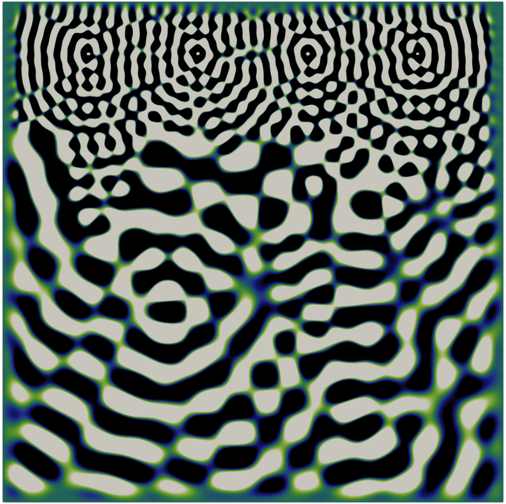
- max. 16 wavelengths in domain
- PML width: 1.25 wavelengths
- 2×2 domain decomposition



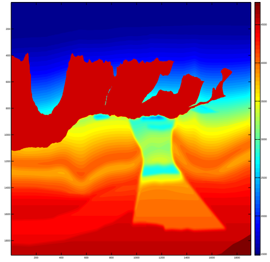
Numerical Example: BP Model



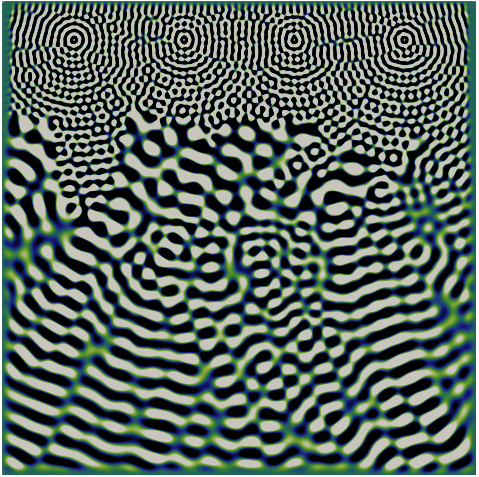
- max. 32 wavelengths in domain
- PML width: 1.5 wavelengths
- 4×4 domain decomposition

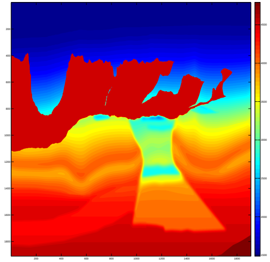


Numerical Example: BP Model

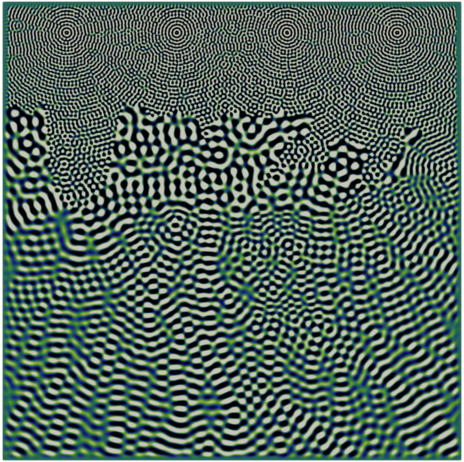


- max. 64 wavelengths in domain
- PML width: 1.75 wavelengths
- 8×8 domain decomposition

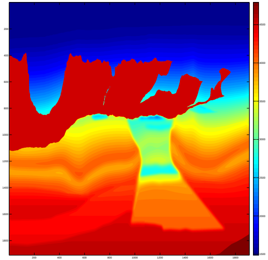


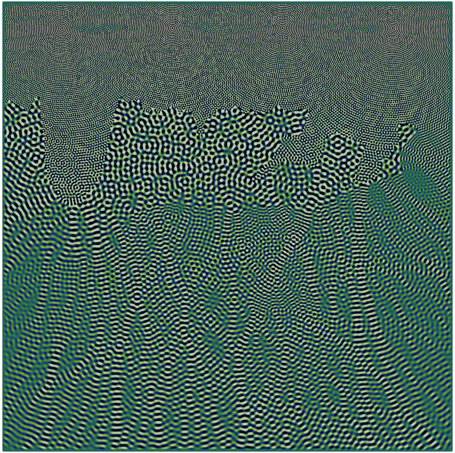


- max. 128 wavelengths in domain
- PML width: 2 wavelengths
- 16×16 domain decomposition



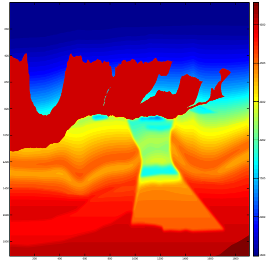
Numerical Example: BP Model



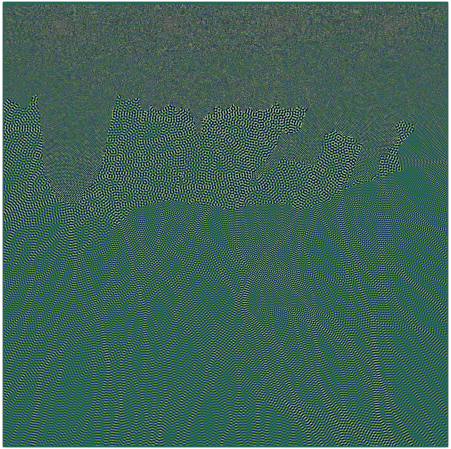


- max. 256 wavelengths in domain
- PML width: 2.25 wavelengths
- 32×32 domain decomposition

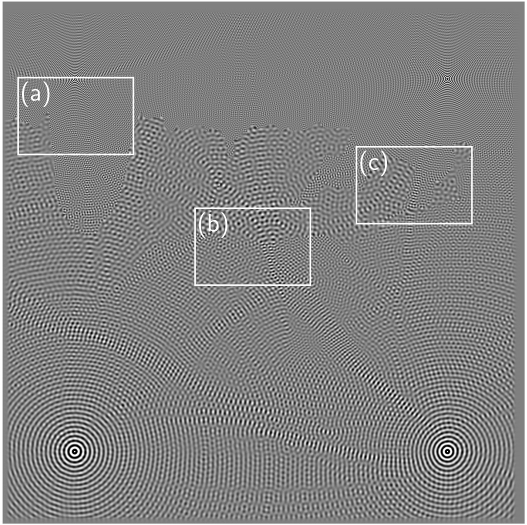
Numerical Example: BP Model

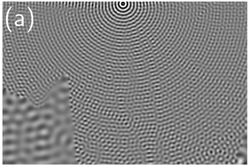


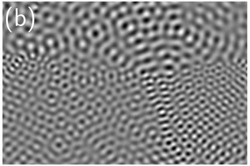
- max. 512 wavelengths in domain
- PML width: 2.5 wavelengths
- 64×64 domain decomposition

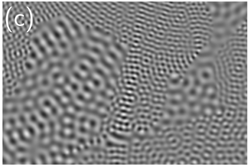


Numerical Example: BP Model

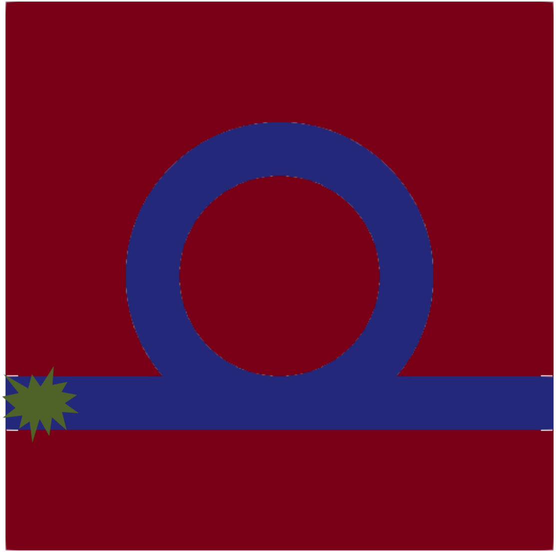








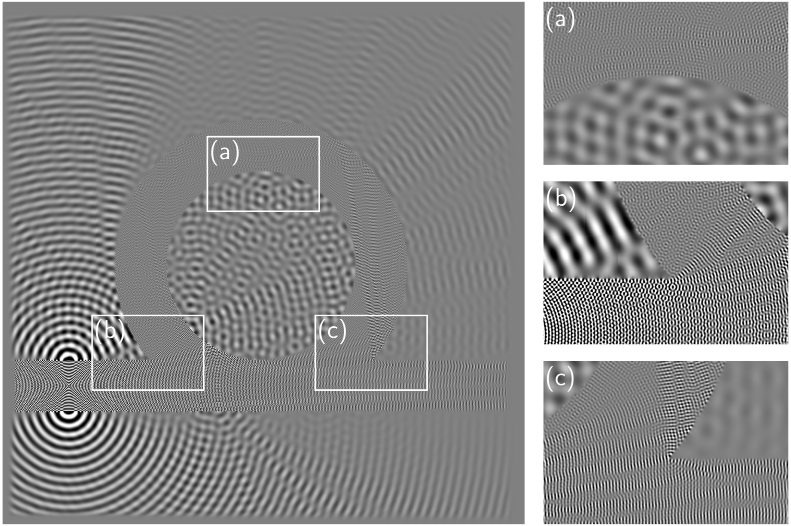
Numerical Example: High-contrast Waveguide



Numerical Example: High-contrast Waveguide

N (without PML)	$\omega/2\pi$	$m = n$	Contrast ratio 2	3	4	5	6
202×202	20.1	2	18	24	24	25	26
404×404	40.3	4	28	29	29	28	30
808×808	80.7	8	30	32	34	33	33
1616×1616	161.5	16	31	33	33	34	35
3232×3232	323.1	32	32	34	36	36	37
6464×6464	646.3	64	32	34	35	36	36

Numerical Example: High-contrast Waveguide



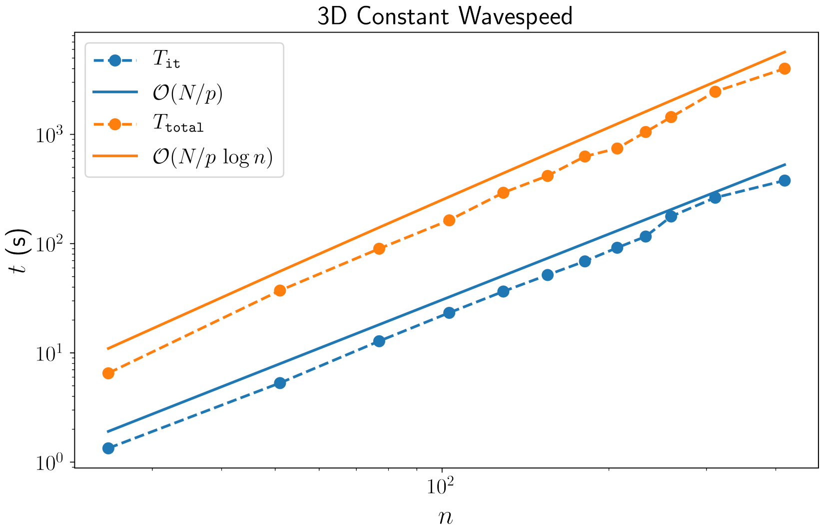
What about 3D?

- There are serious challenges in optimizing parallelism in 3D
- Current implementation uses vertically extruded subdomains
- Consider it a quasi-2D domain decomposition

Numerical Example: Homogeneous Velocity (3D)

N (without PML)	$\omega/2\pi$	p	$T_{\texttt{fact}}$	$N_{\texttt{it}}$	$T_{\texttt{it}}$	$T_{\texttt{total}}$
$26 \times 26 \times 26$	4.17	2	.04	4	1.34	6.52
$52 \times 52 \times 52$	8.50	4	5.54	6	5.30	37.17
$78 \times 78 \times 78$	12.83	6	12.42	6	12.80	89.76
$104 \times 104 \times 104$	17.17	8	22.91	6	23.27	163.62
$130 \times 130 \times 130$	21.50	10	37.53	7	36.47	292.33
$156 \times 156 \times 156$	25.83	12	52.47	7	51.62	417.08
$182 \times 182 \times 182$	30.17	14	71.71	8	68.92	627.23
$208 \times 208 \times 208$	34.50	16	96.14	7	91.65	743.37
$234 \times 234 \times 234$	38.83	18	124.64	8	116.08	1050.31
$260 \times 260 \times 260$	43.17	20	211.87	7	177.21	1438.12
$312 \times 312 \times 312$	51.83	24	314.93	8	263.16	2457.40
$416 \times 416 \times 416$	69.17	32	418.36	9	377.60	3992.63

Numerical Example: Homogeneous Velocity (3D)



Summary and Outlook

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
- Paper: `https://arxiv.org/abs/1909.01467`

Summary and Outlook

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
- Paper: `https://arxiv.org/abs/1909.01467`

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?)