Software and algorithms for large-scale seismic inverse problems

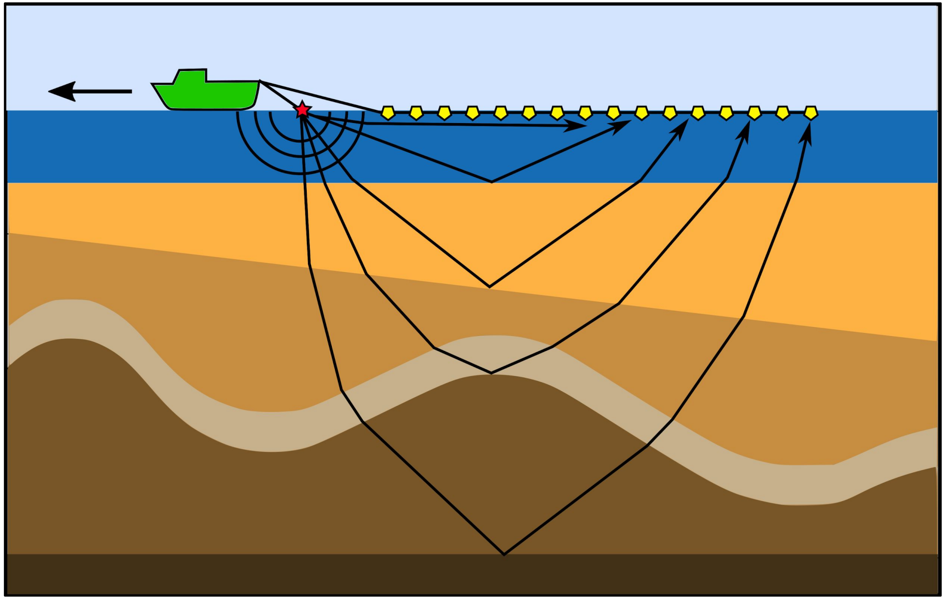
Philipp A. Witte
Ph.D. Defense of Dissertation
February 19, 2020

SLIM
Georgia Institute of Technology

Seismic inverse problems

Estimate unknown subsurface properties from seismic data:

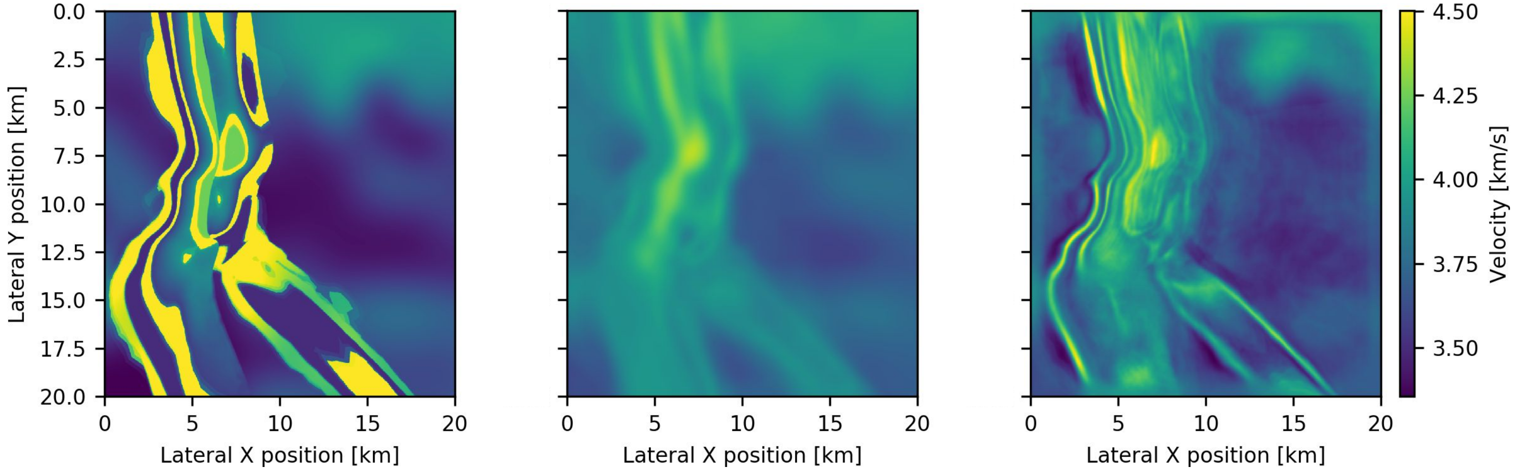
- Image geological structures/discontinuities
- Estimate physical rock properties (wave speed, density, etc.)



Seismic inverse problems

Parameter estimation: non-linear optimization problems

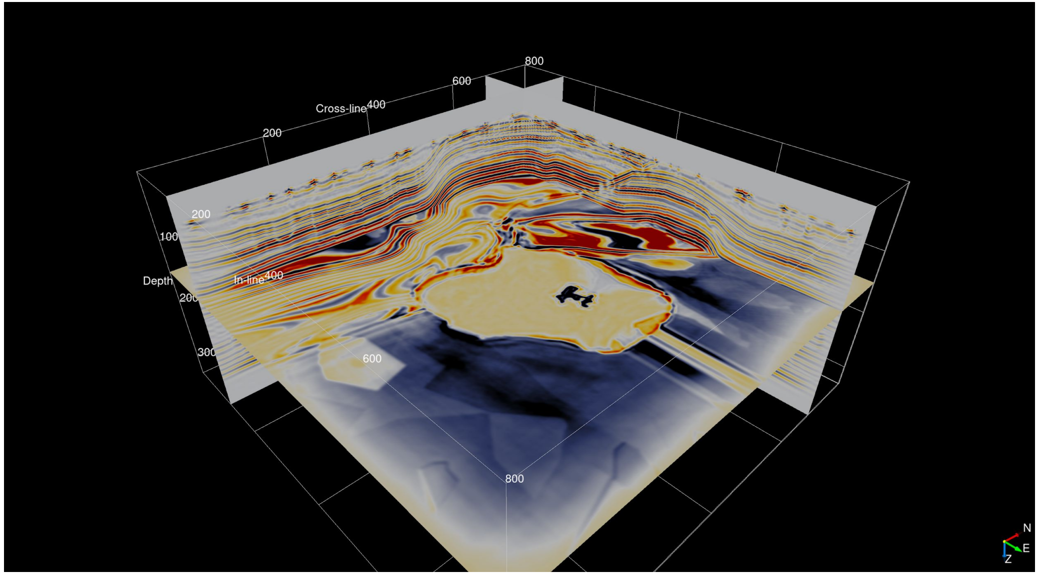
$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2$$



Seismic inverse problems

Seismic imaging: linear least squares optimization problems

$$\underset{\delta\mathbf{m}}{\text{minimize}} \quad \Phi(\delta\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathbf{J}(\mathbf{m}, \mathbf{q}_i)\delta\mathbf{m} - \mathbf{d}_i||_2^2$$



Deploy software to HPC clusters + cloud

amazon web services™

Azure

Software for seismic inverse problems

Computational challenges:

- Up to several billions of unknown variables
- Observations of several magnitudes larger (several TB of data)
- Expensive PDEs: propagate wavefields over thousands of time steps

Mathematical challenges:

- Problems are non-linear, non-convex or ill-conditioned
- Noise + non-uniqueness
- Can only afford few epochs/data passes during optimization

⇨ **Implement sophisticated algorithms + scale to peta/exascale problems**

Software for seismic inverse problems

State of the art software:

- Often trades abstractions for performance, proprietary
- Manually optimized code monoliths in C/Fortran

Contribution of chapter 2:

- High-level, open-source framework in the Julia language
- Manage complexity through layers of abstractions and code generation
- Vertical integration of compiler technologies + finite difference DSLs + abstract user interfaces

Compressive seismic imaging

Further address computational cost through algorithms:

- Conventionally: full gradient methods (GD, GN, QN, PG)
- Exploit redundancies and/or structure in data and parameters (sparsity, low-rank, etc.)

Contribution of chapter 3:

- Seismic imaging in the frequency domain using time-domain modeling
- Leverage ideas from compressive sensing to address prohibitively high cost
- An algorithm whose memory requirements are independent of recording length

Serverless imaging in the cloud

High computational cost of seismic inversion:

- Need access to HPC resources
- Only available to few companies + academic institutions
- Massively parallel applications that run from days to weeks

Cloud computing as alternative to on-premise clusters:

- Virtually unlimited resource availability
- Pay-as-you-go: no upfront cost
- Oftentimes inferior performance to on-premise HPC
- *Lift and shift*: high operating cost + resilience issues

Serverless imaging in the cloud

Contribution of chapter 4:

- An event-driven approach to HPC in the cloud
- *Serverless* algorithms based on map-reduce
- Automatic resource allocation
- Reduce operating cost up to factor of 10x
- Industry uptake: collaborations with cloud providers (GCP, Azure) to validate technology on industry-scale problems

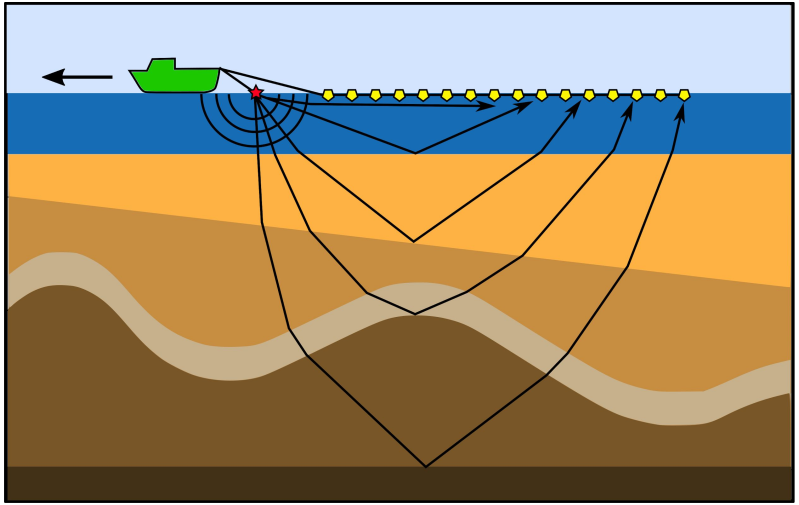
Chapter 2

A large-scale framework for symbolic implementations of seismic inversion algorithms in Julia

Seismic inverse problems

Mathematical formulation of seismic inverse problems: [1],[2]

$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2$$



Software:

- evaluate $\mathcal{F}(\mathbf{m})$ and its gradient
- $\mathcal{F}(\mathbf{m})$ encodes forward problem

[1] A. Tarantola, 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, 49.
[2] J. Virieux and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, 74

Forward problem

The ***forward*** problem:

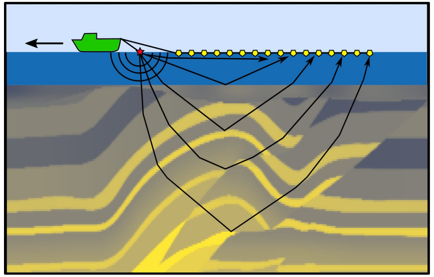
$$\mathcal{F}(\mathbf{m},\mathbf{q}) = \mathbf{P}_r \underbrace{\mathbf{A}(\mathbf{m})^{-1}\mathbf{P}_s^\top \mathbf{q}}_{\mathbf{u}}$$

Discretized acoustic wave equation:

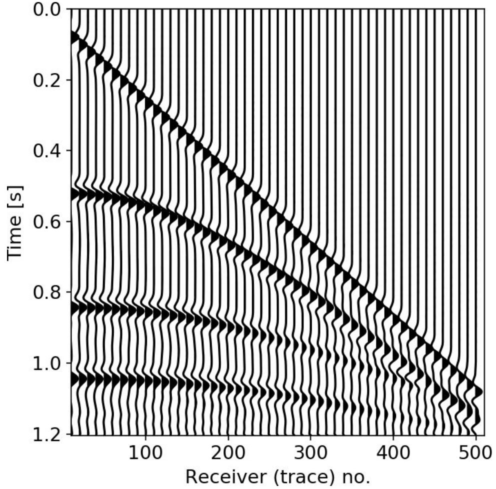
$$\mathbf{A}(\mathbf{m}) = \mathbf{m} \odot \frac{\partial^2}{\partial t^2} - \nabla^2$$

Solve via finite-difference time-stepping:

$$\mathbf{u}^{n+1} = \left[2 + \frac{\Delta t^2}{\mathbf{m}} \odot \mathbf{L}\right]\mathbf{u}^n - \mathbf{u}^{n-1} + \frac{\Delta t^2}{\mathbf{m}} \odot \mathbf{P}_s^\top \mathbf{q}^{n+1}$$



$\mathcal{F}(\mathbf{m})$



Inverse problem

The *inverse* problem:

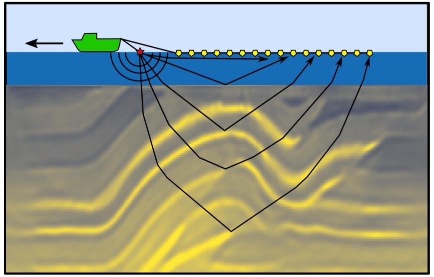
$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2$$

Sensitivities w.r.t. model parameters $\frac{\partial \mathcal{F}(\mathbf{m})}{\partial \mathbf{m}}$

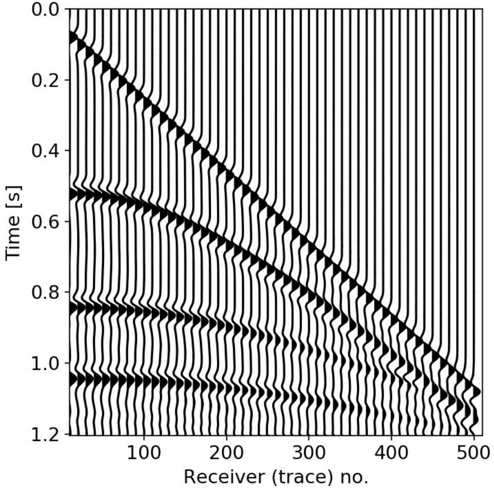
$$\mathbf{J} = -\mathbf{P}_r \mathbf{A}(\mathbf{m})^{-1} \mathrm{diag}\left(\frac{\partial \mathbf{A}(\mathbf{m})}{\partial \mathbf{m}} \mathbf{A}(\mathbf{m})^{-1} \mathbf{P}_s^\top \mathbf{q} \right)$$

Gradient of objective function via backpropagation:

$$\mathbf{g} = \sum_{i=1}^{n_s} \mathbf{J}^\top \left(\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i \right)$$



$\min_{\mathbf{m}} \Phi(\mathbf{m})$



[1] W. Symes, D. Sun and M. Enriquez, 2011, From modelling to inversion: Designing a well-adapted simulator, Geophysical Prospecting, 59.
[2] S. Fomel, P. Sava, I. Vlad, L. Yang, and V. Bashkardin, 2013, Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments: Journal of Open Research Software, 1.
[3] L. Krischer, A. Fichtner, S. Zukauskaite, and H. Igel, 2015, Large-scale seismic inversion framework: Seismological Research Letters, 86.

Motivation

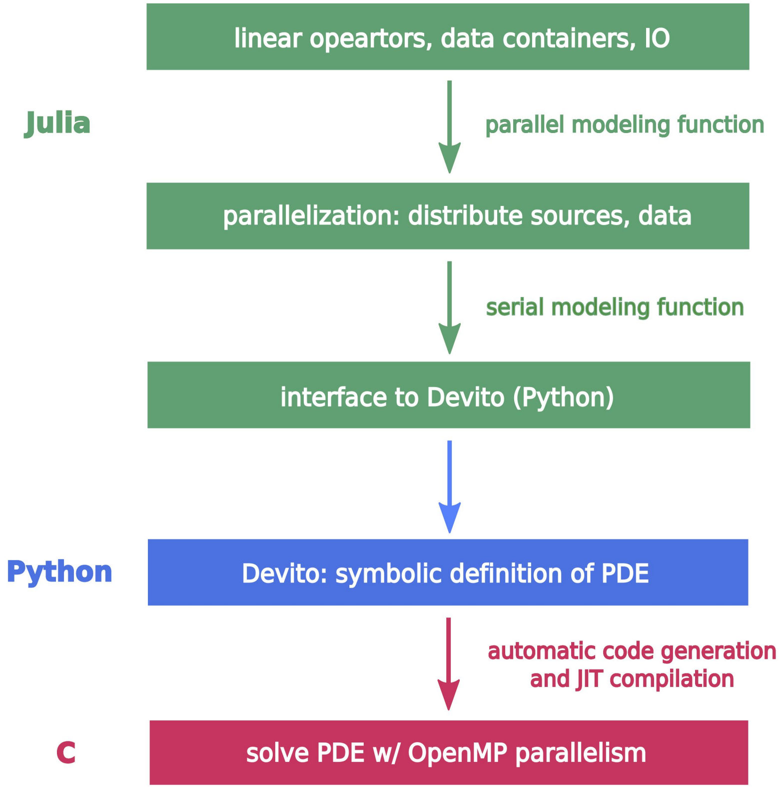
Software for seismic inverse problems:

- Needs fast PDE solvers w/ correct adjoints, gradients
- Robust parallelization (cluster/cloud) w/ resilience
- Manage large seismic data volumes and meta data
- Enable implementations of various optimization algorithms

The reality of seismic inversion codes:

- Academic packages in C, Python, MATLAB not scalable [1],[2],[3]
- Software in O&G companies: low level code in C or FORTRAN
- Mixing of PDE solvers, I/O, parallelization, data processing + algorithms
- Hard to modify

The Julia Devito Inversion Framework



Manage complexity through vertical integration of technologies:

- A domain-specific language + compiler (Devito) to express and solve wave equations [1][2]
- Parallelization in Julia
- High-level abstractions for implementing optimization algorithms
- Interfaces to optimization + deep learning libraries

[1] M. Louboutin, M. Lange, F. Luporini, N. Kukreja, P. A. Witte, F. J. Herrmann, P. Velesko and G. J. Gorman, 2019, Devito 3.1.0: an embedded domain-specific language for finite differences and geophysical exploration: Geoscientific model development, 12, 3.

[2] F. Luporini, M. Lange, M. Louboutin, N. Kukreja, J. Hückelheim, , C. Yount, P. A. Witte, P. H. J. Kelly, F. J. Herrmann and G. J. Gorman, 2019, Architecture and performance of Devito, a system for automated stencil computation, arXiv preprints.

The Julia Devito Inversion framework

Contribution: A framework for symbolic implementations of seismic inversion algorithms

- High-level Julia package built on top of Devito
- Matrix-free linear operators and abstract data vectors
- Formulate algorithms in terms of linear algebra expressions:

$$\mathbf{d}_{\text{pred}} = \mathbf{P}_r \mathbf{A}(\mathbf{m})^{-1} \mathbf{P}_s^\top \mathbf{q}$$

$$f = \frac{1}{2} ||\mathbf{d}_{\text{pred}} - \mathbf{d}_{\text{obs}}||_2^2$$

$$\mathbf{g} = \mathbf{J}^\top \left(\mathbf{d}_{\text{pred}} - \mathbf{d}_{\text{obs}}\right)$$

```
# Model data
d_pred = Pr*A_inv*Ps'*q

# Function value
f = .5f0*norm(d_pred - d_obs)^2

# Gradient
g = J'*(d_pred - d_obs)
```

Example 1: Gauss-Newton method

Gauss-Newton subproblems:

- Pass matrix-free linear operator to third party solvers
- LSQR from Julia's *IterativeSolvers.jl* package [1]
- Overload necessary operations in *lsqr* for JUDI operators/vectors

```
# Main loop
for j=1:maxiter

    # Model predicted data
    d_pred = Pr*A_inv*Ps'*q

    # GN update direction
    p = lsqr(J, d_pred - d_obs; maxiter=10)

    # Update model
    model.m = model.m - reshape(p, model.n)
end
```

[1] IterativeSolvers.jl, https://github.com/JuliaMath/IterativeSolvers.jl

[1] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, V. Shah, 2018, Fashionable Modelling with Flux, Computing Research Repository, arXiv, http://arxiv.org/abs/1811.01457

Example 2: Deep Learning

Combine seismic modeling operators with (deep) CNNs

- Backpropagate through JUDI operators
- Integrate into deep learning libraries (e.g. Julia's Flux.jl[1])

Once again, abstractions pay off:

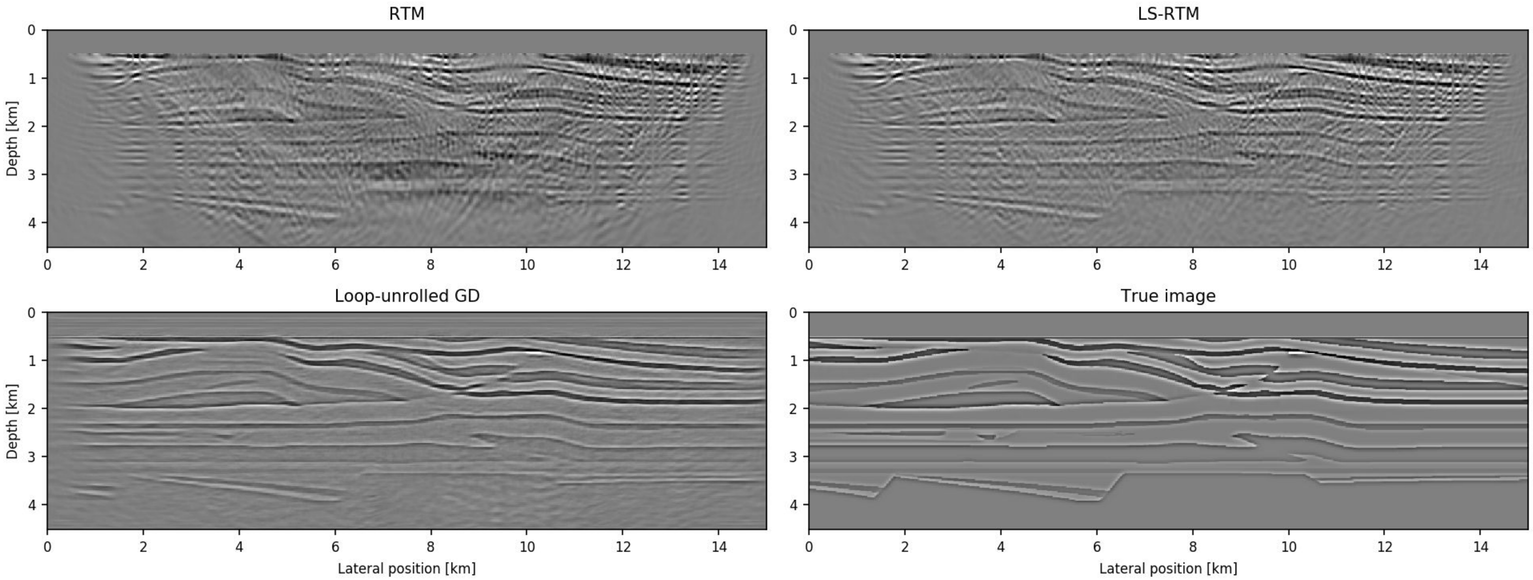
- No need to backpropagate through solvers using automatic differentiation
- Implement backpropagation through JUDI operators
- For linear operators, one line of Julia code:

```
@adjoint *(J::judiJacobian, x) = *(J, x), Δ -> (nothing, transpose(J) * Δ)
```

Example 2: Deep Learning

Use as building blocks for arbitrary complicate CNNs

- E.g. CNNs inspired by loop unrolled optimization algorithms [1]
- Composition of convolutional layers + JUDI operators



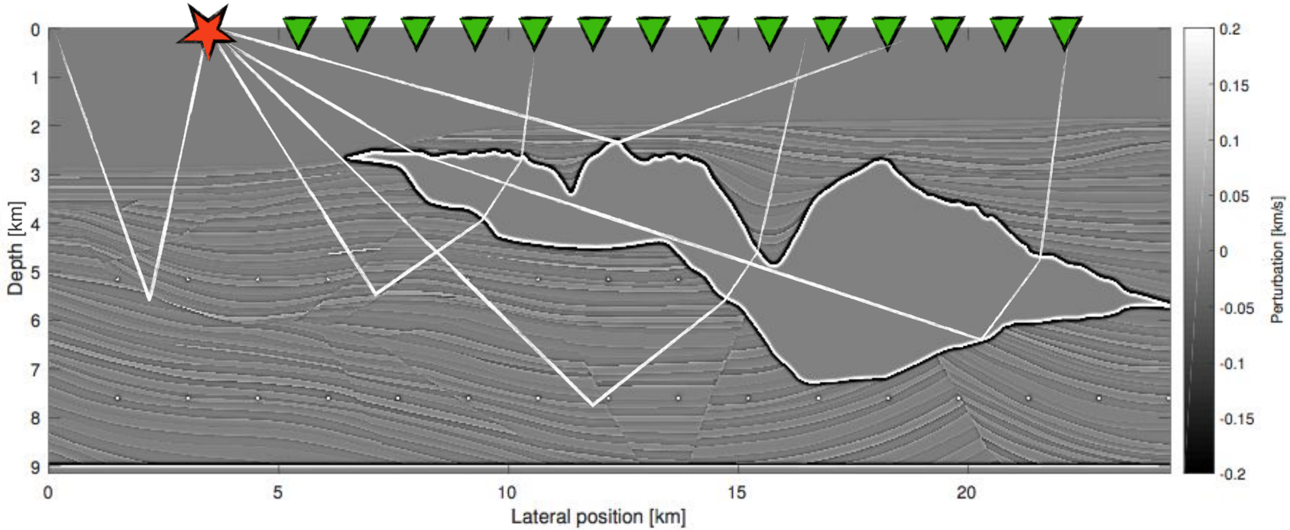
[1] Adler, Jonas, and Ozan Öktem, 2017, Solving ill-posed inverse problems using iterative deep neural networks, Inverse Problems 33, 12.

Chapter 3

Compressive least squares migration with on-the-fly Fourier transforms

Compressive seismic imaging

- Image velocity/impedance contrasts in the subsurface



Motivation

Computational challenges of seismic imaging: [1][2][3]

- LS-RTM: Need to solve large number of PDEs at every iteration
- Expensive PDE solves: propagate waves over ~20,000 time steps
- Large data sets (in the range of TB)
- Large number of variables (between 1e6 and 1e10)
- Large memory requirements for backpropagation
- **Memory demand grows linearly with no. of time steps**

[1] A. A. Valenciano, 2008, Imaging by wave-equation inversion, PhD Thesis, Stanford University.
[2] S. Dong, J. Cai, M. Guo, S. Suh, Z. Zhang, B. Wang and Z. Li, 2012, Least-squares reverse time migration: Towards true amplitude imaging and improving the resolution, SEG, Expanded Abstracts.
[3] F. J. Herrmann, P. Moghaddam and C. C. Stolk, 2008, Sparsity- and continuity-promoting seismic imaging recovery with curvelet frames, Applied and Computational Harmonic Analysis, 24, 2.

Time-to-frequency conversion

Least squares RTM objective function in frequency domain:

$$\underset{\delta \mathbf{m}}{\text{minimize}} \quad \sum_{j=1}^{n_s} \sum_{k=1}^{n_f} \frac{1}{2} \left\| \mathbf{J}(\mathbf{m}_0, \bar{q}_{jk})\, \delta \mathbf{m} - \bar{\mathbf{d}}_{jk}^{\text{obs}} \right\|_2^2 .$$

With:

$\mathbf{J} \in \mathbb{C}^{n_r \times n}$	linearized Born scattering operator
$\delta \mathbf{m} \in \mathbb{C}^{n}$	unknown image
$\mathbf{m}_0 \in \mathbb{C}^{n}$	migration velocity (assumed to be known)
$\bar{\mathbf{d}}_{\text{jk}}^{\text{obs}} \in \mathbb{C}^{n_r}$	observed seismic data
$\bar{q}_{jk} \in \mathbb{C}$	source wavelet (assumed to be known)

Time-to-frequency conversion

Seismic imaging in the frequency domain:

- Avoids backpropagation over time steps
- But: need to solve large-scale 2D/3D Helmholtz equation
- Alternatively: time-domain modeling with time-to-frequency conversion:

$$\bar{\mathbf{d}}_{jk}^{\mathrm{pred}} = -\mathbf{P}_r\mathbf{R}_k\mathbf{F}\mathbf{A}(\mathbf{m}_0)^{-1}\mathrm{diag}\left[\frac{\partial\mathbf{A}(\mathbf{m}_0)}{\partial\mathbf{m}_0}\mathbf{A}(\mathbf{m}_0)^{-1}\mathbf{F}^*\mathbf{R}_k^*\mathbf{p}_s^*\bar{q}_{jk}\right]\delta\mathbf{m}$$

- Gradient given by:

$$\bar{\mathbf{g}}_{jk} = -\mathrm{Re}\left[\mathrm{diag}\left(\omega_k^2\bar{\mathbf{u}}_{jk}\right)^*\bar{\mathbf{v}}_{jk}\right]$$

- With forward and adjoint wavefields:

$$\bar{\mathbf{u}}_{jk} = \mathbf{R}_k\mathbf{F}\mathbf{A}(\mathbf{m}_0)^{-1}\mathbf{F}^*\mathbf{R}_k^*\mathbf{p}_s^*\bar{q}_{jk}$$

$$\bar{\mathbf{v}}_{jk} = \mathbf{R}_k\mathbf{F}\mathbf{A}(\mathbf{m}_0)^{-*}\mathbf{F}^*\mathbf{R}_k^*\mathbf{P}_r^*(\bar{\mathbf{d}}_{jk}^{\mathrm{pred}} - \bar{\mathbf{d}}_{jk}^{\mathrm{obs}})$$

[1] C. M. Furse, 1998, Faster than Fourier - Ultra-efficient time-to-frequency domain conversions for FDTD, Antennas and Propagation Society International Symposium.
[2] L. Sirgue, J. T. Etgen, U. Albertin and S. Brandsberg-Dahl, 2010, Sirgue, L. and Etgen, J.T. and Albertin, U. and Brandsberg-Dahl, S., US Patents.

SLIM

Time-to-frequency conversion

So far: explicit DFTs in modeling expressions

- Replace w/ on-the-fly DFTs[1][2]
- During time stepping compute:

$$\bar{\mathbf{u}}_{jk}^{\text{real}} = \sum_{i=1}^{n_t} \cos(2\pi f_k i \Delta t)\mathbf{u}_i,$$

$$\bar{\mathbf{u}}_{jk}^{\text{imag}} = -\sum_{i=1}^{n_t} \sin(2\pi f_k i \Delta t)\mathbf{u}_i$$

$$\bar{\mathbf{u}}_{jk}^{\text{real}}, \bar{\mathbf{u}}_{jk}^{\text{imag}}, \mathbf{u}_i \in \mathbb{R}^n$$

- Gradient then given by (derivation for impedance in chapter 3):

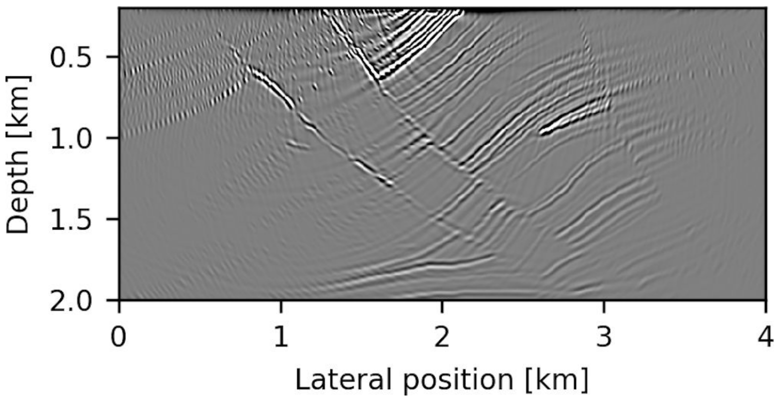
$$\bar{\mathbf{g}}_{jk} = -\sum_{i=1}^{n_t} (2\pi f_k)^2 \text{diag}\left[\bar{\mathbf{u}}_{jk}^{\text{real}} \cos(2\pi f_k i \Delta t) - \bar{\mathbf{u}}_{jk}^{\text{imag}} \sin(2\pi f_k i \Delta)\right] \mathbf{v}_i$$

Sparsity-promoting LS-RTM

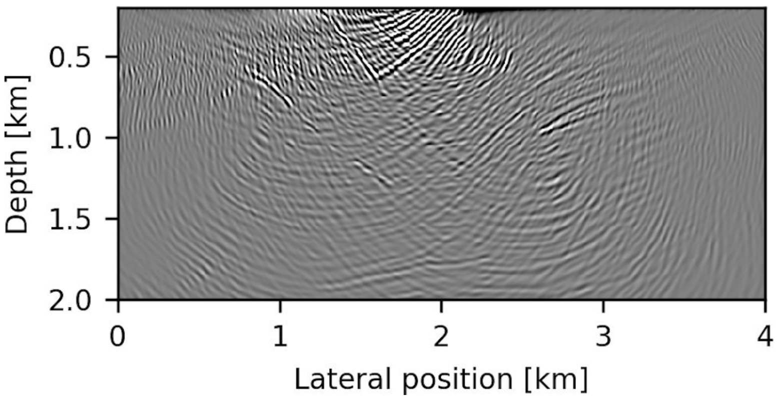
Frequency-domain imaging w/ time modeling:

- Need fine frequency sampling for imaging
- Periodic subsampling causes aliasing/coherent artifacts
- Compressive sensing: random sampling

Time domain: 1 source



Frequency domain: 1 source + 10 frequencies

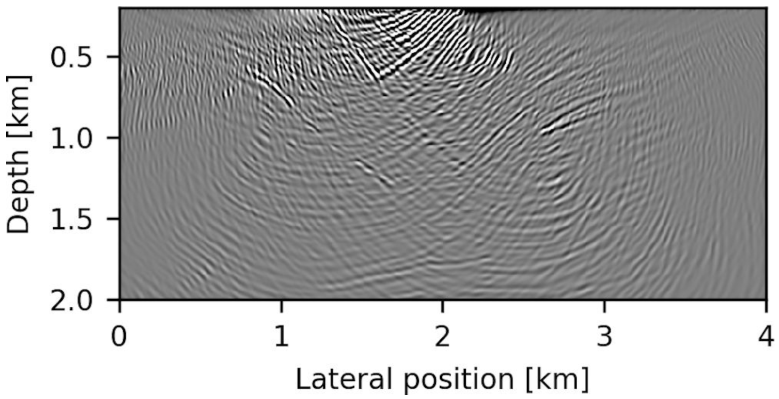


Sparsity-promoting LS-RTM

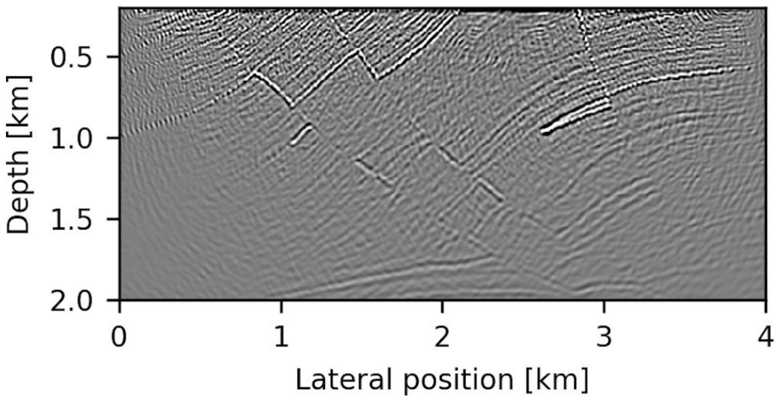
Compressive sensing (CS) inspired imaging:

- Seismic imaging is overdetermined LS problem
- But: work w/ random blocks of rows (i.e. frequencies/sources) in each iteration
- Frequency subsampling -> coherent reflectors + incoherent noise

Frequency domain: 1 source + 10 frequencies



Frequency domain: 10 sources + 10 frequencies



[1] F. J. Herrmann and X. Li, 2012, Efficient least-squares imaging with sparsity promotion and compressive sensing, Geophysical Prospecting, 60.
[2] N. Tu and F. J. Herrmann, 2015, Fast imaging with surface-related multiples by sparse inversion, Geophysical Journal International, 201, 1.

SLIM

Sparsity-promoting LS-RTM

LS-RTM as sparsity-promoting minimization problem:[1][2]

- elastic net objective function (strongly convex)

$$\begin{aligned} \underset{\delta \mathbf{z}}{\text{minimize}} \quad & \lambda ||\mathbf{C}\ \delta \mathbf{z}||_1 + \frac{1}{2} ||\mathbf{C}\ \delta \mathbf{z}||_2^2 \\ \text{subject to:} \quad & \sum_{j=1}^{n_s} \sum_{k=1}^{n_f} \left\| \mathbf{M}_l^{-1} \mathbf{J}(\mathbf{m}_0, \bar{q}_{jk}) \mathbf{M}_r^{-1} \delta \mathbf{z} - \mathbf{M}_l^{-1} \bar{\mathbf{d}}_{jk}^{\text{obs}} \right\|_2 \leq \sigma \end{aligned}$$

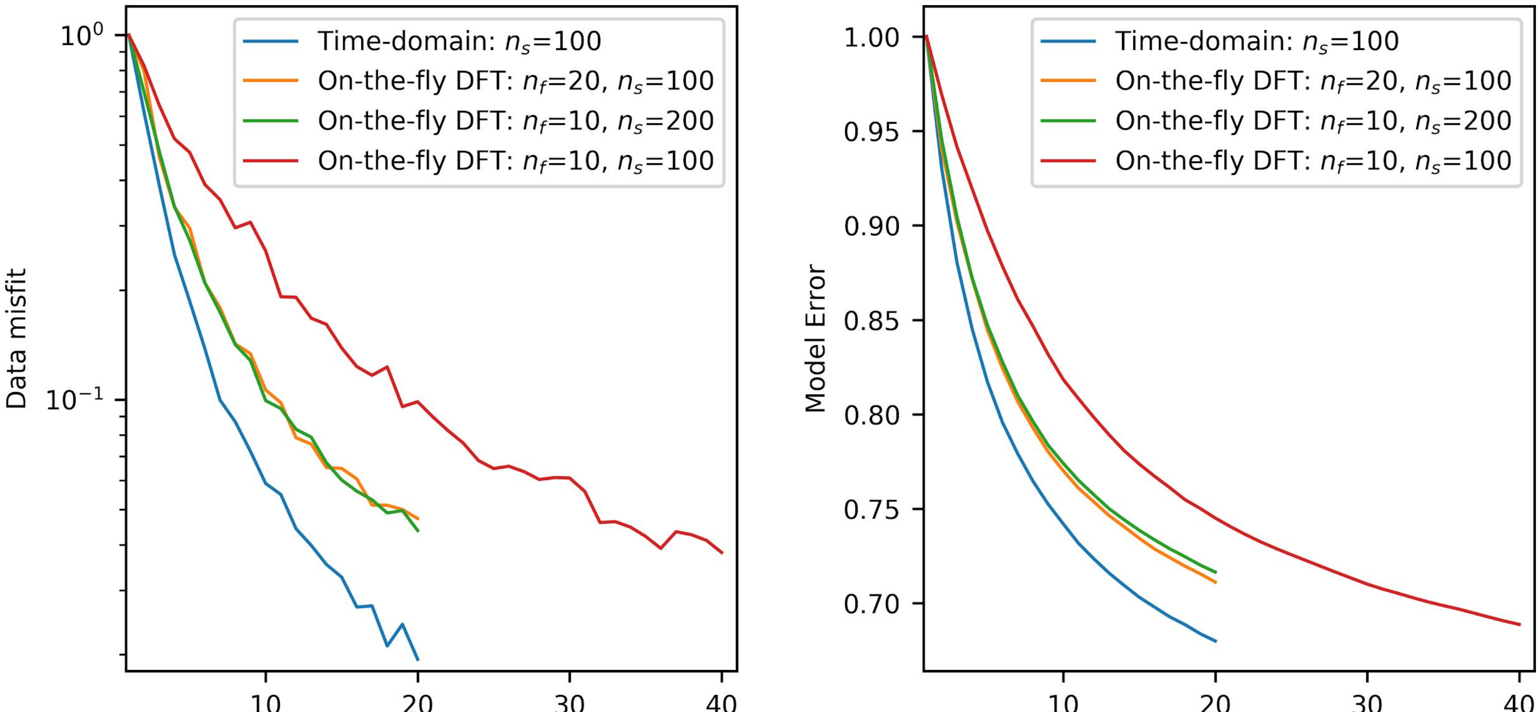
- Solve with linearized Bregman method [1][2]

[1] W. Yin, 2010, Analysis and Generalizations of the Linearized Bregman Method, SIAM Journal on Imaging Sciences, 3, 4.
[2] D. Lorenz, F. Schoepfer and S. Wenger, 2014, The Linearized Bregman Method via Split Feasibility Problems: Analysis and Generalizations, SIAM Journal on Imaging Sciences, 7, 2.

Numerical example

Seismic imaging example (4e6 unknowns, 280e6 data points)

- Run for given batchsize and fixed no. of epochs



Chapter 4

An event-driven approach to seismic imaging in the cloud

Motivation

Seismic imaging and inversion in the cloud?

Pro:

- Pay-as-you-go pricing model, no upfront costs or maintenance
- Theoretically unlimited scalability
- Wide range of hardware available (high-memory nodes, GPUs, etc.)

Con:

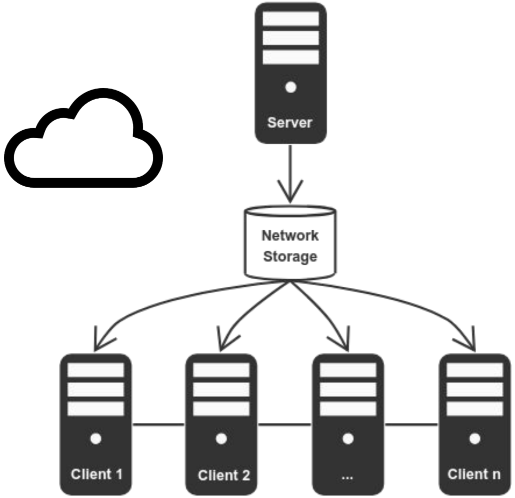
- High cost (depending on hardware)
- High latency and low bandwidth [1][2]
- Poor mean-time-between-failures compared to HPC environment

[1] S. Benedict, 2013, Performance issues and performance analysis tools for HPC cloud applications: a survey, Computing, 95, 2.
[2] C. Kotas, T. Naughton, and N. Imam, 2018, A comparison of Amazon Web Services and Microsoft Azure cloud platforms for high performance computing, IEEE International Conference on Consumer Electronics.

Motivation

Conventional approach of HPC in the cloud:

- Set up cluster of cloud instances
  (MIT StarCluster, AWS ParallelCluster)
- Instances communicate via MPI using Ethernet
- Current approach in O&G[1]



Drawbacks of *lift and shift*:

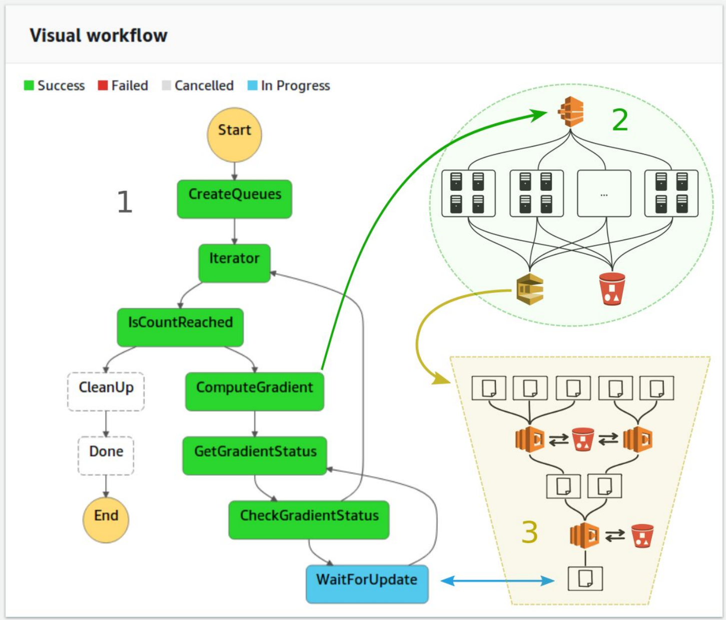
- High cost: need to permanently run (and pay) for instances
- Resilience issues (instances failures, spot-related shut-downs)[2]
- MPI jobs are bad at dealing with node failures
- Reliance on slow connections (bandwidth, latency)

[1] XWI on AWS: Revolutionary earth model building on the cloud, https://www.s-cube.com/xwi-on-the-cloud/
[2] P. Mehrotra, J. Djomehri, S. Heistand, R. Hood, H. Jin, A. Lazanoff, S. Saini, and R. Biswas, Performance evaluation of Amazon EC2 for NASA high-performance computing applications, Concurrency and Computation: Practice and Experience, 28, 4.

Event-driven seismic imaging on AWS

Our approach:

- Event-driven workflow based on AWS services
- Algorithm as serverless visual workflow (AWS Step Functions)
- Serverless map-reduce using AWS Batch + AWS Lambda



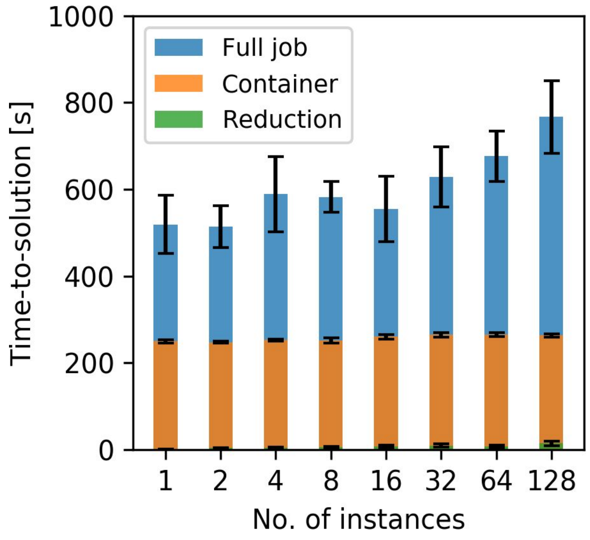
Advantages:

- Serverless: no master/server
- No idle time
- AWS manages resilience and scheduling
- Nested parallelization possible

Performance analysis

Weak scaling:

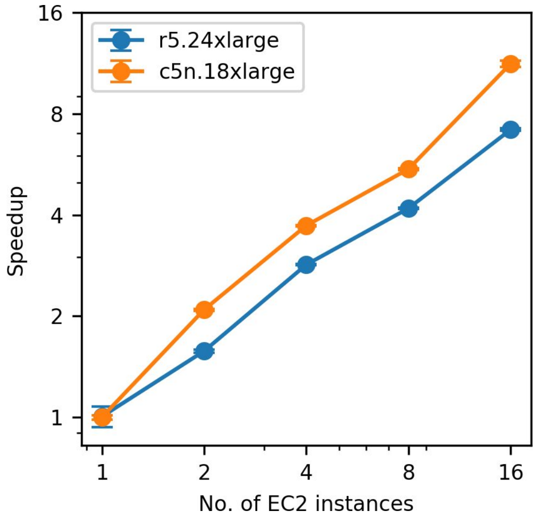
- Compute 1 gradient per instance/node (BP synthetic 2004 model)
- Time-to-solution as function of batchsize (no. of instances)

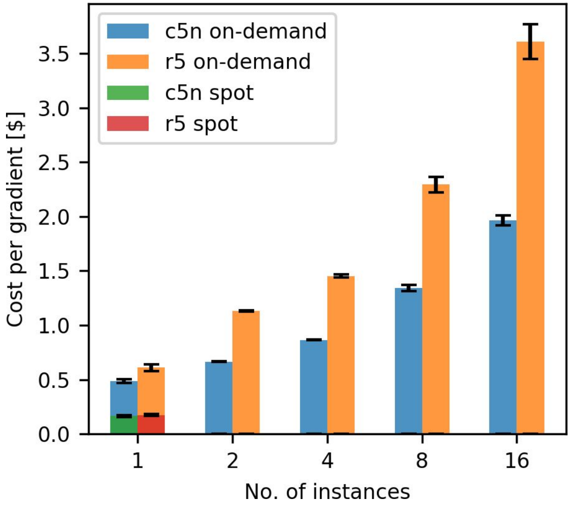


Performance analysis

Strong scaling:

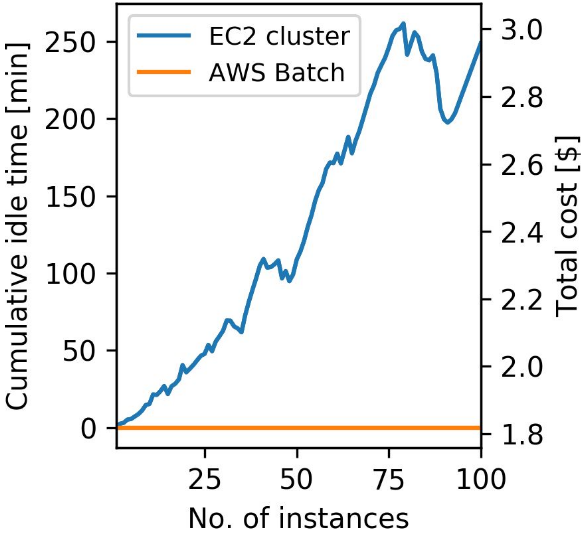
- Compute single gradient with domain decomposition (MPI)
- Interested in both speed-up and cost
- Additional scaling tests in chapter 4 (OMP, MPI, hybrid, resilience)

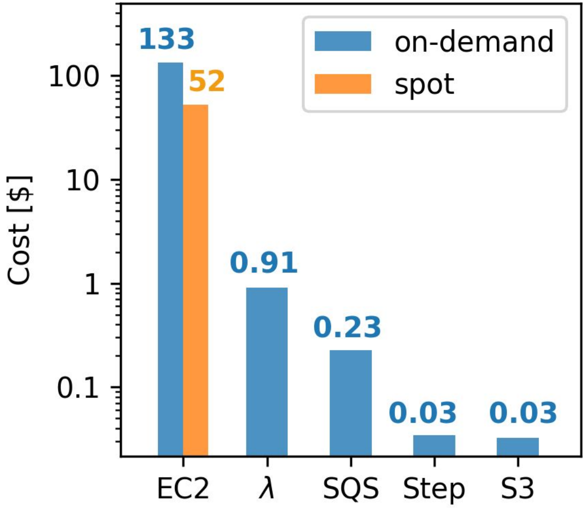




Cost analysis

- Compute gradient for batch size 100
- Model idle-time for fixed cluster and increasing no. of instances
- Compare with AWS Batch: no idle time
- Serverless approach: cost other than EC2 negligible

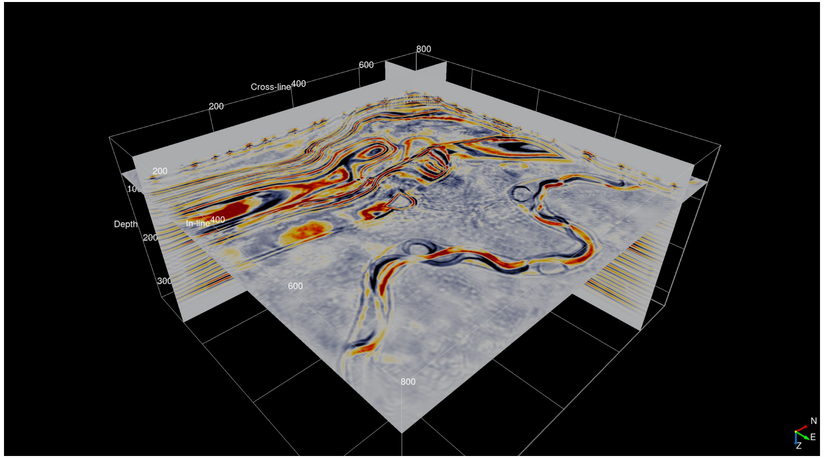


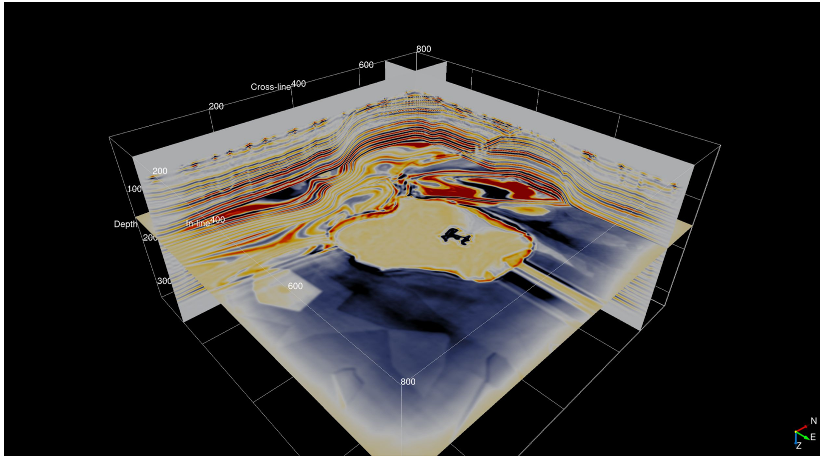


Large-scale case study on Azure

3D Imaging case study

- Data set: 1,500 shot records, each with 638, 401 receivers
- Model: 10 x 10 x 3.325 km (881 x 881 x 347 grid point)
- PDE: tilted transversely isotropic (TTI) wave equation
- Cost: < 10,000$ on 100 E64/E64s instances





Conclusions

Main contributions of this thesis:

- An open-source framework for geophysical inversion in Julia
- Enables (reproducible) research on industry-scale inverse problems
- A practical workflow for seismic imaging using compressive sensing and on-the-fly Fourier transforms
- Enables large-scale imaging w/o restrictions on propagation time
- A serverless approach to seismic imaging in the cloud
- Enables using the cloud for large-scale distributed computing w/o cost and resilience pitfalls

Publications

[1] P. A. Witte, M. Louboutin, N. Kukreja, F. Luporini, M. Lange, G. G. Gorman and Felix J. Herrmann, 2019, *A large-scale framework for symbolic implementations of seismic inversion algorithms in Julia*, Geophysics, 84, 3.

[2] P. A. Witte, M. Louboutin, F. Luporini, G. G. Gorman and Felix J. Herrmann, 2019, *Compressive least-squares migration with on-the-fly Fourier transforms* , Geophysics, 84, 5.

[3] P. A. Witte, M. Louboutin, H. Modzelewski, C. Jones, J. Selvage and Felix J. Herrmann, 2019, *An event-driven approach to serverless seismic imaging in the cloud,* submitted to IEEE Transactions on Parallel and Distributed Systems, arXiv.

[4] P. A. Witte, M. Louboutin, K. Lensink, M. Lange, N. Kukreja, F. Luporini, G. G. Gorman and Felix J. Herrmann, 2019, *Full-Waveform Inversion, Part 3: Optimization*, The Leading Edge, 37, 2.

[5-6] M. Louboutin, P. A. Witte, M. Lange, N. Kukreja, F. Luporini, G. G. Gorman and Felix J. Herrmann, 2017/2018, *Full-Waveform Inversion, Part 1: Forward Modeling, and Part 2: Adjoint Modeling*, The Leading Edge, 36 (12) and 37 (1)

[7] M. Louboutin, M. Lange, F. Luporini, N. Kukreja, P. A. Witte, F. J. Herrmann, P. Velesko and G. J. Gorman, 2019, *Devito 3.1.0: an embedded domain-specific language for finite differences and geophysical exploration*, Geoscientific model development, 12, 3.

[8] F. Luporini, M. Lange, M. Louboutin, N. Kukreja, J. Hückelheim, C. Yount, P. A. Witte, P. H. J. Kelly, F. J. Herrmann and G. J. Gorman, 2019, *Architecture and performance of Devito, a system for automated stencil computation*, accepted for publication in ACM Transactions on Mathematical Software.

Conferences

[9] P. A. Witte, C. C. Stolk and F. J. Herrmann, 2016, *Phase velocity error minimizing scheme for the anisotropic pure p-wave equation*, Society of Exploration Geophysicists (SEG): Expanded Abstracts.

[10] P. A. Witte, M. Yang and F. J. Herrmann, 2017, *Sparsity-promoting least-squares migration with the linearized inverse scattering imaging condition*, European Association of Geoscientists and Engineers (EAGE): Annual Conference Proceedings.

[11] P. A. Witte, M. Louboutin and F. J. Herrmann, 2017, *Large-scale workflows for wave equation-based inversion in Julia*, SIAM Conference on Computational Science and Engineering.

[12] P. A. Witte, M. Louboutin, H. Modzelewski, C. Jones, J. Selvage and Felix J. Herrmann, 2019, *Event-driven workflows for large-scale seismic imaging in the cloud,* Society of Exploration Geophysicists (SEG): Expanded Abstracts.

[13] P. A. Witte, M. Louboutin, F. Luporini, G. J. Gorman, and F. J. Herrmann, 2019, *Compressive least squares migration with on-the-fly Fourier transforms*, SIAM Conference on Computational Science and Engineering.

[14] F. J. Herrmann, C. Jones, G. Gorman, J. Hückelheim, K. Lensink, P. Kelly, N. Kukreja, H. Modzelewski, M. Lange, M. Louboutin, F. Luporini, J. Selvages and P. A. Witte, 2019, *Accelerating ideation & innovation cheaply in the cloud: The power of abstraction, collaboration & reproducibility*, 4th EAGE Workshop on High Performance Computing..

[15] P. A. Witte, M. Louboutin, C. Jones and F. J. Herrmann, 2020, *Serverless seismic imaging in the cloud*, Rice Oil and Gas High Performance Computing Conference.

SLIM

Acknowledgments

Many thanks to:

- Georgia Tech PhD committee
- UBC PhD committee
- Members of SLIM
- Professor Gerard Gorman and his group at Imperial College London
- Charles Jones and his colleagues at Osokey
- Dr. Sverre Brandsberg-Dahl and his colleagues at Microsoft
- Dr. Kenton Prindle and his colleagues at Google
- Current and former industry sponsors of SLIM
- Academic funding agencies (NSERC, Georgia Research Alliance)
- Open-source software community