Recent Developments in Wave-equation Based Inversion Technology

Felix J. Herrmann

SLIM

Seismic Laboratory for Imaging and Modeling
the University of British Columbia

Frugal 3D full-waveform inversion

Felix J. Herrmann and Tristan van Leeuwen

SLIM
Seismic Laboratory for Imaging and Modeling
the University of British Columbia

Frugal strategy

Instead of *insisting* to work with *all* data for *each* iteration

- work on *small* subsets and *grow* subsets *adaptively* to *control* errors & assure *linear* convergence, and
- draw *independent* subsets after each *model* update to avoid *biases* & to eventually touch *all* data

Instead of *capturing* the wave-physics *exactly*

- dial in the *accuracy* as needed

Spend *compute* & *your handholding* time *only* when necessary...

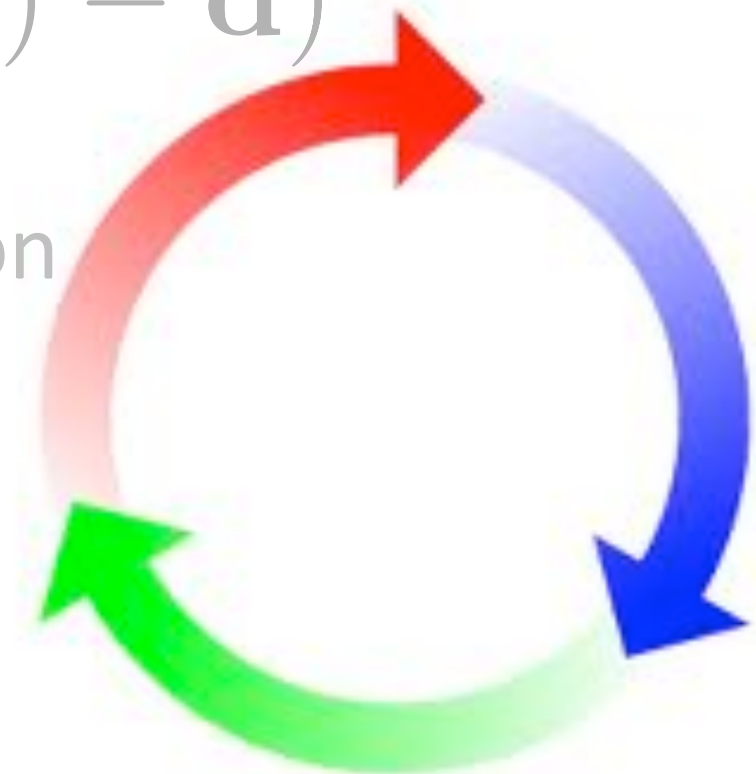
Frugal FWI

$$\min_{\mathbf{m}} \rho(F(\mathbf{m}) - \mathbf{d})$$

robust
formulation

$$A(\mathbf{m})\mathbf{u} = \mathbf{q}$$

versatile
modelling



$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{s}_k$$

***fast* optimization strategies**

computational framework

Fast optimization

Aims:

- to *reduce* work by drawing random *subsets*
- to allow for *inaccurate* PDE solves
- to *control* errors by increasing *subsets* & *accuracy* PDE solves to guarantee *linear* convergence

Outcome:

- scheme that *limits # passes* through *data* & # *iterations* of wave-equation *solver*

[Friedlander & Schmidt ’12, Aravkin et.al. ’12]

SLIM

Fast optimization

with *error* control

Approximate gradients by *sample averages*–i.e.,

$$\nabla\Phi \approx \nabla\widetilde{\Phi} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\nabla\phi_i$$

and *control errors* by *growing* the *sample size* B_k of the *subsets*.

Linear convergence if *error* at *iteration k* bounded by

$$||\mathbf{e}_k|| \leq B_k = \mathcal{O}(\gamma^k)$$

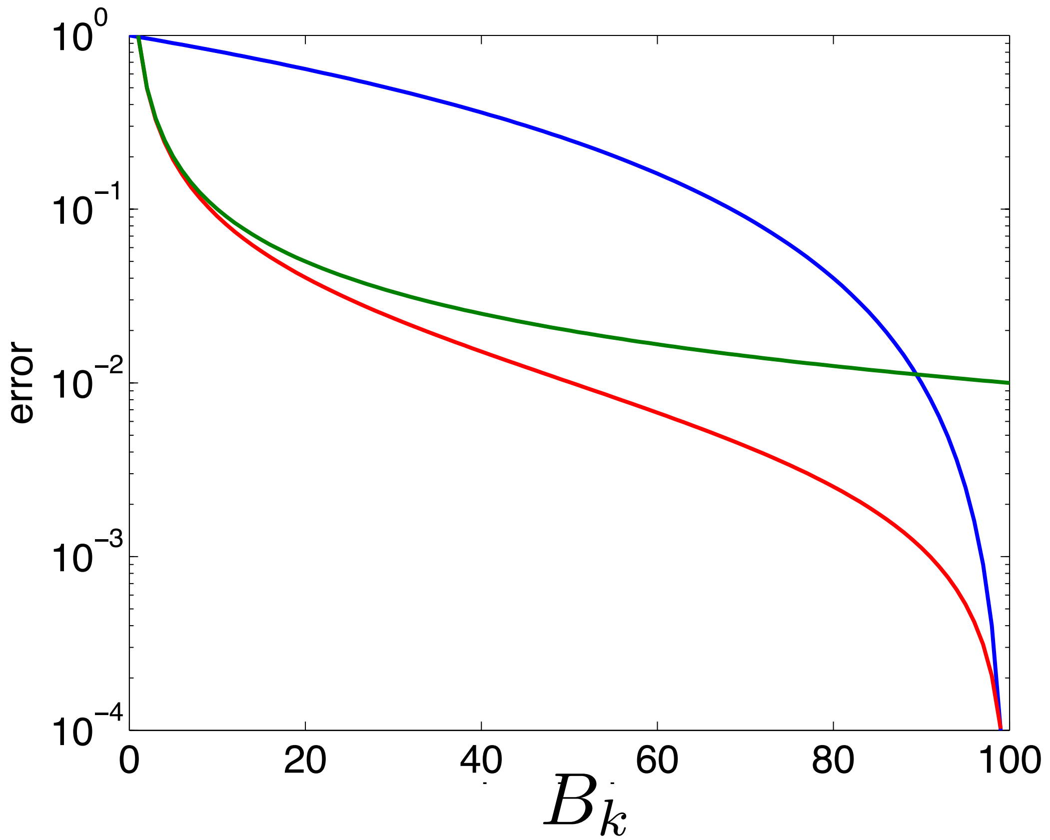
with γ the *convergence* rate.

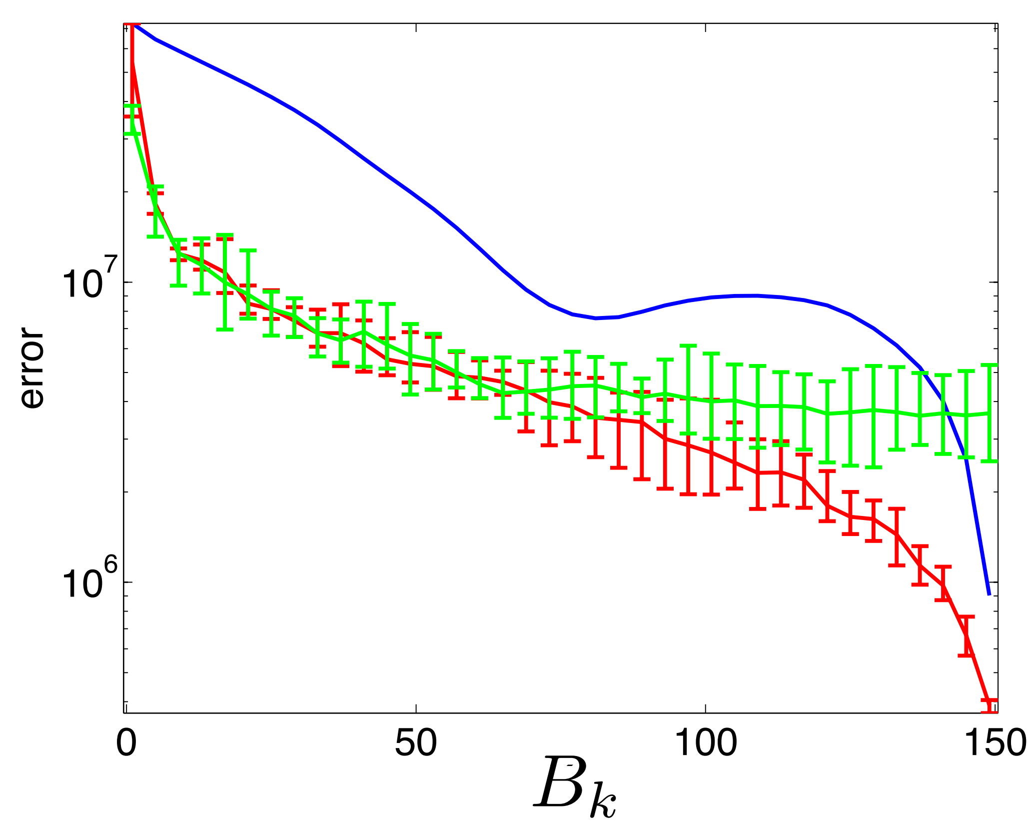
Fast optimization

increase sample size

Select sources

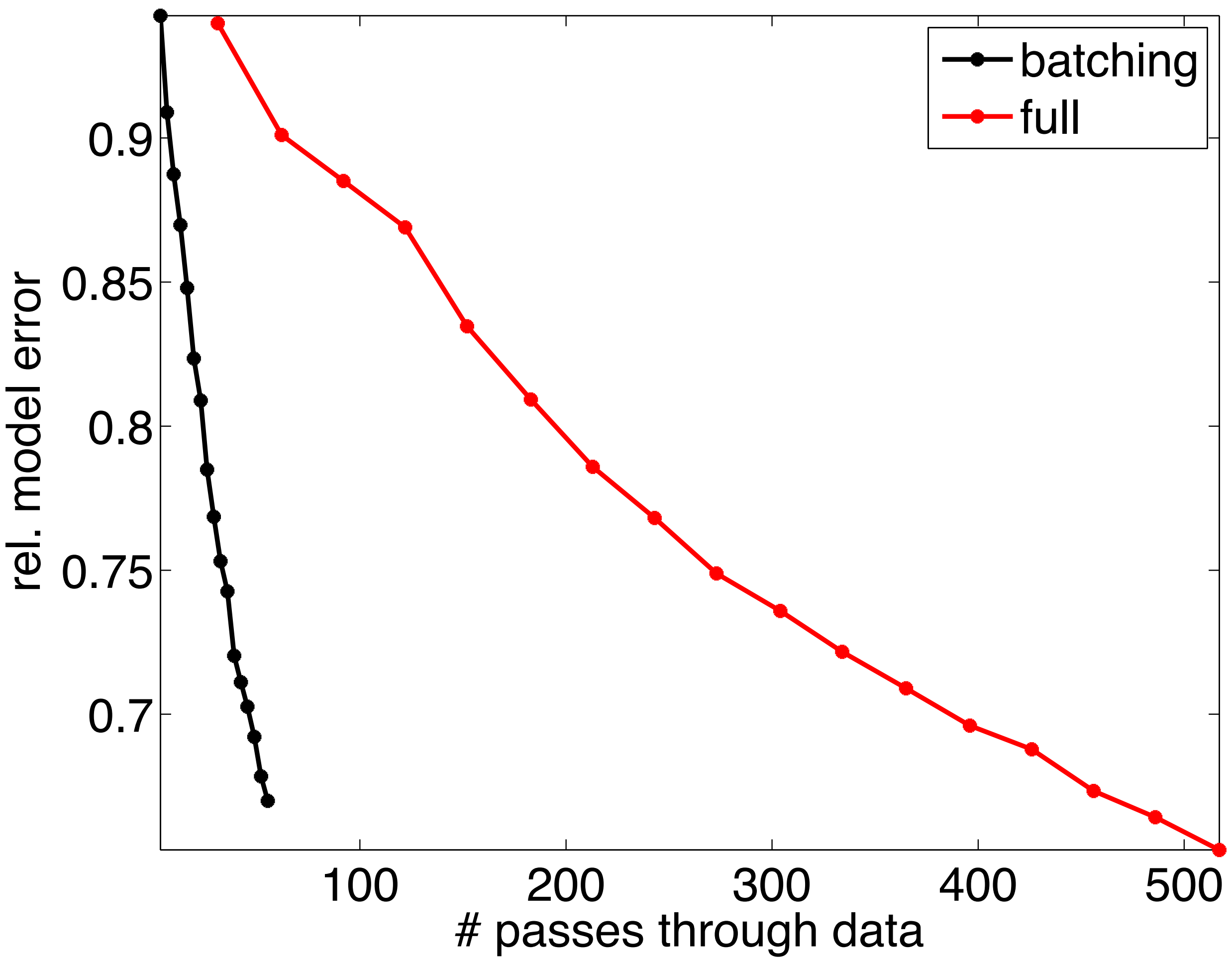
- in a pre-scribed order
- random *without* replacement
- random-*amplitude* source encoding

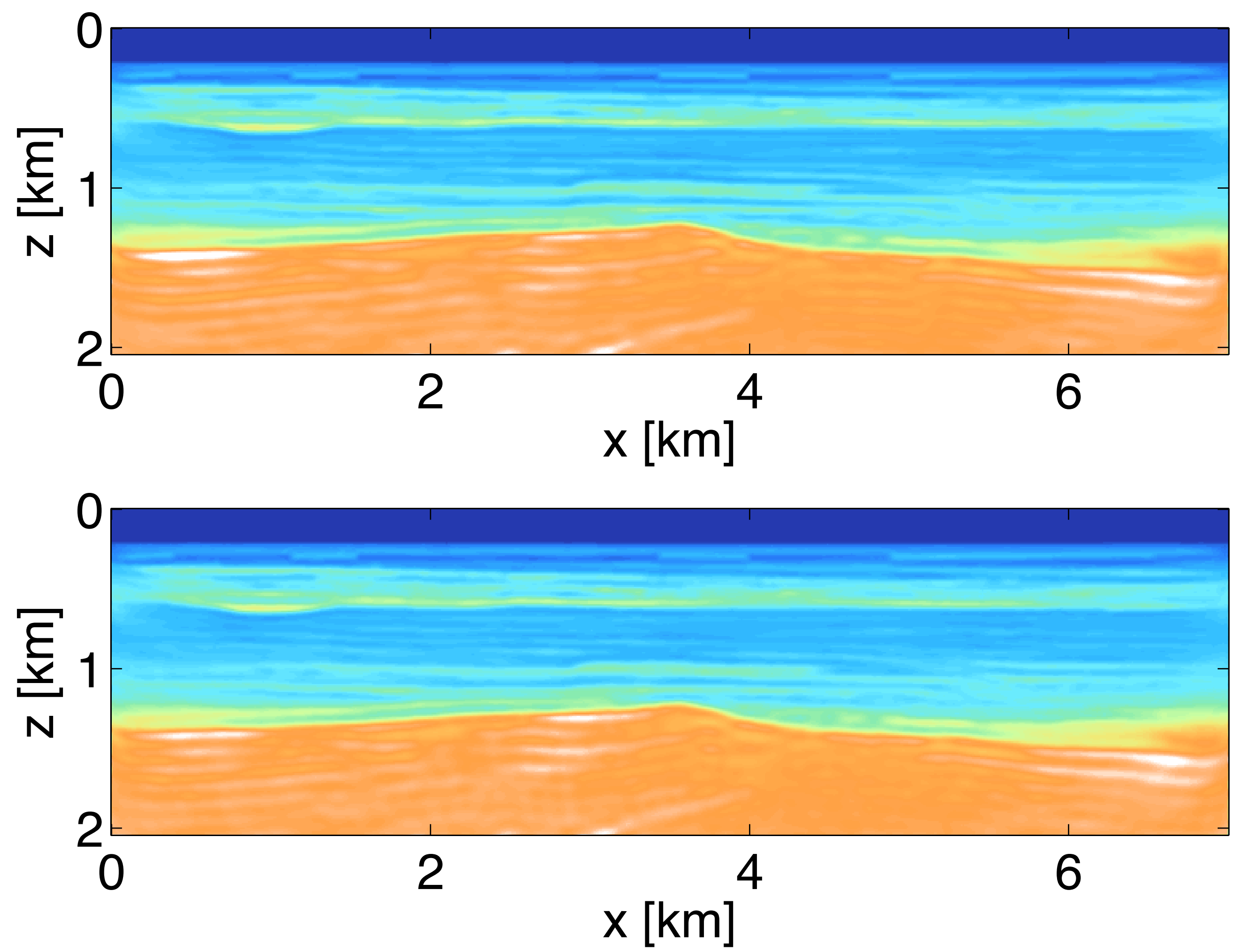




Fast optimization

10 x speedup





[van Leeuwen & FJH ’11]

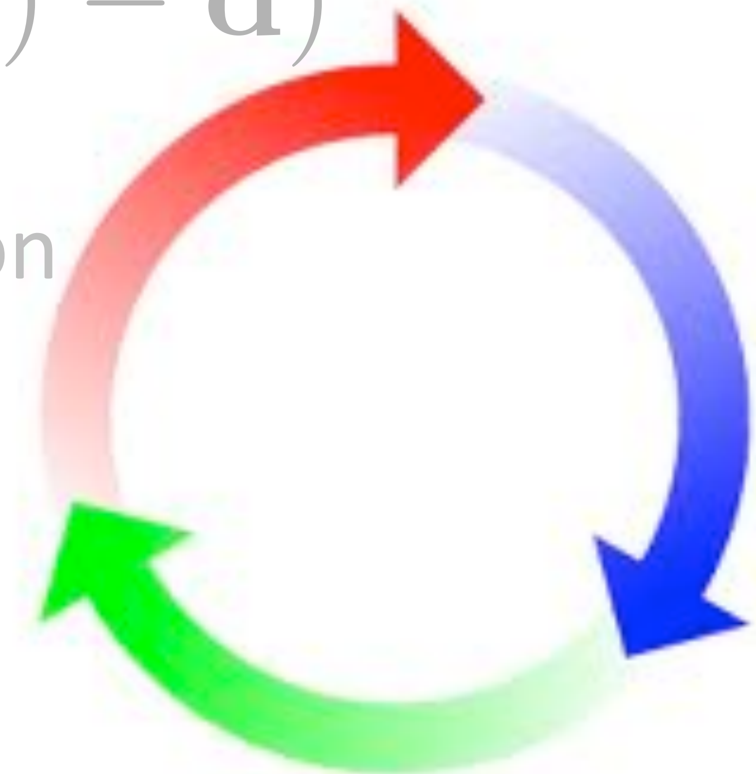
Frugal FWI

$$\min_{\mathbf{m}} \rho(F(\mathbf{m}) - \mathbf{d})$$

robust formulation

$$A(\mathbf{m})\mathbf{u} = \mathbf{q}$$

versatile modelling



$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{s}_k$$

fast optimization strategies

computational framework

Versatile modelling

Aims:

To solve *forward* problems in the *context* of *FWI* by avoiding

- large (e.g., factorization) *setup* & *memory* costs
- methods that depend on *tuning* parameters and that can not *control* their *accuracy*
- *inflexible* preconditioners that *depend* on *specifics* of *physics*

Outcome:

- *scalable* wave simulations w/ prescribed *tolerance*

SLIM

CGMN & CARP-BCG

Use *simple* Kaczmarz *row* projections

$$\mathbf{x} := \mathbf{x} + \frac{\lambda}{||\mathbf{a}_i||_2^2} \left(b_i - \mathbf{a}_i^T \mathbf{x} \right) \mathbf{a}_i,$$

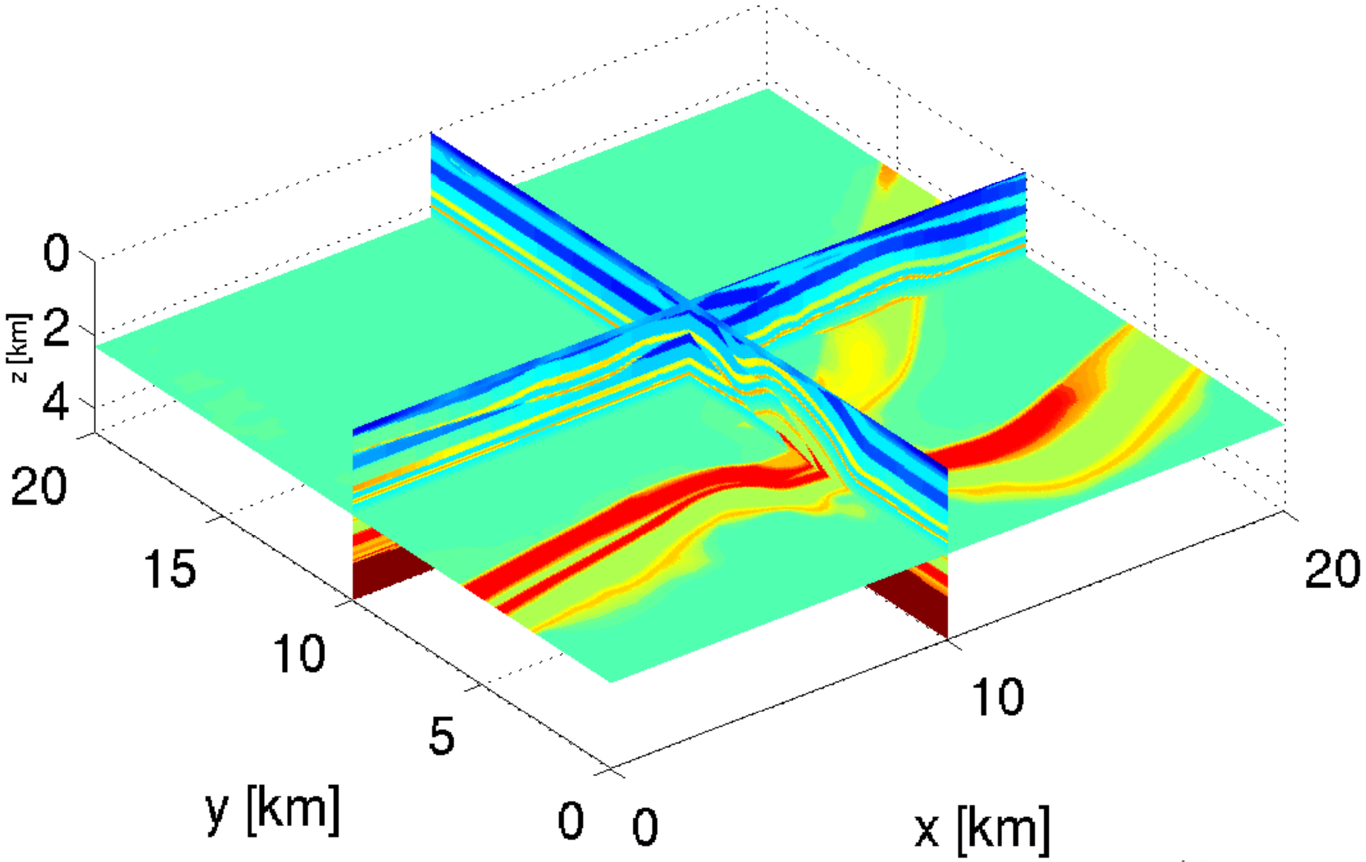
to form a *preconditioner* with *double* sweeps that

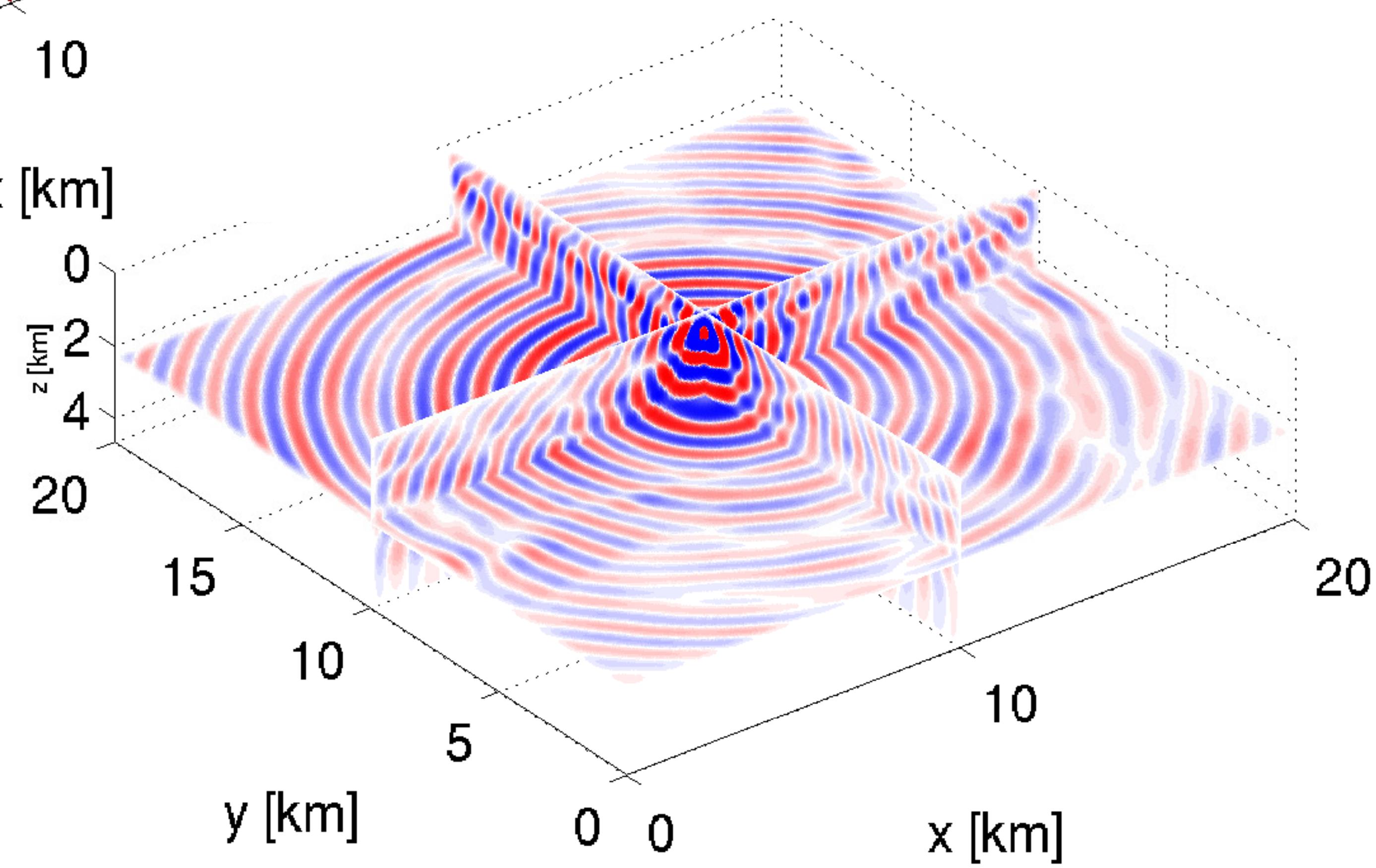
- deals with *multiple* right-hand-sides *simultaneously*
- is *parallelizible* by *projecting* row blocks *independently*
- can be *accelerated* by CG

Simple *scalable* algorithm with *controllable* accuracy...

Overtrust model

4.5 Hz in 1633s



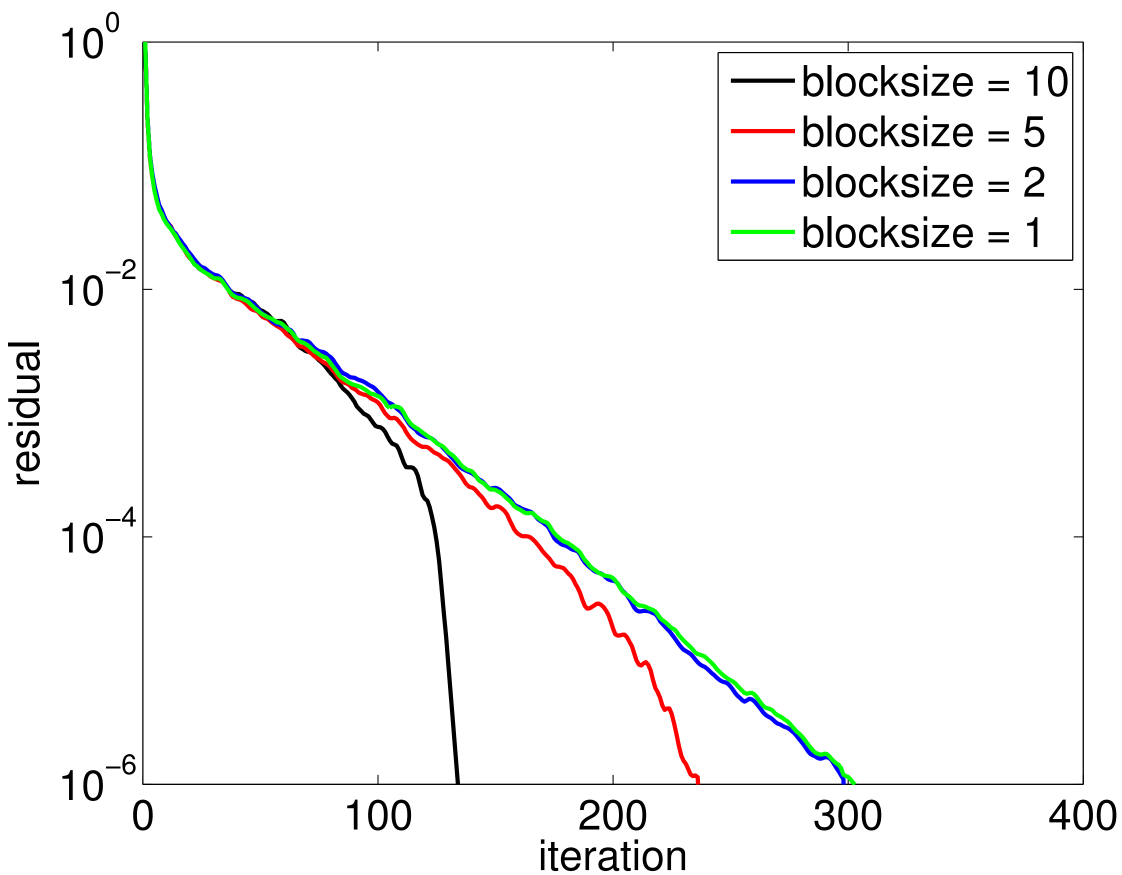


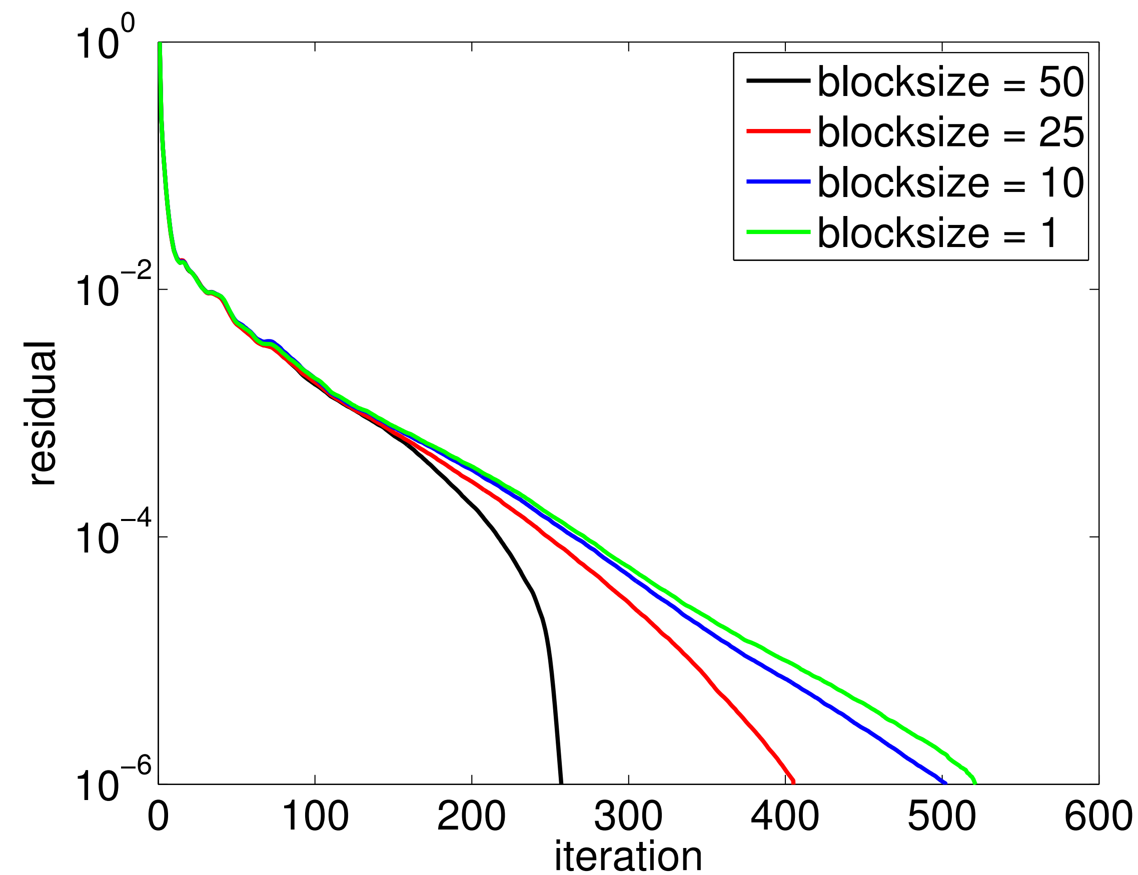
27 point stencil
10 pts per wavelength
PML
5km X 5km X 2.5Km

Block CG

0.5,1 Hz
sources selected *randomly*

multiple right-hand-sides

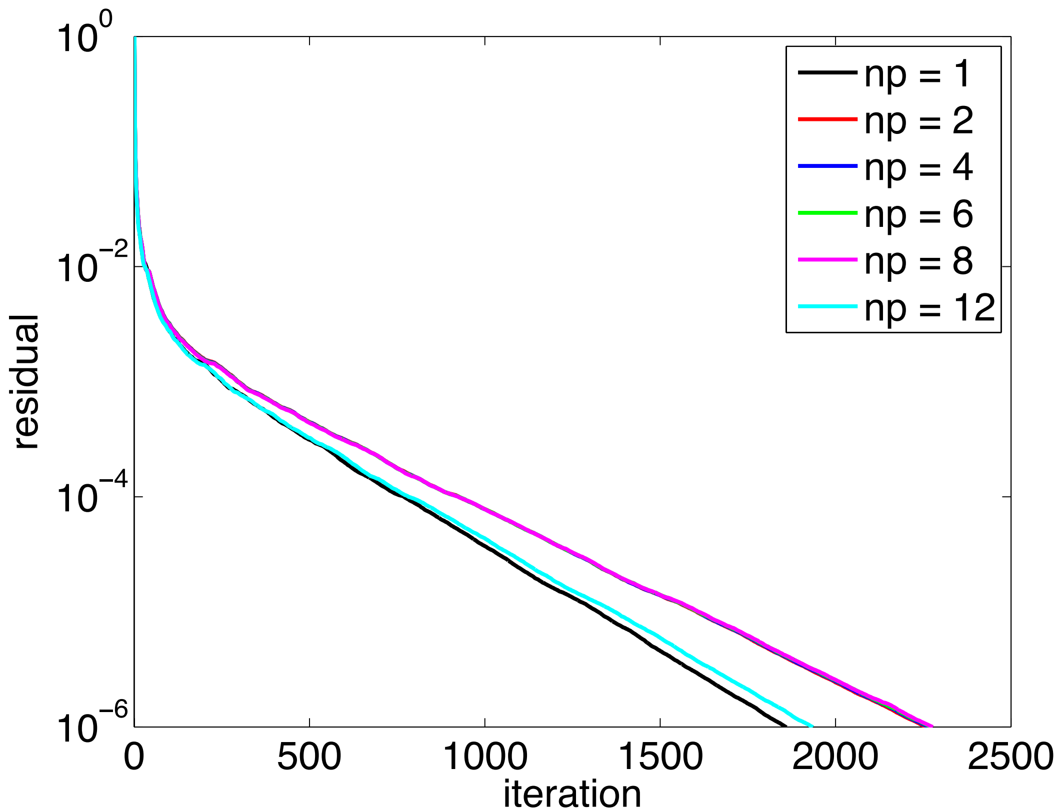




CARP-CG

parallel over *blocks* of *rows*
averaging guarantees *convergence*

multiple cores



np	iter	time [s]	efficiency
1.0	1859	13834.8	1.0
2.0	2256	9141.7	0.9
4.0	2265	4649.0	0.9
6.0	2271	3326.6	0.8
8.0	2276	2598.9	0.8
12.0	1934	1633.0	0.7

Gordon & Gordon ’11-’12

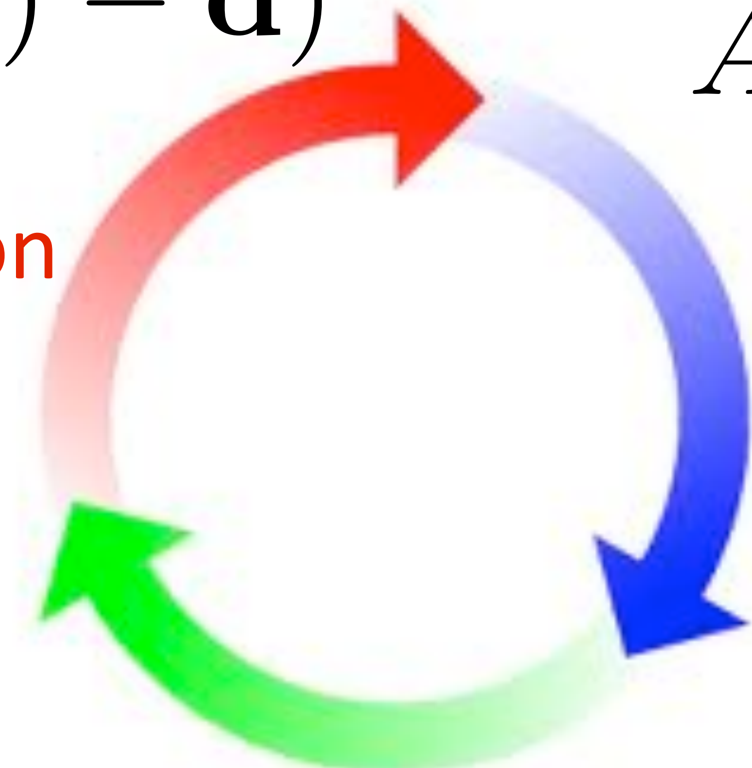
Frugal FWI

$$\min_{\mathbf{m}} \rho(F(\mathbf{m}) - \mathbf{d})$$

robust formulation

$$A(\mathbf{m})\mathbf{u} = \mathbf{q}$$

versatile modelling



$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \mathbf{s}_k$$

fast optimization strategies

computational framework

Frugal FWI

We have

- *randomized* sampling with controllable *error*
- *fast* semi-*stochastic* optimization with *linear convergence*
- *versatile* forward modelling with controllable *tolerance*

Put it *all* together

- *behavior* of the *objective* as a *function* of the *tolerance*
- approximate *residual* by solving PDE up to a *tolerance*

Frugal misfit
w/ approximate PDE solves

Heuristic based on *behavior* of the *misfit* as function of ϵ

$$\phi_i(\mathbf{m},\epsilon) = \rho\big(P_i\mathbf{u}_i(\epsilon) - \mathbf{d}_i\big),$$

by solving PDEs to *tolerance* ϵ.

Ideally find ϵ by guaranteeing

$$|\phi_i(\mathbf{m},\epsilon) - \phi_i(\mathbf{m},0)| \leq \eta\phi_i(\mathbf{m},0)$$

for some fraction η. Instead find *k* such that

$$|\phi_i(\mathbf{m},\alpha^k\epsilon) - \phi_i(\mathbf{m},\alpha^{k+1}\epsilon)| \leq \eta\phi_i(\mathbf{m},\alpha^{k+1}\epsilon) \quad 0<\alpha<1.$$

Frugal misfit

Algorithm 1 $\{f, \mathbf{g}\} = \mathsf{misfit}(\mathbf{m}, \mathcal{I}, \eta)$

$\epsilon = 10^{-2}$, $\alpha = 0.5$// Initialization
for $i \in \mathcal{I}$ **do**
  for $k = 0 \to 10$ **do**
    solve $A(\mathbf{m})\mathbf{u} = \mathbf{s}_i$ up to ϵ // solve forward equation
    $r_k = \rho(P_i\mathbf{u} - \mathbf{d}_i)$ // compute residual
    if $|r_k - r_{k-1}| \leq \eta r_k$ **then**
      break
    else
      $\epsilon = \alpha\epsilon$
    end if
  end for
  solve $A(\mathbf{m})^*\mathbf{v} = {P_i}^*\nabla\rho(P_i\mathbf{u} - \mathbf{d}_i)$ up to ϵ
  $f = f + |\mathcal{I}|^{-1}\rho(P_i\mathbf{u} - \mathbf{d}_i)$ // misfit
  $\mathbf{g} = \mathbf{g} + |\mathcal{I}|^{-1}G(\mathbf{m}, \mathbf{u})^*\mathbf{v}$ // gradient
end for

Stochastic

Quasi-Newton

Final algorithm has the following key ingredients:

- draw *independent* random *subsets* for *each* gradient
- decrease *fraction* η when *linesearch* fails
- increase *sample* size when *average* objective does *not* decrease–i.e, if $(f_{k+1} + f'_{k+1}) \geq (f_k + f'_k)$
- low-rank approximation of *inverse* Quasi-Newton Hessian w/ lBFGS

Stochastic

Quasi-Newton

Algorithm 1 Stochastic L-BFGS method

$\eta = 0.1,\ b = 1,\ \beta = 1,\ b_{\max} = M$ // Initialize

choose $\mathcal{I}_0 \subseteq \{1, 2, \ldots, M\}$ s.t. $|\mathcal{I}_0| = b$

$\{f_0, \mathbf{g}_0\} = \mathsf{misfit}(\mathbf{m}_0, \mathcal{I}_0, \eta)$ // *frugal* misfit & gradient at initial guess

while not converged **do**

  $\delta\mathbf{m}_k = \mathsf{lbfgs}(-\mathbf{g}_k, \{\mathbf{t}_l\}_{l=k-m}^{k}, \{\mathbf{y}_l\}_{l=k-m}^{k})$ // low-rank inverse Hessian

  $\{\mathbf{m}_{k+1}, f_{k+1}, \mathbf{g}_{k+1}\} = \mathsf{linesearch}(f_k, \mathbf{g}_k, \delta\mathbf{m}_k)$

  if linesearch successfull **then**

    $\mathbf{t}_{k+1} = \mathbf{m}_{k+1} - \mathbf{m}_k,\ \mathbf{y}_{k+1} = \mathbf{g}_{k+1} - \mathbf{g}_k$ // update L-BFGS vectors

    choose $\mathcal{I}_{k+1} \subseteq \{1, 2, \ldots, M\}$ s.t. $|\mathcal{I}_{k+1}| = b$ // draw new sample

    $\{f'_{k+1}, \mathbf{g}'_{k+1}\} = \mathsf{misfit}(\mathbf{m}_{k+1}, \mathcal{I}_{k+1}, \eta)$ // misfit & gradient new sample

    if $(f_{k+1} + f'_{k+1}) \geq (f_k + f'_k)$ **then**

      $b = \min(b + \beta, b_{\max})$ // increase batch

    end if

    $f_{k+1} = f'_{k+1},\ \mathbf{g}_{k+1} = \mathbf{g}'_{k+1},\ k = k + 1$ // Use new misfit & gradient

  else

    $\eta = \eta/2$ // narrow tolerenance

  end if

end while

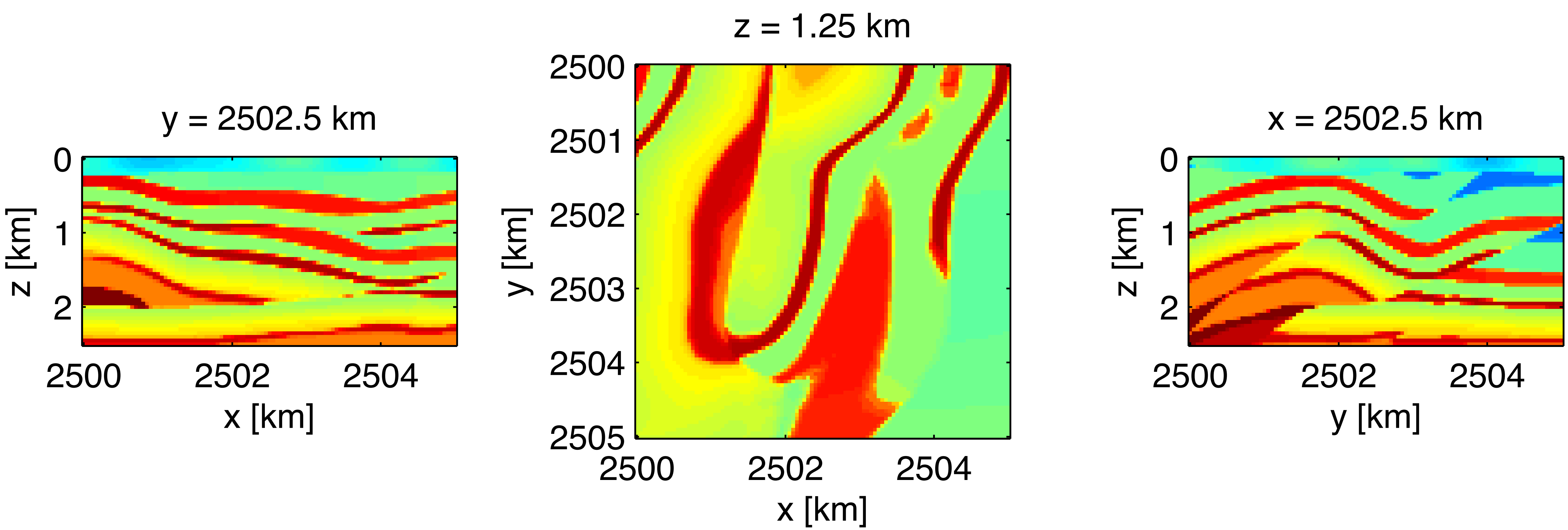
SLIM

Overthrust model

true model

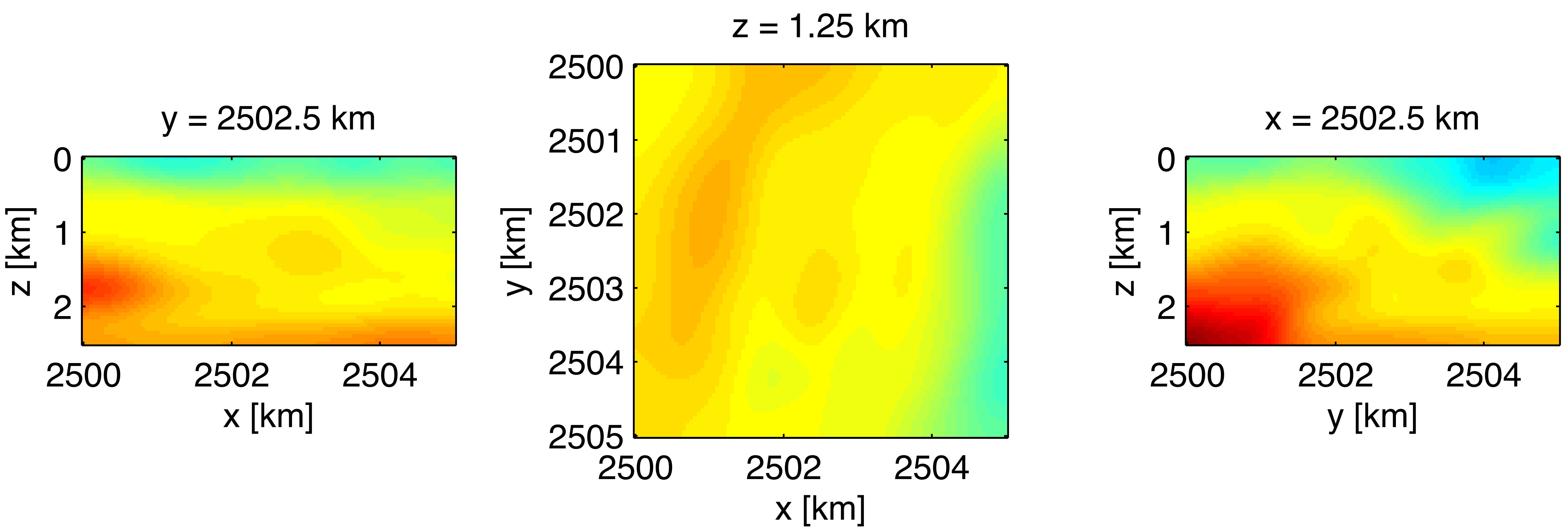
5km X 5km X 2.5Km

121 sources & 2601 receivers



Overthrust model

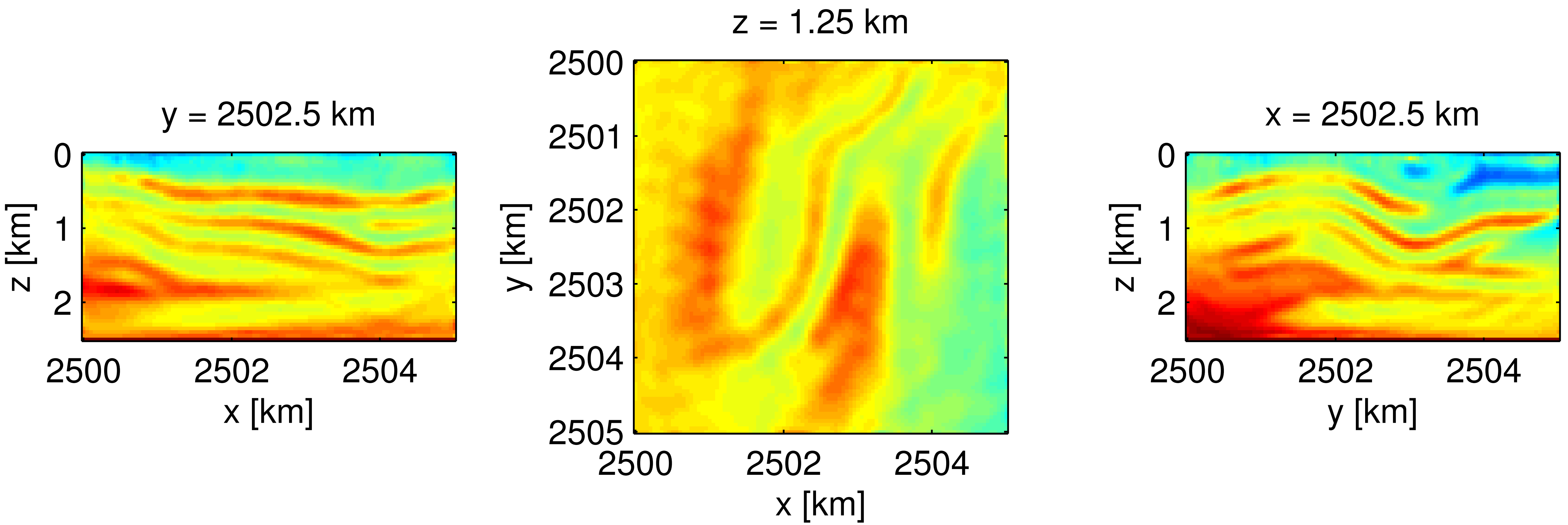
initial model



Overthrust model

recovered model *w/ b=1*

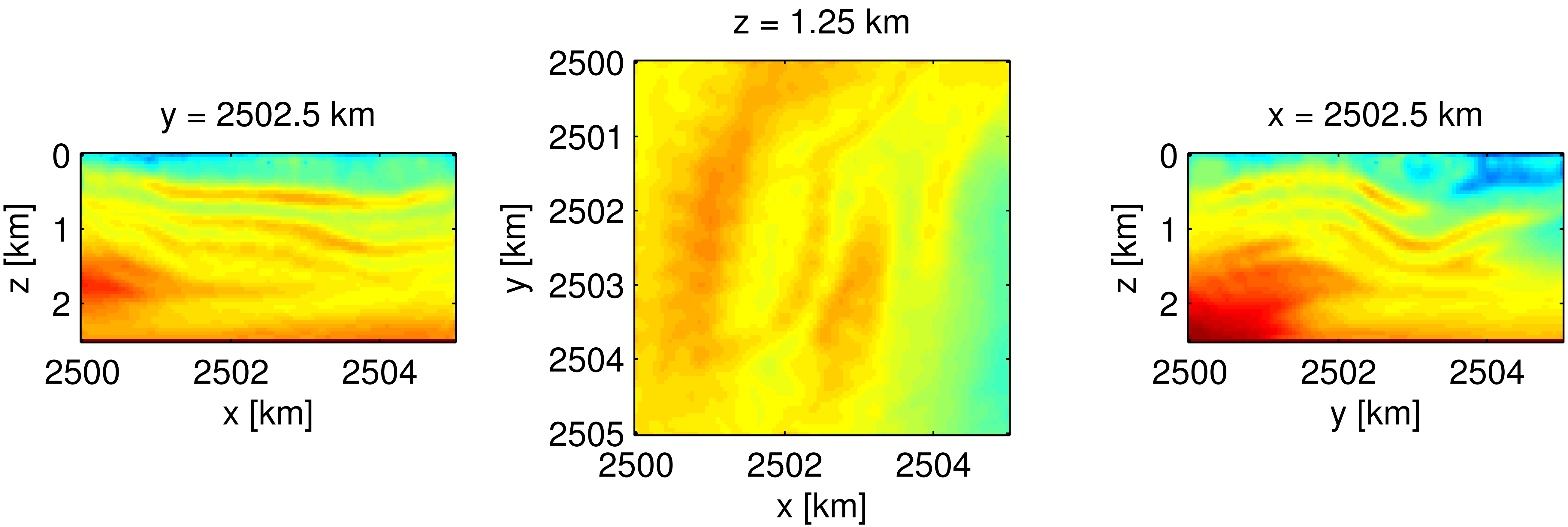
2 passes through *data* for each *(4,6,8) Hz*



Overthrust model

recovered model *w/ b=121*

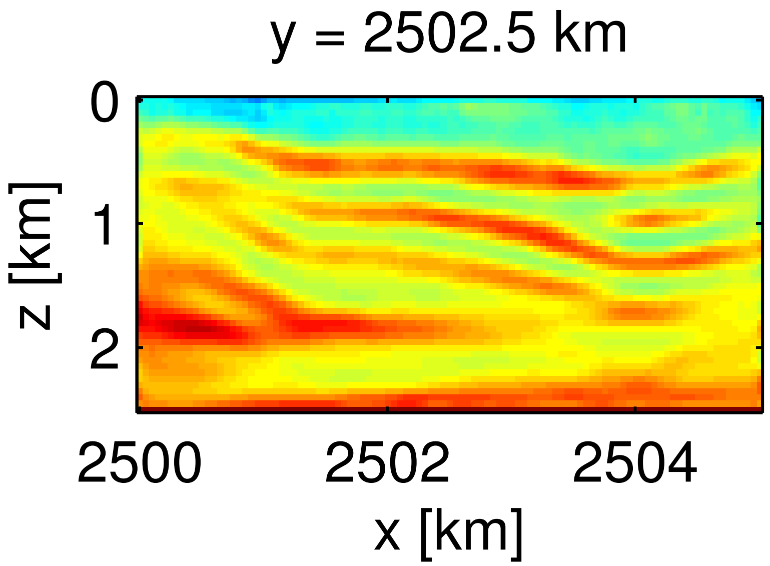
2 passes through *data* for each *(4,6,8) Hz*

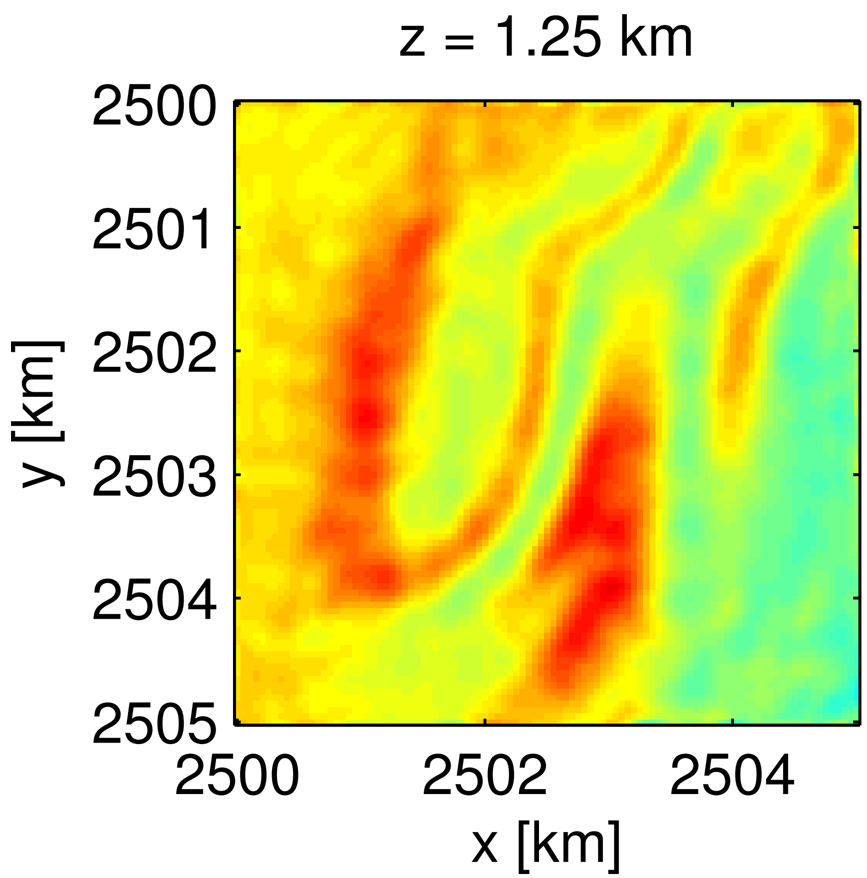


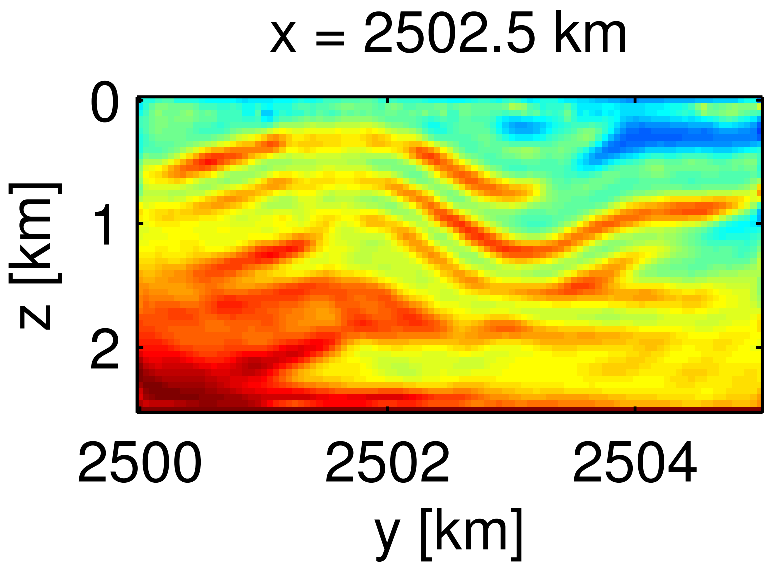
Overthrust model

growing sample size

2 passes through *data* for each *(4,6,8) Hz*

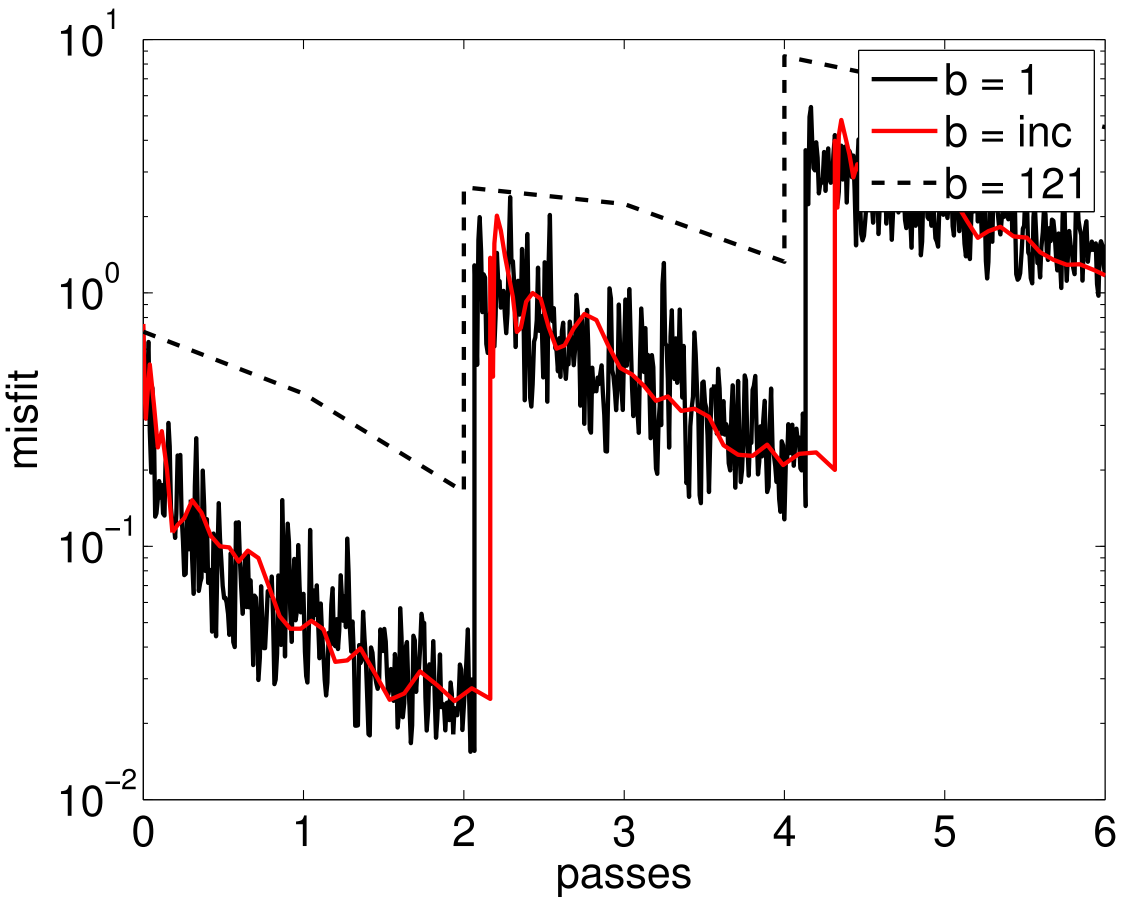


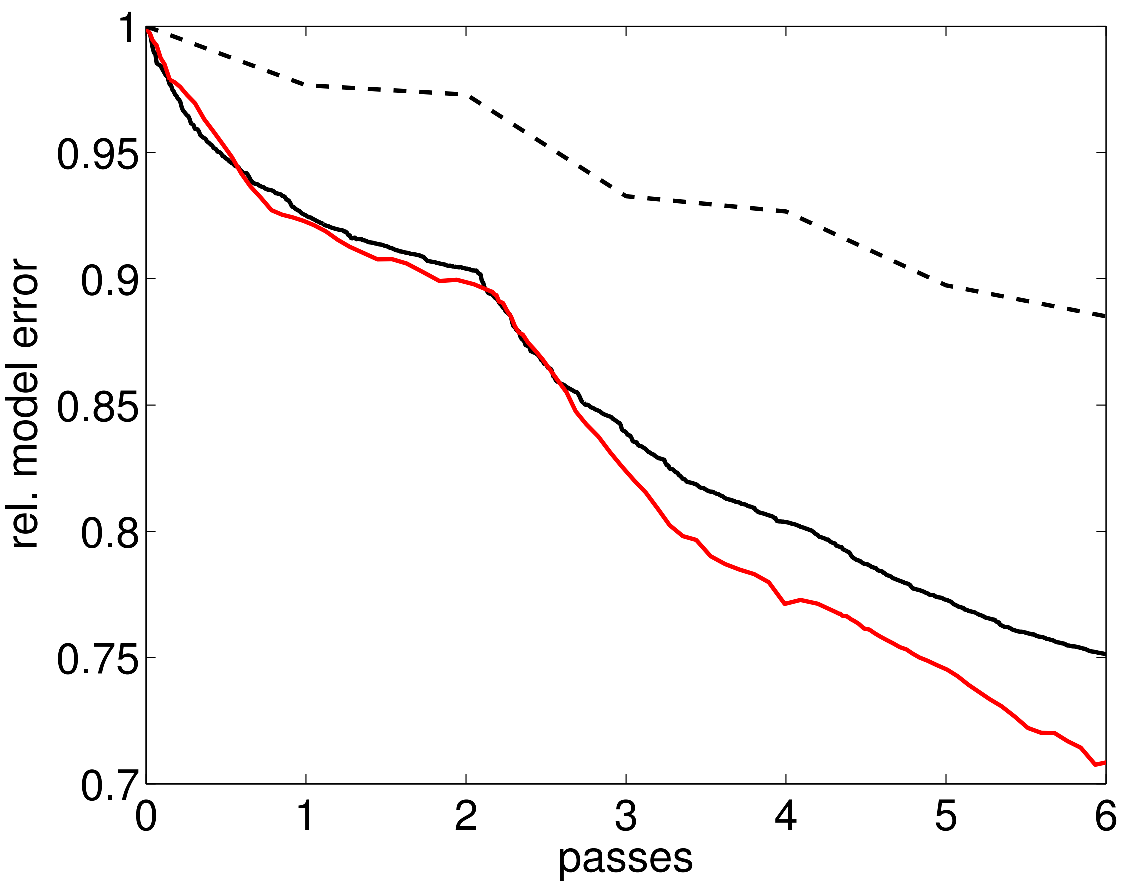




Performance

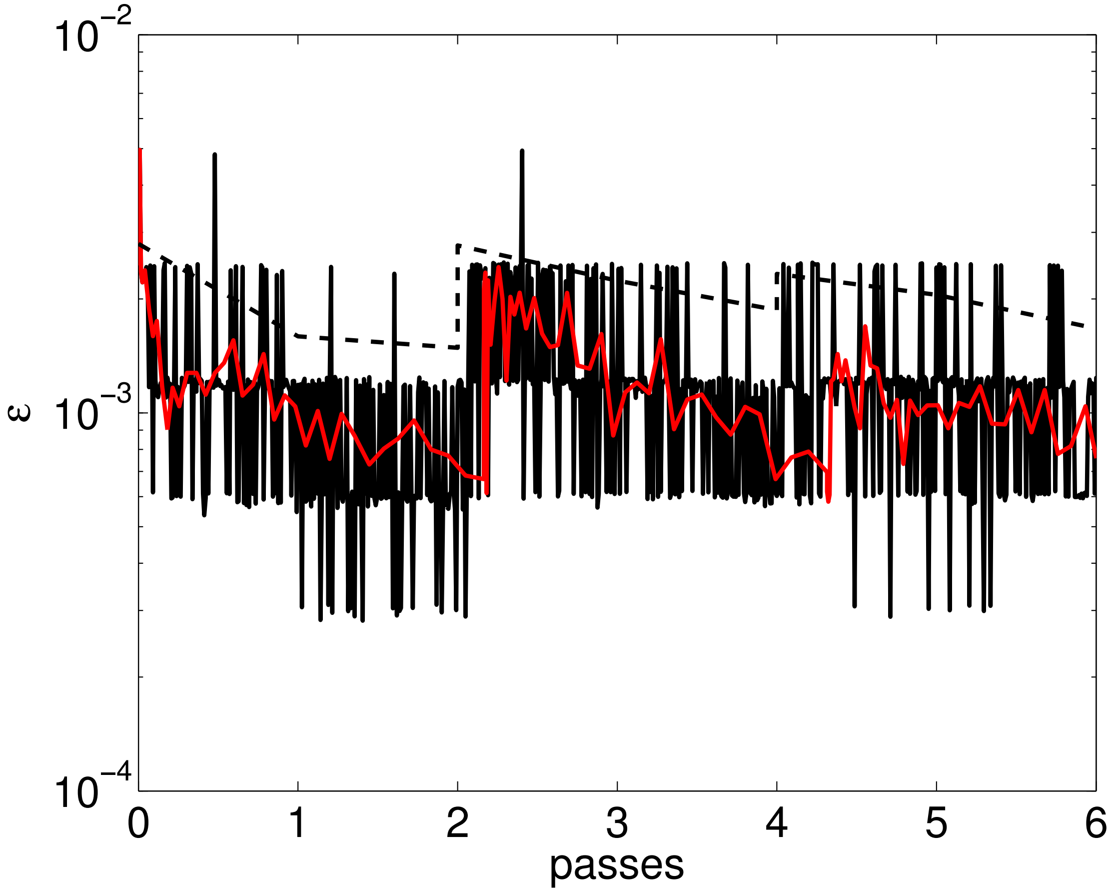
misfit & relative model error

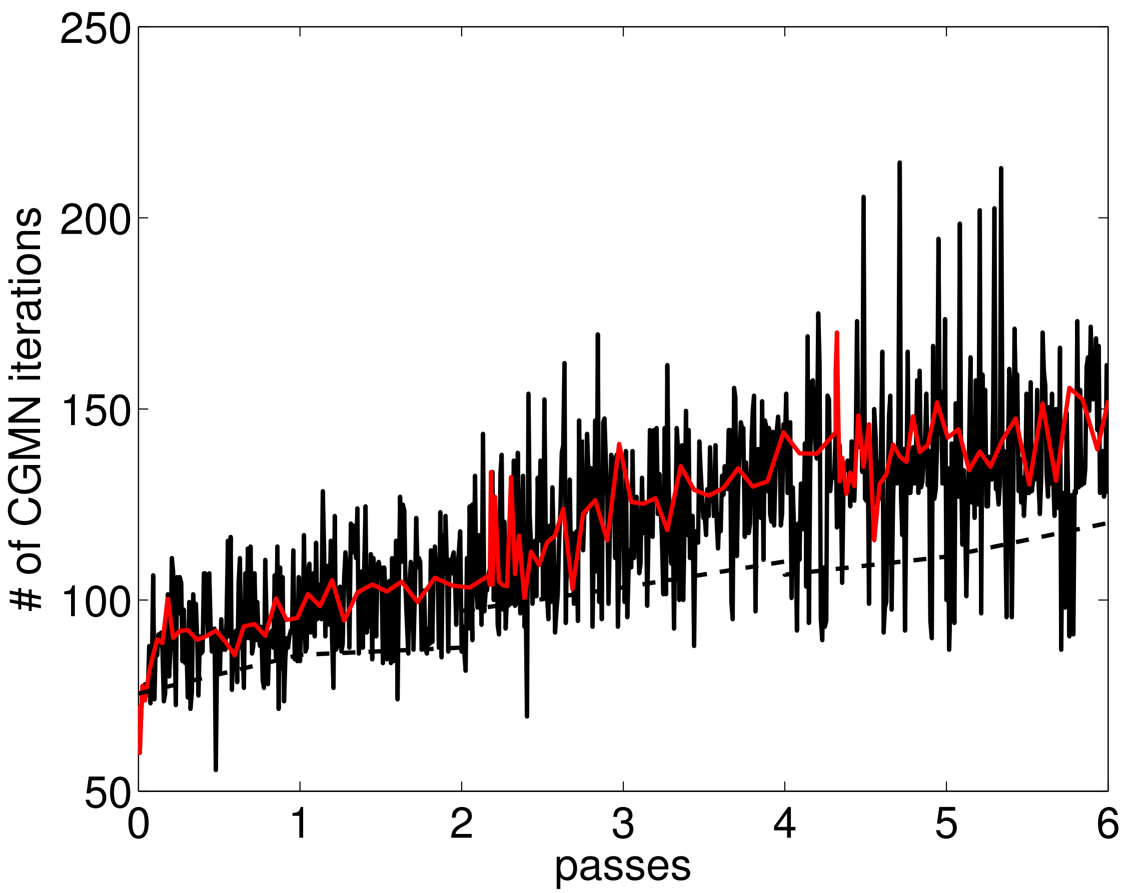




Performance

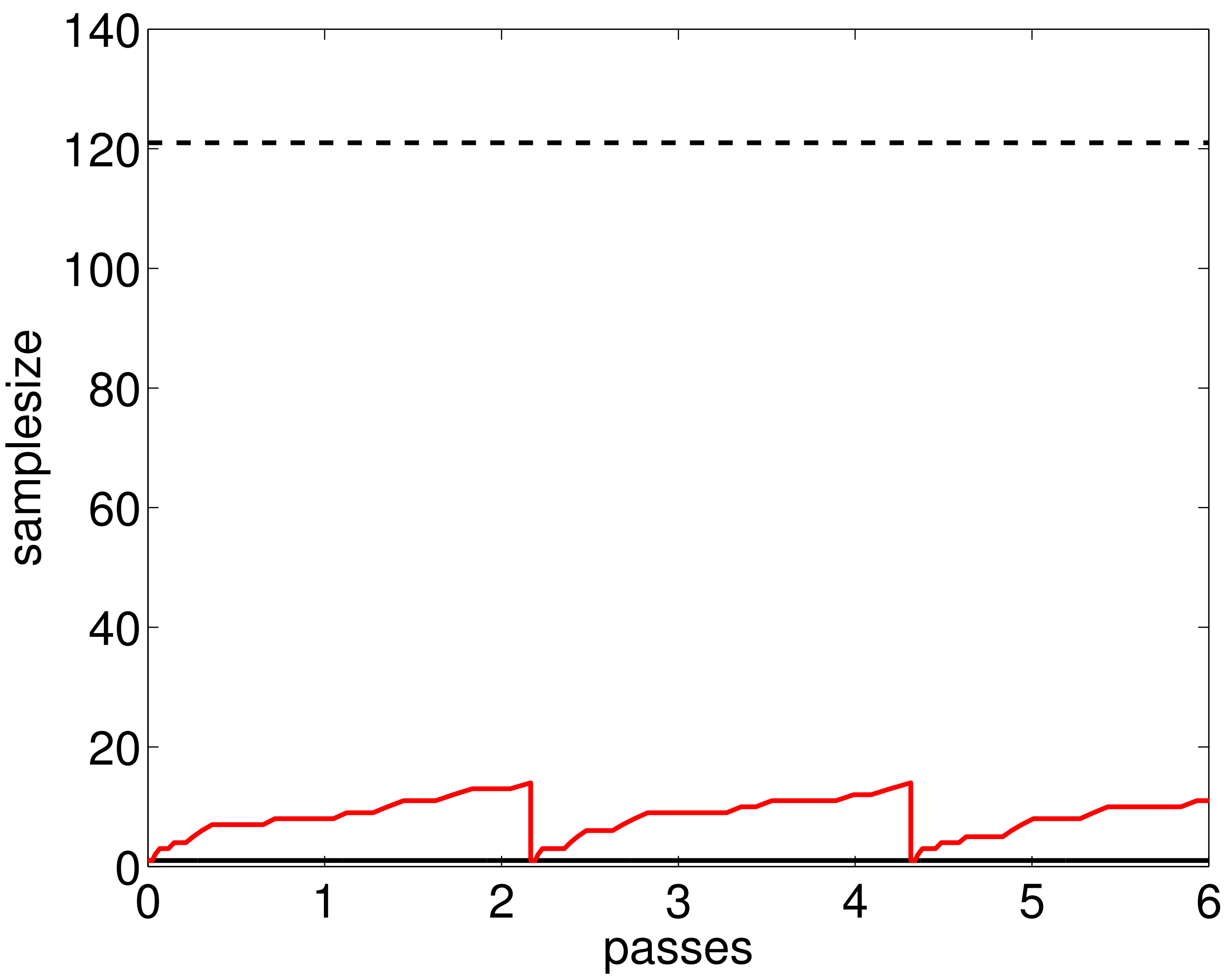
tolerance & # CARP-CG iterations





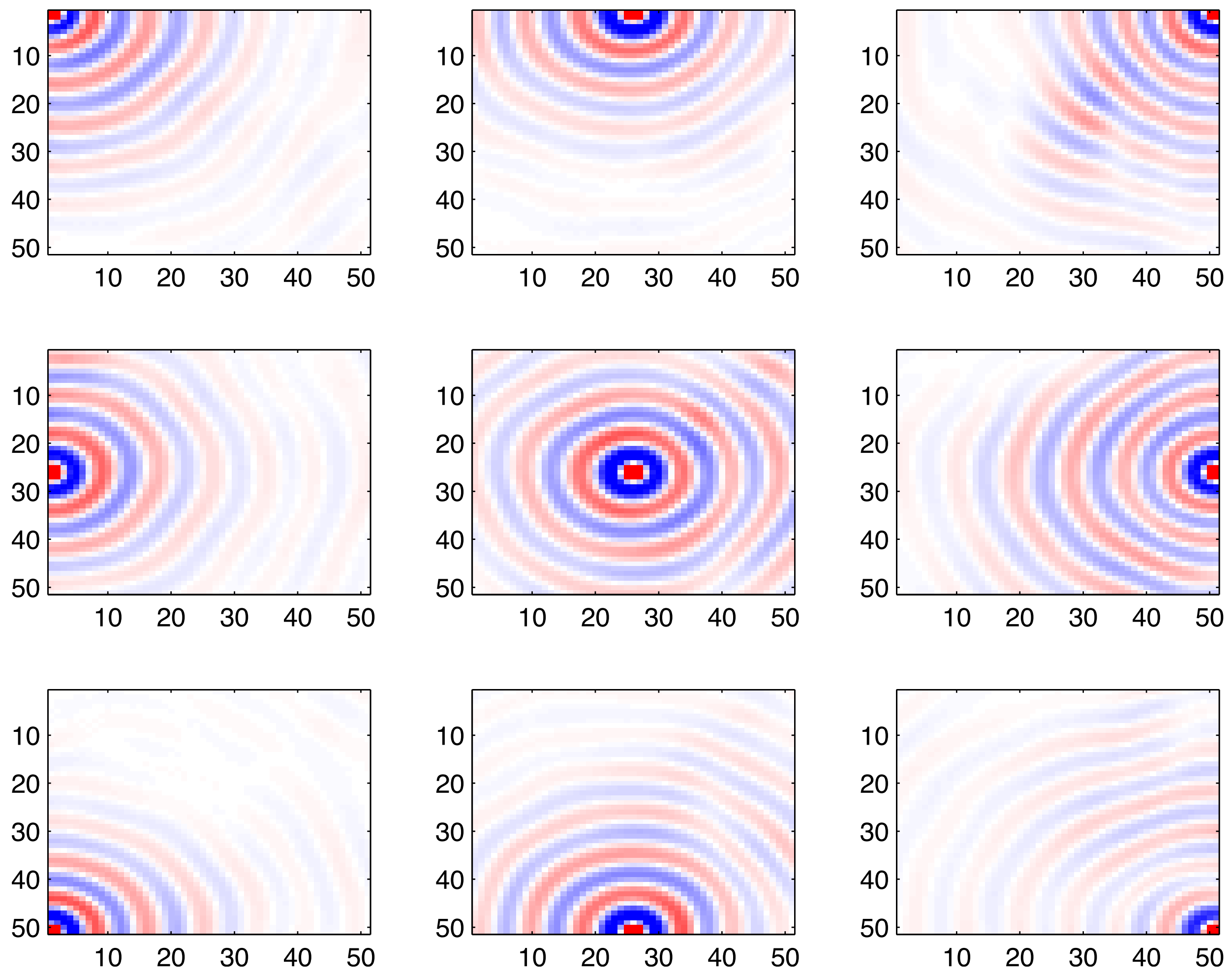
Performance

sample size



Performance

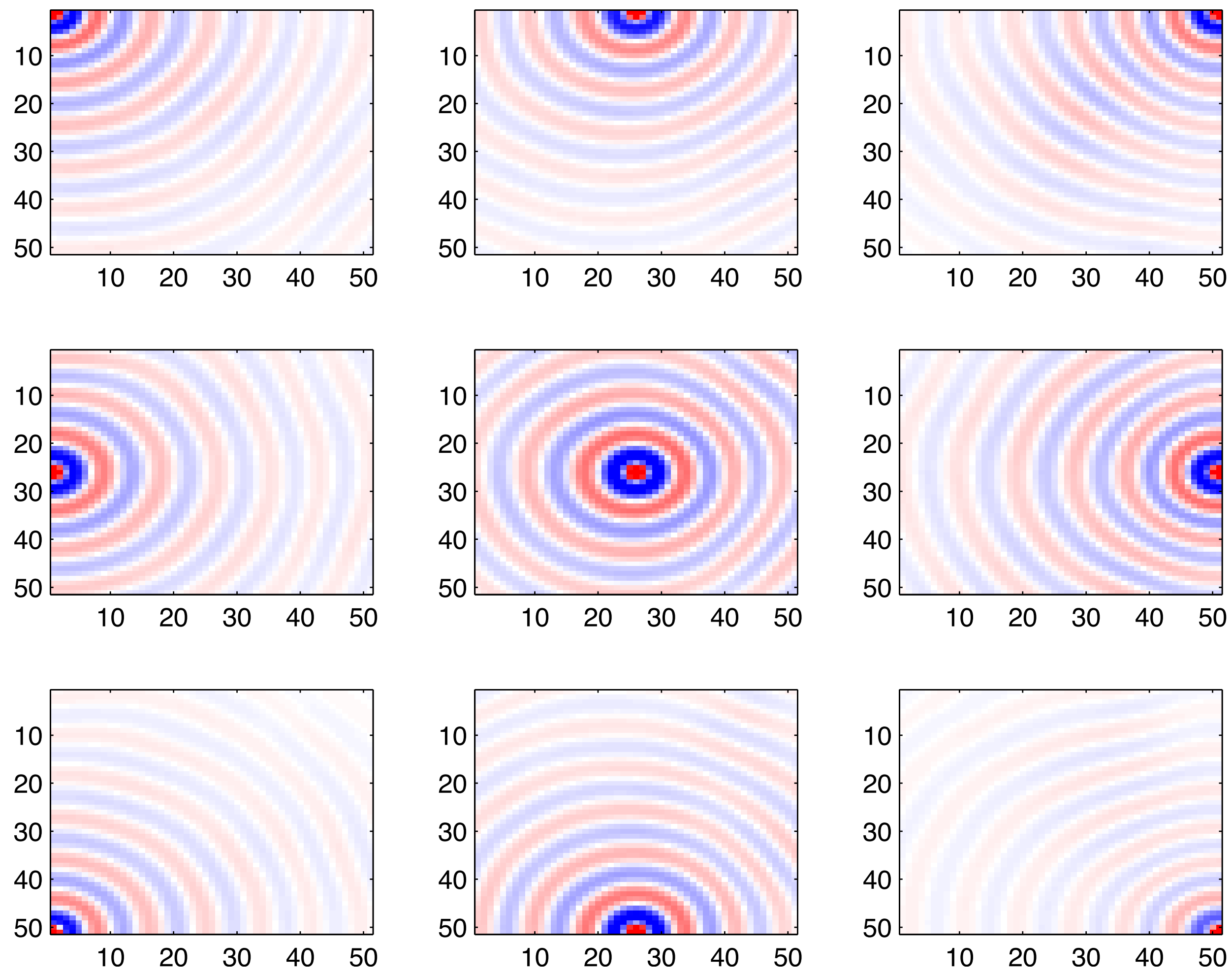
input data @ 4Hz



4 hours

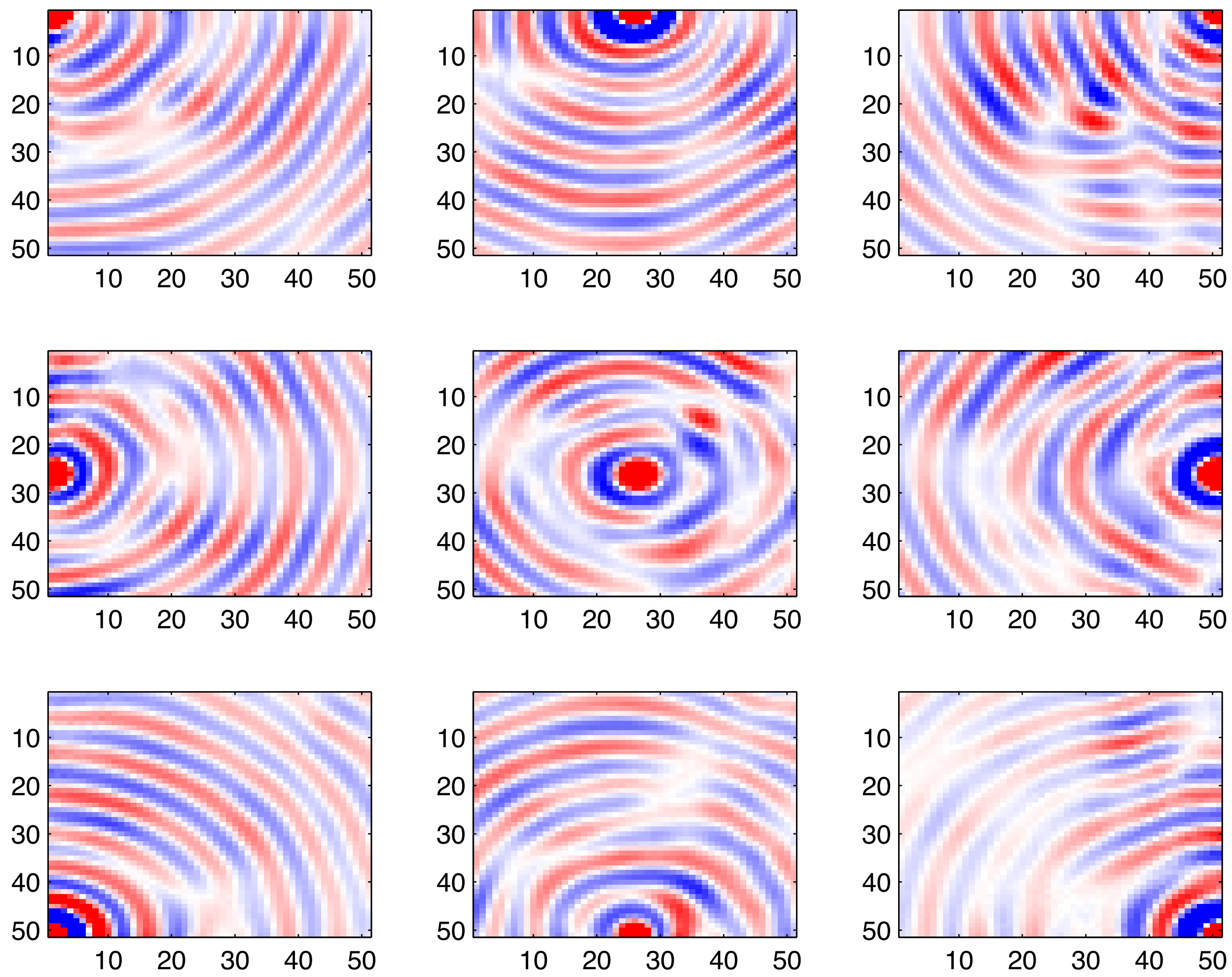
Performance

initial data @ 4Hz



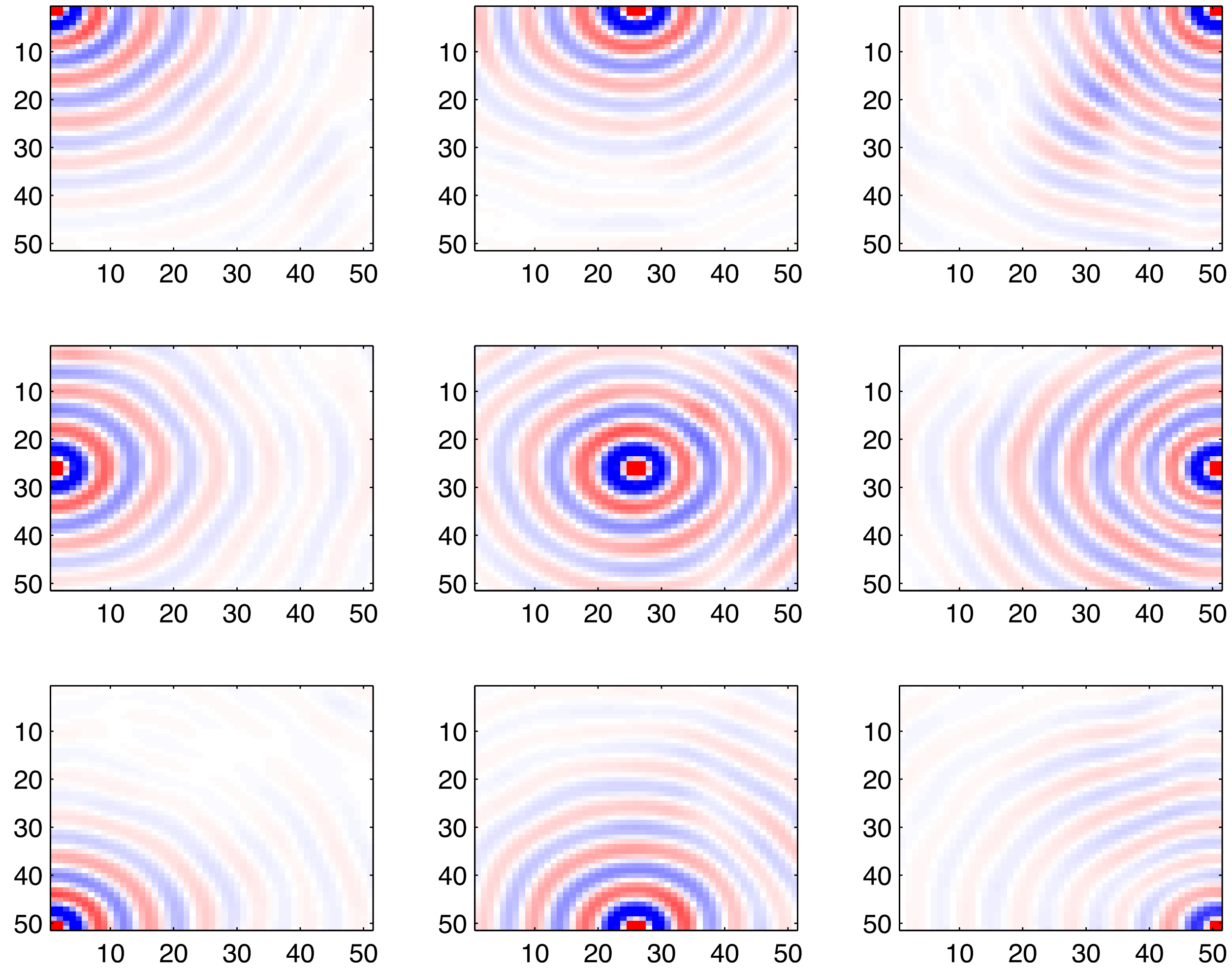
Performance

initial residual @ 4Hz



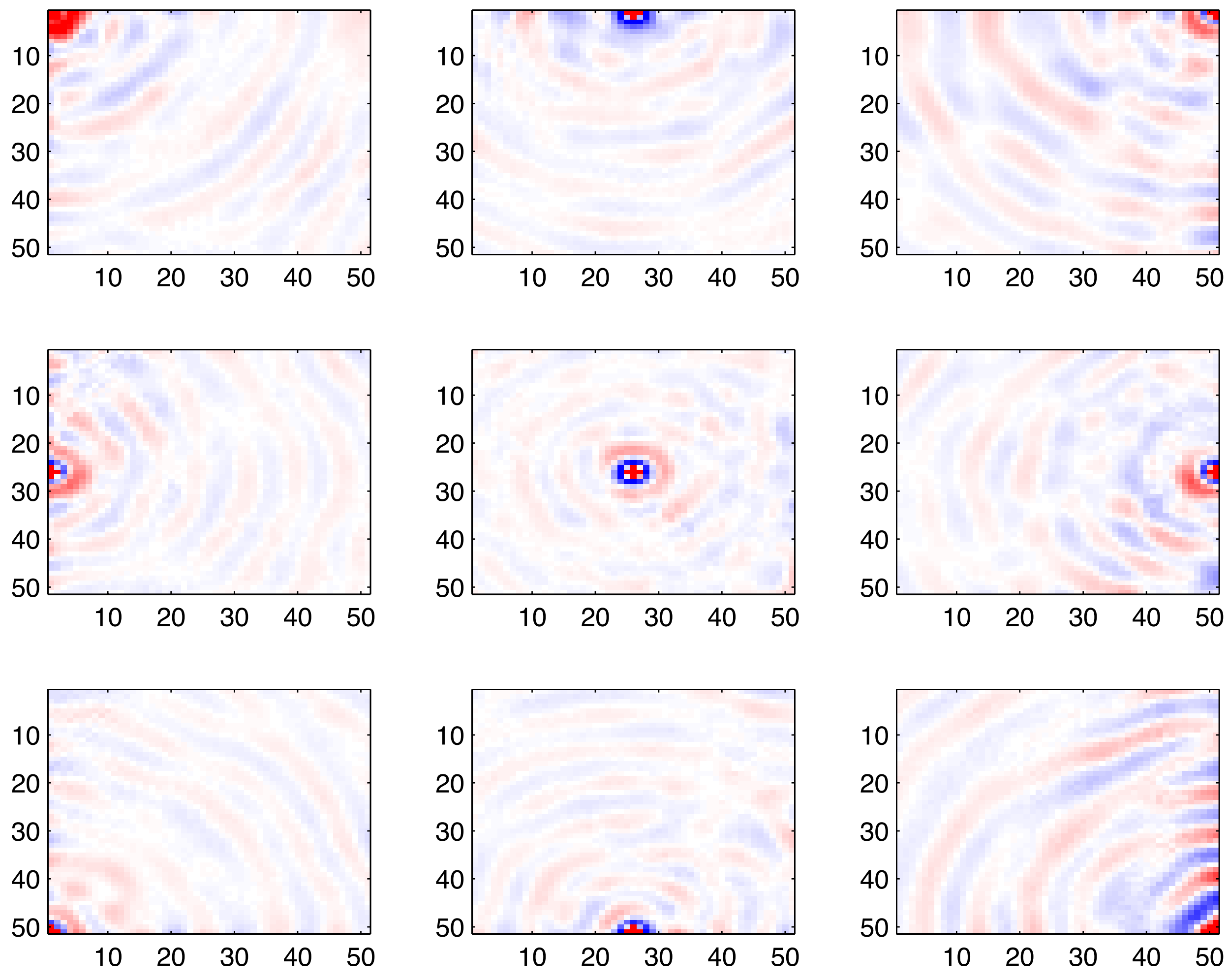
Performance

final data @ 4Hz



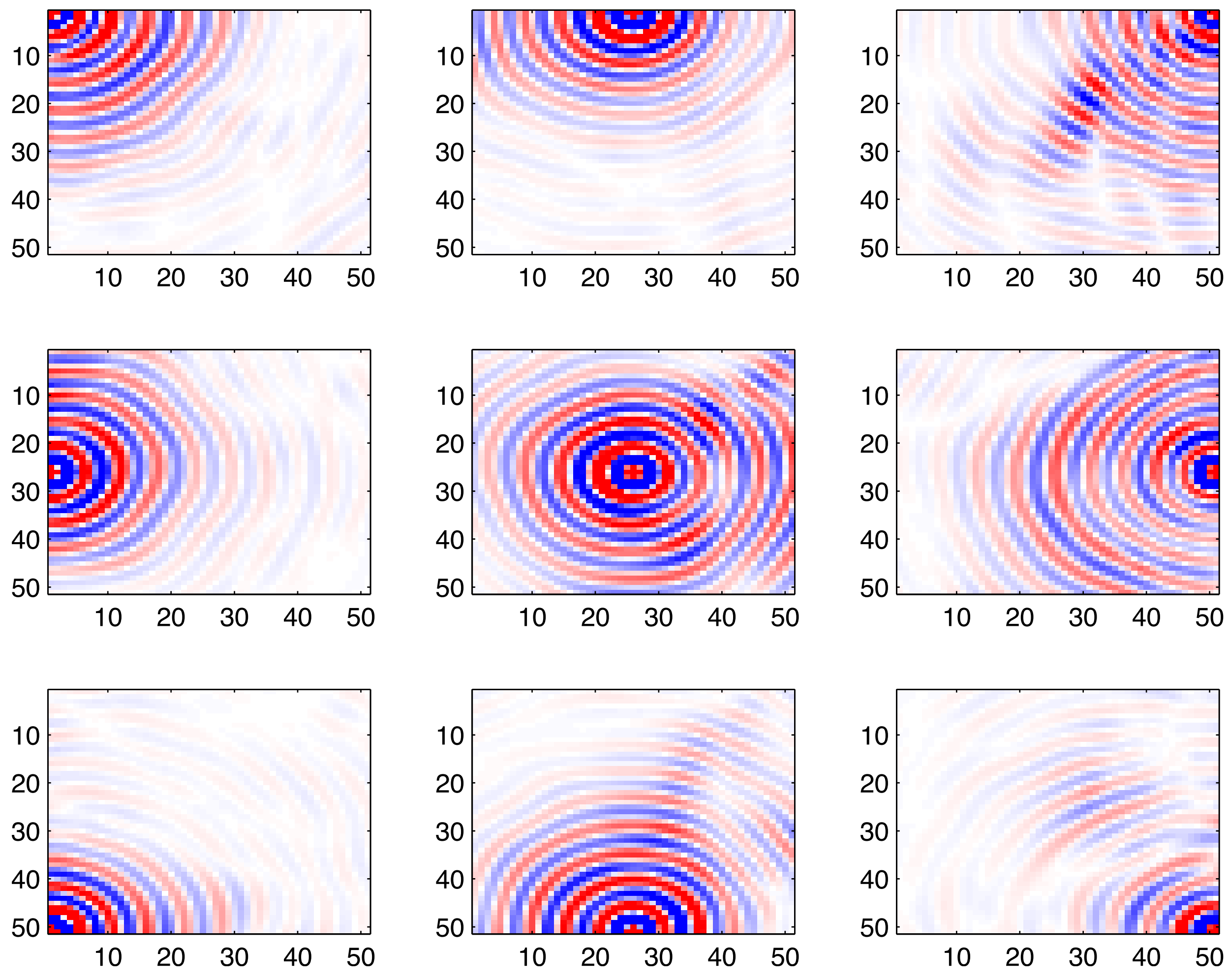
Performance

final residual @ 4Hz



Performance

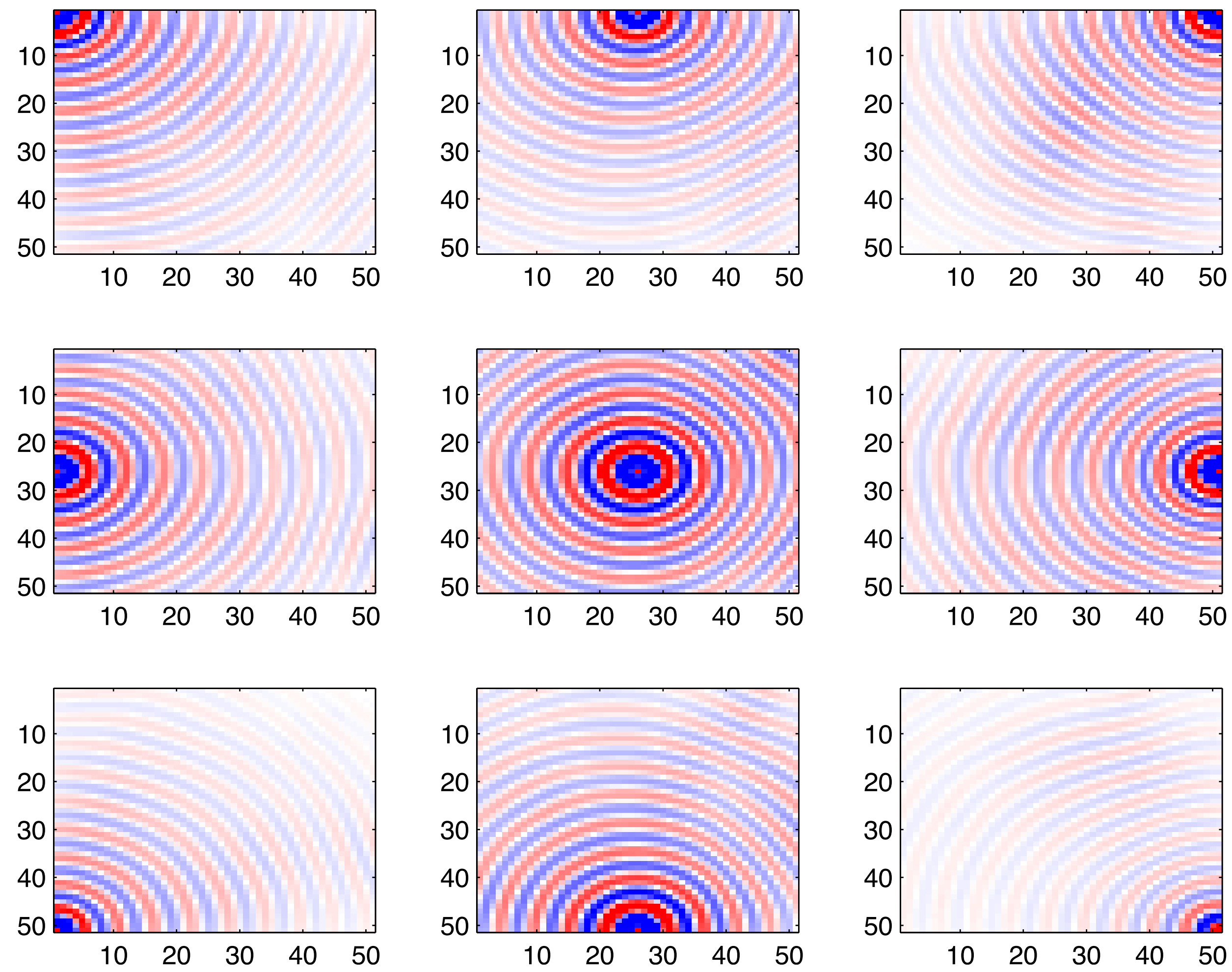
input data @ 6Hz



15 hours

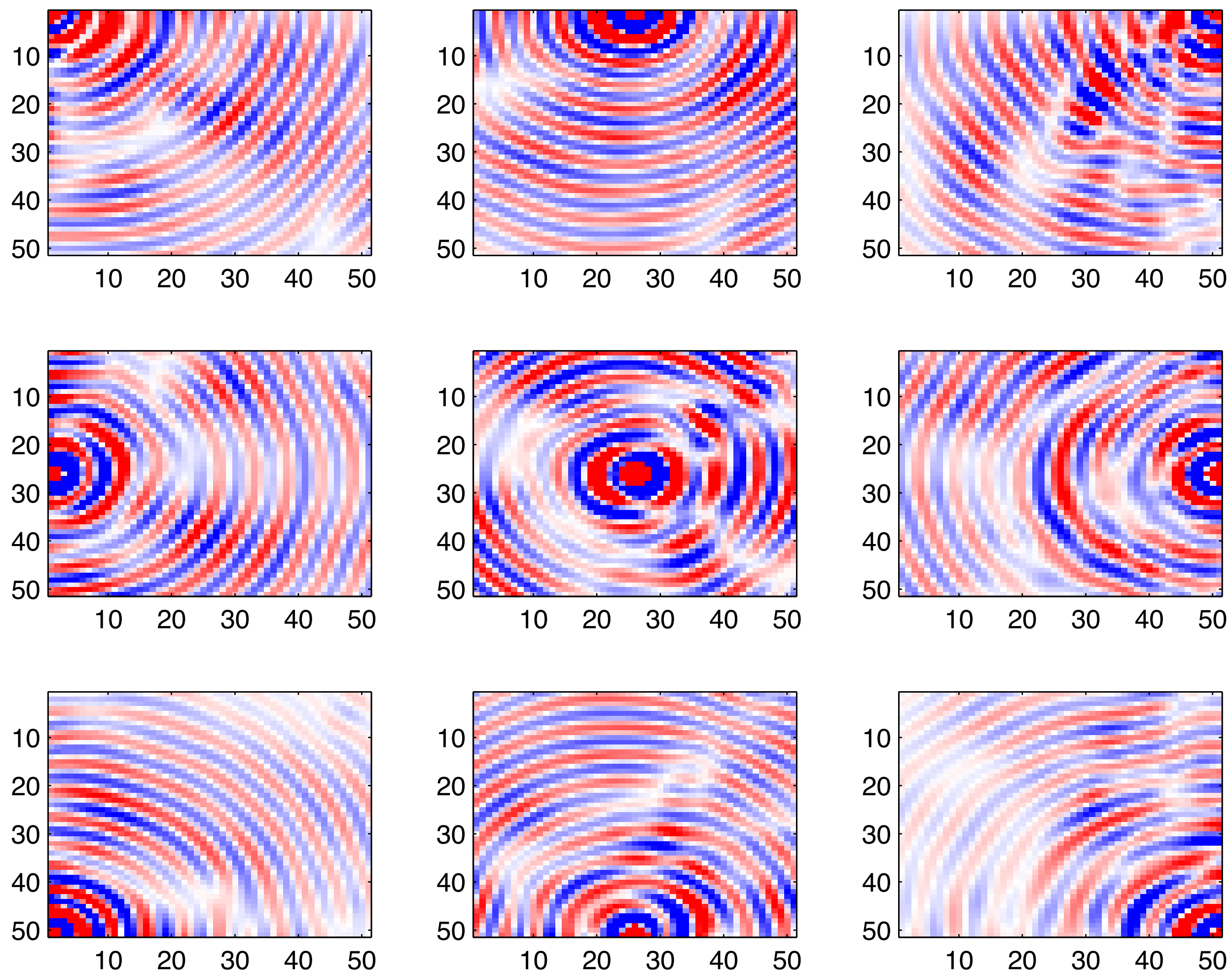
Performance

initial data @ 6Hz



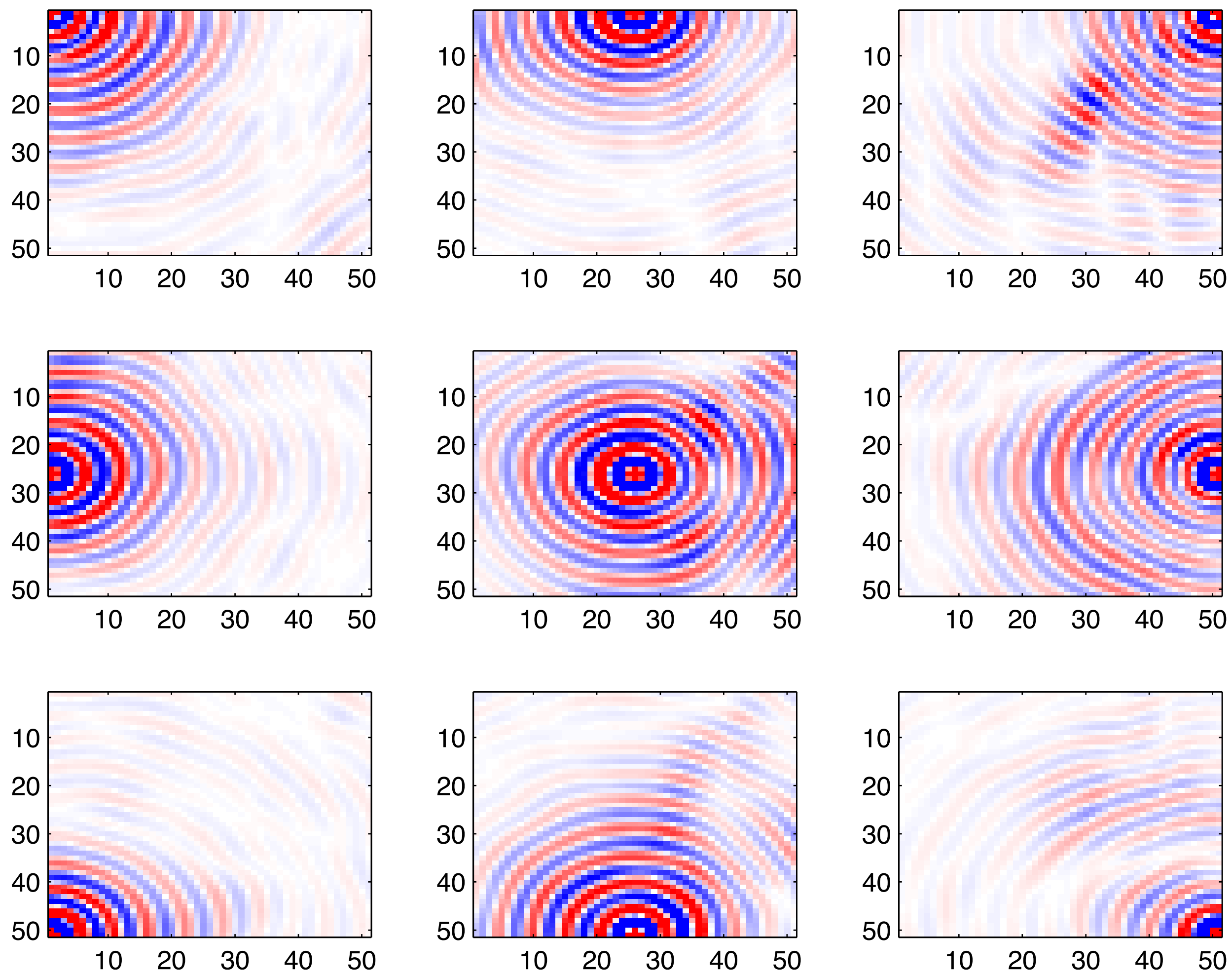
Performance

initial residual @ 6Hz



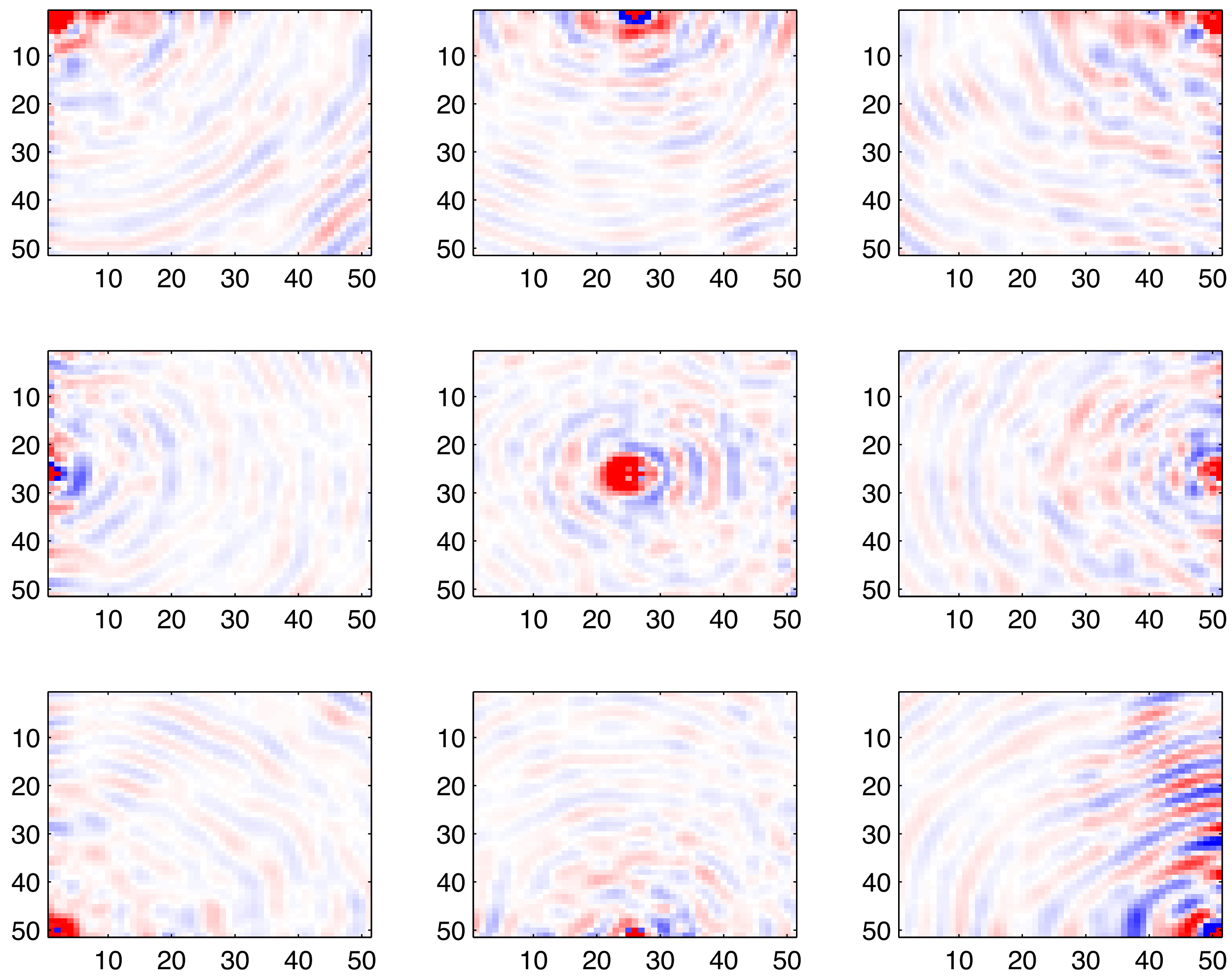
Performance

final data @ 6Hz



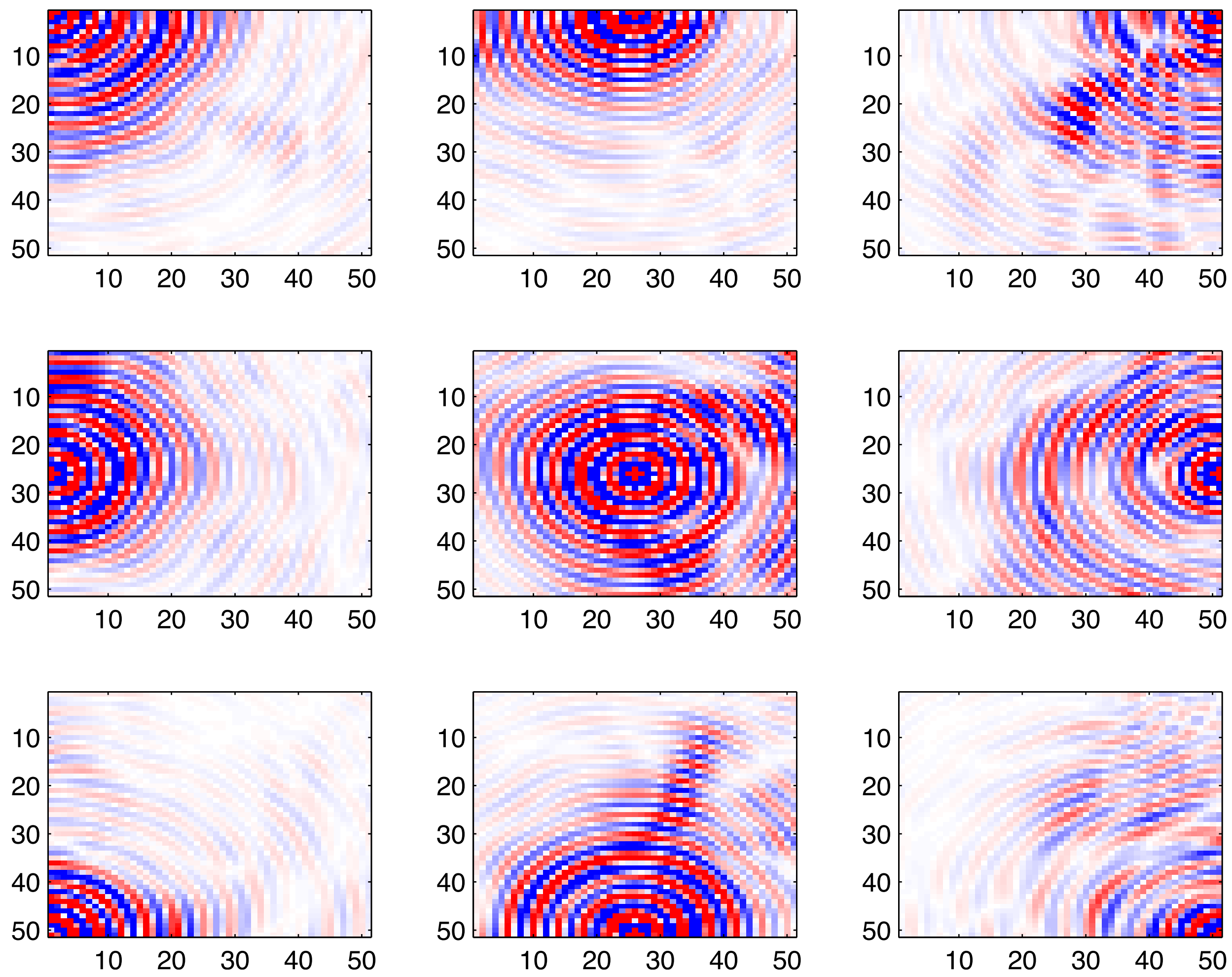
Performance

final residual @ 6Hz



Performance

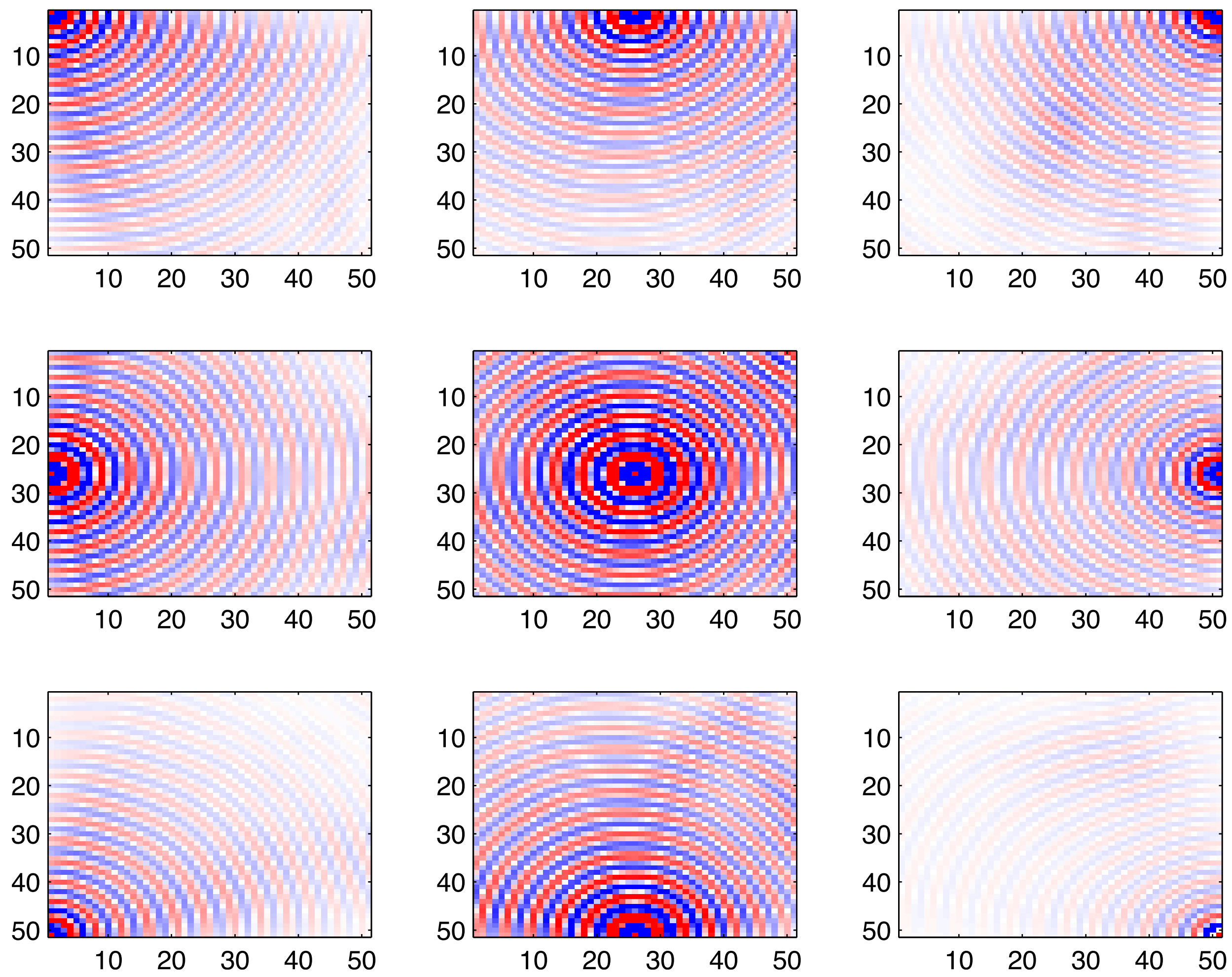
input data @ 8Hz



32 hours

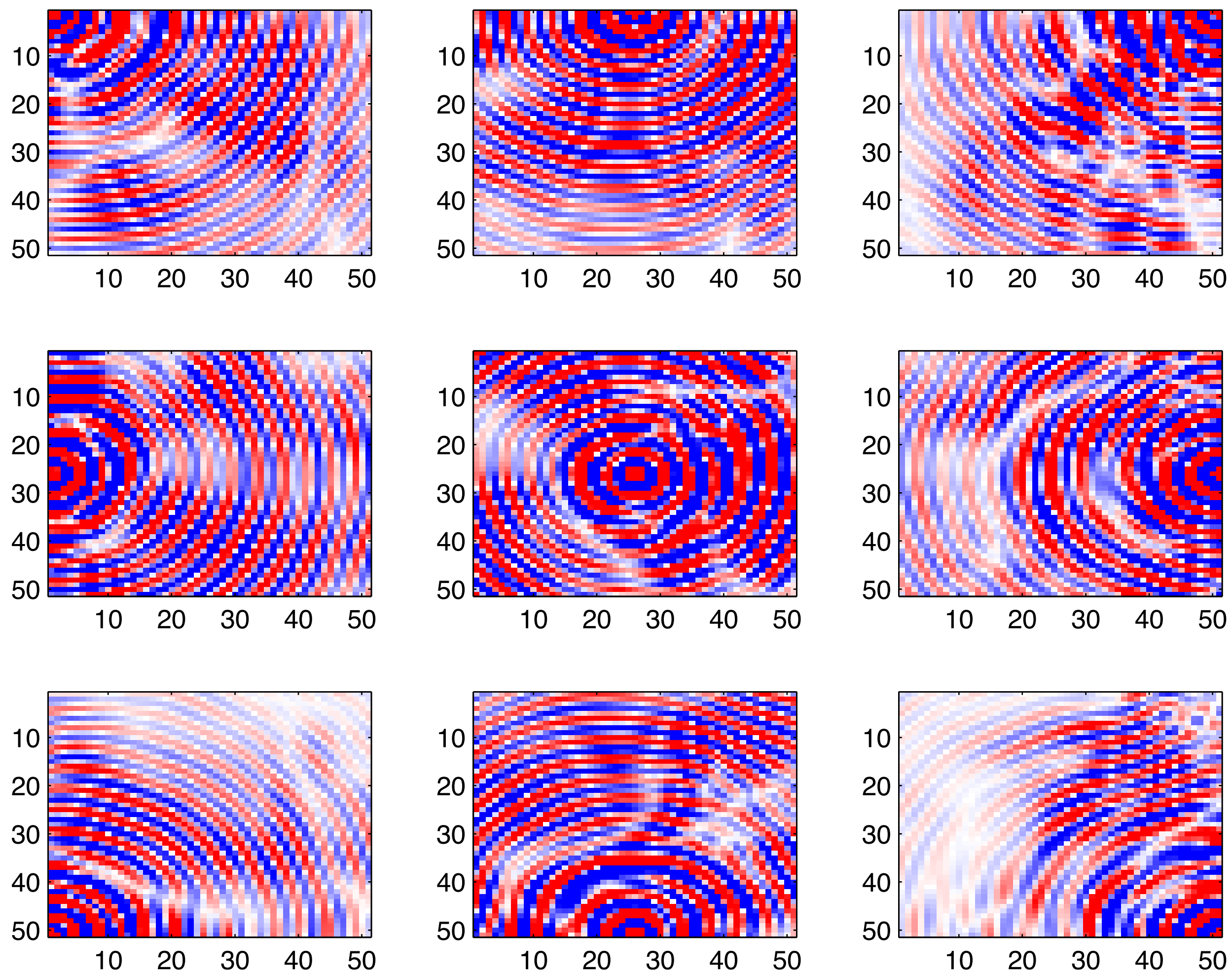
Performance

initial data @ 8Hz



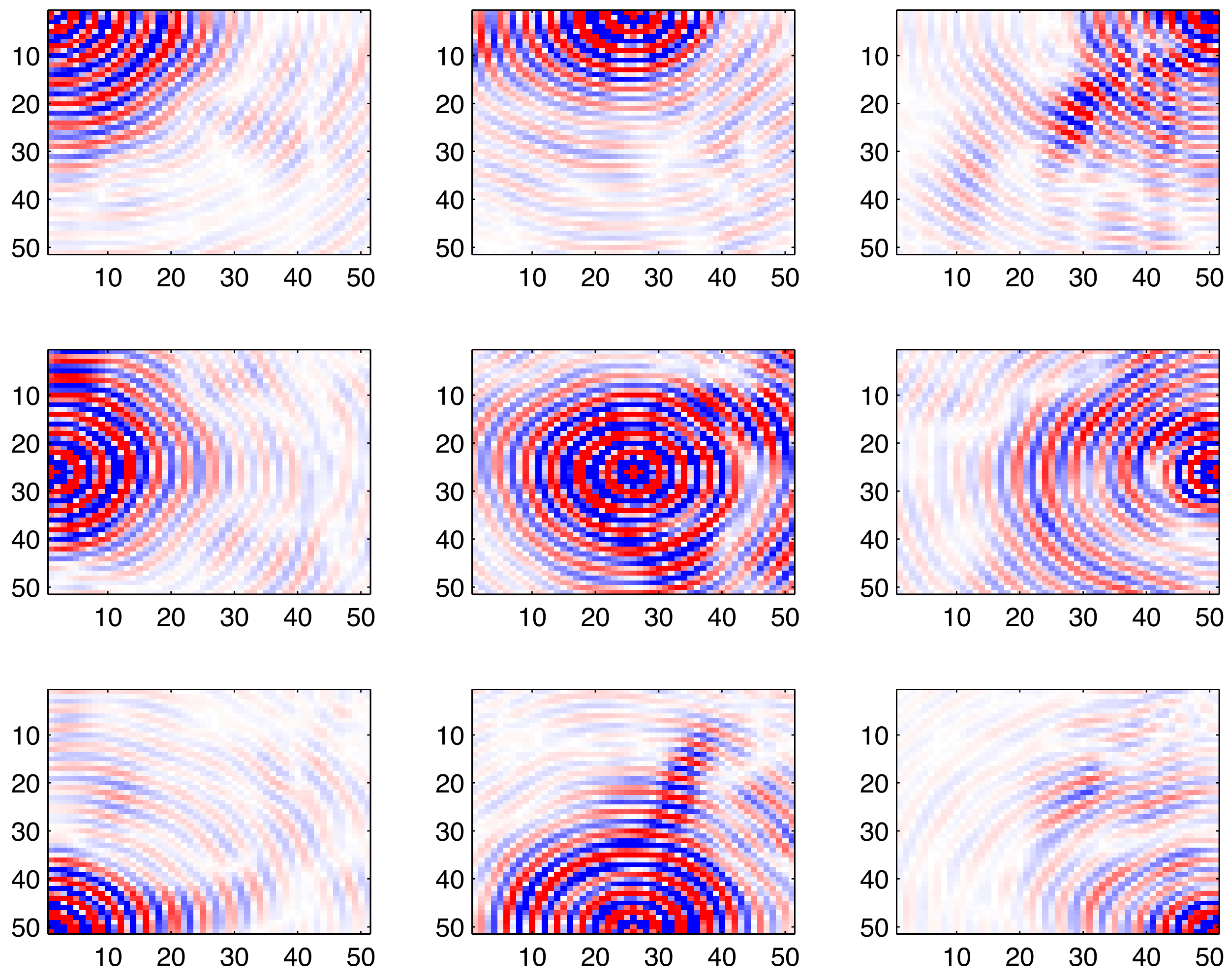
Performance

initial residual @ 8Hz



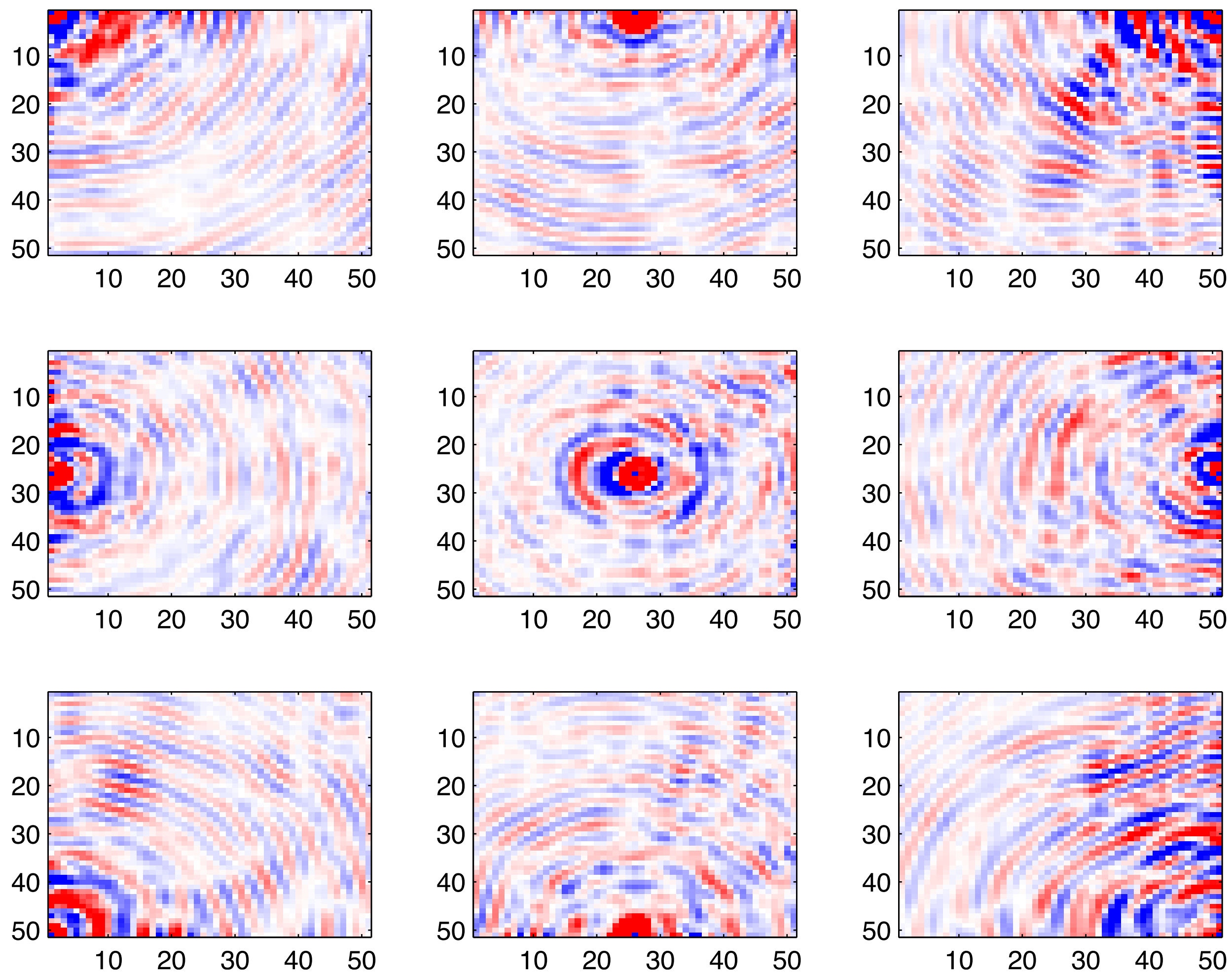
Performance

final data @ 8Hz



Performance

final residual @ 8Hz



Summary

Main *ingredients* for a *scalable* approach to 3D FWI:

- *iterative* Helmholtz solver w/ *little* memory imprint, computational overhead, and model-dependent tuning
- practical *stopping* criterion for wave simulator
- (stochastic) optimization technique that exploits the *separable* structure of FWI by working w/ *small* subsets
- *strategy* to *increase* sample size and accuracy as needed

Carry home message

Insisting on working w/

- *all* data
- *full* accuracy

can be *detrimental* to *FWI.*

When *ill*-conditioned use *less* rather than *more* data & *accuracy.*

Better to *call* for *more* data & *accuracy* only when *strictly* needed.

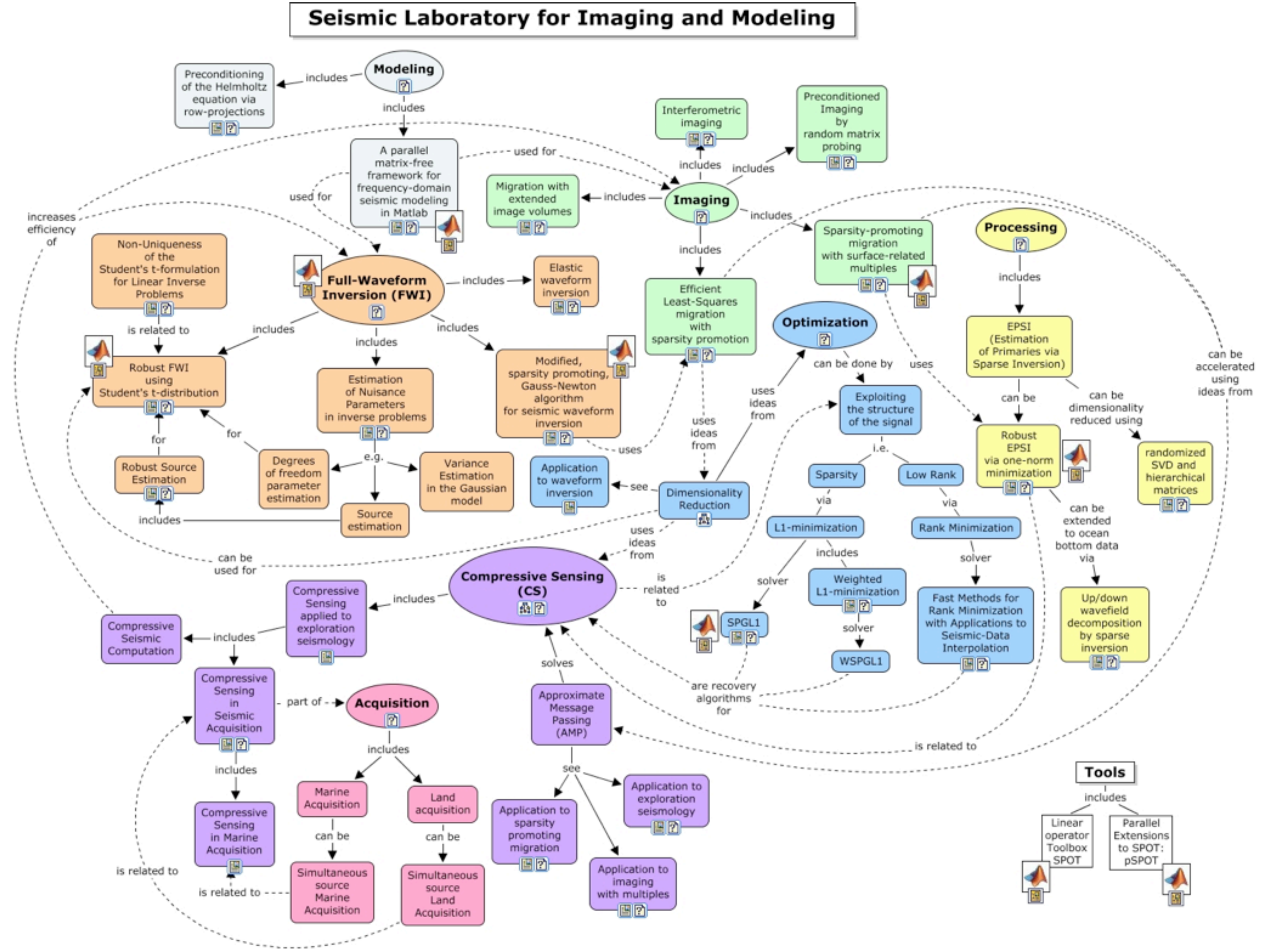
Less is really more...

Acknowledgments

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Collaborative Research and Development Grant DNOISE II (375142-08).

This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BP, BGP, CGGVeritas, Chevron, ConocoPhillips, ION GXT, Petrobras, PGS, Total SA, WesternGeco, and Woodside

NSERC
CRSNG



Seismic Laboratory for Imaging and Modeling
Modeling
includes
Preconditioning of the Helmholtz equation via row-projections
A parallel matrix-free framework for frequency-domain seismic modeling in Matlab
used for
increases efficiency of
Imaging
Interferometric imaging
Preconditioned Imaging by random matrix probing
Migration with extended image volumes
Sparsity-promoting migration with surface-related multiples
Efficient Least-Squares migration with sparsity promotion
Full-Waveform Inversion (FWI)
Elastic waveform inversion
Non-Uniqueness of the Student's t-formulation for Linear Inverse Problems
is related to
Robust FWI using Student's t-distribution
Estimation of Nuisance Parameters in inverse problems
Modified, sparsity promoting, Gauss-Newton algorithm for seismic waveform inversion
uses
for
Robust Source Estimation
Degrees of freedom parameter estimation
e.g.
Variance Estimation in the Gaussian model
Source estimation
Optimization
can be done by
Exploiting the structure of the signal
i.e.
Sparsity
Low Rank
via
L1-minimization
Rank Minimization
Weighted L1-minimization
solver
SPGL1
WSPGL1
Fast Methods for Rank Minimization with Applications to Seismic-Data Interpolation
Application to waveform inversion
see
Dimensionality Reduction
uses ideas from
Processing
EPSI (Estimation of Primaries via Sparse Inversion)
can be
Robust EPSI via one-norm minimization
can be dimensionality reduced using
randomized SVD and hierarchical matrices
can be extended to ocean bottom data via
Up/down wavefield decomposition by sparse inversion
can be accelerated using ideas from
Compressive Sensing (CS)
can be used for
is related to
are recovery algorithms for
Compressive Sensing applied to exploration seismology
Compressive Seismic Computation
Compressive Sensing in Seismic Acquisition
part of
Acquisition
Compressive Sensing in Marine Acquisition
Marine Acquisition
Land acquisition
Simultaneous source Marine Acquisition
Simultaneous source Land Acquisition
solves
Approximate Message Passing (AMP)
Application to sparsity promoting migration
Application to exploration seismology
Application to imaging with multiples
Tools
Linear operator Toolbox SPOT
Parallel Extensions to SPOT: pSPOT

Thank you

www.slim.eos.ubc.ca