3D WRI

Bas Peters
Joint work with Chen Greif & Felix J. Herrmann

2015 SINBAD Fall Consortium meeting. October 26.

SLIM
University of British Columbia

Problem of interest

Time-harmonic Wavefield Reconstruction Inversion (WRI) requires to solve:

$$\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}) \\ P \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ \mathbf{d} \end{pmatrix} \right\|_2$$

$H(\mathbf{m}) \in \mathbb{C}^{N \times N}$ discrete PDE

$\mathbf{m} \in \mathbb{R}^N$ medium parameters

$P \in \mathbb{R}^{m \times N}$ selects field at receivers

$\mathbf{u} \in \mathbb{C}^N$ field

$\mathbf{d} \in \mathbb{C}^m$ observed data

$\mathbf{q} \in \mathbb{C}^N$ source

$$\min_{\mathbf{m},\mathbf{u}} \frac{1}{2}\|P\mathbf{u}-\mathbf{d}\|_2^2 + \frac{\lambda^2}{2}\|H(\mathbf{m})\mathbf{u}-\mathbf{q}\|_2^2$$

A few algorithms are based on the quadratic-penalty form:

[R.E. Kleinman & P.M.van den Berg, 1992 ;
P.M. Van Den Berg & and R.E. Kleinman, 1997;
A. Abubakar et. al., 2002;
T. van Leeuwen & F.J. Herrmann, 2013]

$\downarrow$

eliminate field variables $\nabla_{\mathbf{u}}\phi(\mathbf{m},\bar{\mathbf{u}},\lambda)=0$

[T. van Leeuwen & F.J. Herrmann, 2013]

$\downarrow$

$$\min_{\mathbf{m}} \frac{1}{2}\|P\bar{\mathbf{u}}-\mathbf{d}\|_2^2 + \frac{\lambda^2}{2}\|H(\mathbf{m})\bar{\mathbf{u}}-\mathbf{q}\|_2^2$$

reduced quadratic-penalty

[T. van Leeuwen & F.J. Herrmann, 2013]

WRI

To minimize: $$\bar{\phi}(\mathbf{m}, \bar{\mathbf{u}}, \lambda) = \frac{1}{2}\|P\bar{\mathbf{u}} - \mathbf{d}\|_2^2 + \frac{\lambda^2}{2}\|H(\mathbf{m})\bar{\mathbf{u}} - \mathbf{q}\|_2^2$$

at every outer iteration:

- compute $\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}) \\ P \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ \mathbf{d} \end{pmatrix} \right\|_2$
- evaluate $\bar{\phi}(\mathbf{m}, \bar{\mathbf{u}}, \lambda)$ & $\nabla_{\mathbf{m}}\bar{\phi}(\mathbf{m}, \bar{\mathbf{u}}, \lambda)$
- update $\mathbf{m}$

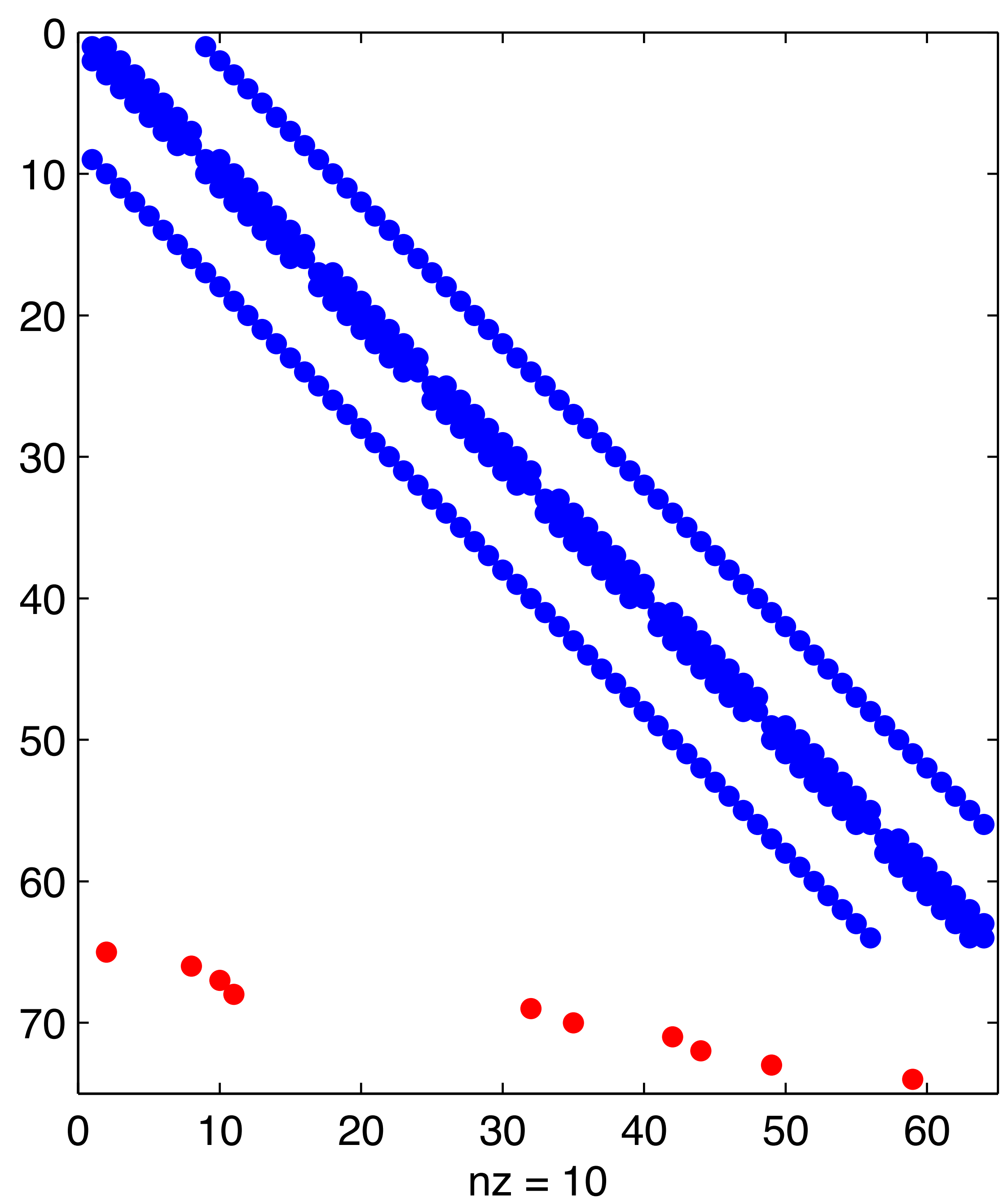
Properties of the problem

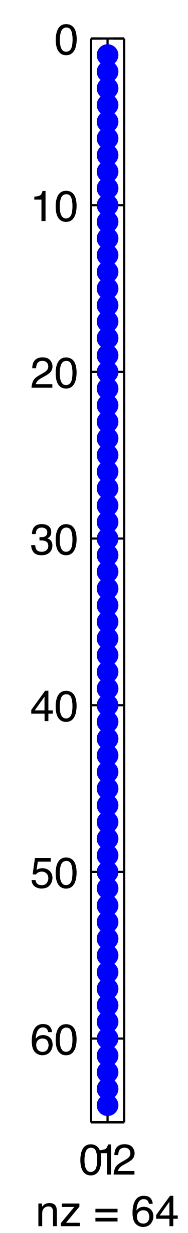
$$\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}) \\ P \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ \mathbf{d} \end{pmatrix} \right\|_2$$

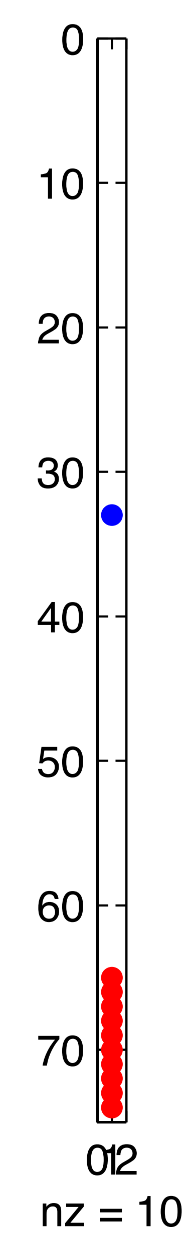
- H is indefinite, non-Hermitian
- inconsistent
- full column rank

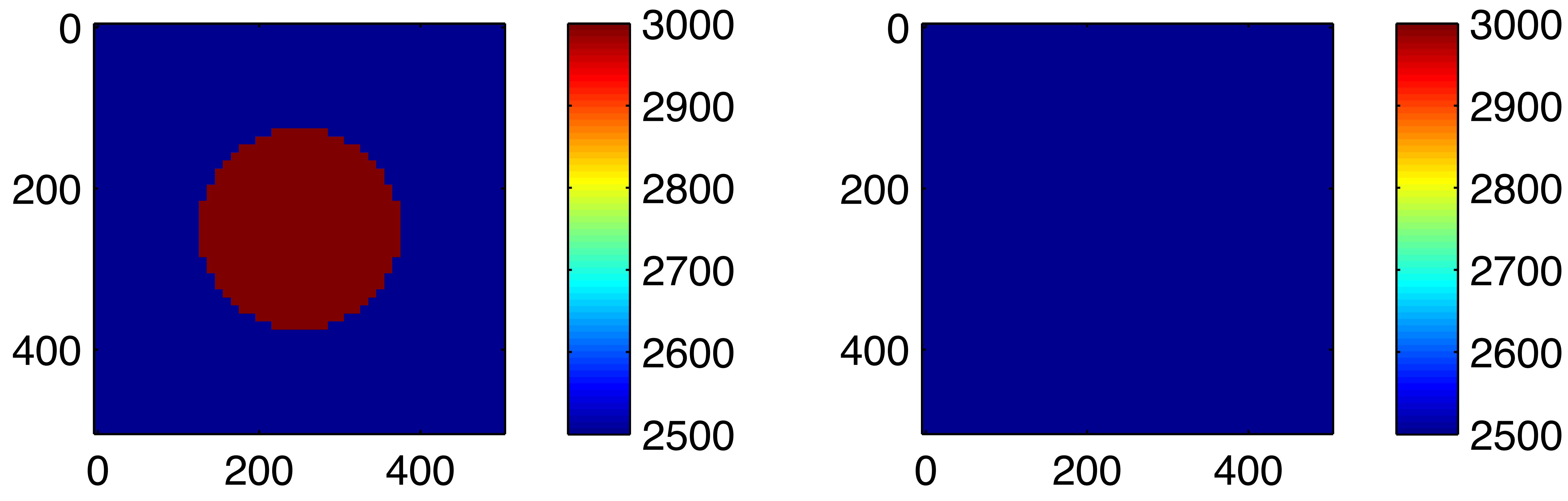
Properties of the problem

$$\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}) \\ {\color{red}P} \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ {\color{red}\mathbf{d}} \end{pmatrix} \right\|_2$$





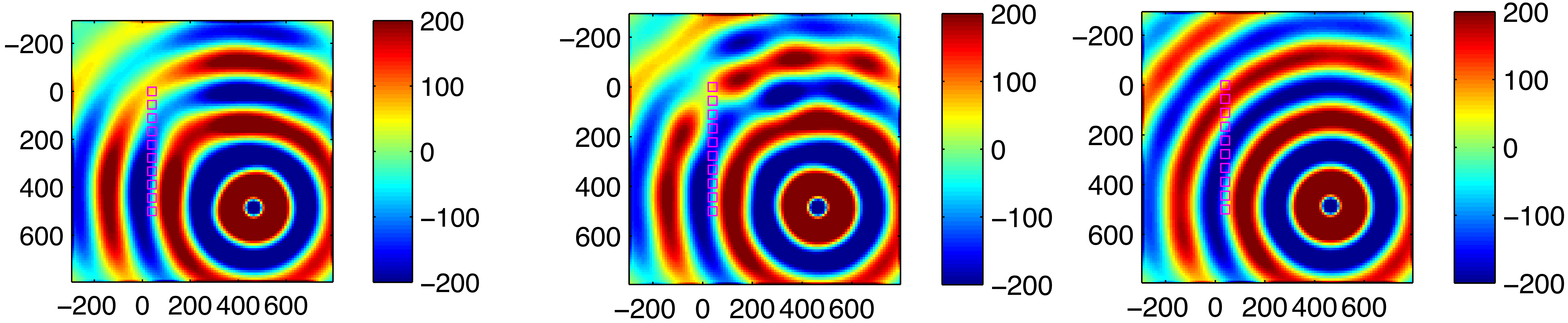




2D example

$$\mathbf{u} = H(\mathbf{m}_*)^{-1}\mathbf{q}$$

$$\mathbf{u} = H(\mathbf{m}_0)^{-1}\mathbf{q}$$



$$\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}_0) \\ P \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ \mathbf{d} \end{pmatrix} \right\|_2$$

Solution of the sub-problem

Main challenge: solve $\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}) \\ P \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ \mathbf{d} \end{pmatrix} \right\|_2$

- 2D: direct factorization **[L. M. Delves & I. Barrodale, 1979 ; T. A. Davis, 2011]**

In 3D we want:

- iteratively & matrix-free
- no QR or LU factorizations
- at cost of a few PDE solves

Derivation of the proposed algorithm

Manuscript is in preparation.

Some details on the derivation of the proposed algorithm:

Bas Peters, Chen Greif, and Felix J. Herrmann,

"An algorithm for solving least-squares problems with a Helmholtz block and multiple right-hand-sides",

International Conference On Preconditioning Techniques For Scientific And Industrial Applications, 2015

Proposed algorithm

for angular frequency ω **do**

$\quad$ // solve m Helmholtz problems

$\quad$ $H_\lambda^* W = P^*$

$\quad$ $M = (I + W^*W)^{-1}$

$\quad$ **for** right hand side i **do**

$\quad\quad$ $\mathbf{y}_i = \big(I - WMW^*\big)\big(\lambda \mathbf{q}_i + W\mathbf{d}_i\big)$

$\quad\quad$ // solve for $\bar{\mathbf{u}}_i$

$\quad\quad$ $H_\lambda \bar{\mathbf{u}}_i = \mathbf{y}_i$

$\quad$ **end for**

end for

Proposed algorithm

Matrix-free algorithm

- no direct solves
- related mildly overdetermined systems **[L. M. Delves & I. Barrodale, 1979]**

Computational cost:

- 1 PDE per receiver
- 1 PDE per source

Memory requirements:

- 1 vector per receiver (W)
- system matrix (H)
- storage for solving systems with H

Proposed algorithm

Inexact solutions to the linear systems:

for angular frequency ω **do**

// solve m Helmholtz problems inexactly

$$\boxed{H_\lambda^* \hat{W} = P^* + R_W}$$

$$\hat{M} = (I + \hat{W}^* \hat{W})^{-1}$$

for right hand side $\mathbf{b}_i$ **do**

$$\hat{\mathbf{y}}_i = \left(I - \hat{W}\hat{M}\hat{W}^*\right)\left(\lambda \mathbf{q}_i + \hat{W}\mathbf{d}_i\right)$$

// solve for $\bar{\mathbf{u}}_i$ inexactly

$$\boxed{H_\lambda \hat{\mathbf{u}}_i = \hat{\mathbf{y}}_i + \mathbf{r_u}}$$

end for

end for

Proposed algorithm

error propagation (1 right-hand-side, 1 receiver case):

$$H_\lambda^* \hat{\mathbf{w}} = \mathbf{p}^* + \mathbf{r}_{\mathbf{w}}$$

$$(I + \hat{\mathbf{w}}\hat{\mathbf{w}}^*)\hat{\mathbf{y}} = \lambda \mathbf{q} + \hat{\mathbf{w}} d$$

$$H_\lambda \hat{\mathbf{u}} = \hat{\mathbf{y}} + \mathbf{r}_{\mathbf{u}}$$

Proposed algorithm

error propagation (1 right-hand-side, 1 receiver case):

$$H_\lambda^* \hat{\mathbf{w}} = \mathbf{p}^* + \mathbf{r}_{\mathbf{w}}$$

$$(I + \hat{\mathbf{w}}\hat{\mathbf{w}}^*)\hat{\mathbf{y}} = \lambda \mathbf{q} + \hat{\mathbf{w}} d$$

$\longrightarrow$ solve as: $\hat{\mathbf{y}} = (I - \hat{m}\hat{\mathbf{w}}\hat{\mathbf{w}}^*)(\lambda \mathbf{q} + \hat{\mathbf{w}} d)$

with $\hat{m} = \dfrac{1}{1 + \hat{\mathbf{w}}^* \hat{\mathbf{w}}}$

$$H_\lambda \hat{\mathbf{u}} = \hat{\mathbf{y}} + \mathbf{r}_{\mathbf{u}}$$

Suggested PDE-solver

Store 1 vector per receiver

-> use PDE-solver w/ low-memory & setup requirements

Helmholtz:

- CGMN (only 4 vectors) / CARP-CG [A. Bjorck & T. Elfving, 1979; D. Gordon & R. Gordon, 2010; T. van Leeuwen & F.J. Herrmann, 2014]
- shifted-Laplacian w/ multi-grid [Y.A. Erlangga, 2008; H. Calandra et al., 2013]
- combination of the above [R. Lago & F.J. Herrmann, 2015]

Simultaneous receivers

Simultaneous sources reduce the number of sources to be modeled.

Can we use similar ideas with the proposed algorithm?

Memory and computational cost now depends on the number of sources + receivers.

Simultaneous receivers

What is the number of receivers is too large, storage wise?

Can we approximate the least-squares problem using randomization & subsampling (simultaneous receivers)?

Use ideas from algorithms such as

- **[V Rokhlin & M Tygert, 2008]**
- Blendenpik **[H. Avron et. al., 2010]**
- LSRN **[X. Meng, M. A. Saunders, M. W. Mahoney, 2014]**

Simultaneous receivers

Blendenpik:

- randomize (mix the rows) & subsample very overdetermined systems
- use *R* from *QR* of the approximated and well conditioned problem as a preconditioner for LSQR to solve the original problem.
- define randomize & subsample matrix as: $V = SFD$,

$$V \in \mathbb{C}^{l \times m}, \quad l < m$$

$$D \in \mathbb{R}^{m \times m} \quad \text{random [+1 , -1] on the diagonal}$$

$$F \in \mathbb{C}^{m \times m} \quad \text{DFT matrix}$$

$$S \in \mathbb{R}^{l \times m} \quad \text{subsampling matrix, restriction of the identity}$$

Simultaneous receivers

Initial attempt in this work:

Apply randomization and subsampling to the receiver block only for a one-step approximation:

$$\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}) \\ VP \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ V\mathbf{d} \end{pmatrix} \right\|_2$$

$$V \in \mathbb{C}^{l \times m}, \quad l < m$$

What should V be ? ongoing research, use $V = SFD$

to illustrate the principle

SLIM

Simultaneous receivers

Initial attempt in this work:

Apply randomization and subsampling to the receiver block only for a one-step approximation:

$$\bar{\mathbf{u}} = \arg\min_{\mathbf{u}} \left\| \begin{pmatrix} \lambda H(\mathbf{m}) \\ VP \end{pmatrix} \mathbf{u} - \begin{pmatrix} \lambda \mathbf{q} \\ V\mathbf{d} \end{pmatrix} \right\|_2$$

$$V \in \mathbb{C}^{l \times m}, \quad l < m$$

reduces

- # of PDE solves
- # vectors to be stored

Simultaneous receivers

Investigate error

- in the fields and
- in the resulting gradient,

introduced by subsampling and randomization.

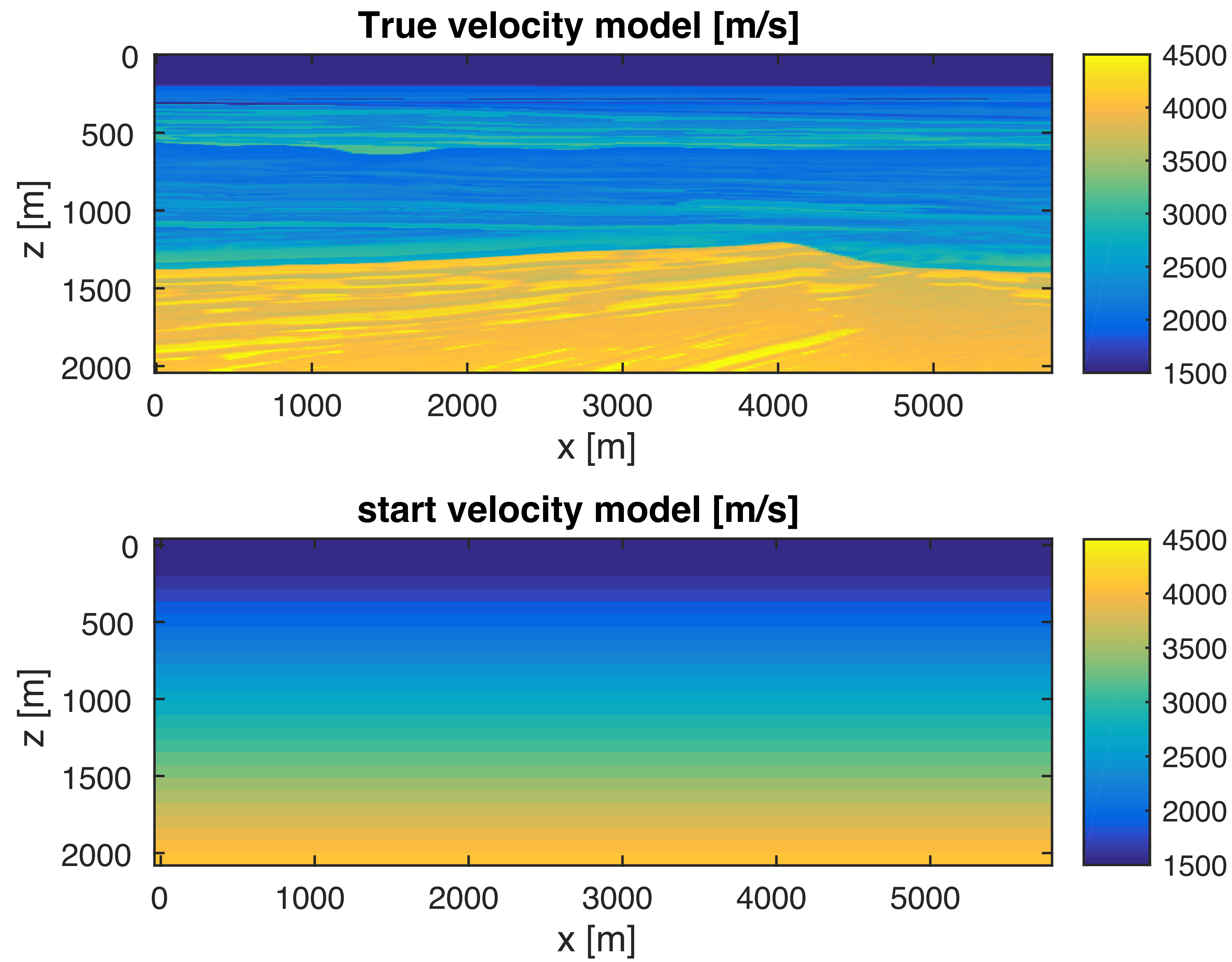
Assume:

- fixed number of sources (20)
- 1 frequency (3Hz)

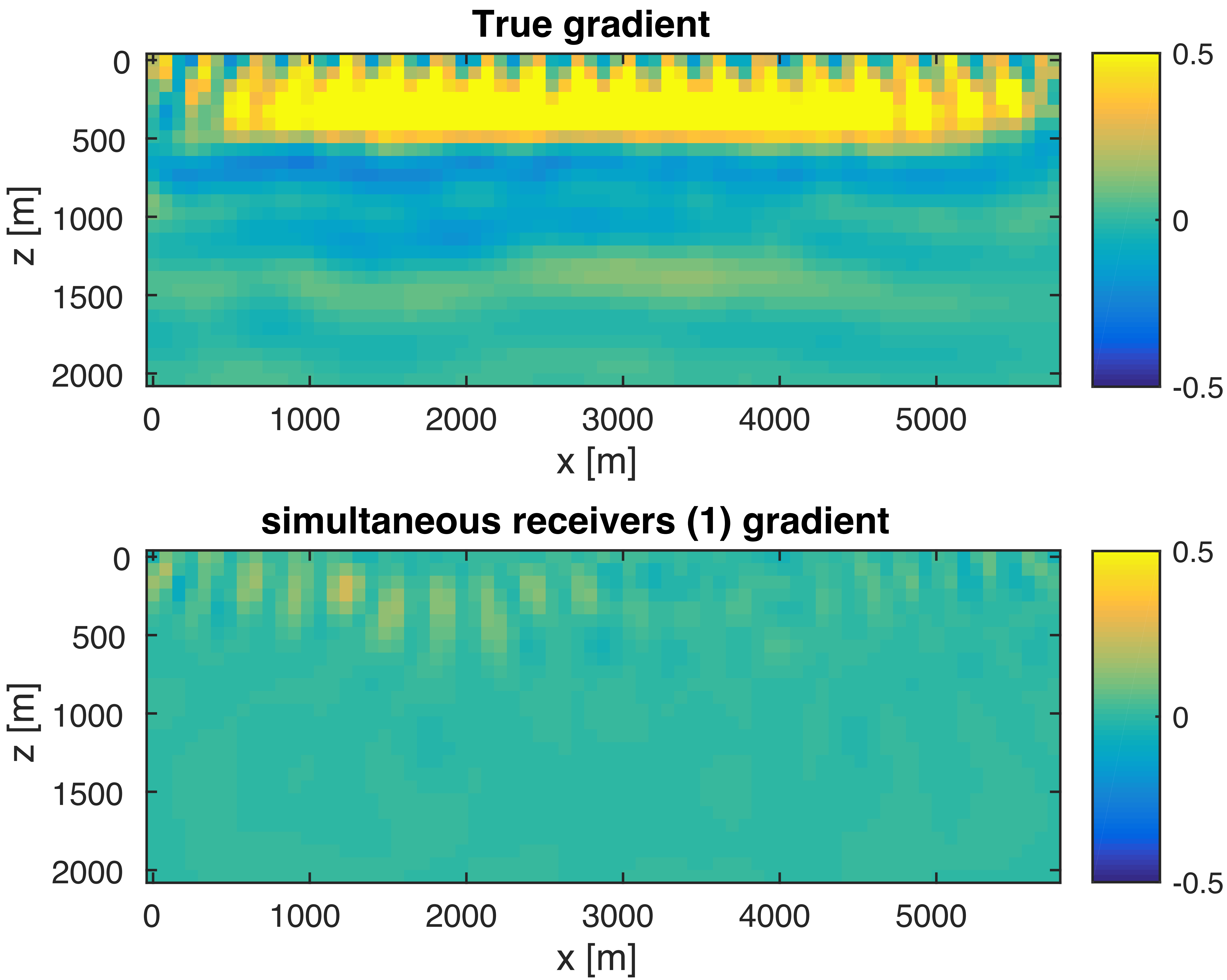
Simultaneous receivers

Example:

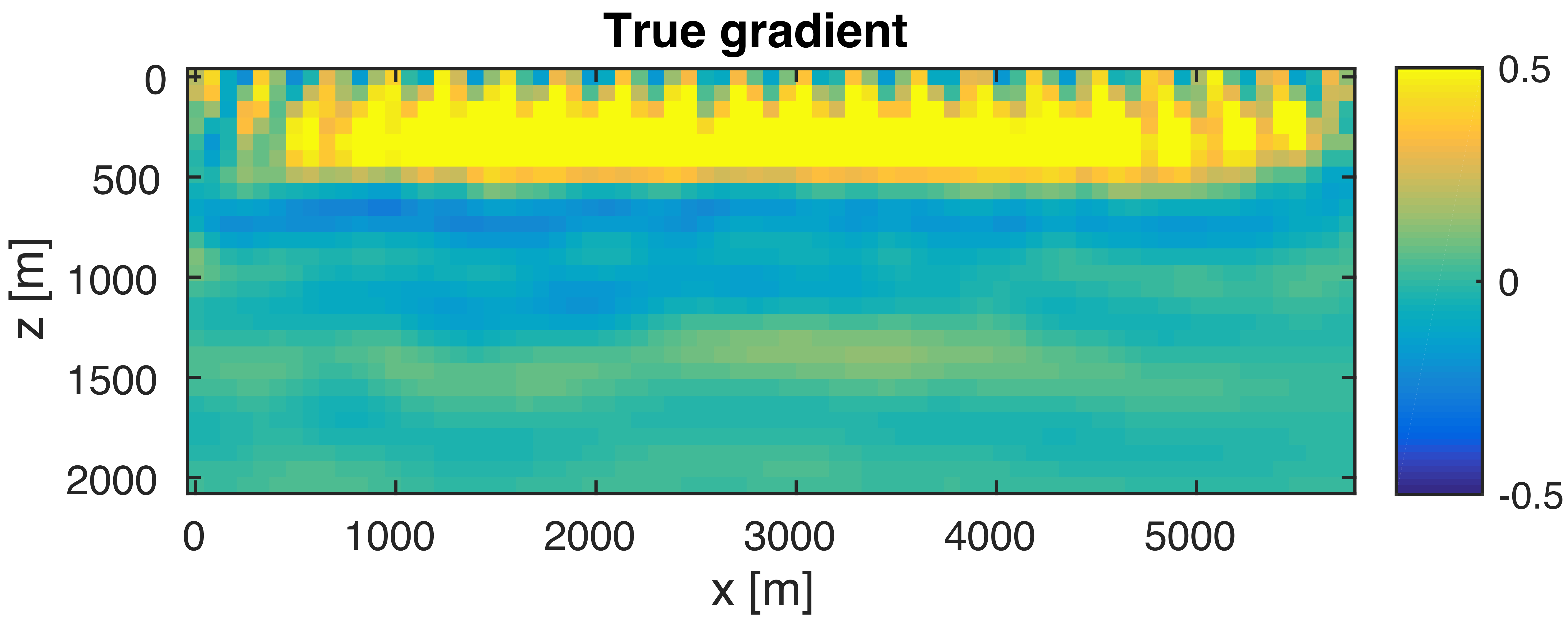
- 115 receivers in total
  (50 m interval)
- only work with randomized subsets of varying size
  (previous slides)
- 20 sources
  (300 m interval)
- 1 frequency (3Hz)

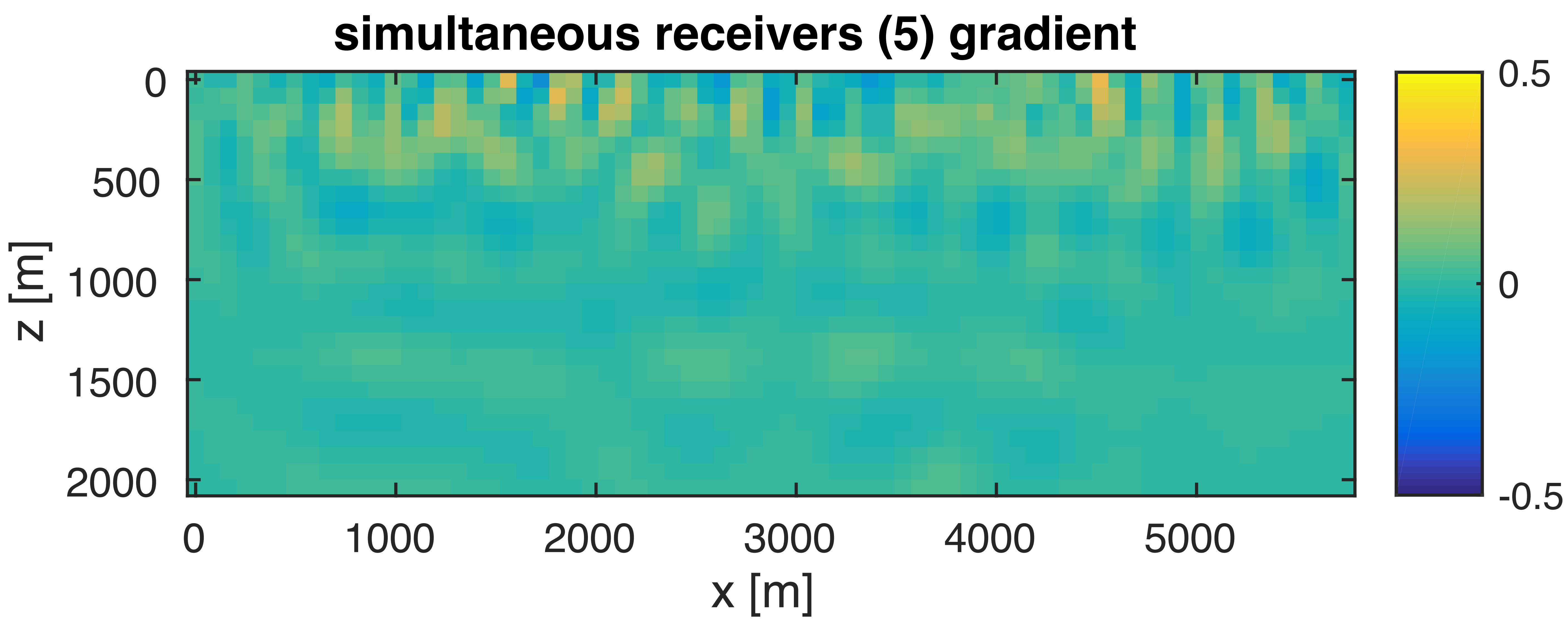


Simultaneous receivers

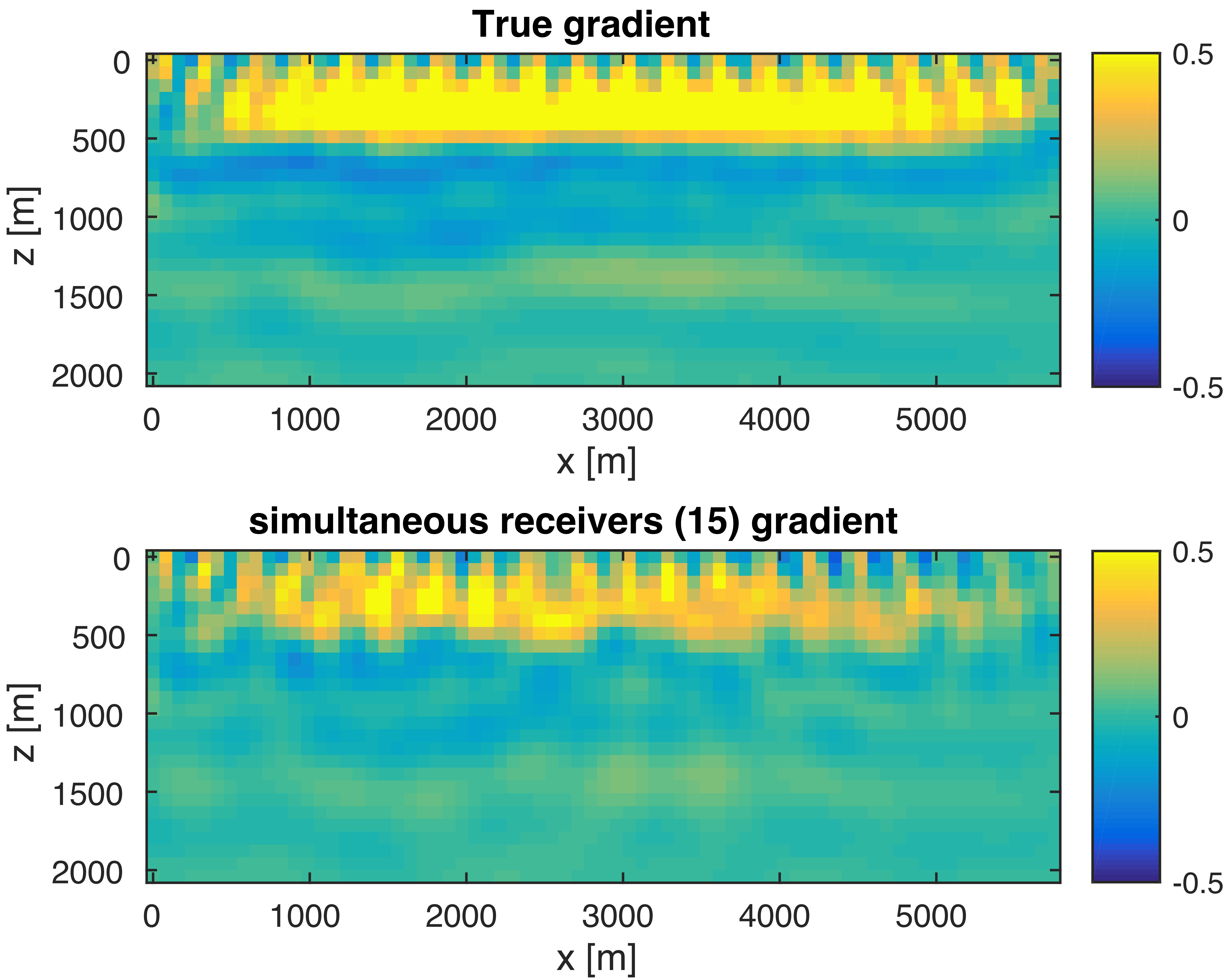


Simultaneous receivers

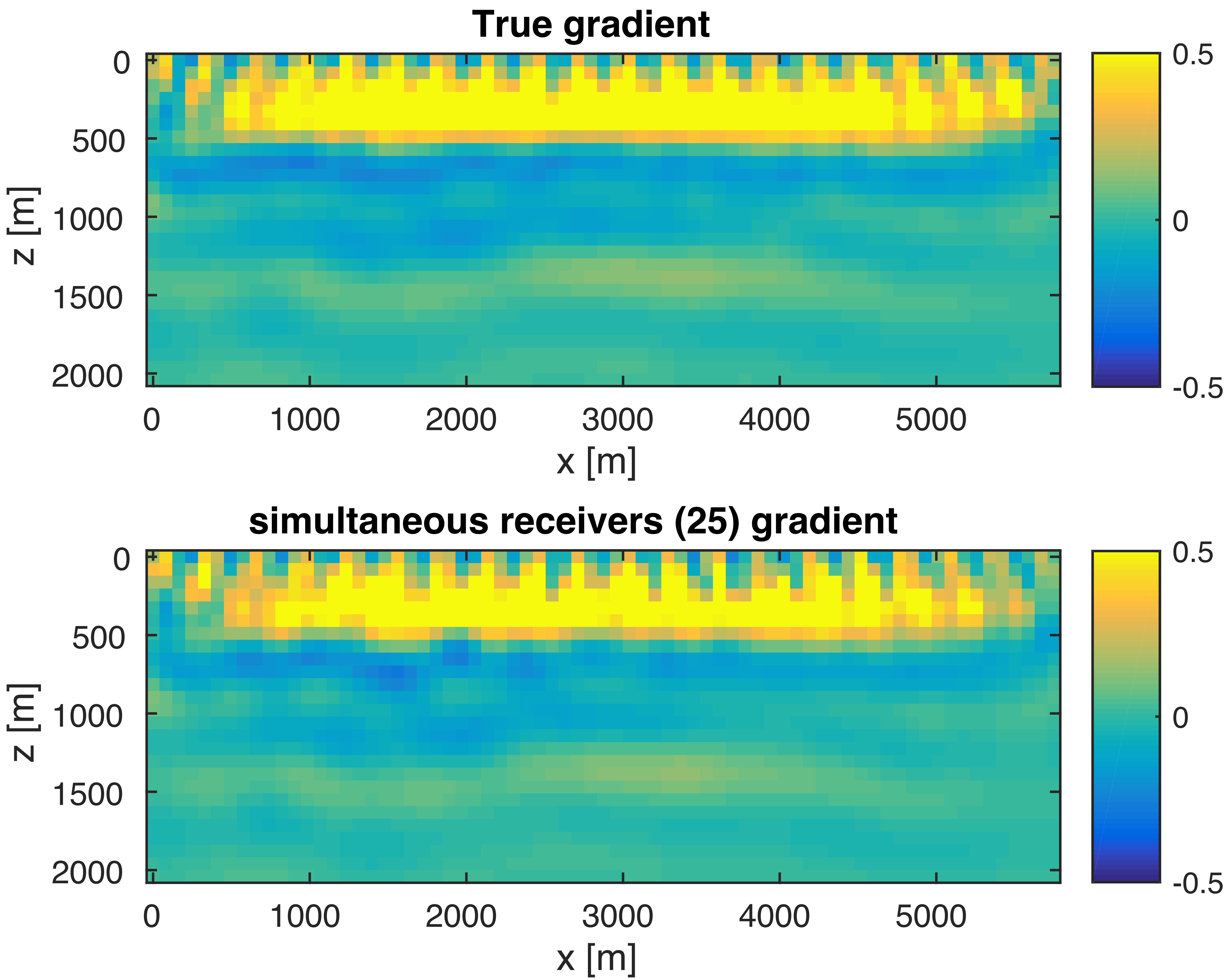




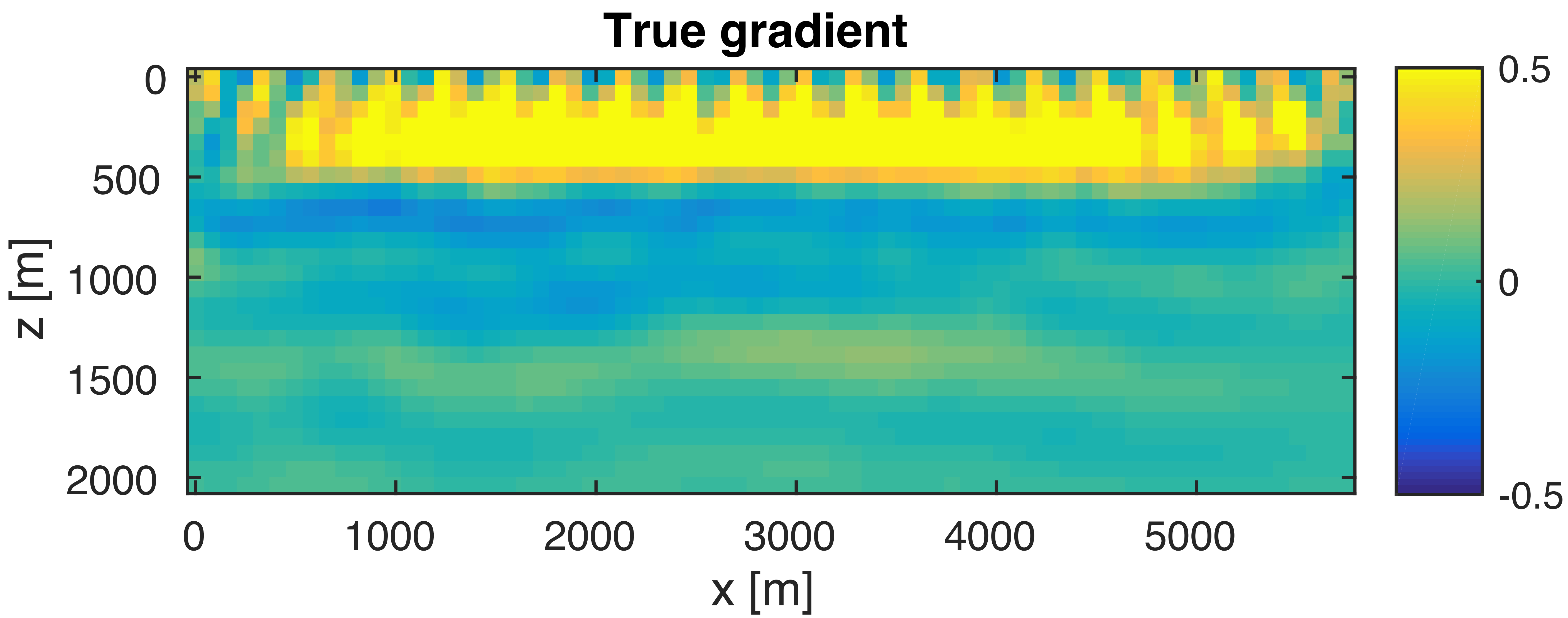
Simultaneous receivers

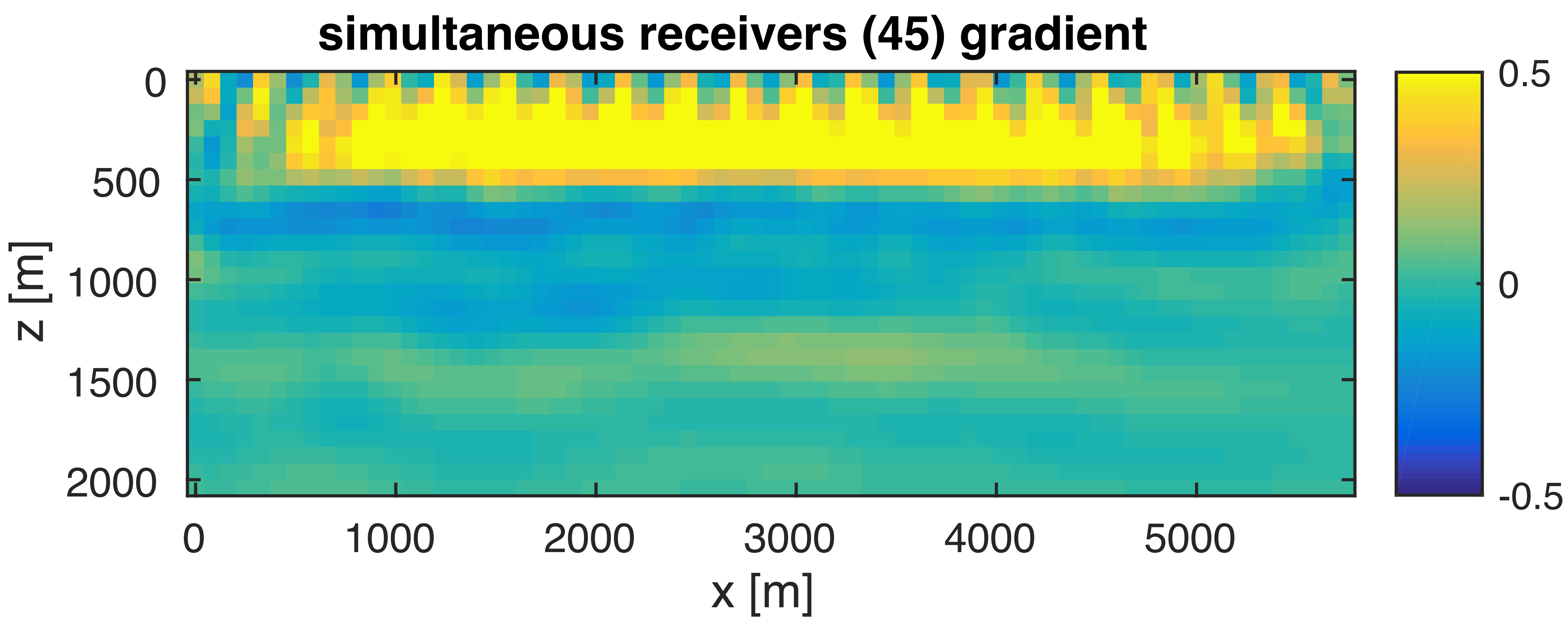


Simultaneous receivers

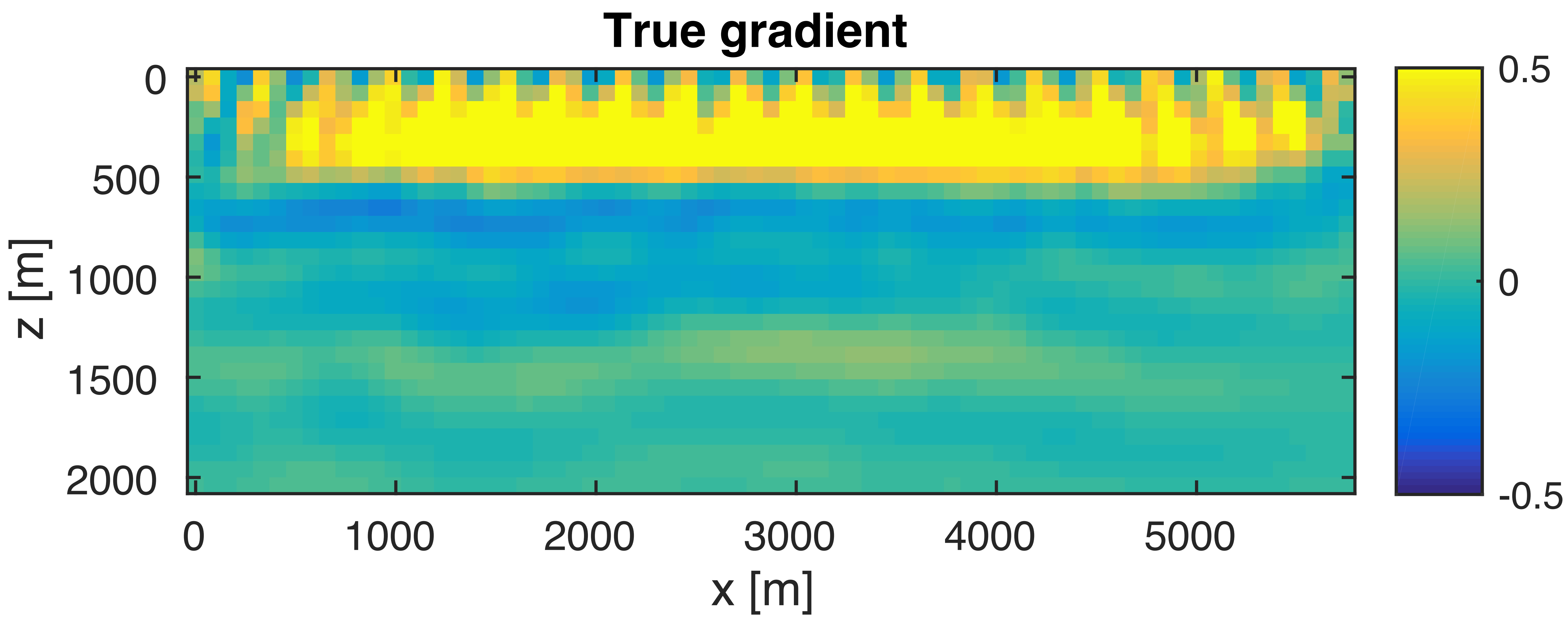


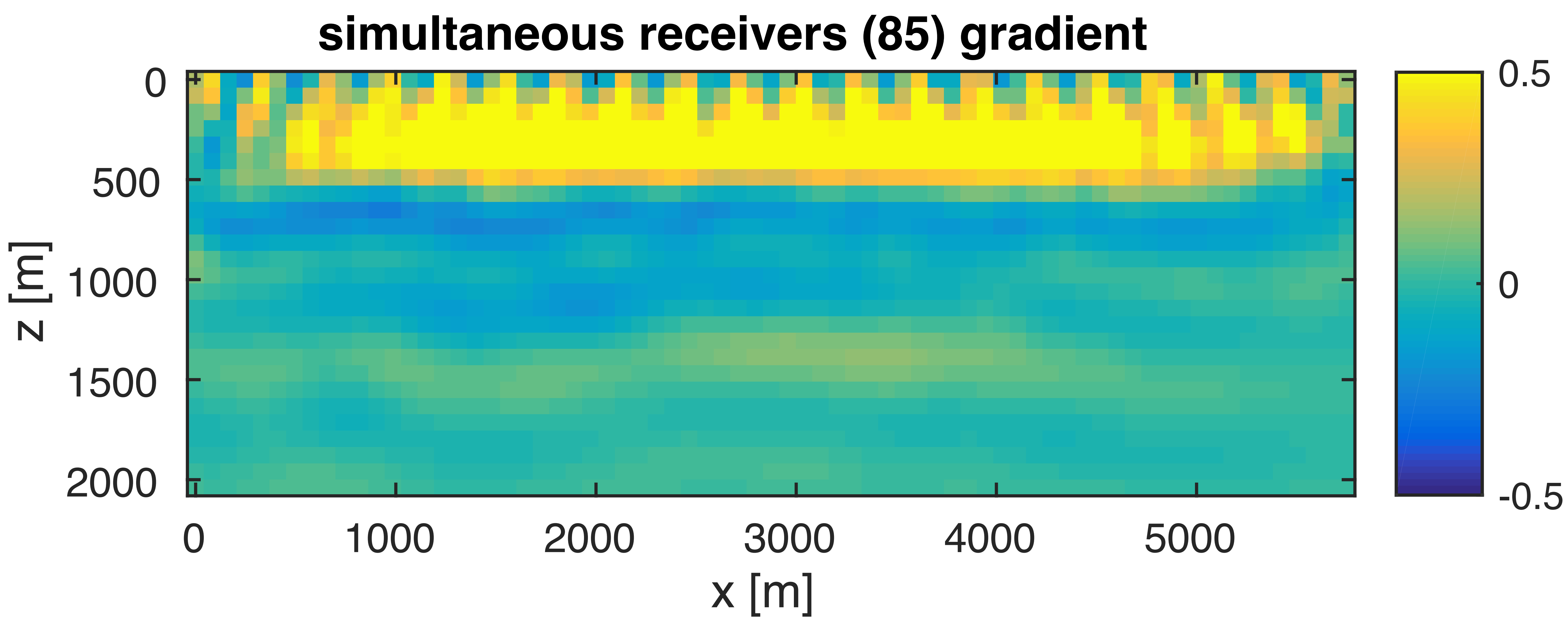
Simultaneous receivers



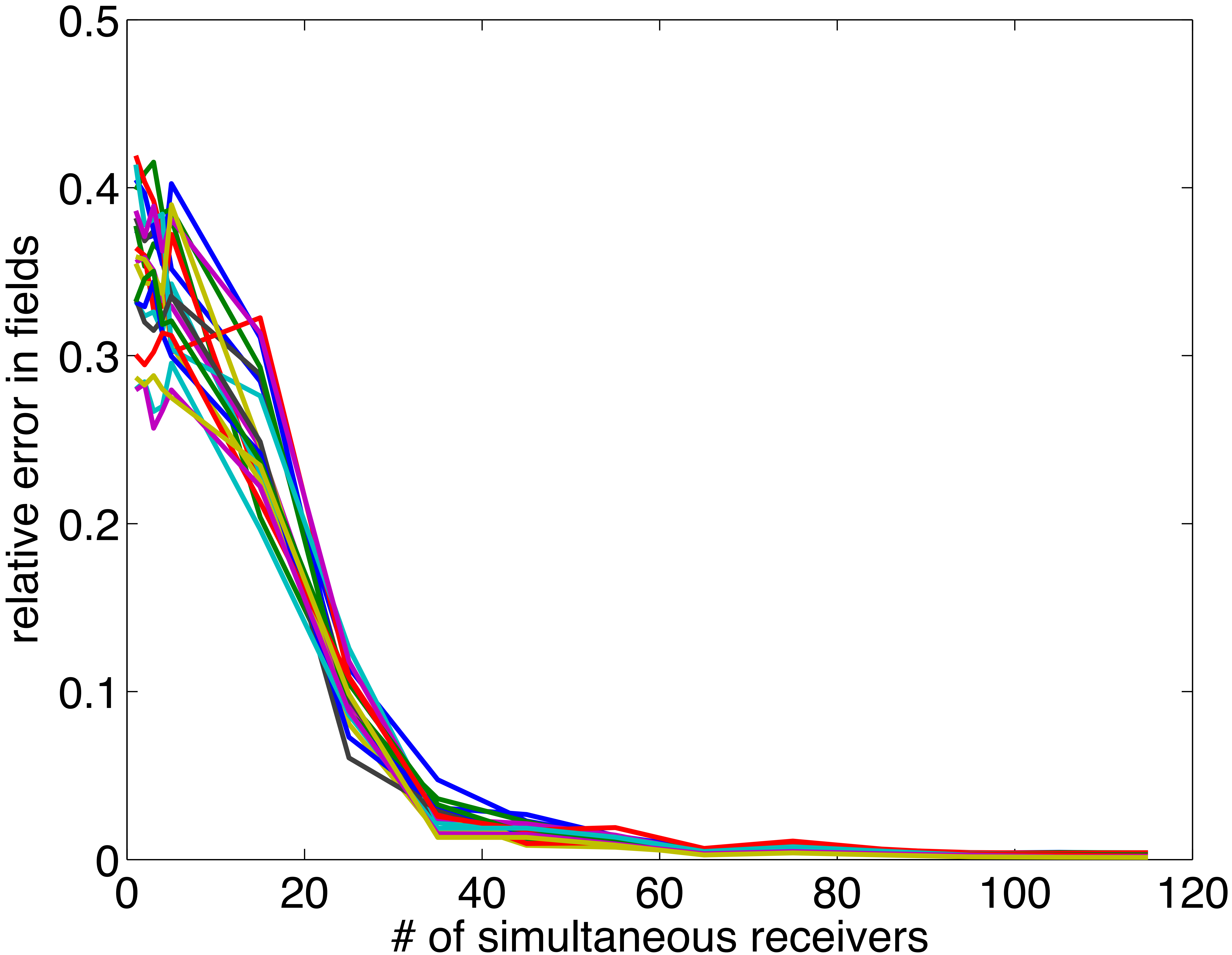


Simultaneous receivers

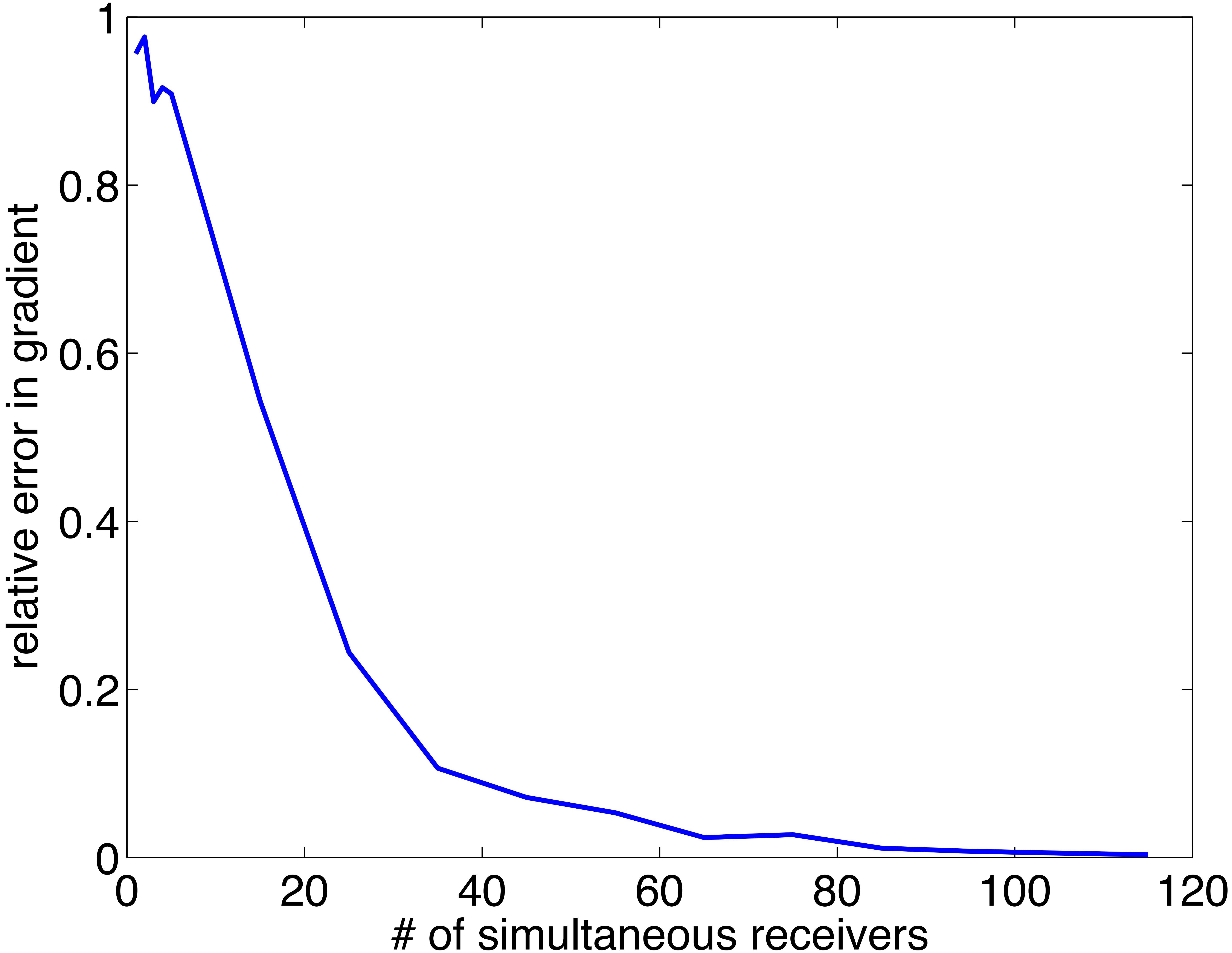




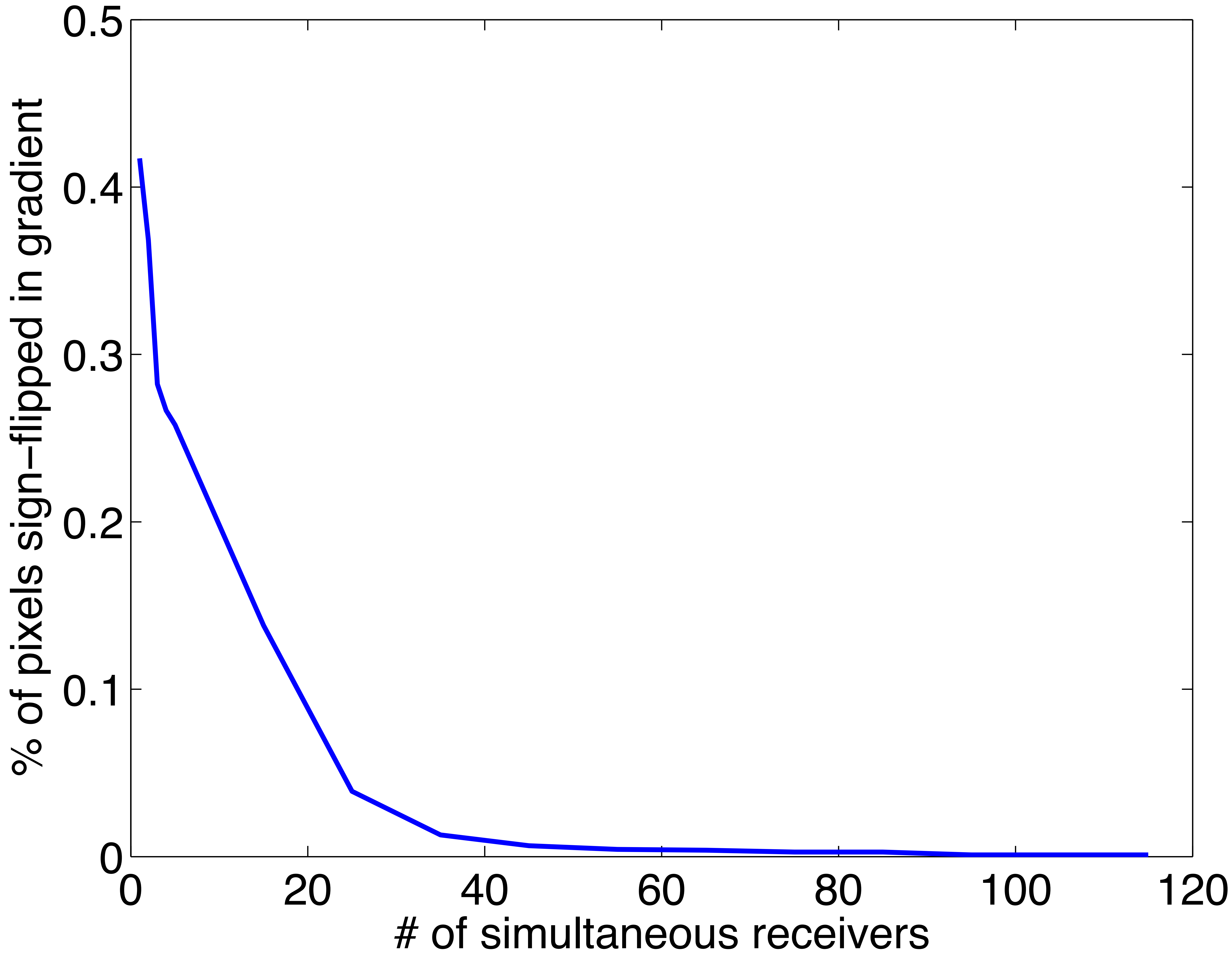
Simultaneous receivers -
Relative error in each of the 20 fields



Simultaneous receivers -
Relative error in the gradient



Simultaneous receivers -
pixels in the gradient with incorrect sign



Simultaneous receivers

Simultaneous receivers can

- reduce the number of PDE solves and
- memory

without losing much accuracy.

Performance depends on model, frequency, sources & receivers.

Parallel implementation

The basic parallel flow is:

1. solve PDE’s in parallel
2. Sherman-Morrison using distributed arrays (communication intensive):

solve $$(I + WW^*)\mathbf{y} = \lambda\mathbf{q} + W\mathbf{d}$$

as $$\mathbf{y} = (I - W(I + W^*W)^{-1}W^*)(\lambda\mathbf{q} + W\mathbf{d})$$

3. solve PDE’s in parallel

Parallel implementation

Helmholtz problems are solved using CGMN.

On each compute node, CGMN solves for multiple sources simultaneously using multi-threading implemented in C.

Store only 1 system matrix per node. (Stencil based matrix-free mat-vec and Kaczmarz sweep are work in progress by Curt).

Requires 1 Matlab worker per node.

Everything except the Helmholtz solver uses Matlab Parallel Computing toolbox.

Compute node 1

Compute node 2

Solve $H^*\mathbf{w_1} = \mathbf{p_1}^*$

Solve $H^*\mathbf{w_2} = \mathbf{p_2}^*$

distributed array (no gathers): $W = (\mathbf{w_1}, \mathbf{w_2})$

Compute (communication):

$$\mathbf{y}_1 = (I - W(I + W^*W)^{-1}W^*)(\lambda\mathbf{q}_1 + W\mathbf{d}_1)$$

Compute (communication):

$$\mathbf{y}_2 = (I - W(I + W^*W)^{-1}W^*)(\lambda\mathbf{q}_2 + W\mathbf{d}_2)$$

Solve $H\bar{\mathbf{u}}_1 = \mathbf{y}_1$

Solve $H\bar{\mathbf{u}}_2 = \mathbf{y}_2$

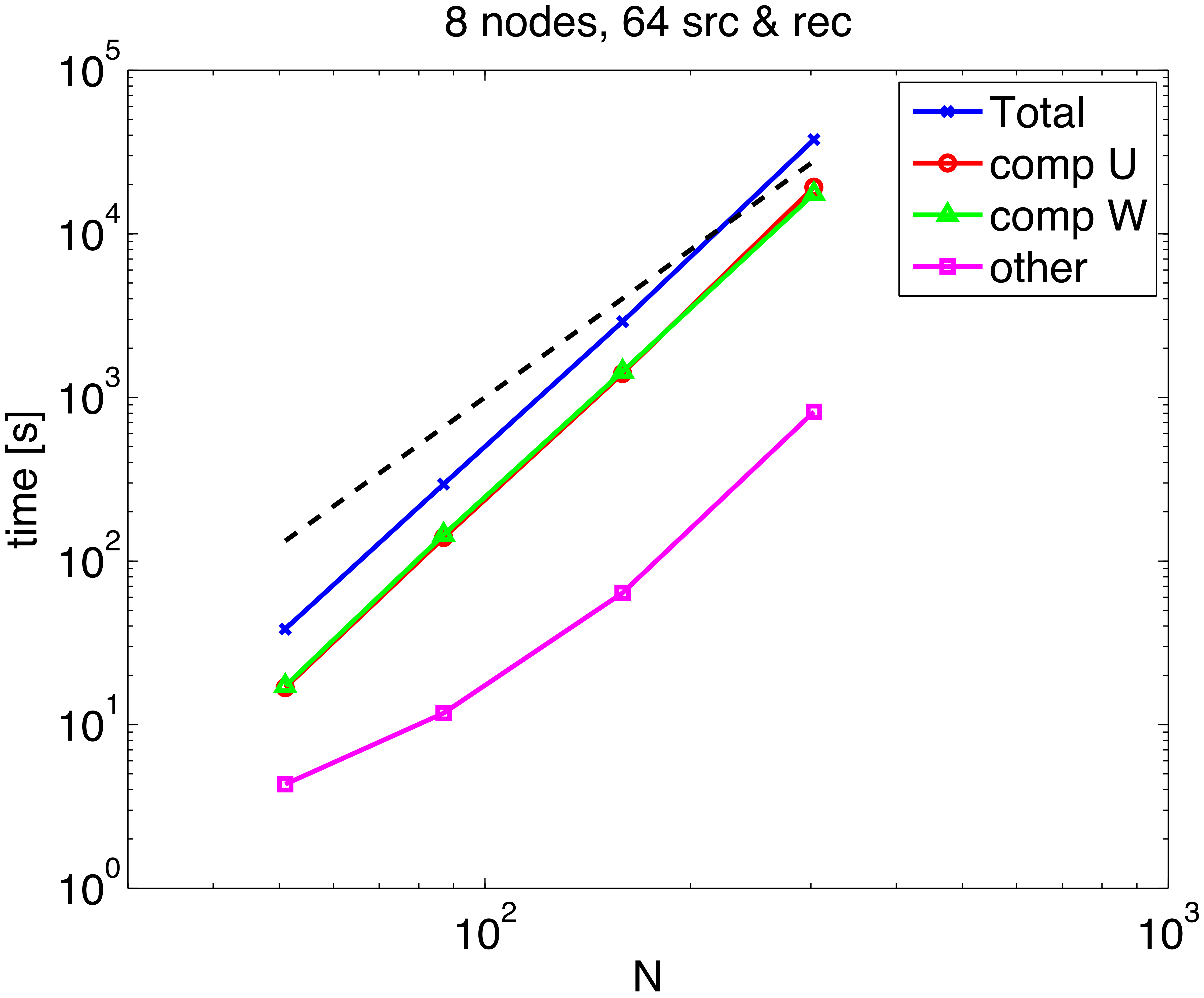
Parallel implementation

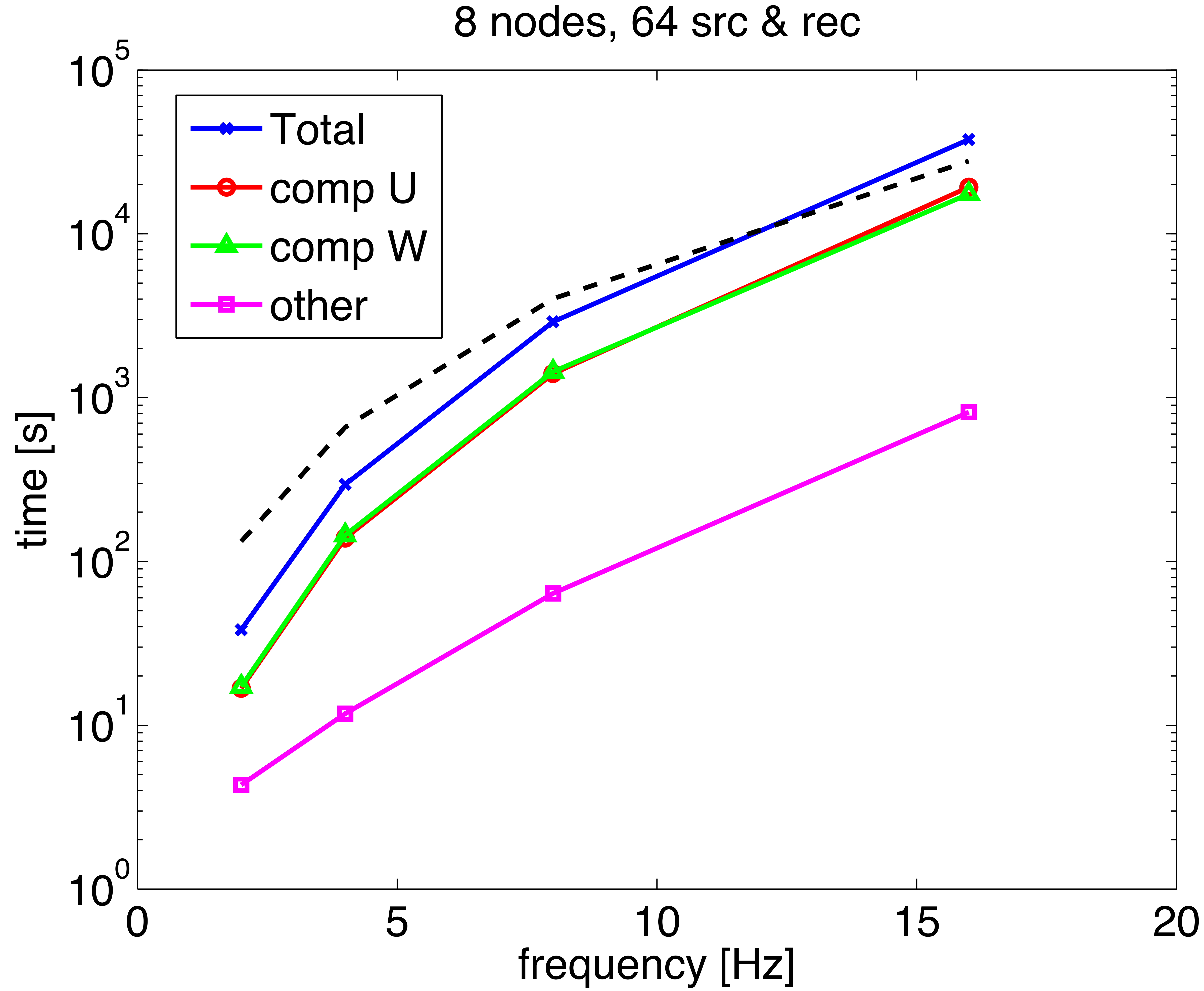
More nodes allow the Helmholtz problems to be solved in less total time.

How does the communication scale?

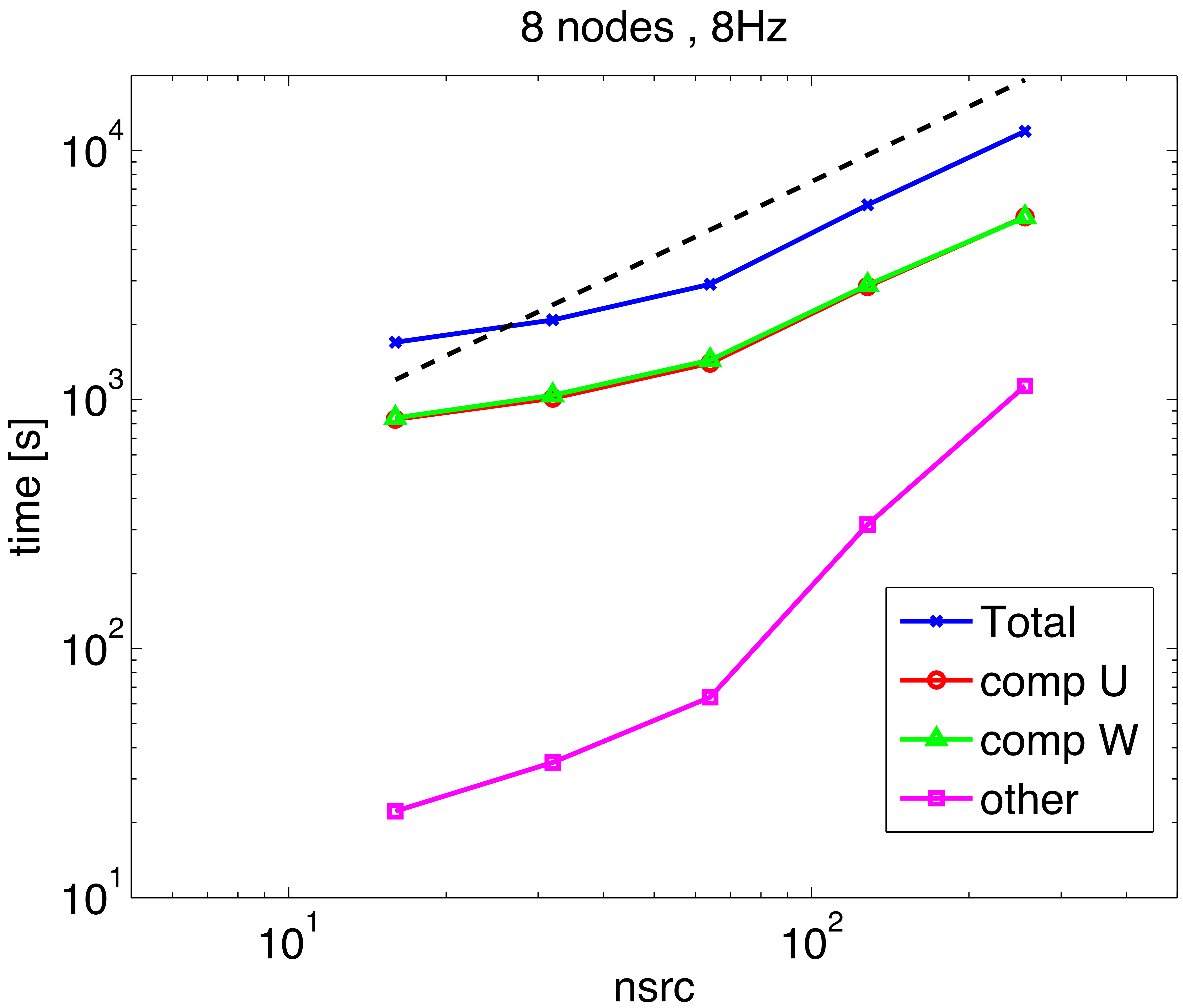
Scaling

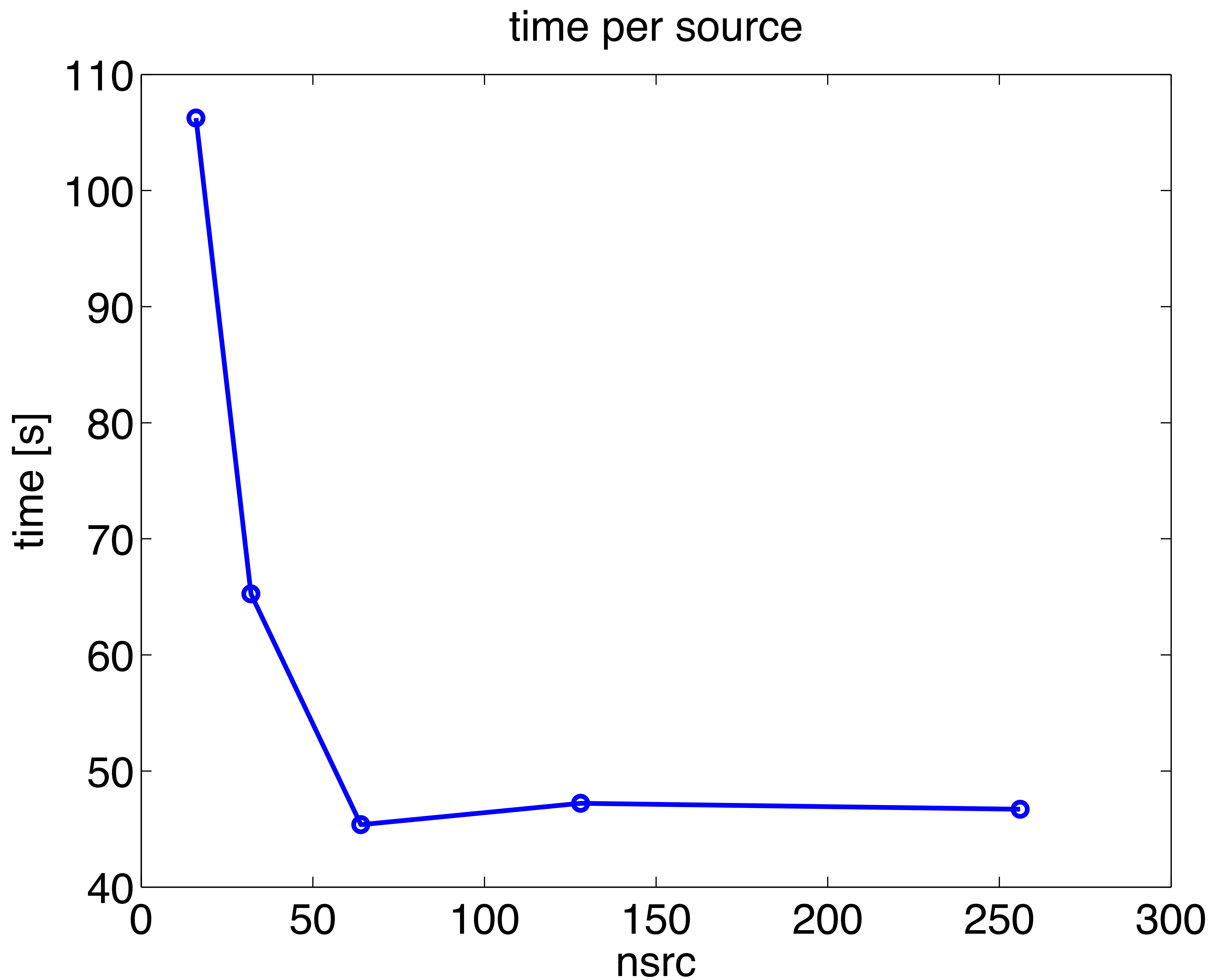
- 6 km cube model
- ~40 wavelengths propagated between source & receiver
- 8 nodes
- Each node solves the PDE's in the sub-problems for 8 right-hand-sides simultaneously.
- This setup can process 8 x 8 = 64 PDE solves simultaneously.
- Fixed tolerance for all PDE solves.



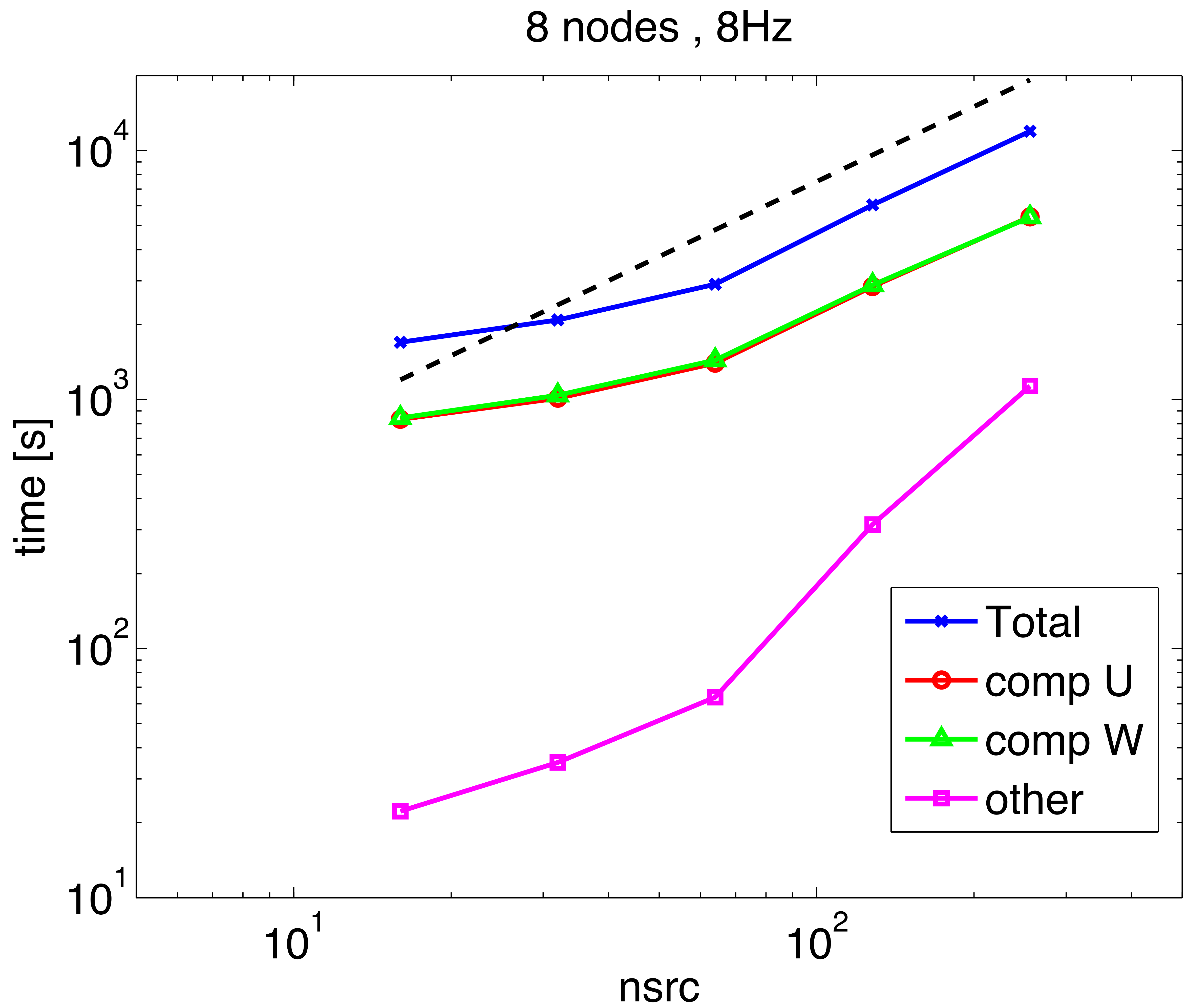


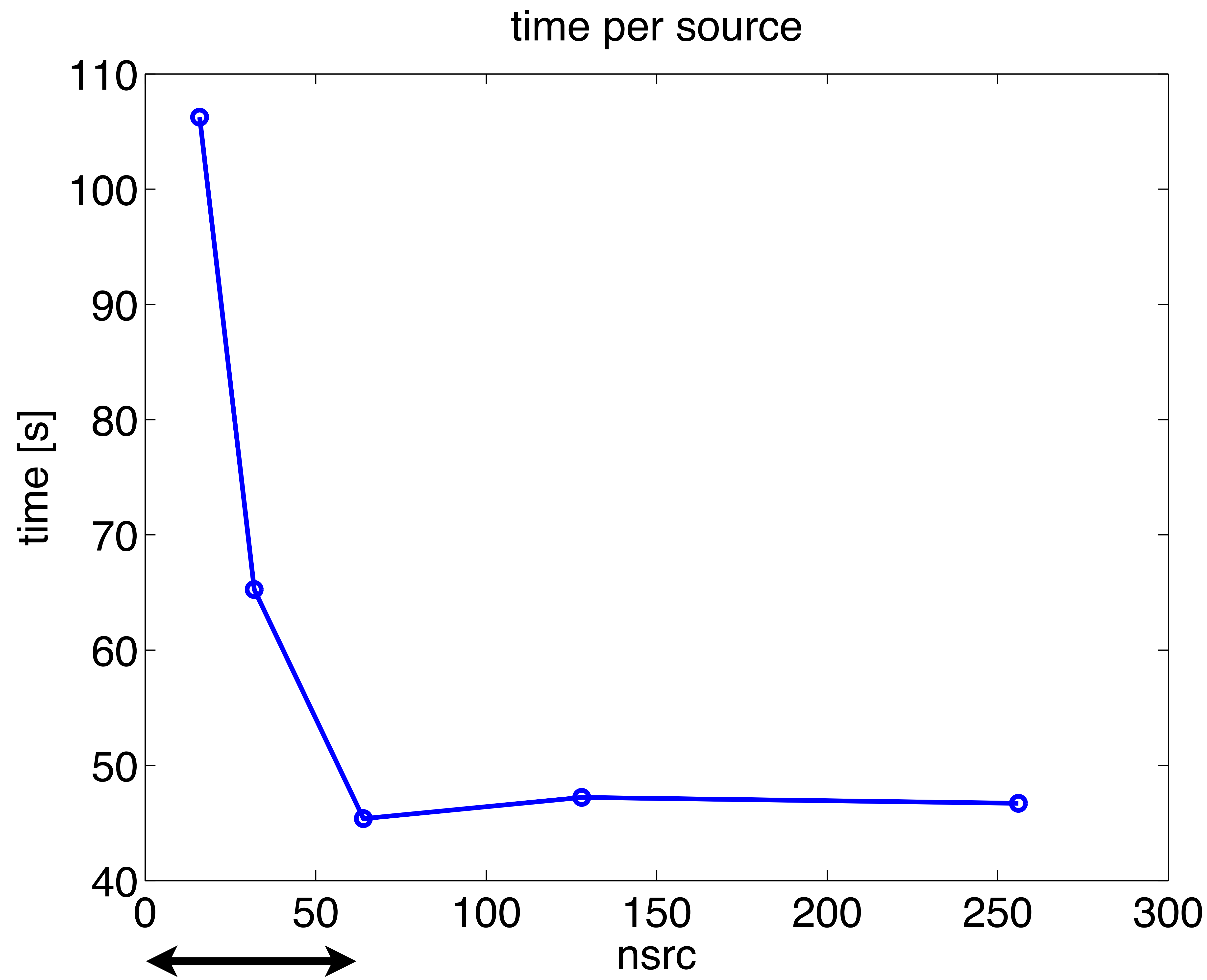
64 sources, 64 receivers. Varying frequency & number of grid points



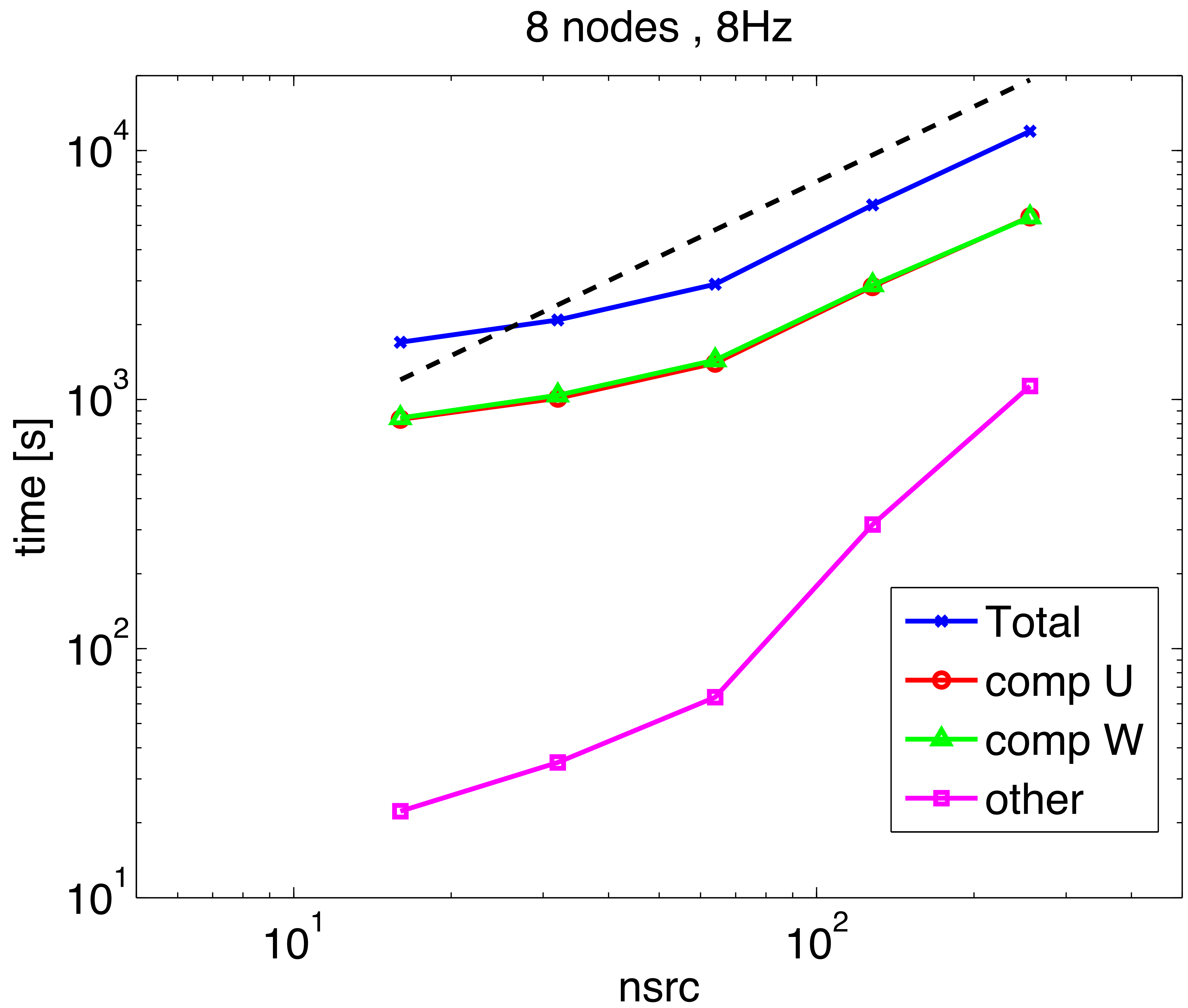


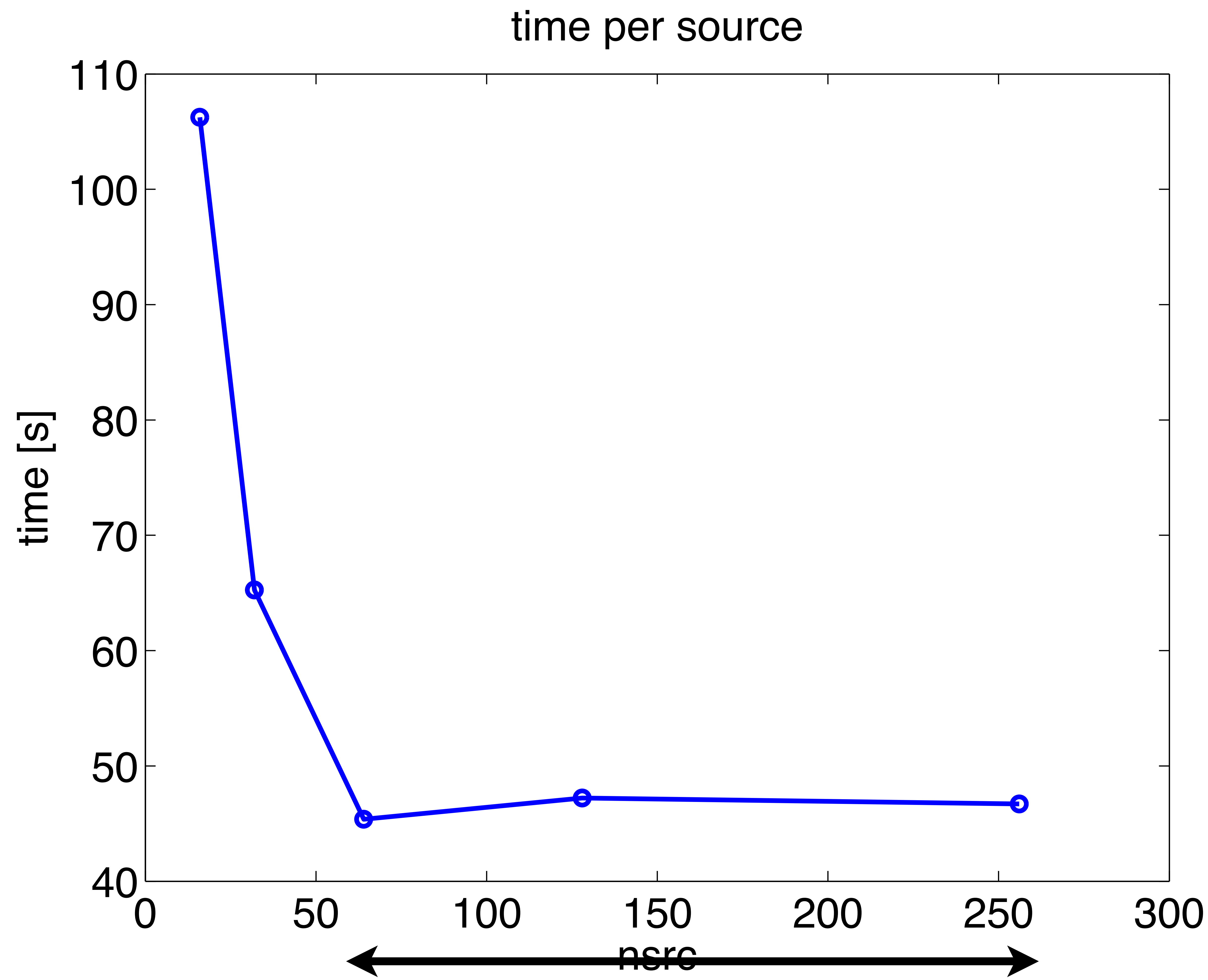
8 Hz. Varying number of sources & receivers (8 - 256).



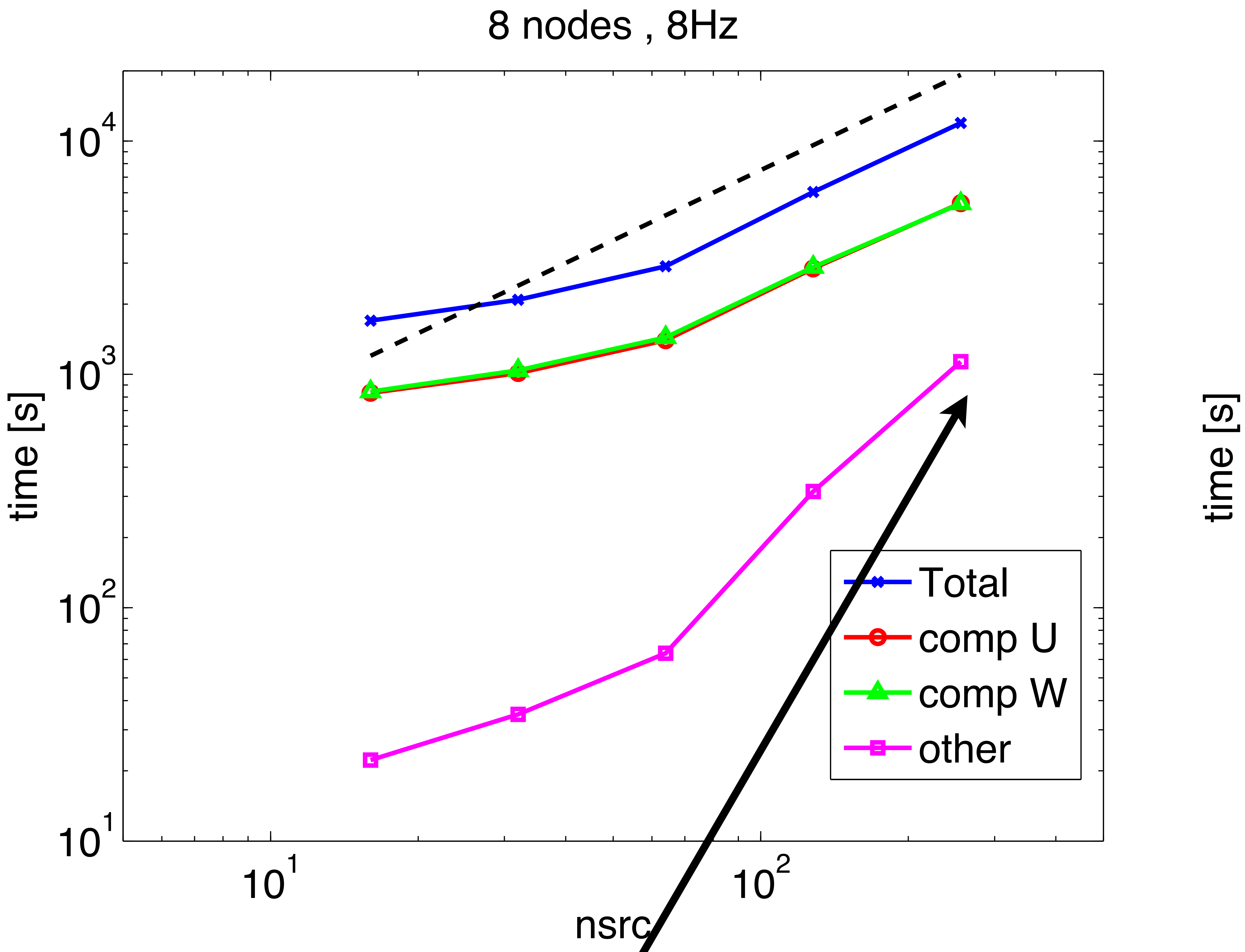


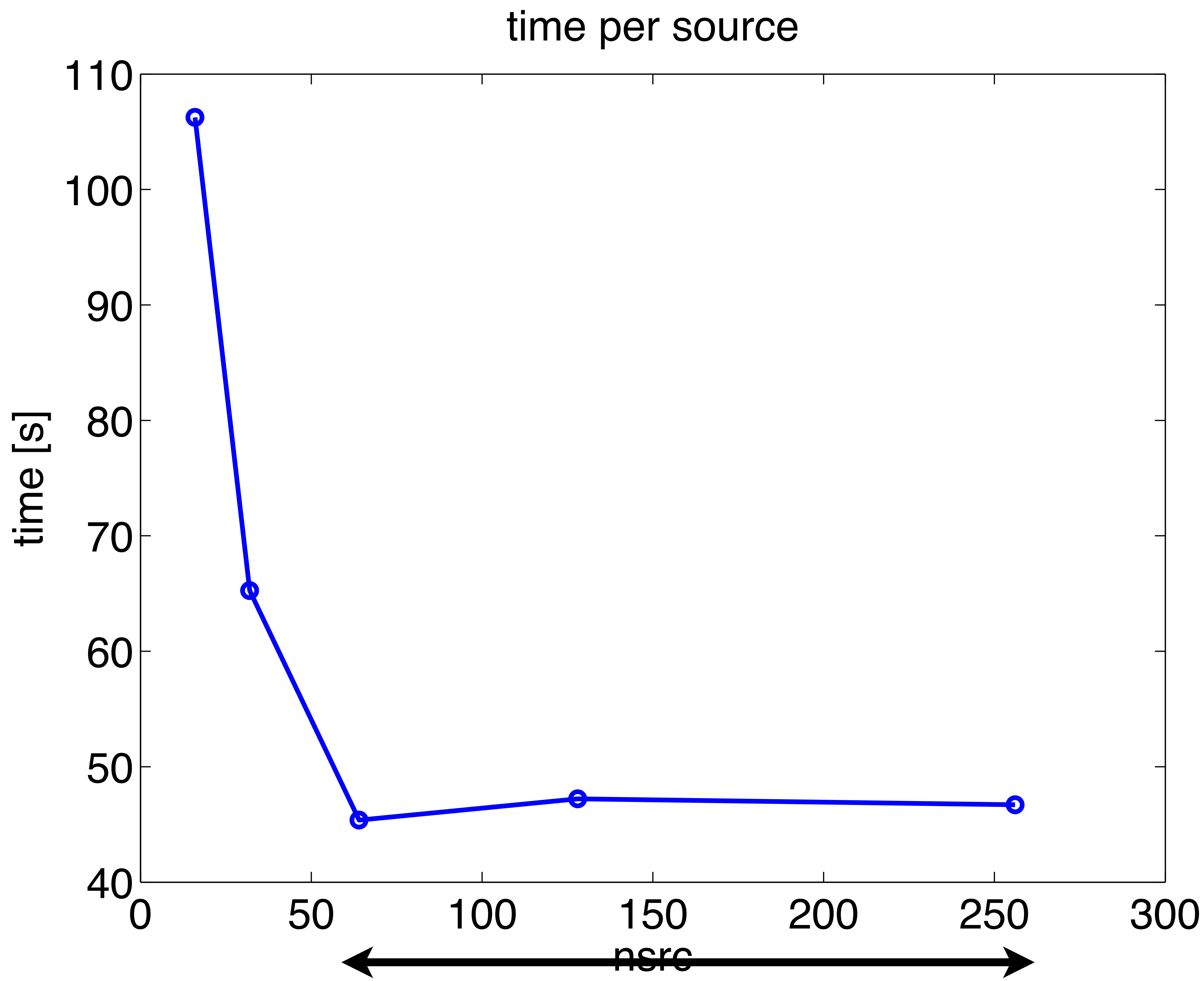
not enough sources & receivers
to use computational capacity of the nodes





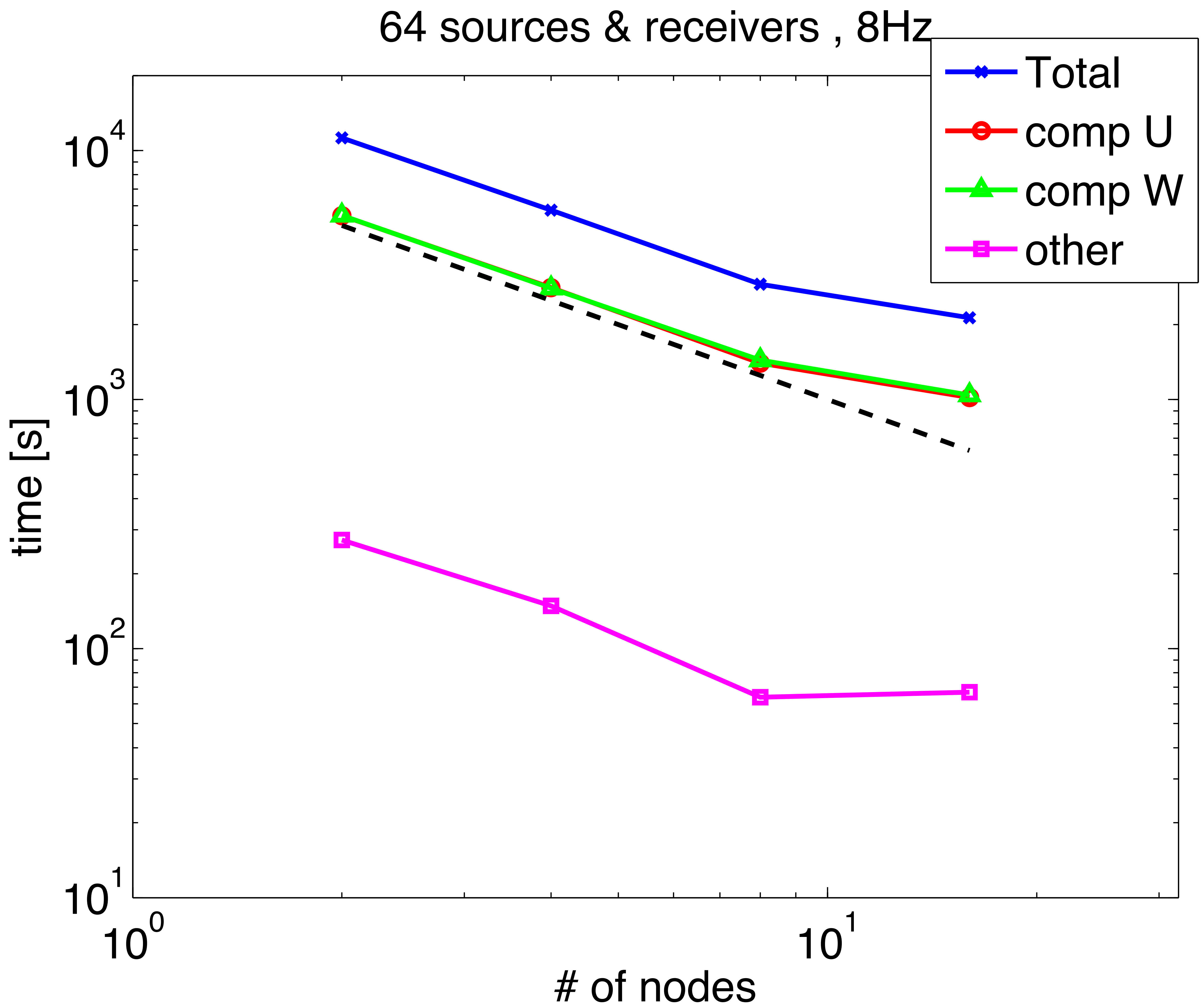
more sources & receivers,
still close to constant time per source



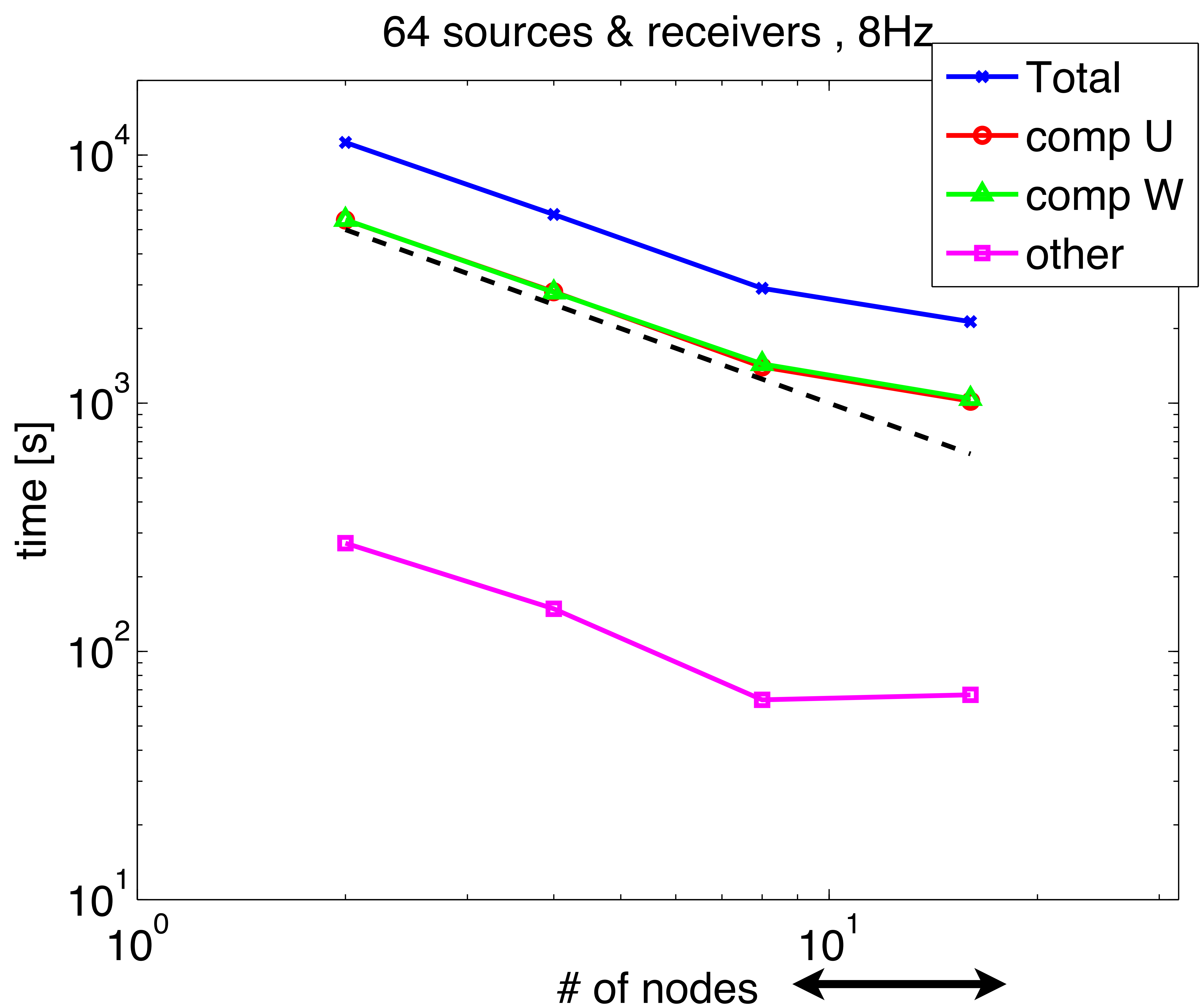


other costs (including communication) increase, but remain relatively small

more sources & receivers, still close to constant time per source



8 Hz. 64 sources & 64 receivers. Varying number of nodes (2 - 16).



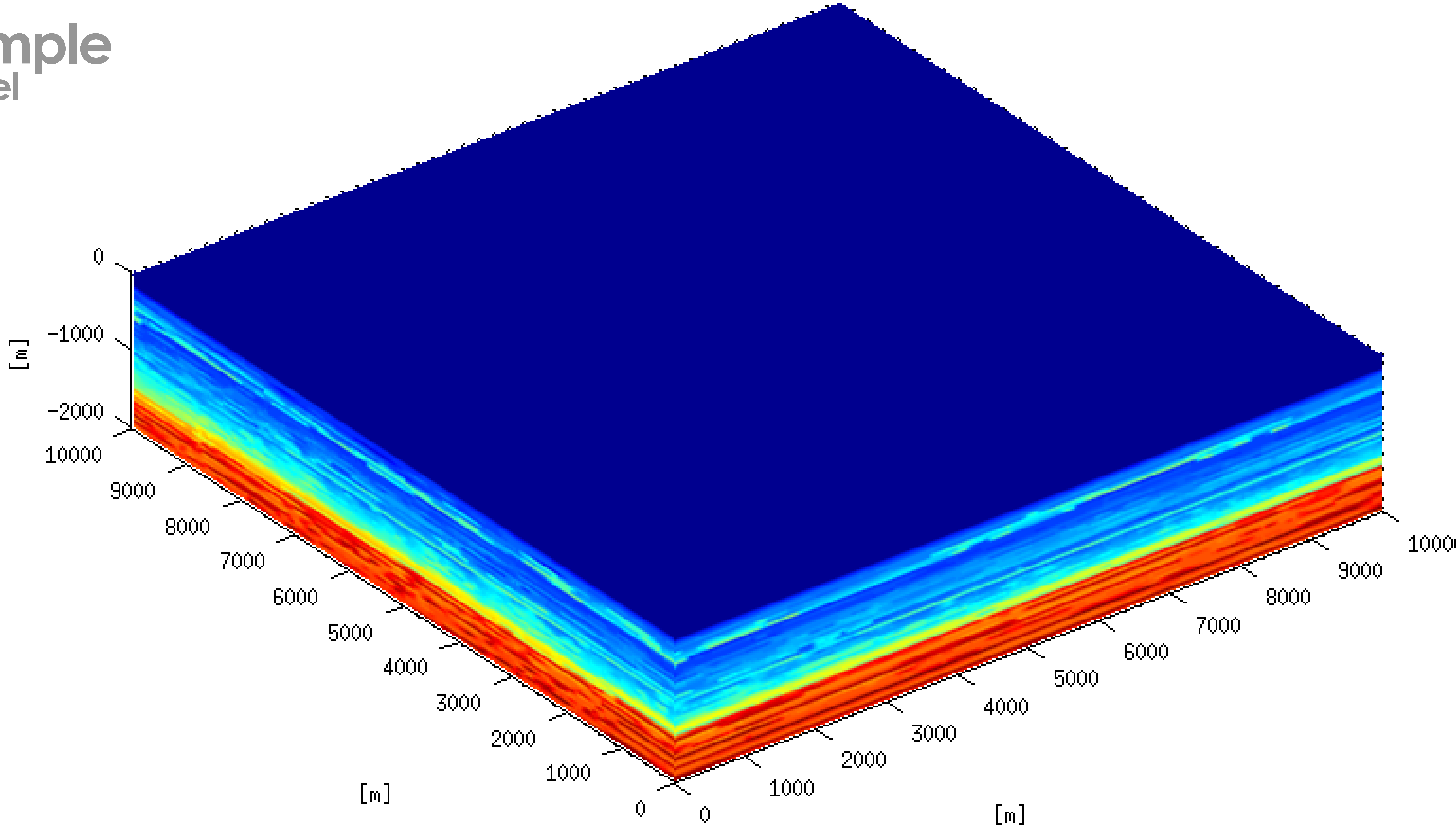
not enough sources & receivers to use computational capacity of the nodes; results in smaller speedup

Scaling

Results depend on the number PDE's solved simultaneously on each node using multithreading.

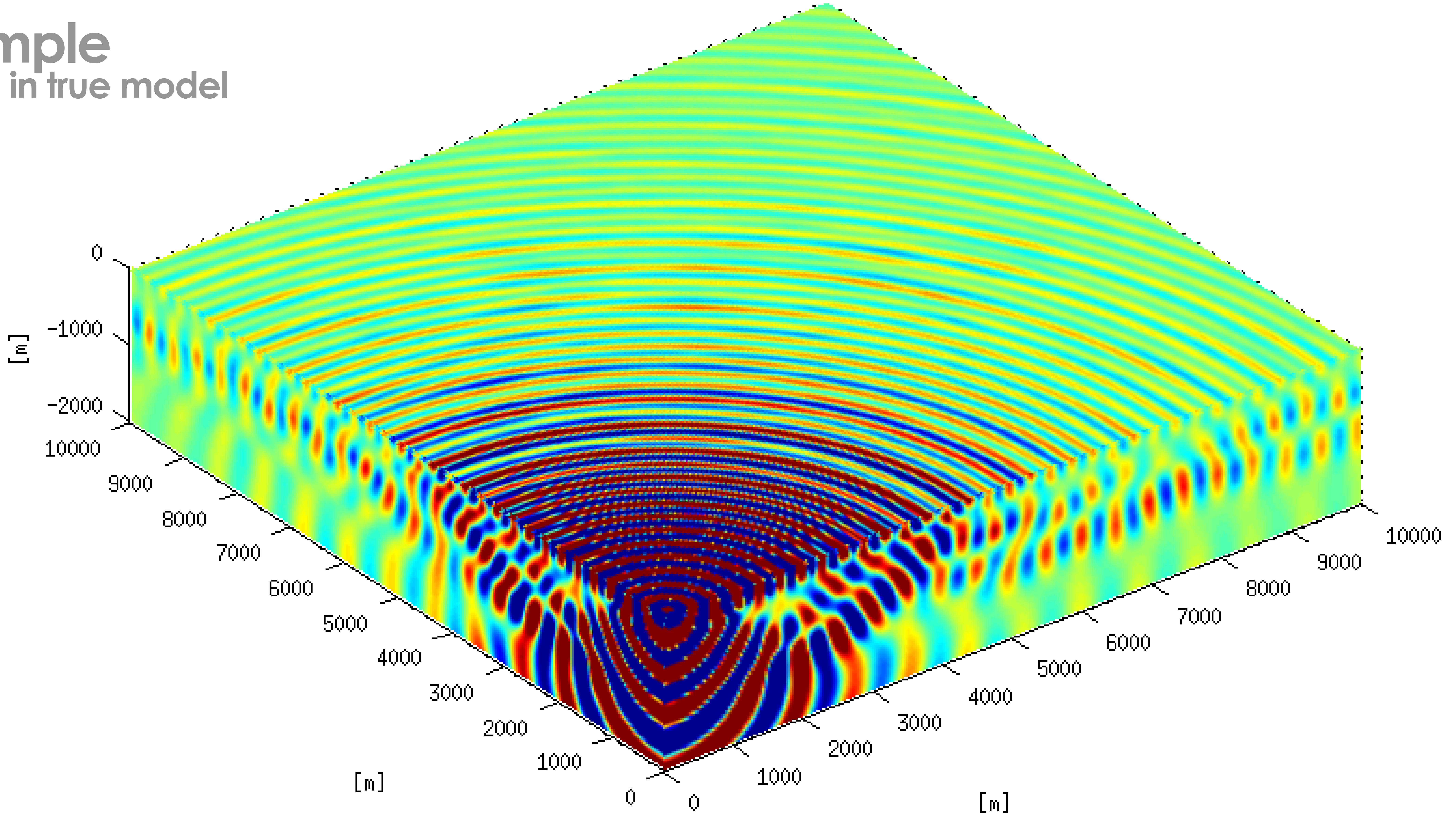
More than 8 is possible, but uses more memory.

3D Example
– true model

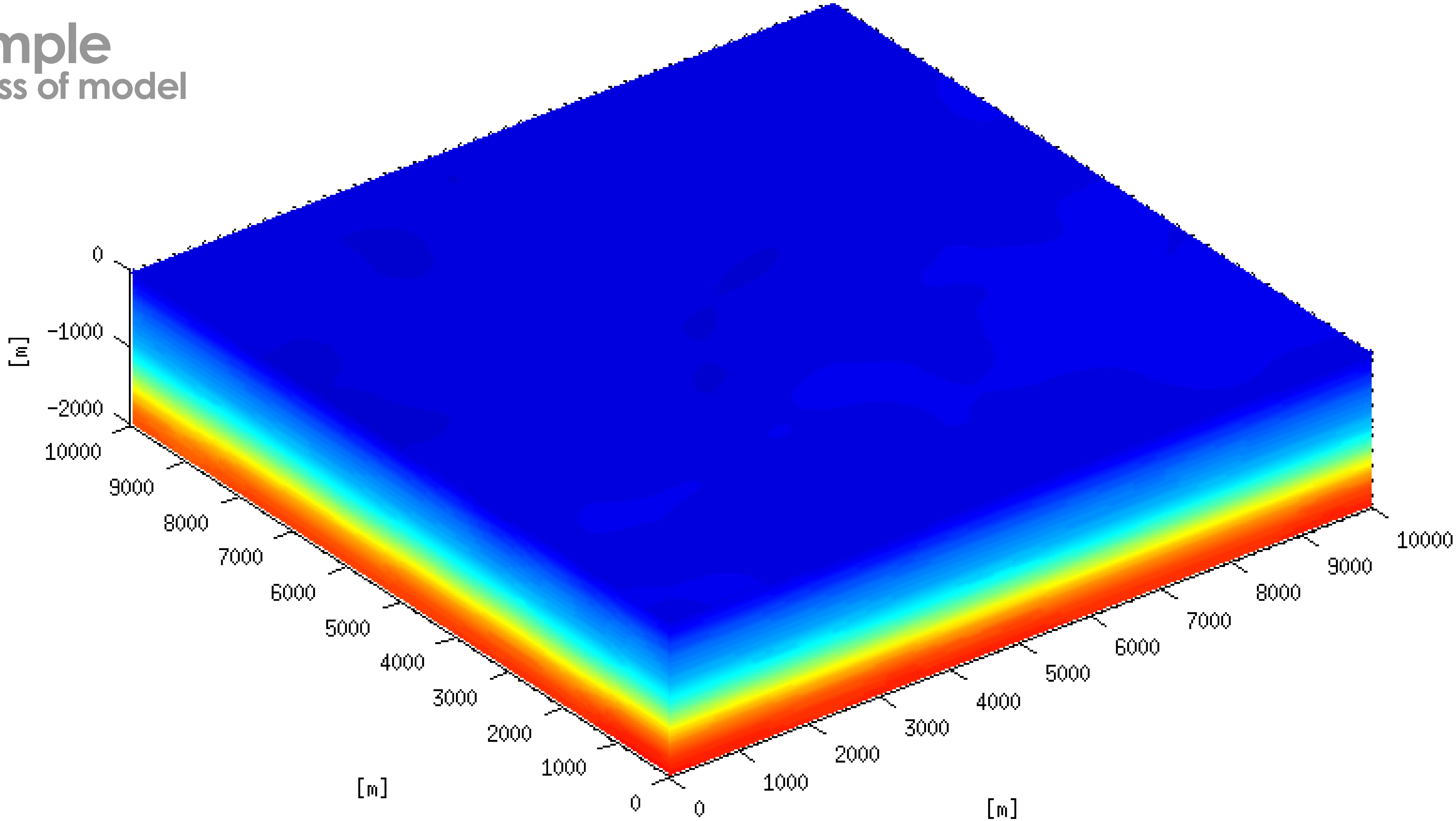


10 x 10 x 2 km, 5 Hz, 27-point discretization , source at [0,0,0]

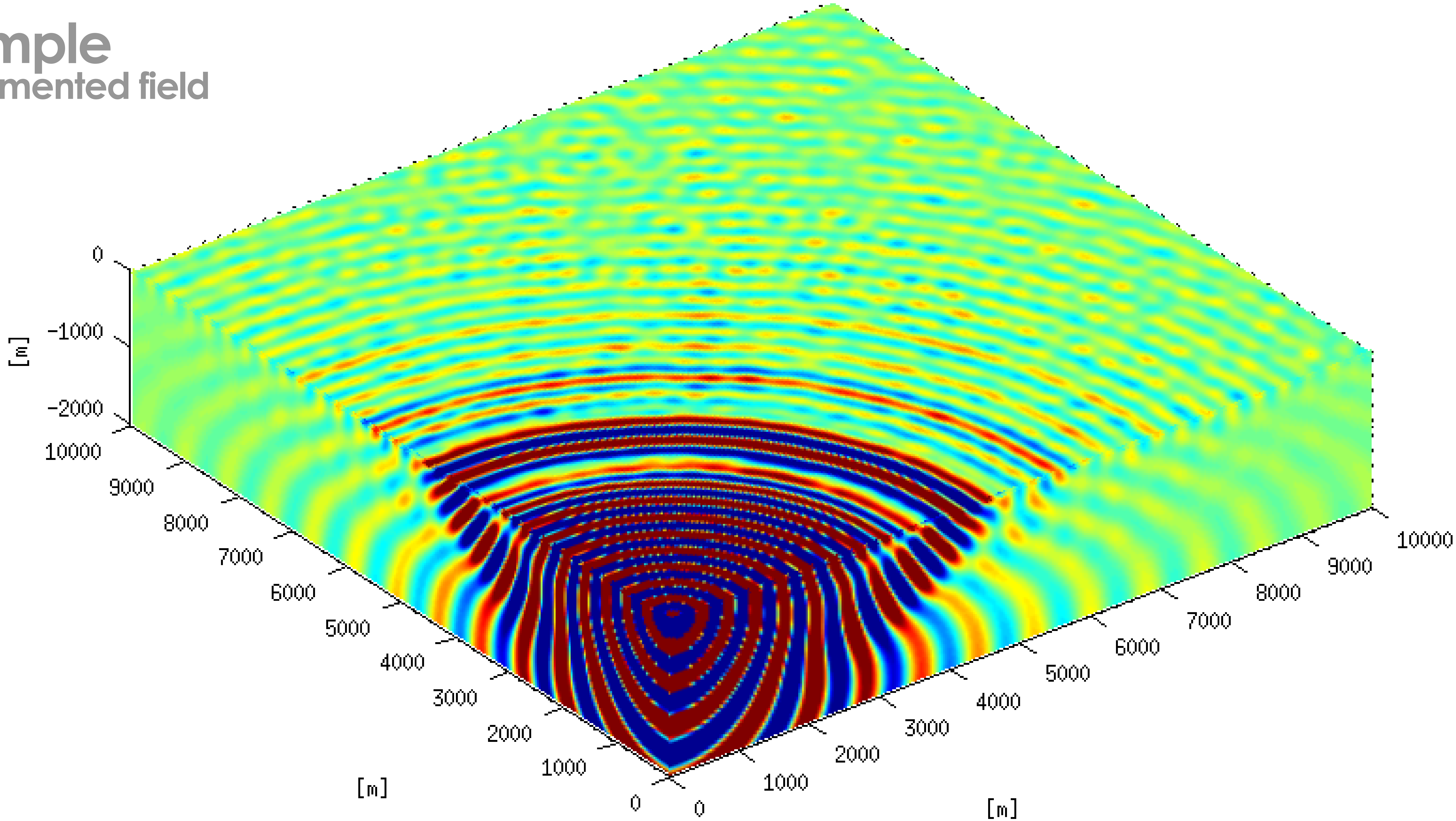
3D Example
– wavefield in true model



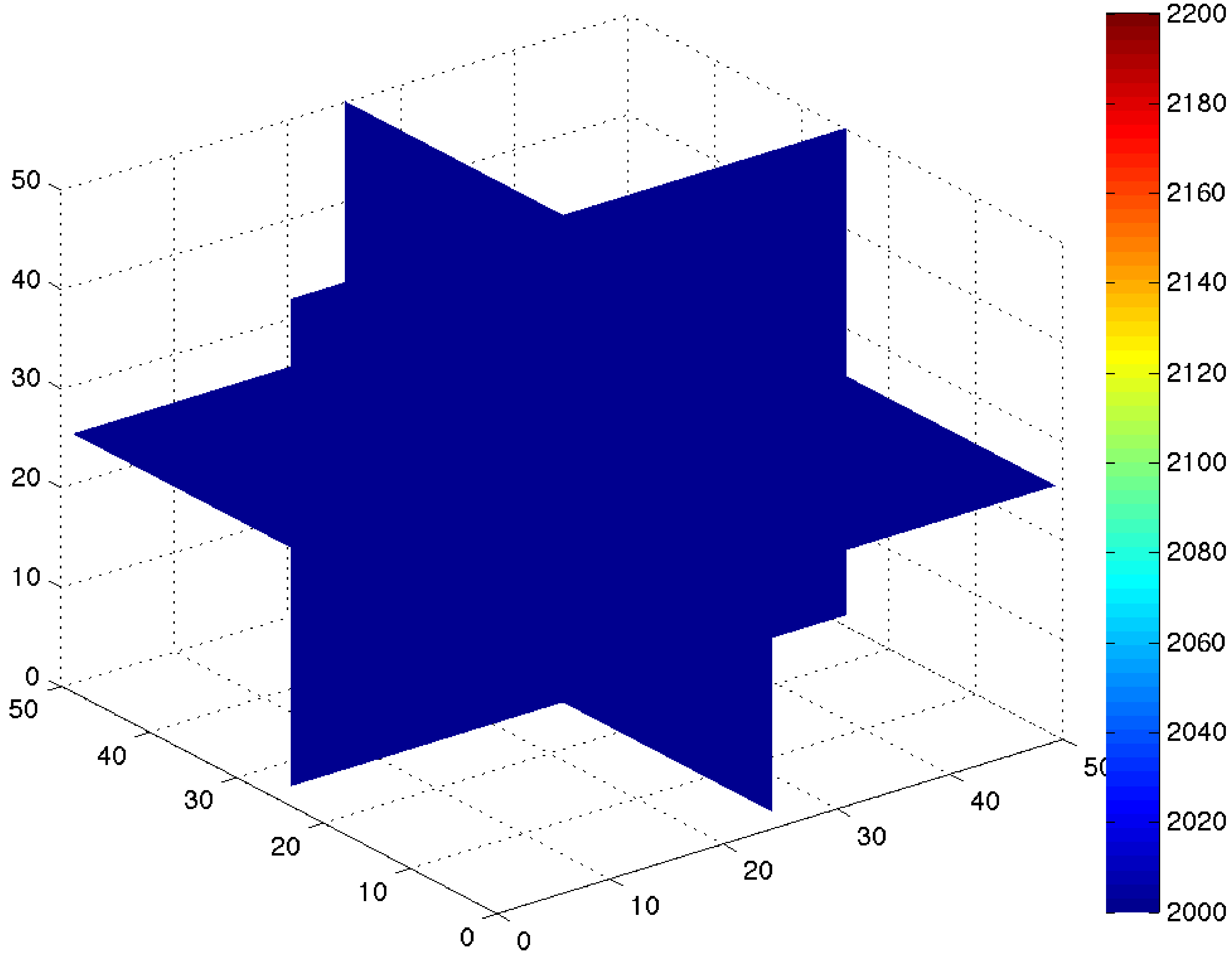
3D Example
– initial guess of model

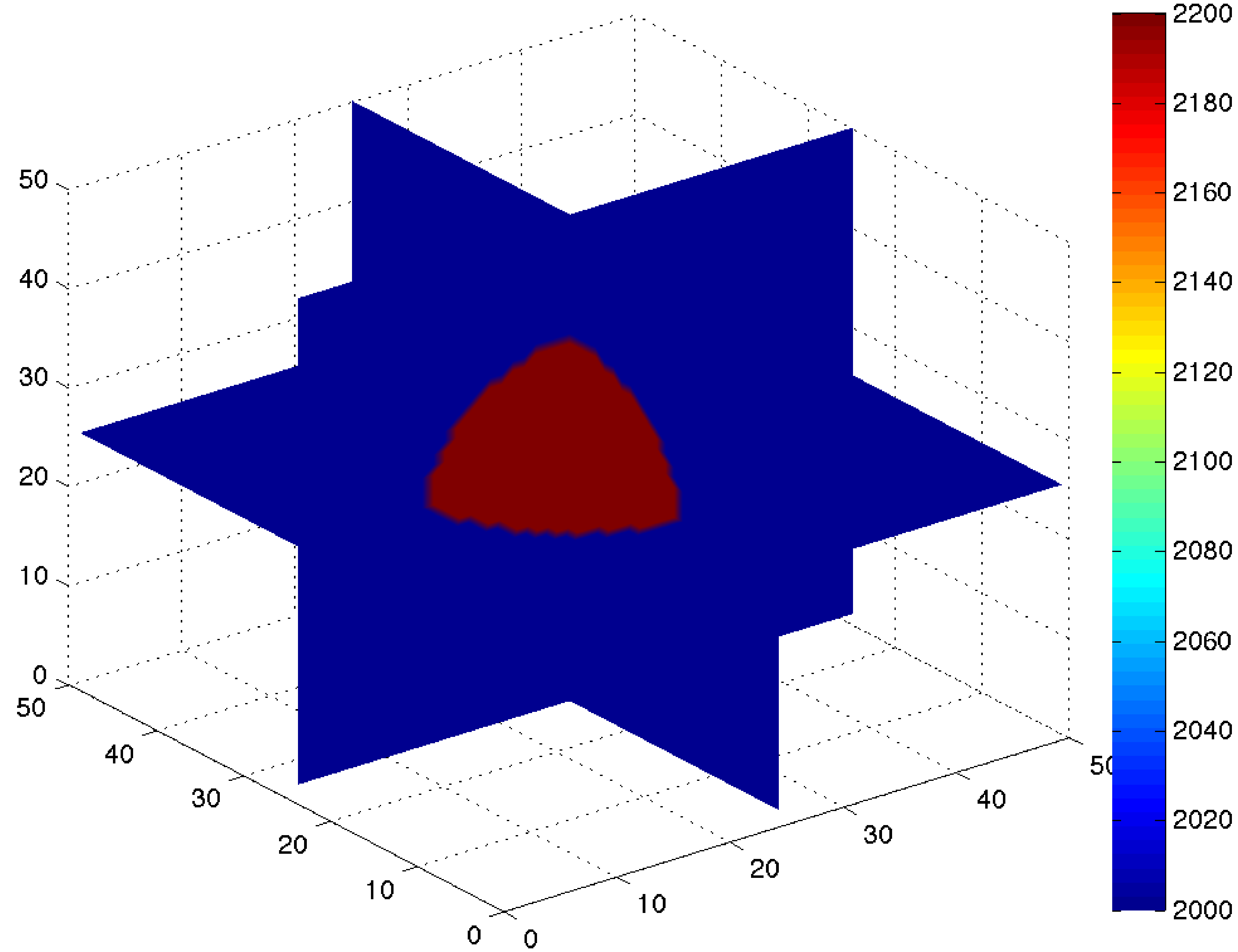


3D Example
– data-augmented field

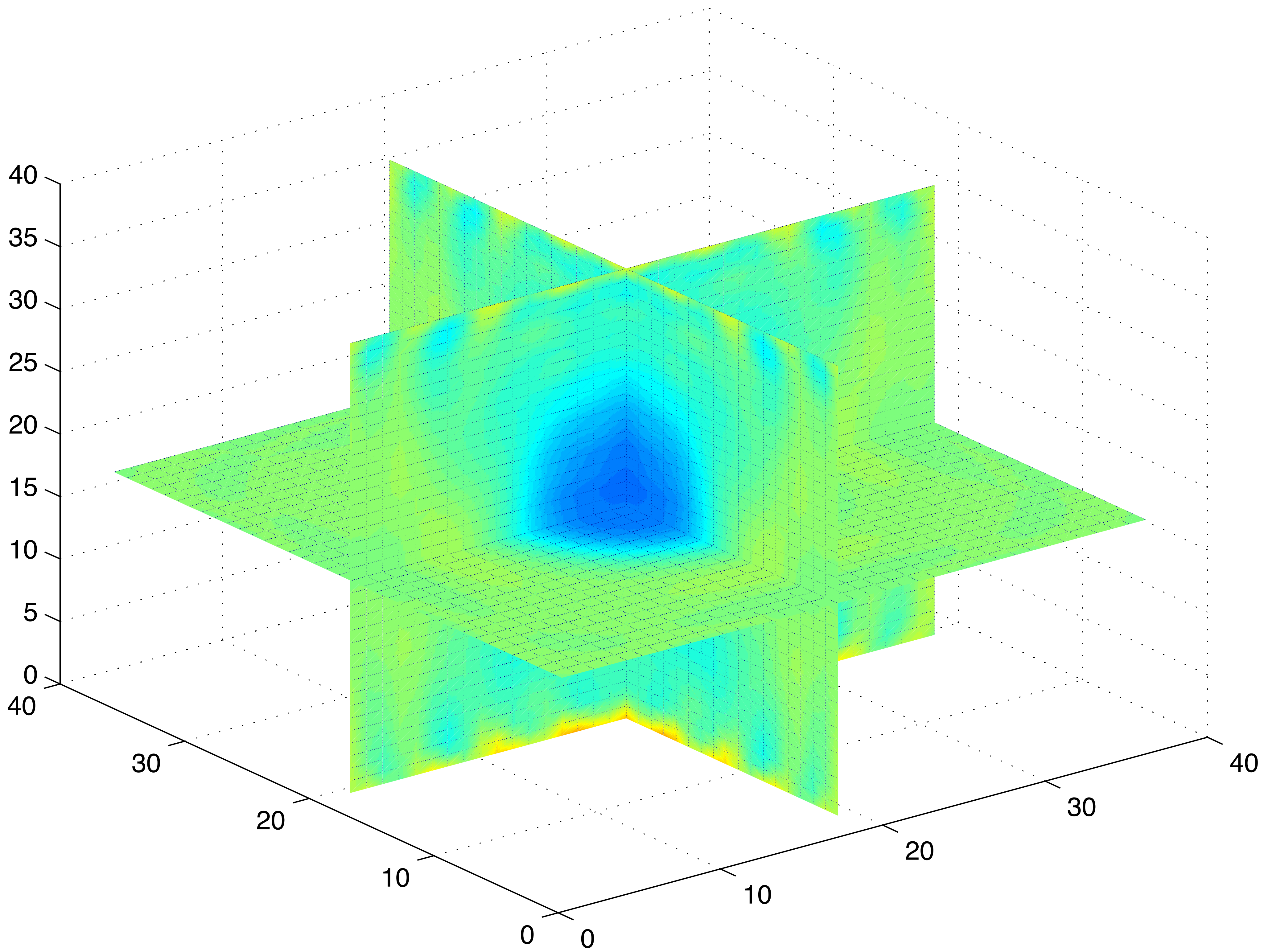


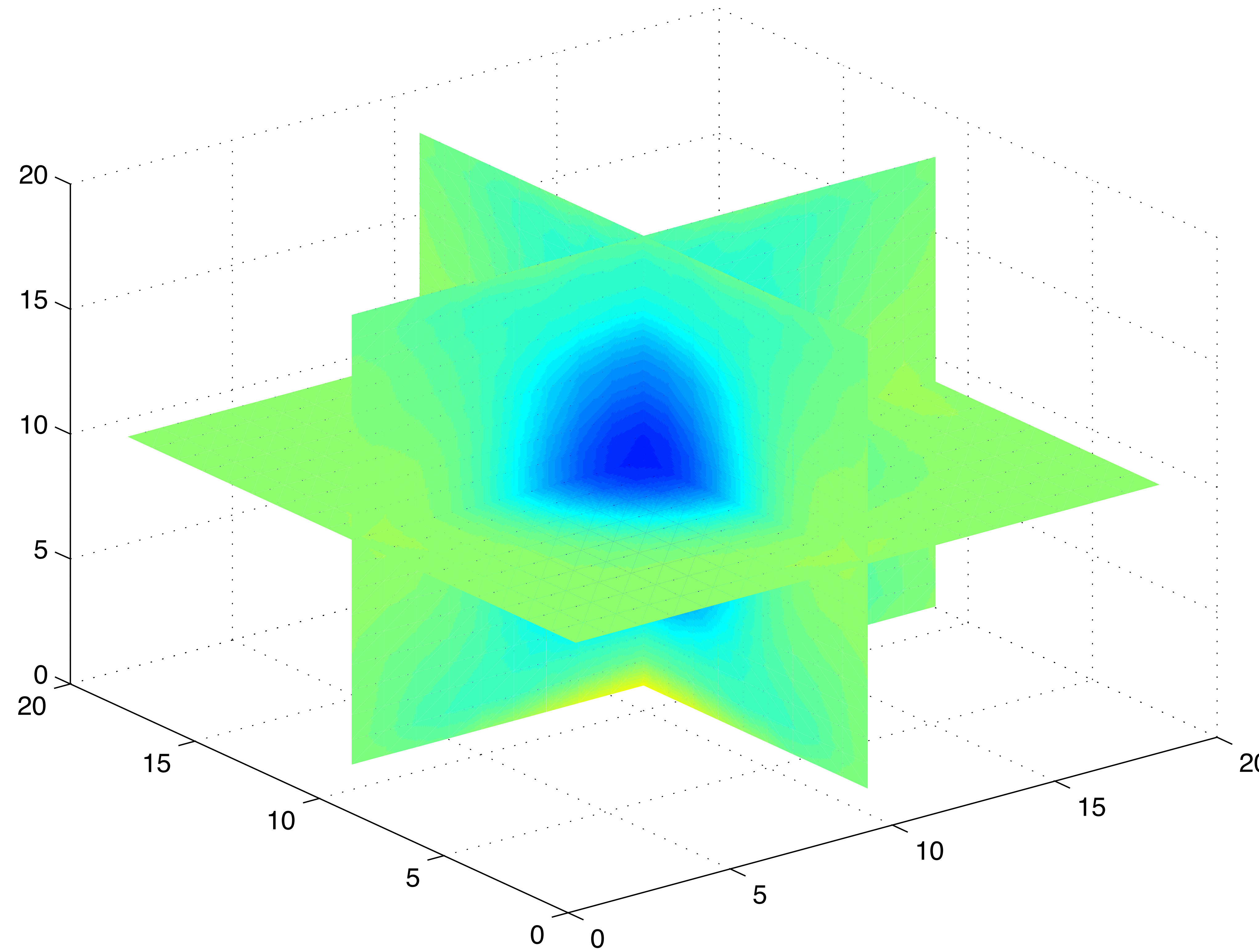
True and Initial model
8 sources at the top, 8 receivers at the bottom





gradients at 4Hz and 2Hz





Conclusions

All tools to use WRI in 3D are available.

This algorithm can also be used for other applications.

(tomorrow, 10:45—11:10 AM, Bas Peters, A quadratic-penalty full-space method for waveform inversion)

Accepts any Helmholtz solver for the sub-problems.

Compute 1 Helmholtz problem inexactly per source and 1 per receiver.

Store 1 vector per receiver.

Can use simultaneous receivers to reduce computational cost and memory use.

Currently in progress...

Used native Matlab implementation for several computations with distributed arrays, for example W^*W .

$W \in \mathbb{C}^{n_{\mathrm{grid}} \times n_{\mathrm{rec}}}$ is very tall and formed distributed over the columns.

Currently testing memory optimized (slow) and time optimized (heavy on memory) implementations of this operation.

Acknowledgements

Tristan van Leeuwen, Art Petrenko, Rafael Lago, Mathias Louboutin & Curt da Silva for the CGMN & CARP-CG implementation

SINBAD

NSERC CRSNG

This work was financially supported by SINBAD Consortium members BG Group, BGP, CGG, Chevron, ConocoPhillips, DownUnder GeoSolutions, Hess, Petrobras, PGS, Schlumberger, Statoil, Sub Salt Solutions and Woodside; and by the Natural Sciences and Engineering Research Council of Canada via NSERC Collaborative Research and Development Grant DNOISEII (CRDPJ 375142--08).

References

1. A. Bjorck and T. Elfving, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations, BIT, 19 (1979), pp. 145–163.
2. D. Gordon and R. Gordon, CARP-CG: A robust and efficient parallel solver for linear systems, applied to strongly convection dominated pdes, Parallel Computing, 36 (2010), pp. 495– 515.
3. Calandra, H., Gratton, S., Pinel, X. and Vasseur, X. [2013] An improved two-grid preconditioner for the solution of three-dimensional Helmholtz problems in heterogeneous media. Numerical Linear Algebra with Applications.
4. Erlangga, Y.A. [2008] Advances in iterative methods and preconditioners for the Helmholtz equation. Archives of Computational Methods in Engineering, 15, 37–66.
5. Delves, L. M., and I. Barrodale. "A fast direct method for the least squares solution of slightly overdetermined sets of linear equations." IMA Journal of Applied Mathematics 24.2 (1979): 149-156.
6. Rafael Lago, Felix J. Herrmann. Towards a robust geometric multigrid scheme for {Helmholtz} equation, Tech Report, UBC, 2015.
7. Avron, Haim, Petar Maymounkov, and Sivan Toledo. "Blendenpik: Supercharging LAPACK's least-squares solver." SIAM Journal on Scientific Computing 32.3 (2010): 1217-1236.
8. Meng, Xiangrui, Michael A. Saunders, and Michael W. Mahoney. "LSRN: A parallel iterative solver for strongly over-or underdetermined systems." SIAM Journal on Scientific Computing 36.2 (2014): C95-C118.
9. Van Den Berg, Peter M., and Ralph E. Kleinman. "A contrast source inversion method." Inverse problems 13.6 (1997): 1607.
10. Abubakar, Aria, Peter M. Van den Berg, and Jordi J. Mallorqui. "Imaging of biomedical data using a multiplicative regularized contrast source inversion method." Microwave Theory and Techniques, IEEE Transactions on 50.7 (2002): 1761-1771.
11. Leeuwen, Tristan van, and Felix J. Herrmann. 2013a. “Mitigating Local Minima in Full-Waveform Inversion by Expanding the Search Space.” Geophysical Journal International 195: 661–67.