Prospects for Seismic Inversion on Petaflop Systems

Carsten Burstedde
Institute for Computational Engineering & Sciences
The University of Texas at Austin

Research Group UT-Austin
Omar Ghattas
Seong-Won Na, Georg Stadler, Lucas Wilcox, Pearl Flath

Collaborators
Volkan Akcelik (SLAC), Aysegul Askan (CMU), Jacobo Bielak (CMU),
George Biros (Penn), Andrei Draganescu (UMBC),
Ioannis Epanomeritakis, Judy Hill (Sandia),
Loukas Kallivokas (UT), Eui Joong Kim (Samsung),
Julio Lopez (CMU), Kwan-Liu Ma (UC-Davis),
David O'Hallaron (CMU), Leonardo Ramirez-Guzman (CMU),
Ricardo Taborda (CMU), Tiankai Tu (CMU), Bart van Bloemen
Waanders (Sandia), Karen Willcox (MIT), Hongfeng Yu (UC-Davis)

Prospects for Seismic Inversion on Petaflop Systems

- Prospects
  - Current situation: funding, expectations, challenges
- of Seismic Inversion
  - full waveform inversion in 3D
  - least-squares observation and modifications
  - inexact Newton-Krylov iterative algorithm
- on Petaflop Systems
  - large problems on large clusters of computers
  - a very specialized view on computation:

"Does it scale?"

“Does it scale?” – What is the meaning?

“Strong scaling” – often unrealistic

- Fix problem size (e.g. size of the domain, frequency):
- Runtime decrease in proportion to resource increase?

“Weak scaling” – widely accepted criterion

- Fix problem size per processor (e.g. space/time variables)
- Runtime stays constant while increasing #processors?

Some things that do not scale:

- Hard disk read/write (limited resource, slow bandwidth)
- Dense matrices (assembly, storage, matrix-vector product N^2)
- Matrix inversion (storage N^2, runtime N^3)
- Implicit PDE solvers (limited by network topology, bandwidth)
- Steepest descent minimization (#iterations depend on N)

Prospects for Seismic Inversion on Petaflop Systems

→ This talk is about parallel computing.

Kilo – 1,000 Mega – 1,000,000 Giga – 1,000,000,000

Tera – 1,000,000,000,000 Peta – 1,000,000,000,000,000

Flop – "Floating Point Operation"

Example Earthquake Simulation:

- 600km x 600km x 70km region
- Average velocity 2km/s, Max frequency 2Hz
- Wavelength 1km - 25 million grid boxes
- 10 variables per wavelength - 25 billion variables
- Simulated waves travel 400km - 4000 time steps
- Executing on average 10 arithmetic operations per variable

25 billion x 4000 x 10 = 10^15 → 1 Petaflop

On the Road to Petascale Computation

Identification of possible scientific breakthroughs

- SCaLeS report, Petascale Collaboratory for Geosciences
- President's American Competitiveness Initiative (ACI)

Aggressive programs for petaflops performance

- NSF Track 1 Petascale Acquisition, DOE SC @ ORNL, ANL
- Cost for NSF and DOE/SC Petaflop systems alone: >$1B
- First peak petaflop machine should appear in 2008

Opportunities

- Now it is up to the science & engineering communities to make effective use of these systems
- Many of the grand challenge problems facing society today are related to the geosciences
- High performance computing is "popular" – but for how long?

Performance is Flop per Second: How expensive are sustained Flop/s?

Human
- 0.1 Flop/s $100,000

Laptop
- 10 Gigaflop/s $1,500

Selfmade cluster
- 1 Teraflop/s $200,000

The world's biggest systems in 2011
- 1 Petaflop/s $200,000,000 each

Does it scale (hardware/money)? – Probably.

Does it scale (potential for waste/failure)? – Definitely.

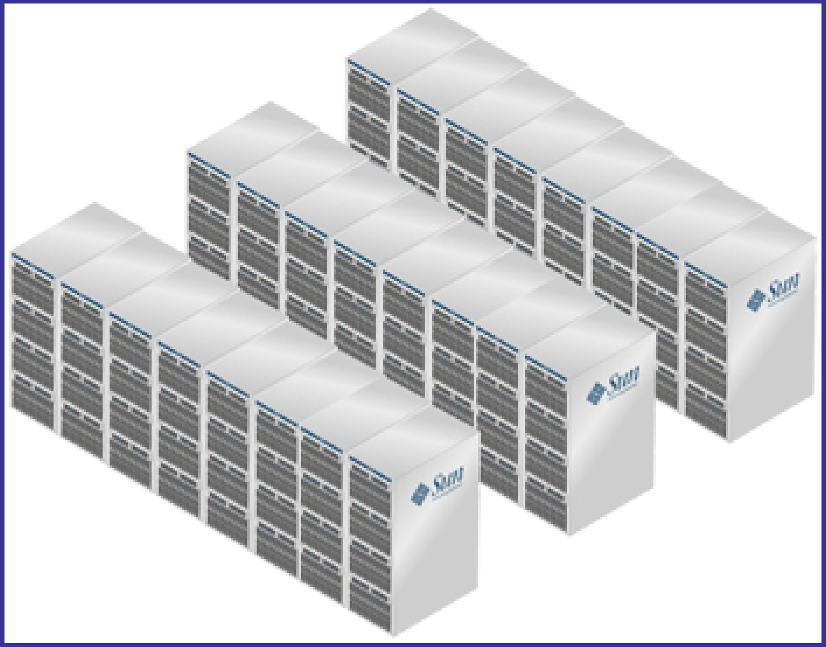
Challenges of Petascale Computation

- Does it scale (unprecedented hardware complexity)? – hopefully.
  - o 500K – 1M cores
  - o Multicore processors, specialized accelerator chips
    - AMD: GP-GPU
    - Intel: 80-core teraflop box
    - IBM: cell processor
    - FPGAs
    - Cray: multithreading + multicore scalar
    - Multiple processors inside SMP box
    - Complex network topology
- Does it scale (numerical algorithms)? ⇨ that's us ⇦
  - o Good news: in principle, PDE solvers should scale… but in practice?
    - anisotropies, heterogeneities, multiphysics, nonlinearities, …
    - dynamic adaptivity, free boundary/interface problems, …
  - o Growing algorithmic and programming complexity
    - Deep nesting of control flow / data structures

Building a petaflops machine is unquestionably hard.
But using it is even harder!

New Sun/AMD system at Texas Advanced Computing Center, UT-Austin

- New $59M NSF award to UT
  deploy and maintain Sun/AMD “Track 2” supercomputer
- Collaboration between TACC, ICES, Cornell, and Arizona St.
- Expected final configuration in November 2007:
  - Deerhound-based (quad-core, 4-way)
  - 530 Teraflop/s
  - >62,000 cores
  - 125 TB memory
  - 1.9 PB disk



The billion dollar question: Can we scale up our simulations to capitalize on $O(10^5)$ CPUs?

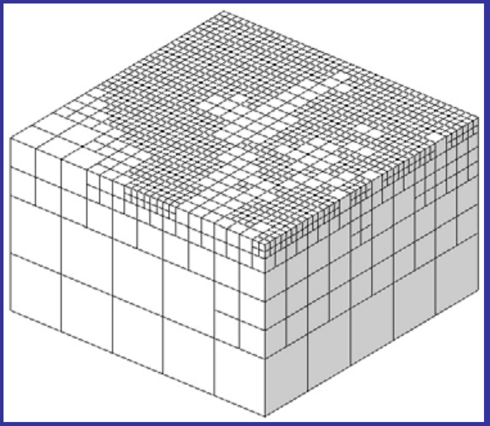
Overall scalability requires:

- System scalability
- Implementation scalability
- Algorithmic scalability
  - Some of the most powerful and algorithmically-scalable numerical algorithms and schemes also have the most complex data structures and implementation requirements
    - Multigrid, AMR, fast multipole, ...
  - Focus of talk: how scalable is seismic inversion?
    - Part 1: scalability of forward solver
    - Part 2: scalability of inverse method

Earthquake wave propagation model

$$
\begin{aligned}
\nabla \cdot \left[\mu \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\mathsf{T}}\right) + \lambda (\nabla \cdot \boldsymbol{u}) \boldsymbol{I}\right] &= \rho \ddot{\boldsymbol{u}} - \boldsymbol{b} \text{ in } \Omega \times (0, T) \\
\left[\mu \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\mathsf{T}}\right) + \lambda (\nabla \cdot \boldsymbol{u}) \boldsymbol{I}\right] \boldsymbol{n} &= \boldsymbol{L}^{AB} \boldsymbol{u} \text{ on } \partial\Omega \times (0, T) \\
\boldsymbol{u} &= \boldsymbol{0} \text{ on } \Omega \times \{t = 0\} \\
\dot{\boldsymbol{u}} &= \boldsymbol{0} \text{ on } \Omega \times \{t = 0\}
\end{aligned}
$$

+Rayleigh attenuation model

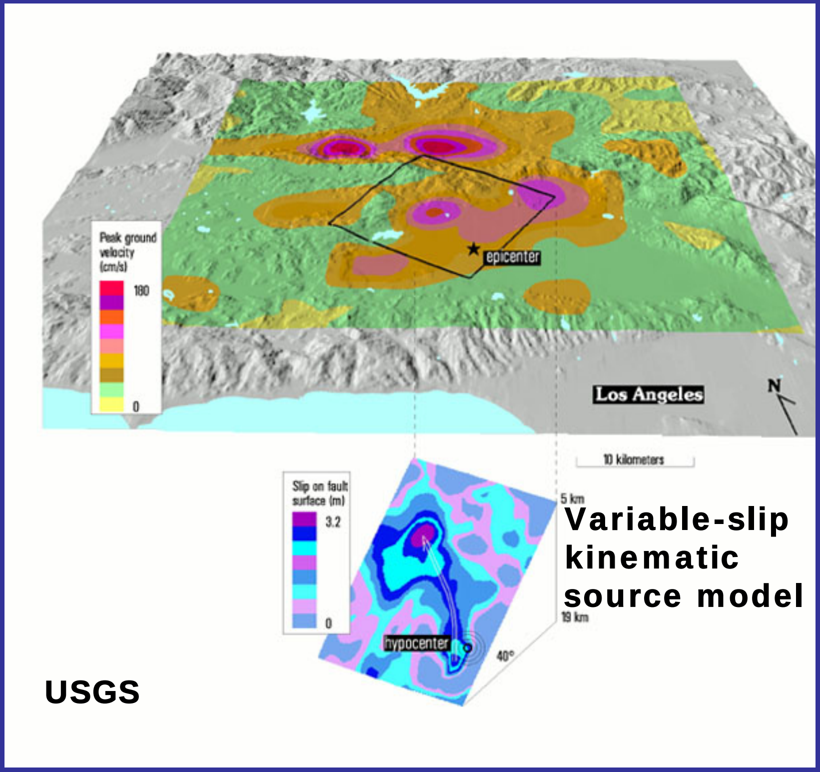


$$
\begin{aligned}
\boldsymbol{u}(\boldsymbol{x}, t) &:= \text{displacement} \\
\rho &:= \text{material density} \\
\mu, \lambda &:= \text{elastic parameters} \\
\boldsymbol{b}(\boldsymbol{x}, t) &:= \text{rupture force, e.g. for point source} \\
&:= -\mu v A f(t) \boldsymbol{M} \nabla \delta(\boldsymbol{x} - \boldsymbol{\xi})
\end{aligned}
$$

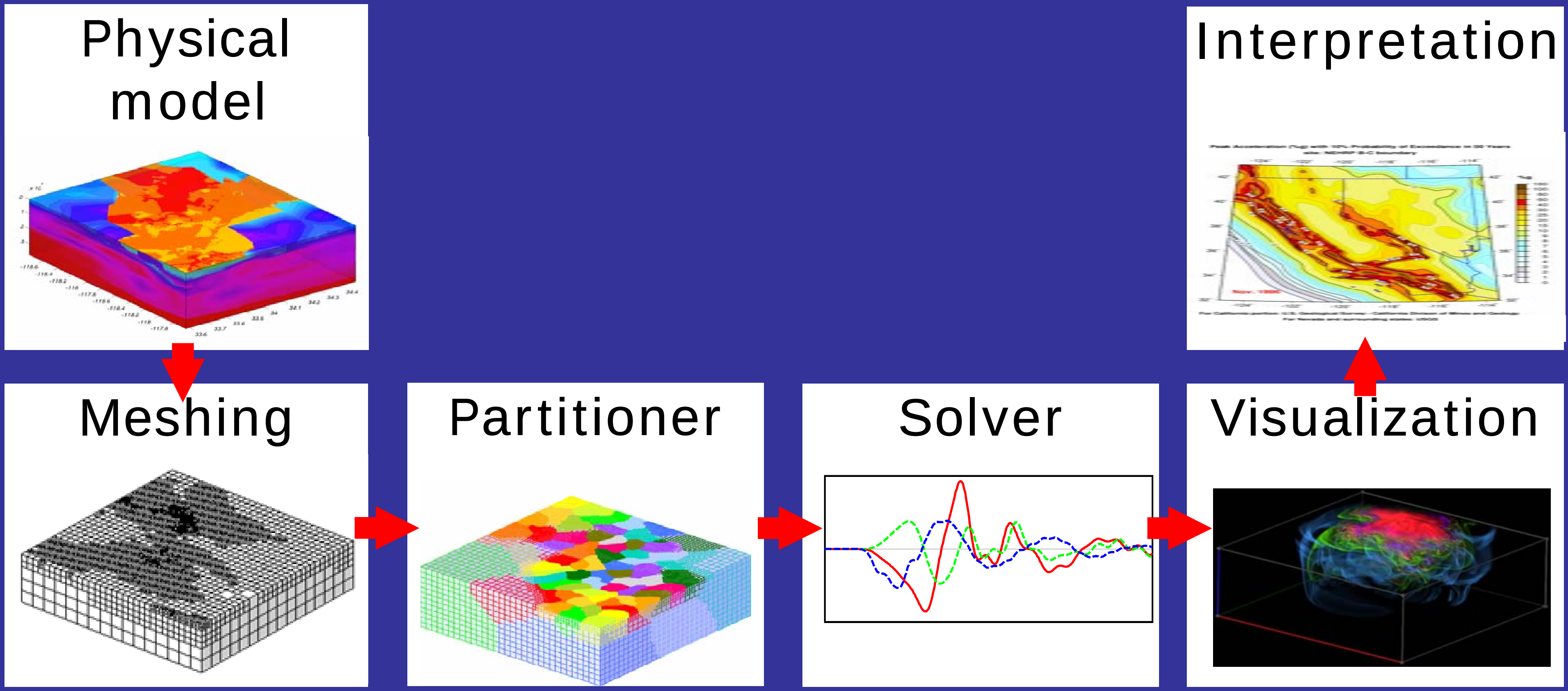
$\boldsymbol{L}^{AB}$ is $\boldsymbol{0}$ on free surfaces, and is given by Stacy's absorbing boundary condition on truncated surfaces:

$$
\boldsymbol{L}^{AB} \boldsymbol{u} \equiv \begin{bmatrix} -d_1 \frac{\partial}{\partial t} & c_1 \frac{\partial}{\partial \tau_1} & c_1 \frac{\partial}{\partial \tau_2} \\ -c_1 \frac{\partial}{\partial \tau_1} & -d_2 \frac{\partial}{\partial t} & 0 \\ -c_1 \frac{\partial}{\partial \tau_2} & 0 & -d_2 \frac{\partial}{\partial t} \end{bmatrix} \begin{Bmatrix} u_n \\ u_{\tau_1} \\ u_{\tau_2} \end{Bmatrix}
$$

$$
\begin{aligned}
c_1 &= -2\mu + \sqrt{\mu(\lambda + 2\mu)}, \\
d_1 &= \sqrt{\rho(\lambda + 2\mu)}, \\
d_2 &= \sqrt{\rho\mu}.
\end{aligned}
$$

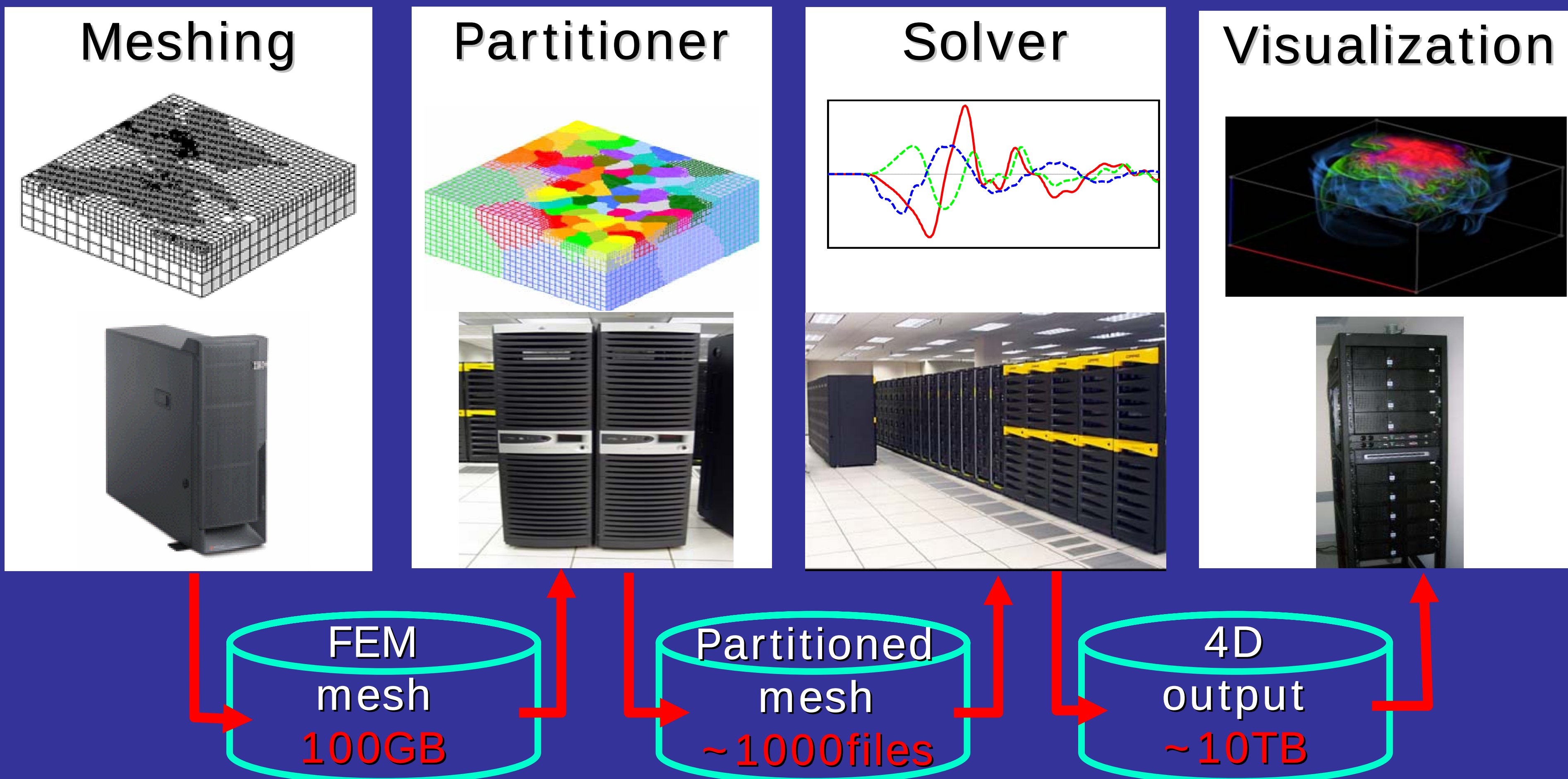


Forward modeling toolchain



Pitfalls of old system

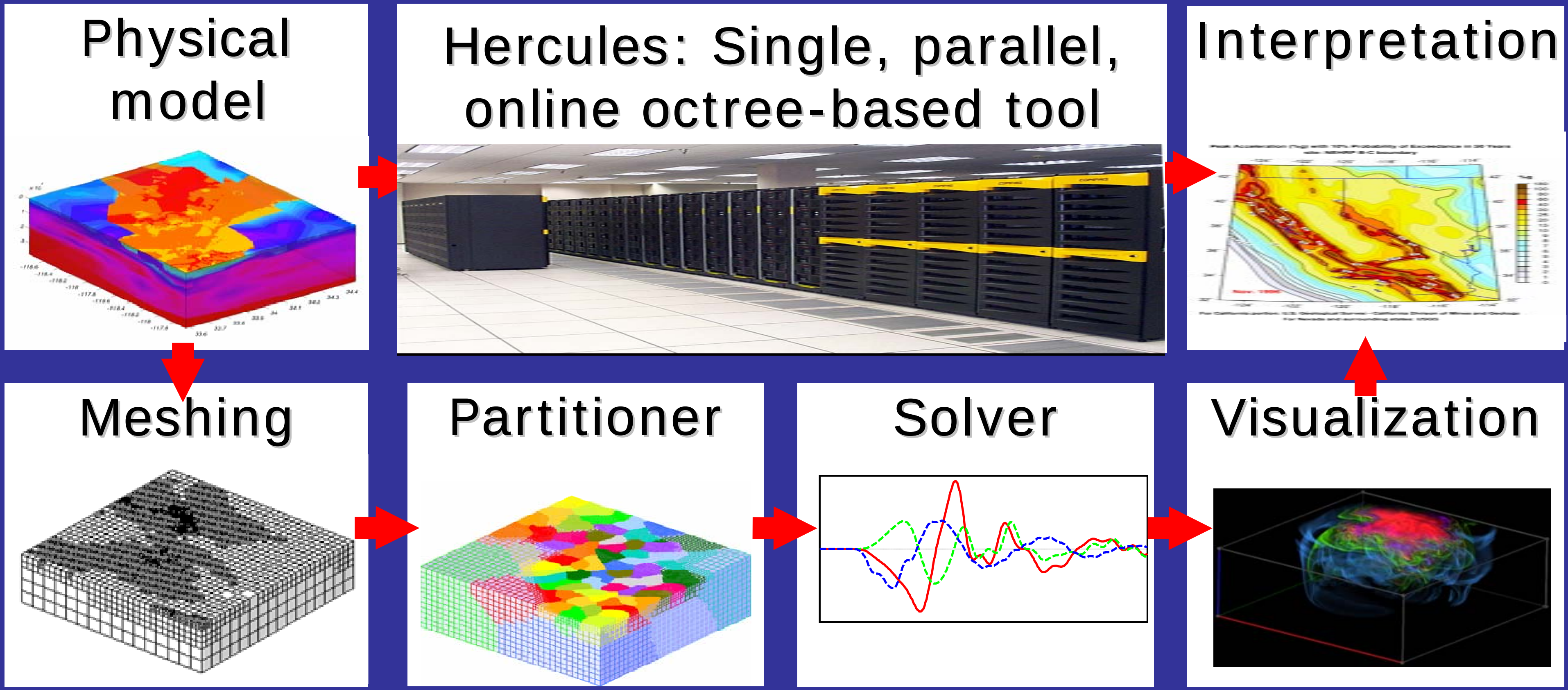
- Different computing systems
- I/O and network bandwidth bound
- Complex data format conversions



End-to-end parallelism

(Tiankai Tu and Hongfeng Yu)

- Avoid large file I/O
- JPEG outputs
- Enable online steering



Isogranular Scalability (55 TF/s TACC Lonestar)

LA Basin (100km x 100km x 37.5 km), SCEC CVM Version 2.0; minimum shear wave velocity 100m/s

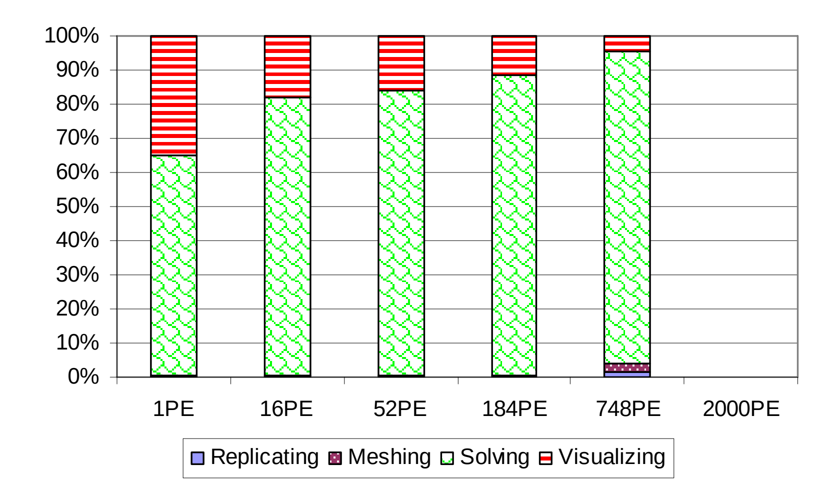
Meshing: $O(N \log E) = O(N \log N)$

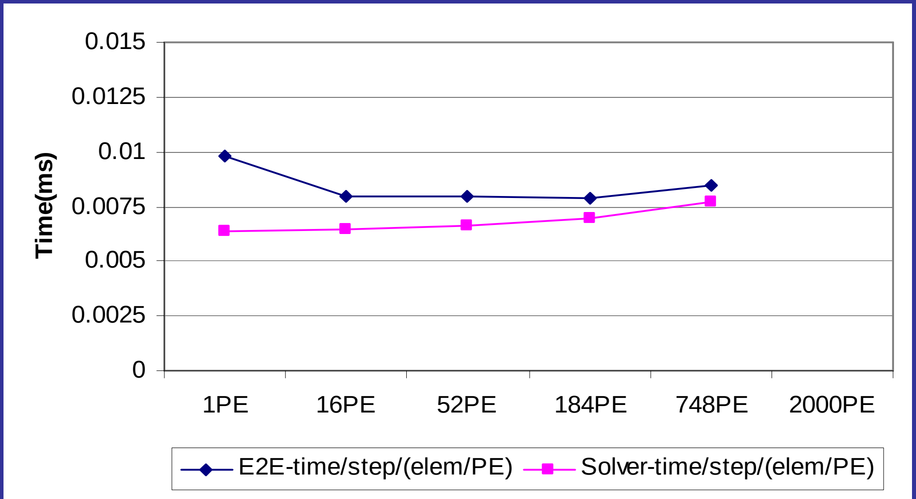
Wave propagation solver: $O(N\,T) = O(N^{4/3})$

Volume rendering: $O(XY\,E^{1/3} \log E) = O(XY\,N^{1/3} \log N)$

N = # grid points; E = # elements; T = # time steps; (X,Y) = pixels in canvas

Performance data (AlphaServer at PSC)





Isogranular scalability:

- End-to-end: up to 534M grid pts, 81% efficiency from 1 to 784 processors

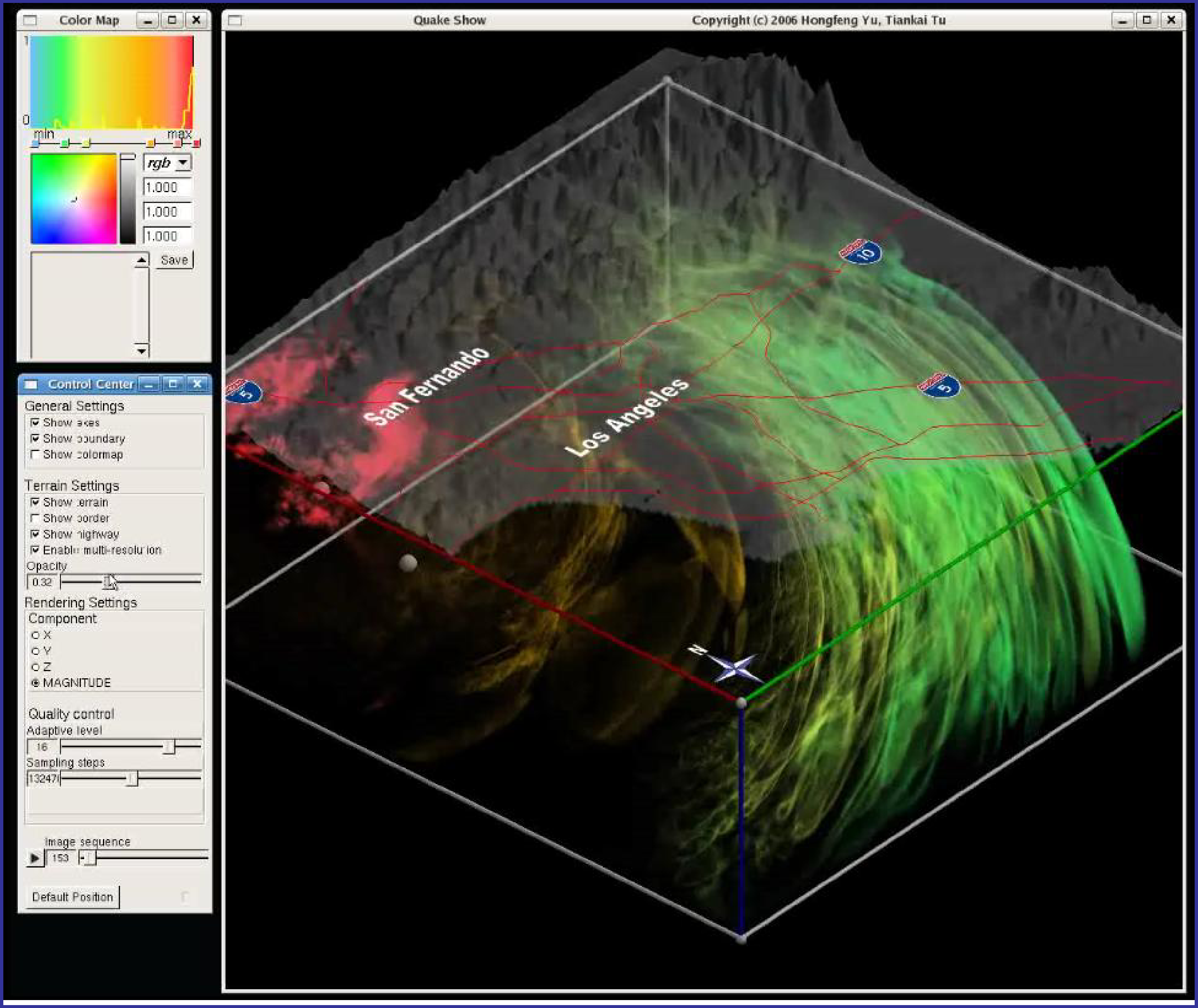
Fixed-size scalability:

- Meshing + solver: 134M grid pts, 84% efficiency on 2048 processors (base 128PEs)
- End-to-end: 134M grid pts, 76% efficiency on 748 processors (base 128PEs)

Processor utilization:

- ~655 MFlops/sec/PE; 33% of the peak performance (2GFlops/sec/PE).

Example: 1994 Northridge earthquake simulation (2006)
(real-time on 2000 cores of TACC Dell/Woodcrest Lonestar)



Full waveform inversion: Challenges

- Inverse problem can be highly nonlinear even when forward problem is linear
- Inverse operator is non-local, non-causal
- Need to build on large body of algorithms, libraries, and software for forward simulation
- Numerous forward simulations required for an inverse solution
- Numerous inverse solutions required to estimate best regularization parameter
- Want to estimate not just mean of parameters, but also variance (or better: distribution)
- Significant work by: Symes, Tarantola, Chavent, Mora, Jordan, Tromp, Pratt, etc.

Least squares parameter estimation formulation of inverse wave propagation

sources · receivers · data misfit · inversion fields · displacements

$$\min_{\boldsymbol{u}_i, \lambda, \mu} \mathcal{F} = \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_r^i} \int_0^T \int_\Omega \left(\boldsymbol{u}_i^* - \boldsymbol{u}_i\right)^2 \delta\left(\boldsymbol{x} - \boldsymbol{x}_{ij}\right) d\boldsymbol{x}\, dt$$

$$+ \int_\Omega \beta_\mu \left(\nabla\mu \cdot \nabla\mu + \varepsilon\right)^{\frac{1}{2}} + \beta_\lambda \left(\nabla\lambda \cdot \nabla\lambda + \varepsilon\right)^{\frac{1}{2}} d\boldsymbol{x}$$

subject to $\forall i$:

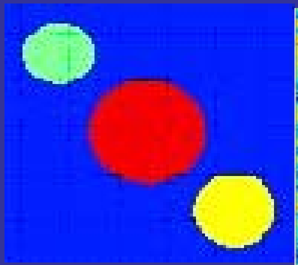
$$\rho \ddot{\boldsymbol{u}}_i - \nabla \cdot \left[\mu\left(\nabla \boldsymbol{u}_i + \nabla \boldsymbol{u}_i^{\mathsf{T}}\right) + \lambda(\nabla \cdot \boldsymbol{u}_i)\boldsymbol{I}\right] = \boldsymbol{b}_i \quad \text{in } \Omega \times (0, T)$$

$$\left[\mu\left(\nabla \boldsymbol{u}_i + \nabla \boldsymbol{u}_i^{\mathsf{T}}\right) + \lambda(\nabla \cdot \boldsymbol{u}_i)\boldsymbol{I}\right] \boldsymbol{n} = \boldsymbol{L}_{\mathrm{AB}} \boldsymbol{u}_i \quad \text{on } \partial\Omega \times (0, T)$$

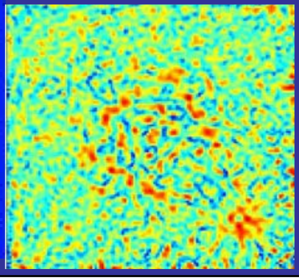
$$\boldsymbol{u}_i = \boldsymbol{0} \quad \text{on } \Omega \times \{t = 0\}$$

$$\dot{\boldsymbol{u}}_i = \boldsymbol{0} \quad \text{on } \Omega \times \{t = 0\}$$

forward wave propagation model

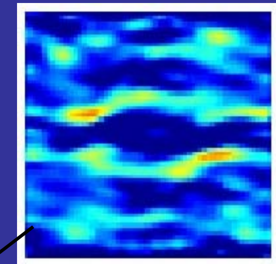


target

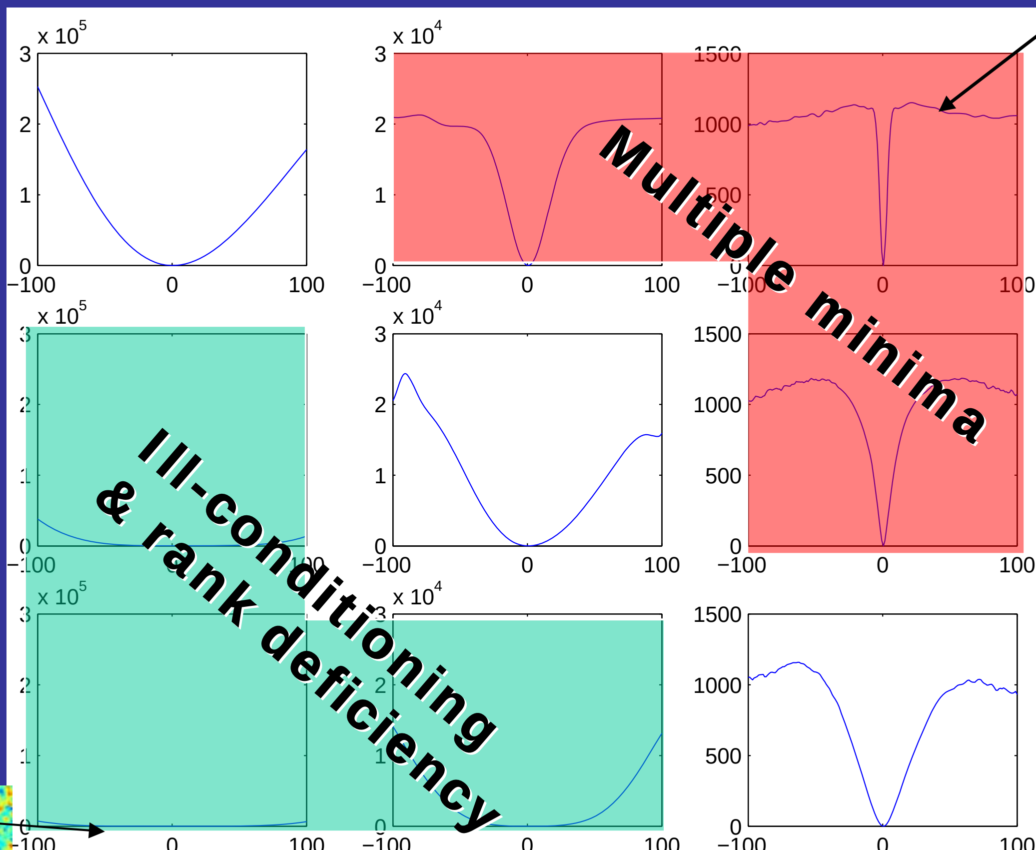


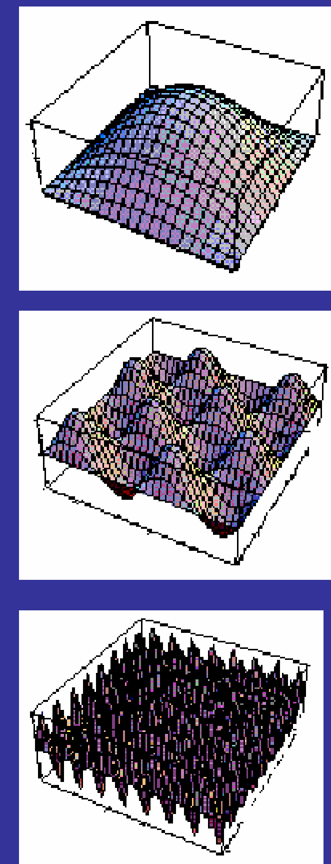
unregularized solution

Behavior of misfit function F in direction of material perturbation



Source frequency



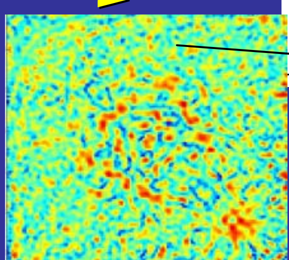


low

med

high

Wavenumber of perturbation of material field



low   medium   high

$$\mathcal{F} = \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_r^i} \int_0^T \int_\Omega (\boldsymbol{u}_i^* - \boldsymbol{u}_i)^2 \delta \left(\boldsymbol{x} - \boldsymbol{x}_{ij} \right) d\boldsymbol{x} \, dt$$

Least squares parameter estimation formulation of inverse wave propagation

inversion fields — sources — receivers — data misfit — displacements

$$\min_{\boldsymbol{u}_i,\lambda,\mu} \mathcal{F} = \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_r^i} \int_0^T \int_\Omega \left(\boldsymbol{u}_i^* - \boldsymbol{u}_i\right)^2 \delta\left(\boldsymbol{x} - \boldsymbol{x}_{ij}\right) d\boldsymbol{x}\, dt$$

$$+ \int_\Omega \beta_\mu \left(\boldsymbol{\nabla}\mu \cdot \boldsymbol{\nabla}\mu + \varepsilon\right)^{\frac{1}{2}} + \beta_\lambda \left(\boldsymbol{\nabla}\lambda \cdot \boldsymbol{\nabla}\lambda + \varepsilon\right)^{\frac{1}{2}} d\boldsymbol{x}$$

subject to $\forall i$:

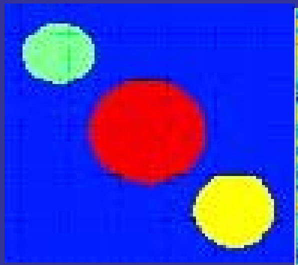
$$\rho \ddot{\boldsymbol{u}}_i - \boldsymbol{\nabla} \cdot \left[\mu\left(\boldsymbol{\nabla}\boldsymbol{u}_i + \boldsymbol{\nabla}\boldsymbol{u}_i^{\mathsf{T}}\right) + \lambda(\boldsymbol{\nabla}\cdot\boldsymbol{u}_i)\boldsymbol{I}\right] = \boldsymbol{b}_i \quad \text{in } \Omega \times (0,T)$$

$$\left[\mu\left(\boldsymbol{\nabla}\boldsymbol{u}_i + \boldsymbol{\nabla}\boldsymbol{u}_i^{\mathsf{T}}\right) + \lambda(\boldsymbol{\nabla}\cdot\boldsymbol{u}_i)\boldsymbol{I}\right]\boldsymbol{n} = \boldsymbol{L}_{\mathsf{AB}}\boldsymbol{u}_i \quad \text{on } \partial\Omega \times (0,T)$$

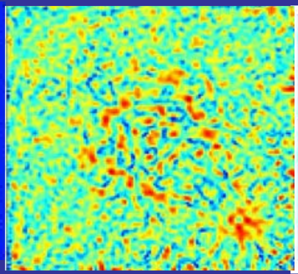
$$\boldsymbol{u}_i = \boldsymbol{0} \quad \text{on } \Omega \times \{t = 0\}$$

$$\dot{\boldsymbol{u}}_i = \boldsymbol{0} \quad \text{on } \Omega \times \{t = 0\}$$

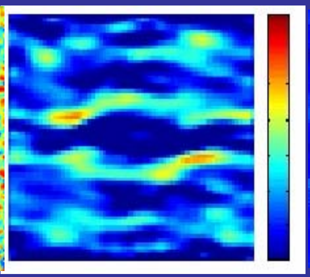
forward wave propagation model



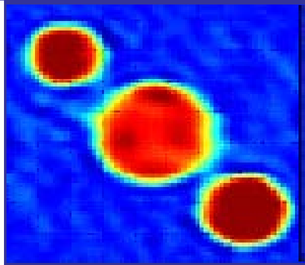
target



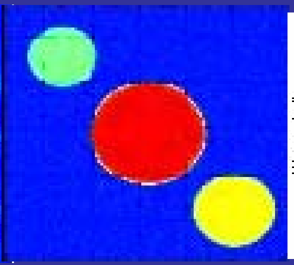
unregularized solution



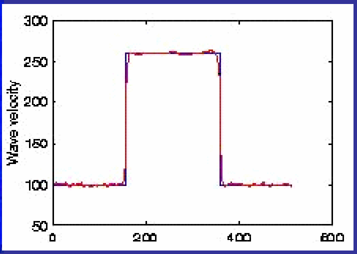
local minimum



Tikhonov reg. + multiscale



BV regular. + multiscale



BV reg., section

Least squares parameter estimation formulation of inverse wave propagation

inversion fields | sources | receivers | data misfit

$$\min_{\boldsymbol{u}_i,\lambda,\mu} \mathcal{F} = \frac{1}{2}\sum_{i=1}^{N_s}\sum_{j=1}^{N_r^i}\int_0^T\int_\Omega (\boldsymbol{u}_i^* - \boldsymbol{u}_i)^2 \delta\left(\boldsymbol{x} - \boldsymbol{x}_{ij}\right) d\boldsymbol{x}\, dt$$

$$+ \int_\Omega \beta_\mu \left(\boldsymbol{\nabla}\mu \cdot \boldsymbol{\nabla}\mu + \varepsilon\right)^{\frac{1}{2}} + \beta_\lambda \left(\boldsymbol{\nabla}\lambda \cdot \boldsymbol{\nabla}\lambda + \varepsilon\right)^{\frac{1}{2}} d\boldsymbol{x}$$

displacements

subject to $\forall i$:

$$\rho \ddot{\boldsymbol{u}}_i - \boldsymbol{\nabla}\cdot\left[\mu\left(\boldsymbol{\nabla}\boldsymbol{u}_i + \boldsymbol{\nabla}\boldsymbol{u}_i^{\mathsf{T}}\right) + \lambda(\boldsymbol{\nabla}\cdot\boldsymbol{u}_i)\boldsymbol{I}\right] = \boldsymbol{b}_i \quad \text{in } \Omega\times(0,T)$$

$$\left[\mu\left(\boldsymbol{\nabla}\boldsymbol{u}_i + \boldsymbol{\nabla}\boldsymbol{u}_i^{\mathsf{T}}\right) + \lambda(\boldsymbol{\nabla}\cdot\boldsymbol{u}_i)\boldsymbol{I}\right]\boldsymbol{n} = \boldsymbol{L}_{\mathrm{AB}}\boldsymbol{u}_i \quad \text{on } \partial\Omega\times(0,T)$$

$$\boldsymbol{u}_i = \boldsymbol{0} \quad \text{on } \Omega\times\{t=0\}$$

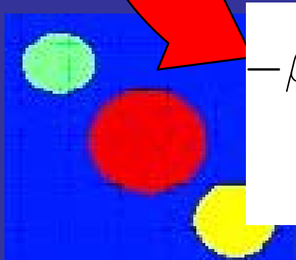
$$\dot{\boldsymbol{u}}_i = \boldsymbol{0} \quad \text{on } \Omega\times\{t=0\}$$

forward wave propagation model

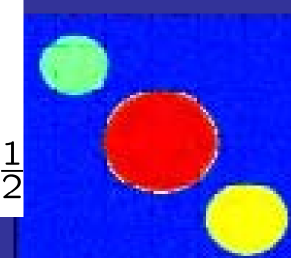
second variation

$$-\beta_\mu \boldsymbol{\nabla}\cdot\left[\frac{1}{|\boldsymbol{\nabla}\mu|_\varepsilon}\left(\boldsymbol{I} - \frac{\boldsymbol{\nabla}\mu\otimes\boldsymbol{\nabla}\mu}{|\boldsymbol{\nabla}\mu|_\varepsilon^2}\right)\boldsymbol{\nabla}\tilde{\mu}\right],$$

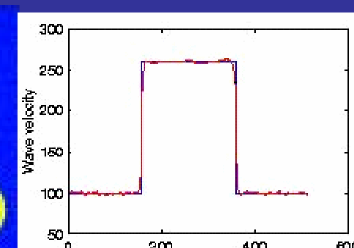
where $|\boldsymbol{\nabla}\mu|_\varepsilon := \left(\boldsymbol{\nabla}\mu\cdot\boldsymbol{\nabla}\mu + \varepsilon\right)^{\frac{1}{2}}$



target



BV regular. + multiscale



BV reg., section

Lagrangian and weak form of optimality system

$$\mathcal{L}(u,p,\mu) := \frac{1}{2}\sum_{j=1}^{N_r}\int_0^T\int_\Omega (u-u^*)^2\delta(\boldsymbol{x}-\boldsymbol{x}_j)\,d\boldsymbol{x}\,dt \;+\; \beta\int_\Omega |\boldsymbol{\nabla}\mu|_\varepsilon\,d\boldsymbol{x}$$

$$+\int_0^T\int_\Omega (\mu\boldsymbol{\nabla}u\cdot\boldsymbol{\nabla}p - \rho u_t\,p_t)\,d\boldsymbol{x}\,dt \;-\int_0^T\int_\Sigma \mu u_0 g\boldsymbol{\nabla}p\cdot\boldsymbol{n}_\Sigma\,d\boldsymbol{s}\,dt \;-\int_0^T\int_{\Gamma_{AB}} u_t p\sqrt{\rho\mu}\,d\boldsymbol{s}\,dt$$

Strong form of first order necessary conditions

A Gauss-Newton-Schur-CG method

The Gauss-Newton step:

$$\begin{bmatrix} \mathcal{B} & \mathcal{C}^*(p) & \mathcal{A}^*(\mu) \\ \mathcal{C}(p) & \mathcal{R}(\mu) & \mathcal{D}^*(u) \\ \mathcal{A}(\mu) & \mathcal{D}(u) & 0 \end{bmatrix} \begin{Bmatrix} \tilde{u} \\ \tilde{\mu} \\ \tilde{p} \end{Bmatrix} = - \begin{Bmatrix} \mathcal{L}_u(u,\mu,p) \\ \mathcal{L}_\mu(u,\mu,p) \\ \mathcal{L}_p(u,\mu) \end{Bmatrix}$$

A linear Schur complement method:

$$\left(\mathcal{D}^*\mathcal{A}^{*^{-1}}\mathcal{B}\mathcal{A}^{-1}\mathcal{D} - \mathcal{D}^*\mathcal{A}^{*^{-1}}\mathcal{C} - \mathcal{C}\mathcal{A}^{-1}\mathcal{D} + \mathcal{R}\right)\tilde{\mu} = \mathcal{D}^*\mathcal{A}^{*^{-1}}\mathcal{L}_u - \mathcal{L}_\mu$$

- How to solve? Straightforward approach intractable, e.g. for largest problem, we solve:
  - o 17 million wave propagations to set up linear system
  - o 2 petabytes to store it
  - o 3 hours on an exaflops/s machine for one Newton iteration
  - o (for 1Hz LA model, zettascale/yottascale computing needed; arrives in ~2050 per Demi Moore's law)

- Instead, solve by conjugate gradients:
  - o form Hessian-vector products on the fly, terminate early per Eisenstat-Walker
  - o globalize with line search or trust region
  - o each CG iteration requires 1 forward, 1 adjoint wave propagation -> parallelizes as well as forward problem (N_s forw/adj wave propagations for multiple sources)
  - o construction of preconditioner difficult, since Hessian not formed explicitly

Algorithmic scalability for 3D acoustic inversion example

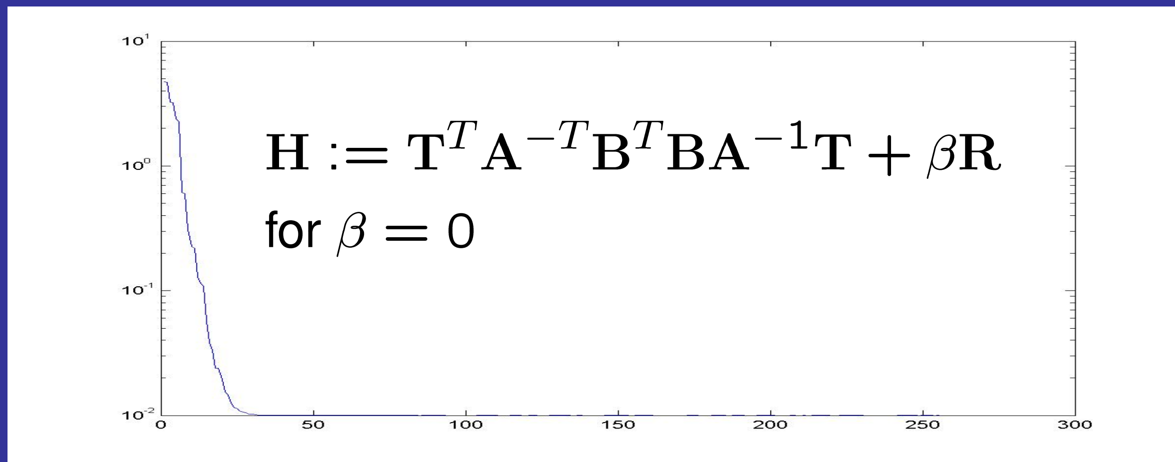
material grid	Picard-Gauss-Newton-Krylov iter, no PC			Picard-Gauss-Newton-Krylov iter, LBFGS/2SR PC		
	nonlinear iter	total linear iter	avg linear iter	nonlinear iter	total linear iter	avg linear iter
2^3	6	31	5.2	6	13	2.2
3^3	11	121	11.0	11	39	3.5
5^3	18	321	17.8	17	144	8.5
9^3	13	614	47.2	12	249	21.0
17^3	11	1413	128.5	12	396	33.0
33^3	17	1445	85.0	25	439	17.6
65^3	19	1923	101.2	19	370	19.5
129^3	21	2003	95.4	22	436	19.8

Mesh independence of nonlinear iterations

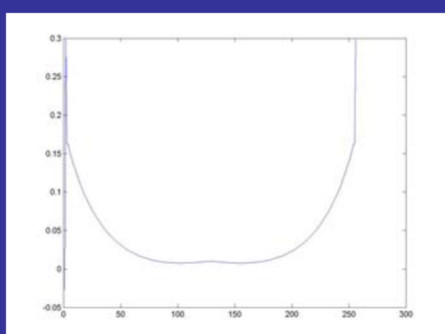
Mesh independence of linear iterations

But even with mesh independence, # of wave propagations still large!

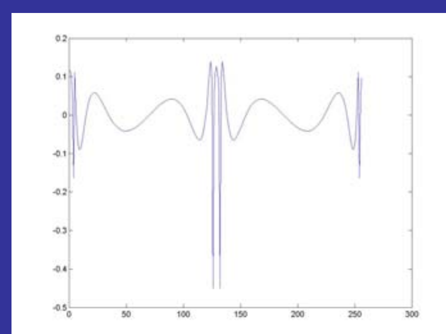
Motivation for reduced space CG solver



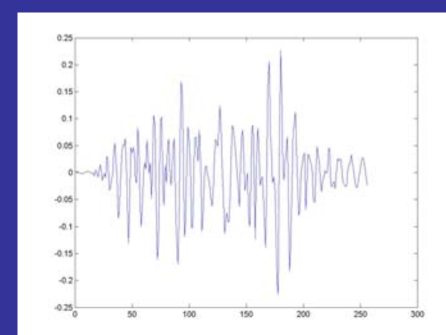
Spectrum of discrete reduced Hessian



Eigenvector for large eigenvalue



Eigenvector for intermediate eigenvalue



Eigenvector for small eigenvalue

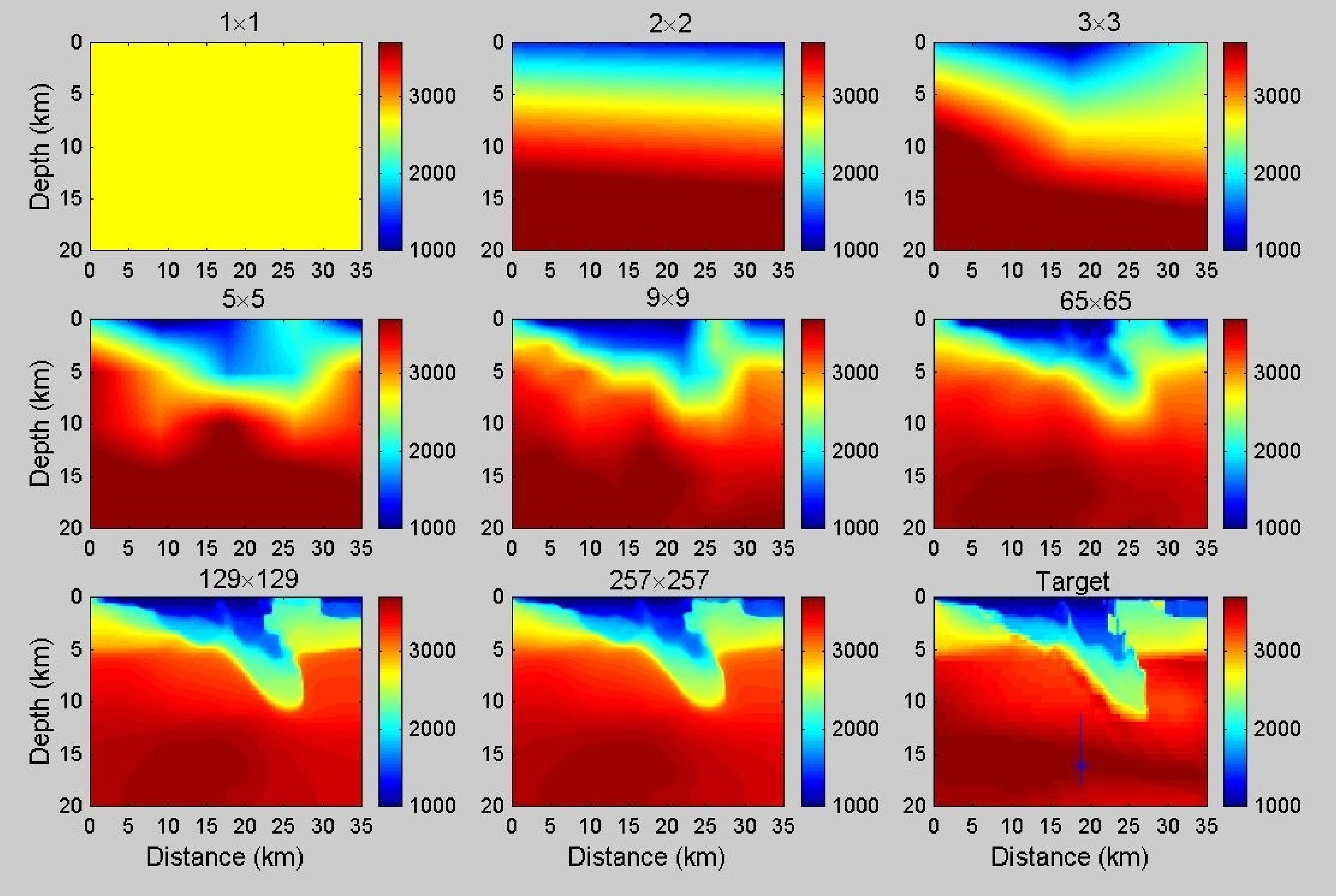
Solution algorithm:
Multilevel inexact Gauss-Newton-PCG

- Multilevel continuation over grid and source frequency
  - o Inexact Gauss-Newton nonlinear iteration
    - Conjugate gradient solution of reduced Hessian system (each matvec requires N_s forward & adjoint wave propagation solutions)
      - – Preconditioner:
        - » limited memory BFGS (Morales-Nocedal)
        - » initialized with several iterations of Frankel's method (two-step stationary method) to "invert" $\alpha\mathcal{I} + \beta\mathcal{F}$
        - » Multigrid

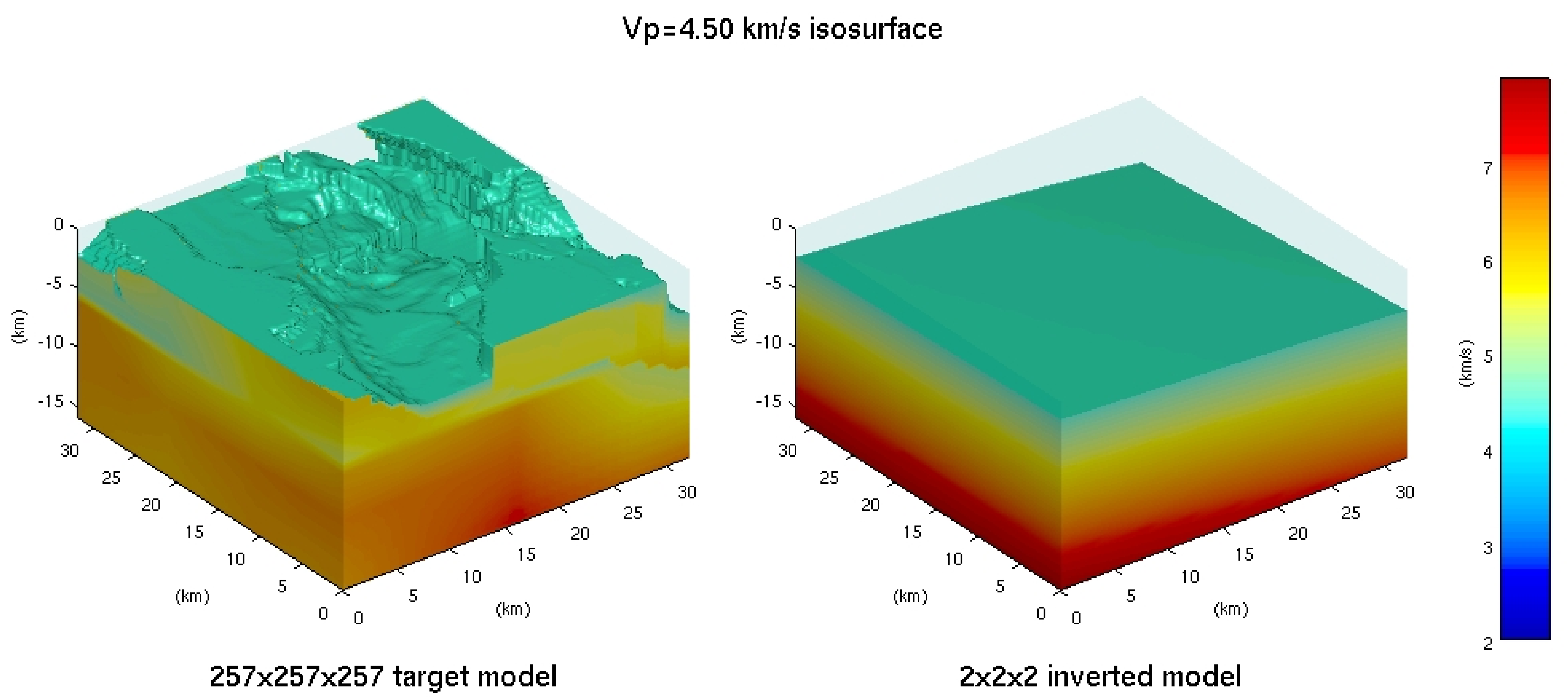
Inversion examples

- 2D shear, 3D acoustic, and 3D elastic models
- Synthetic inversion (some with 5% added noise) using SCEC community velocity model
- Piecewise bi/trilinear finite element approximation of state, adjoint, and material property in space
- Explicit central difference time integration
- PETSc (www.petsc.anl.gov) implementation
- Up to 257x257x257 grid (17 million inversion parameters) on 2048 processors (~12h)
- Up to 225 surface receivers

Material inversion: multiscale continuation (64 receivers)

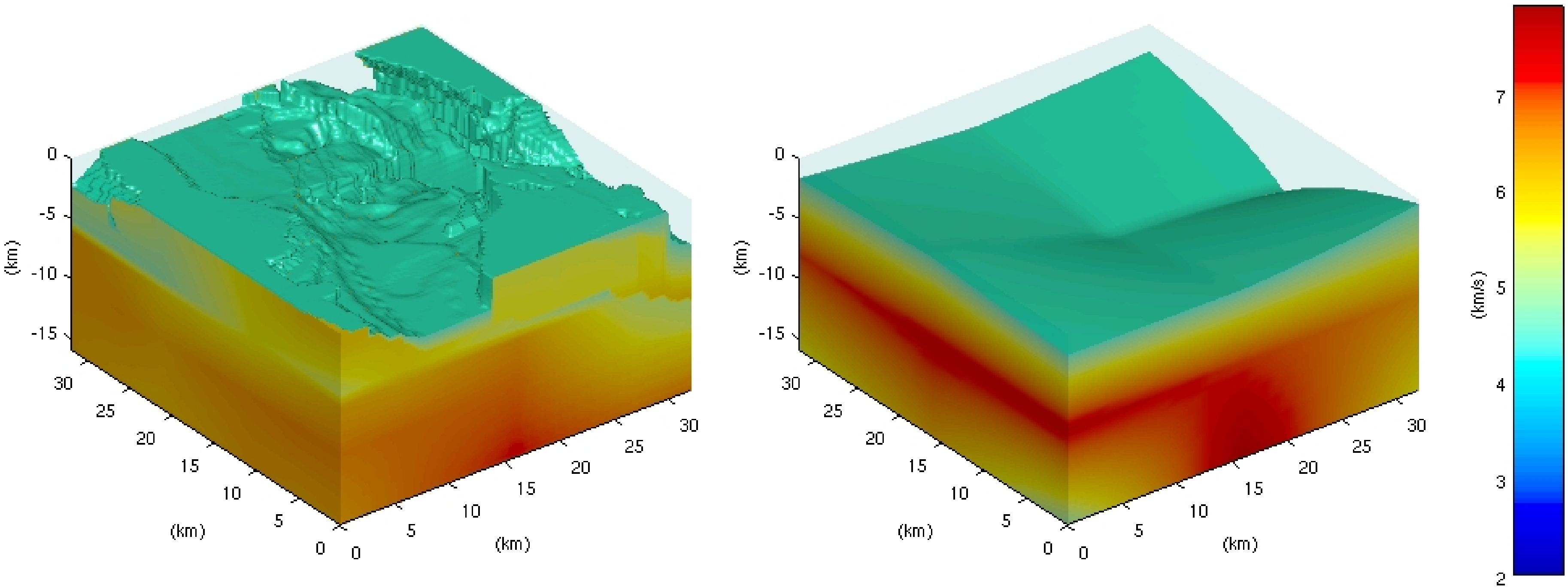


Multiscale inversion:
Target vs. inverted isosurfaces, level 1



Multiscale inversion:
Target vs. inverted isosurfaces, level 2

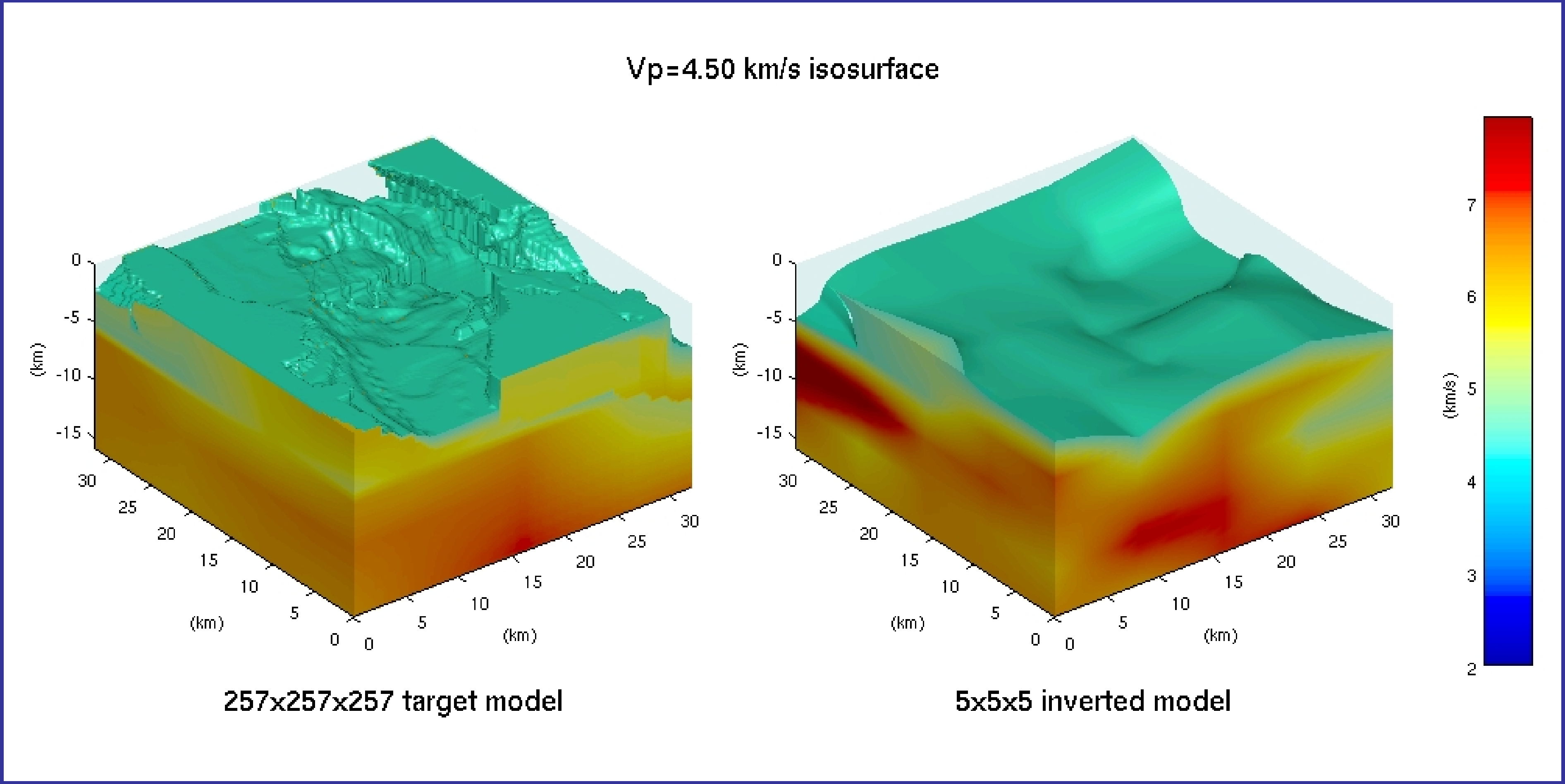
Vp=4.50 km/s isosurface



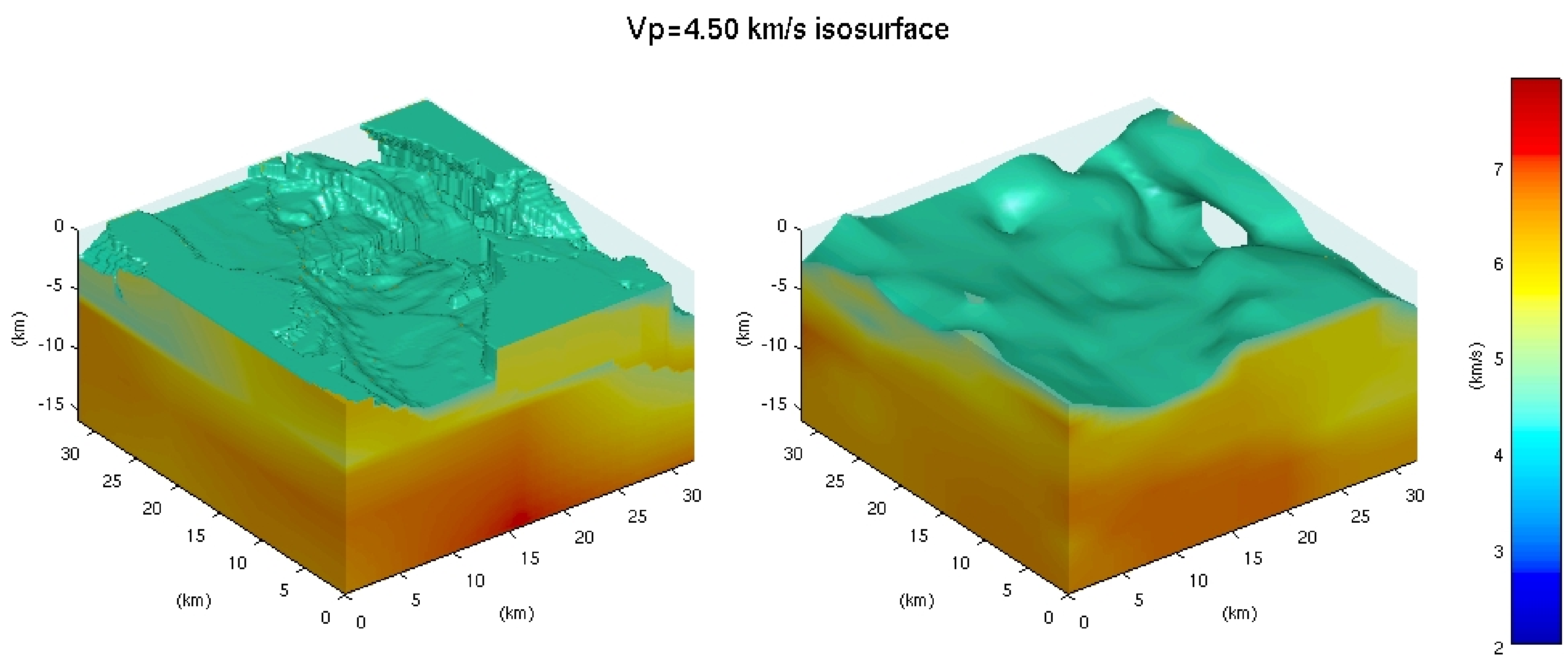
257x257x257 target model

3x3x3 inverted model

Multiscale inversion:
Target vs. inverted isosurfaces, level 3



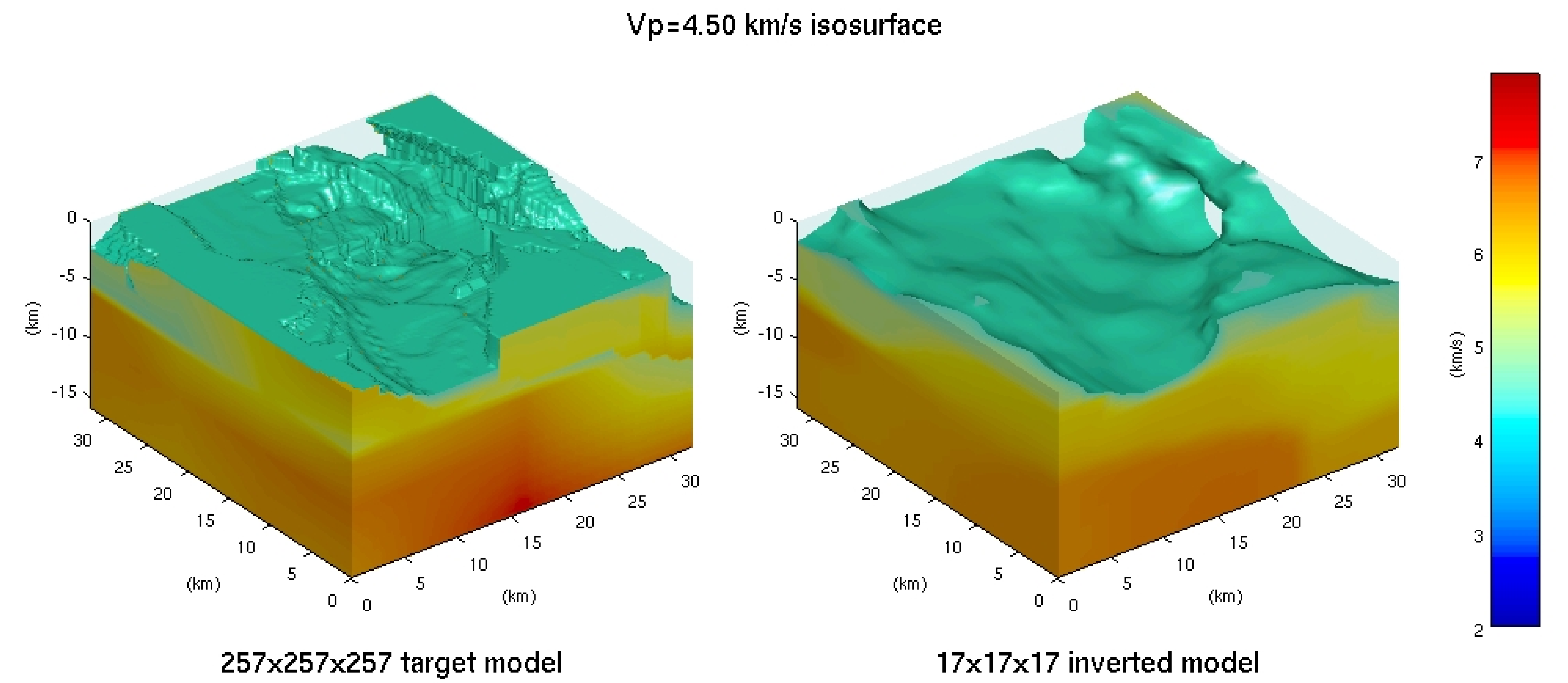
Multiscale inversion:
Target vs. inverted isosurfaces, level 4



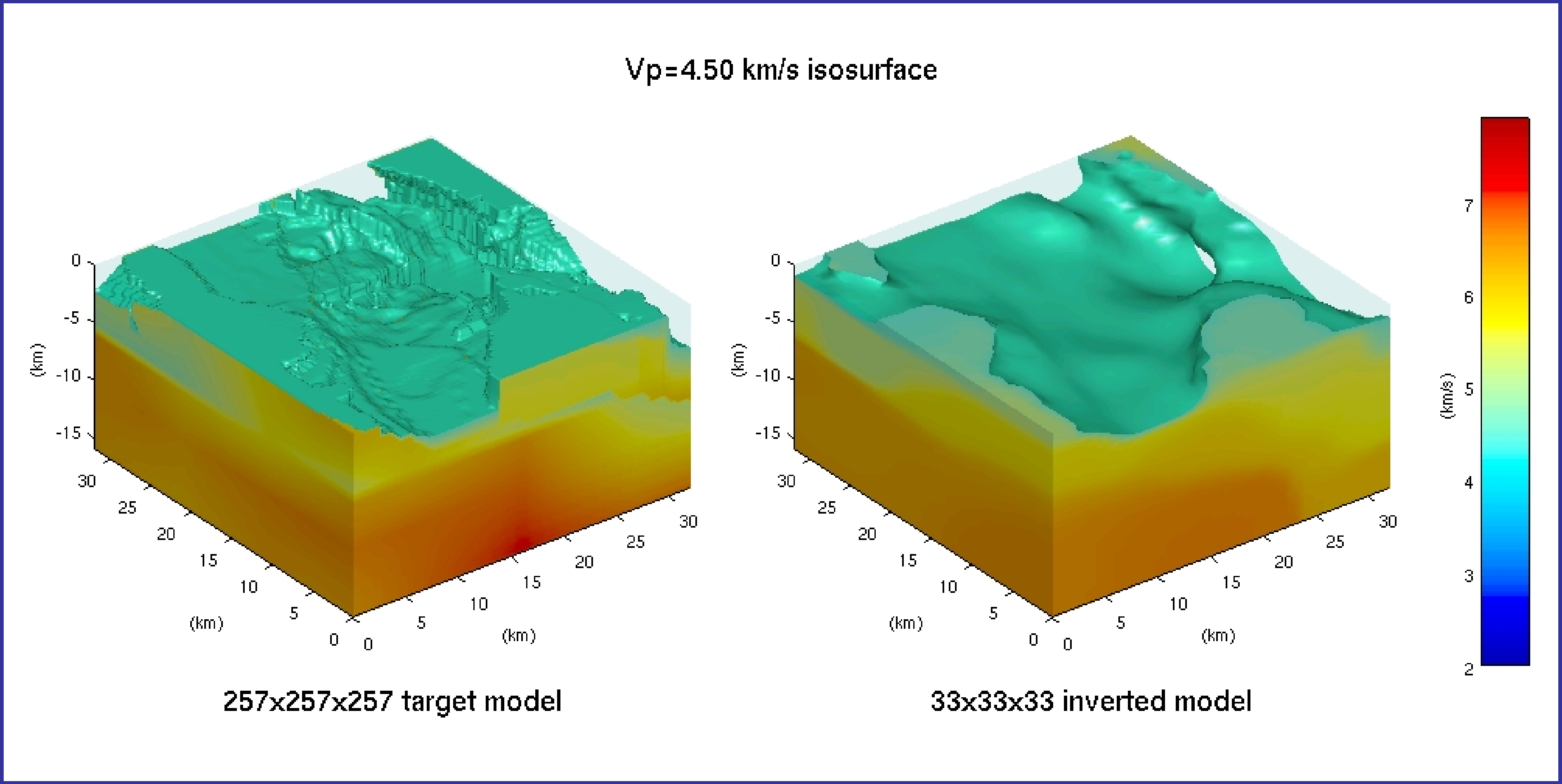
257x257x257 target model

9x9x9 inverted model

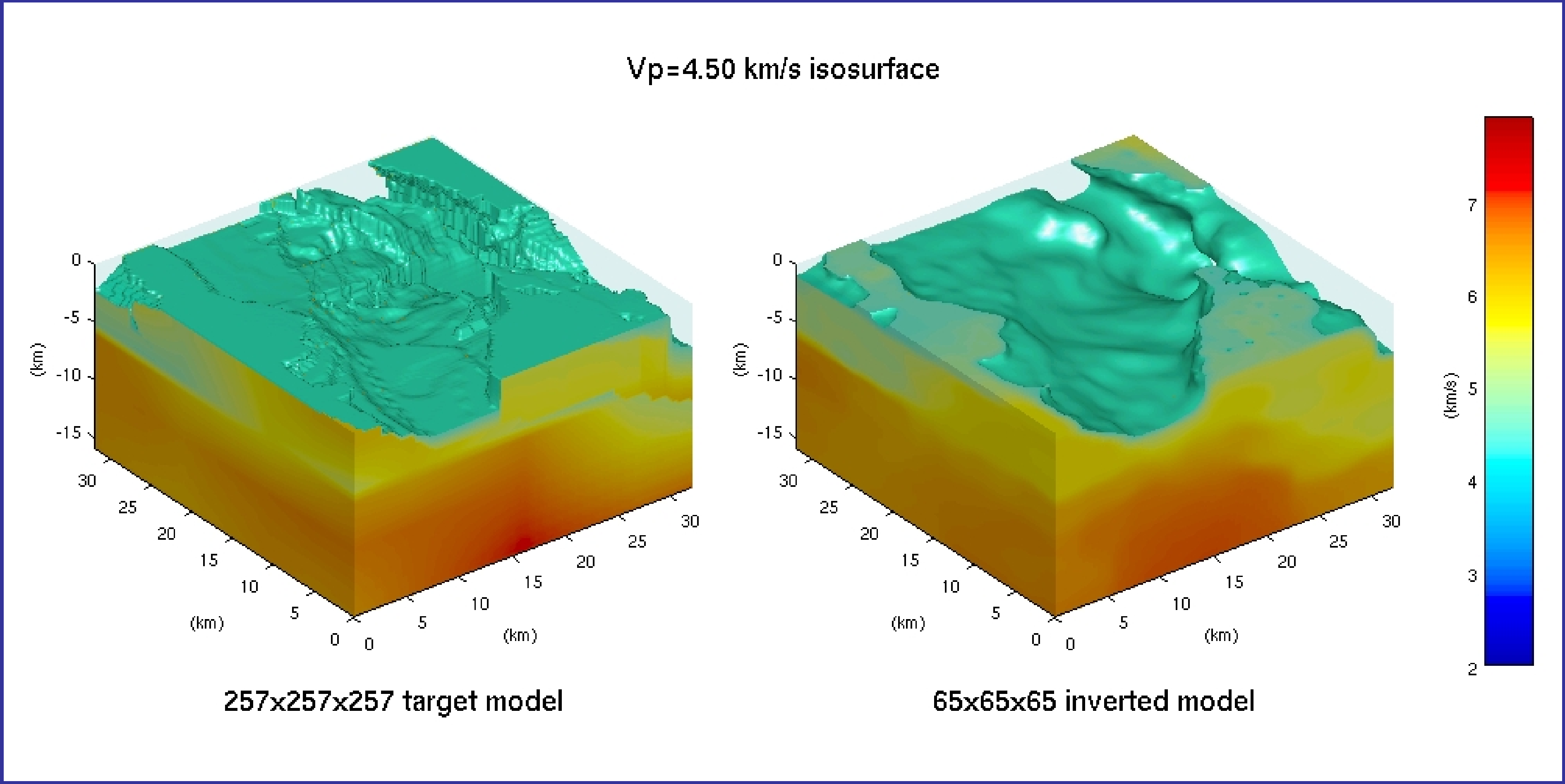
Multiscale inversion:
Target vs. inverted isosurfaces, level 5



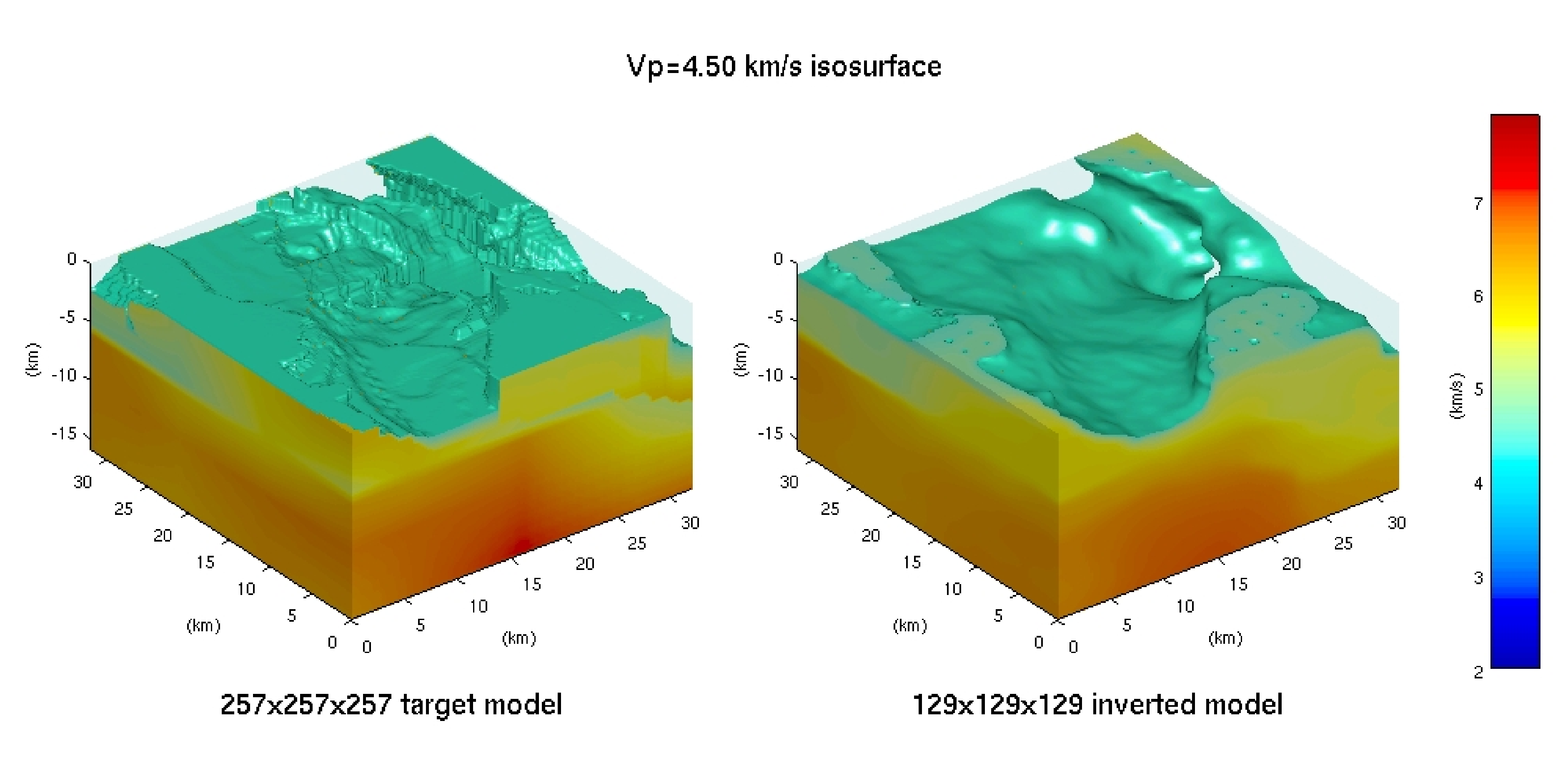
Multiscale inversion:
Target vs. inverted isosurfaces, level 6



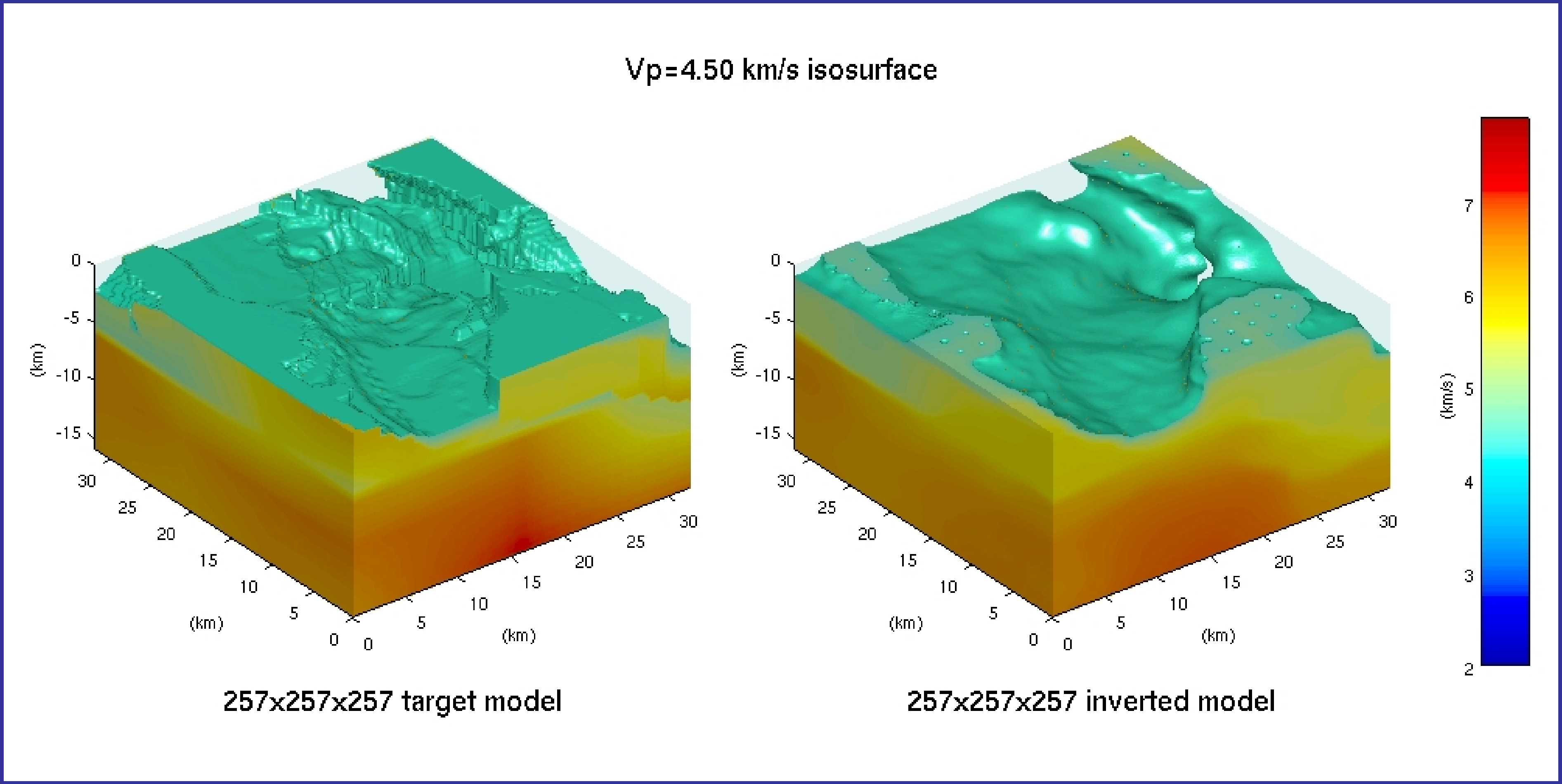
Multiscale inversion:
Target vs. inverted isosurfaces, level 7



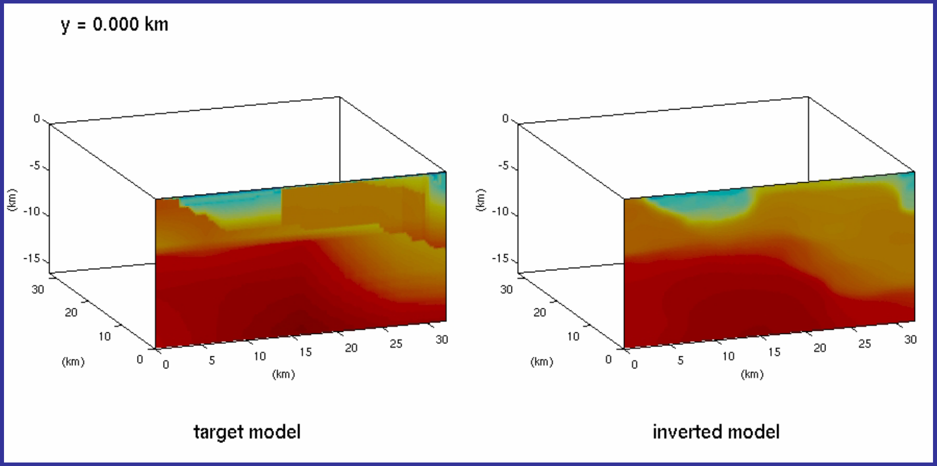
Multiscale inversion: Target vs. inverted isosurfaces, level 8

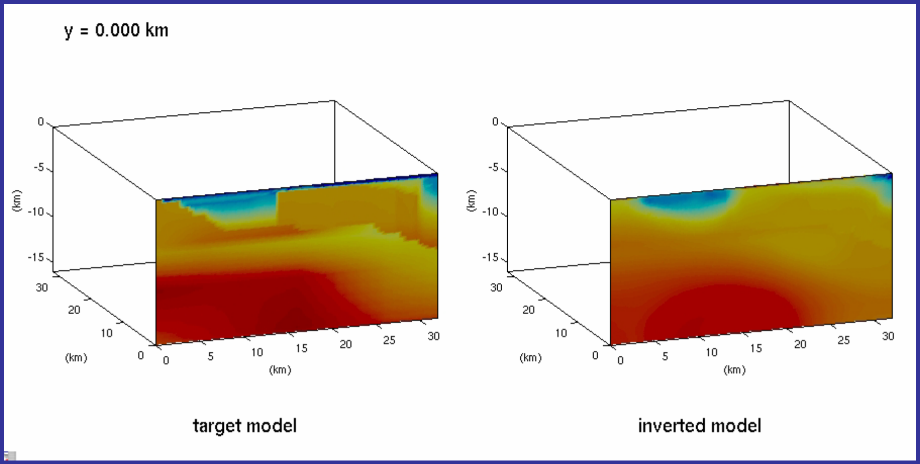


Multiscale inversion:
Target vs. inverted isosurfaces, level 9



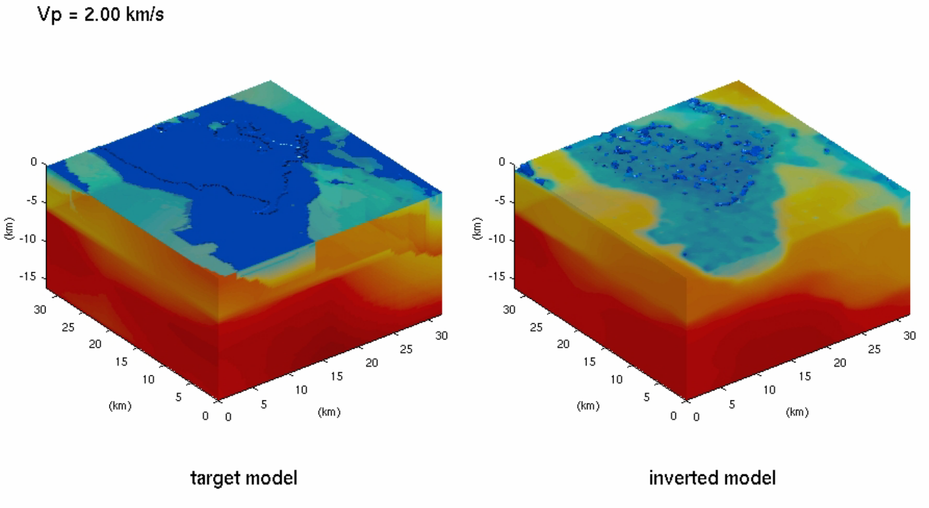
Comparison of target and inverted material models: 3D acoustic and elastic

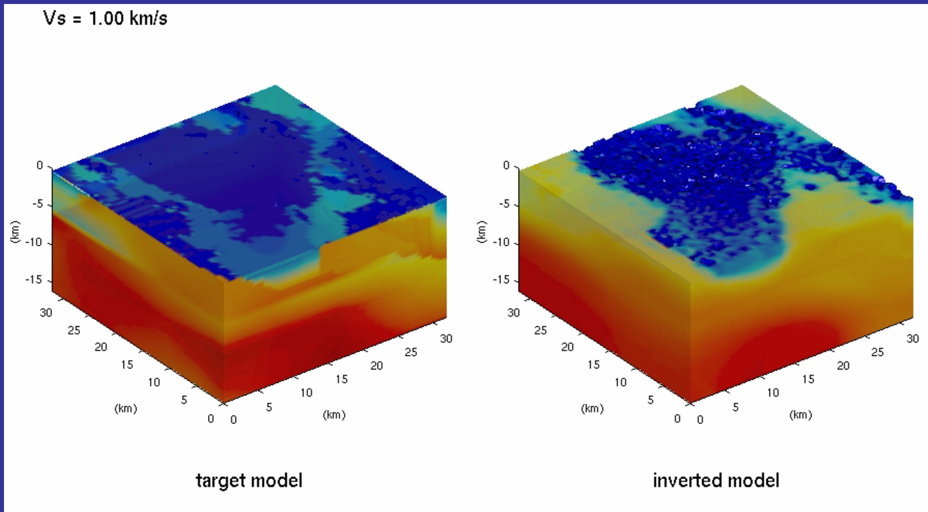


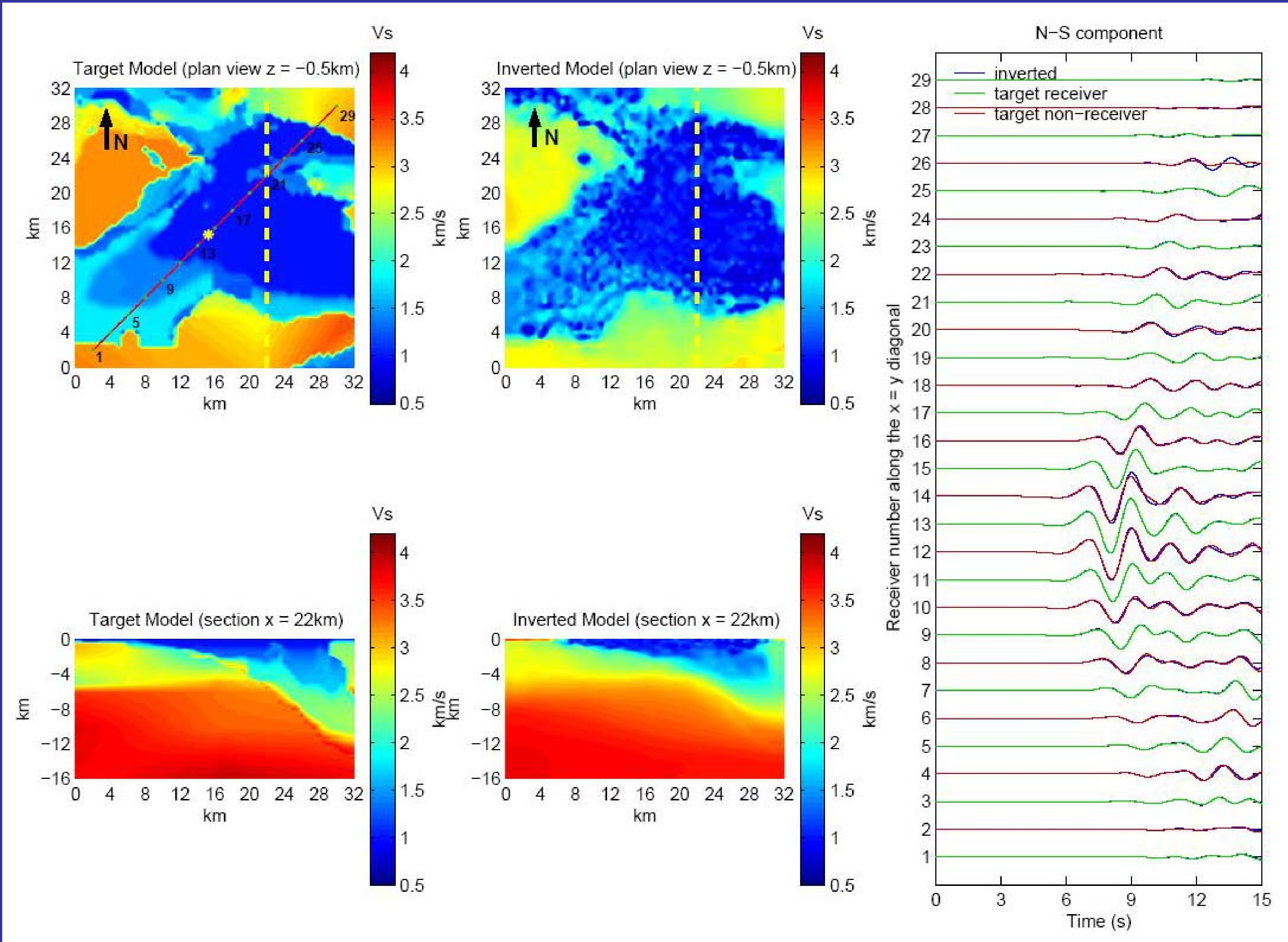


Acoustic medium, p-wave velocity

Elastic medium, s-wave velocity







Target Model (plan view z = −0.5km)
Inverted Model (plan view z = −0.5km)
Target Model (section x = 22km)
Inverted Model (section x = 22km)
Vs
km/s
km
N
N−S component
inverted
target receiver
target non−receiver
Receiver number along the x = y diagonal
Time (s)

Prospects for 3D elastodynamic inversion with observations from multiple events?

- 4x2 earthquake simulations per CG iteration
- 20 CG iterations per Gauss-Newton iteration
- 20 Gauss-Newton iterations
- Inversion costs 4000x a single forward simulation
- Assume sustained petaflops machine has 100,000 8X faster CPUs (than Lonestar's Woodcrest)
- Inverse algorithm can absorb 50x increase in CPUs (assuming network keep up with faster processors; granularity will be 70k element/CPU)
- Therefore inverse problem can be solved in 4000*15hr/400 or ~ 1 week on 8 Pflops machine

Overall remarks:

- Multilevel continuation forces successive iterates to remain within basin of attraction of global minimum
- Total variation regularization very effective at localizing sharp material interfaces
- Outer and inner iterations are mesh-independent, once nonlinearities have been resolved
- Algorithmic, parallel, and overall scalability follow
- Despite algorithmic and parallel scalability, number of forward/adjoint solutions is large (equivalent to ~800 wave propagations for 129^3 grid)
- Multigrid preconditioner is promising
- High-fidelity inverse earthquake modeling w/multiple earthquake sources remains a petaflops-level challenge

Ongoing and future work

- Incorporation of parallel adaptive octree grids
- GCV for regularization parameter selection
- Uncertainty estimation via inverse Hessian approximation of covariance matrix
- Incorporation of prior (SCEC community velocity model)
- Inversion for attenuation parameters
- Inversion for fault parameters
- Inversion for fault location (shape optimization problem)

Acknowledgments

- Quake Project: Forward/Inverse Earthquake Modeling in Large Basins (www.cs.cmu.edu/~quake)
  - NSF/KDI CMS-9980063, NSF/ITR ATM-0326449
  - Other Quake group members: Steve Day and Harold Magistrale (SDSU), Jonahan Shewchuk (Berkeley)
- TOPS Center: Toward Optimal Petascale Simulations (www.tops-scidac.org)
  - Supported under DOE SciDAC program
  - Collaboration with LLNL, ANL, LBNL, SNL + 5 universities
- Caliente Project: Dynamic Inversion and Control (www.cs.cmu.edu/~caliente)
  - NSF/ITR ACI-0121667
  - Collaborators: Larry Biegler (CMU), David Keyes (Columbia), Matthias Heinkenschloss (Rice), Roscoe Bartlett, Kevin Long, and Bart van Bloemen Waanders (Sandia), David Young (Boeing), Frank Fendell (TRW)
- DDDAS Project: Real-time Inversion, Prediction, and Sensor Steering
  - NSF/CNS-0540372
- Southern California Earthquake Center (SCEC)
  - EAR-0122464
- Computer Science Research Institute (CSRI), Sandia
- Institute for Scientific Computing Research (ISCR), Lawrence Livermore
- Special thanks to staff at Pittsburgh Supercomputing Center (PSC) and Texas Advanced Computing Center (TACC)
  - ASC-010025P, ASC-010036P, MCA01S002P, BCS020001P, MCA04N026P
- Special thanks to PETSc group at Argonne
  - Satish Balay, Kris Buschelman, Victor Eijkhout, William Gropp, Dmitry Karpeev, Dinesh Kaushik, Matt Knepley, Lois McInnes, Barry Smith, Hong Zhang

Collaborators


Volkan Akcelik


Aysegul Askan


Jacobo Bielak


George Biros

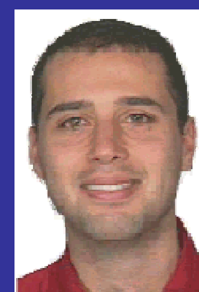

Andrei Draganescu


Ioannis Epanomeritakis


Pearl Flath


Judy Hill


Loukas Kallivokas


Julio Lopez


Kwan-Liu Ma


Dave O'Hallaron


Leonardo Ramirez-Guzman


Ricardo Taborda


Tiankai Tu


Bart van Bloemen Waanders


Karen Willcox


Hongfeng Yu

Research Group UT-Austin