Download material from github

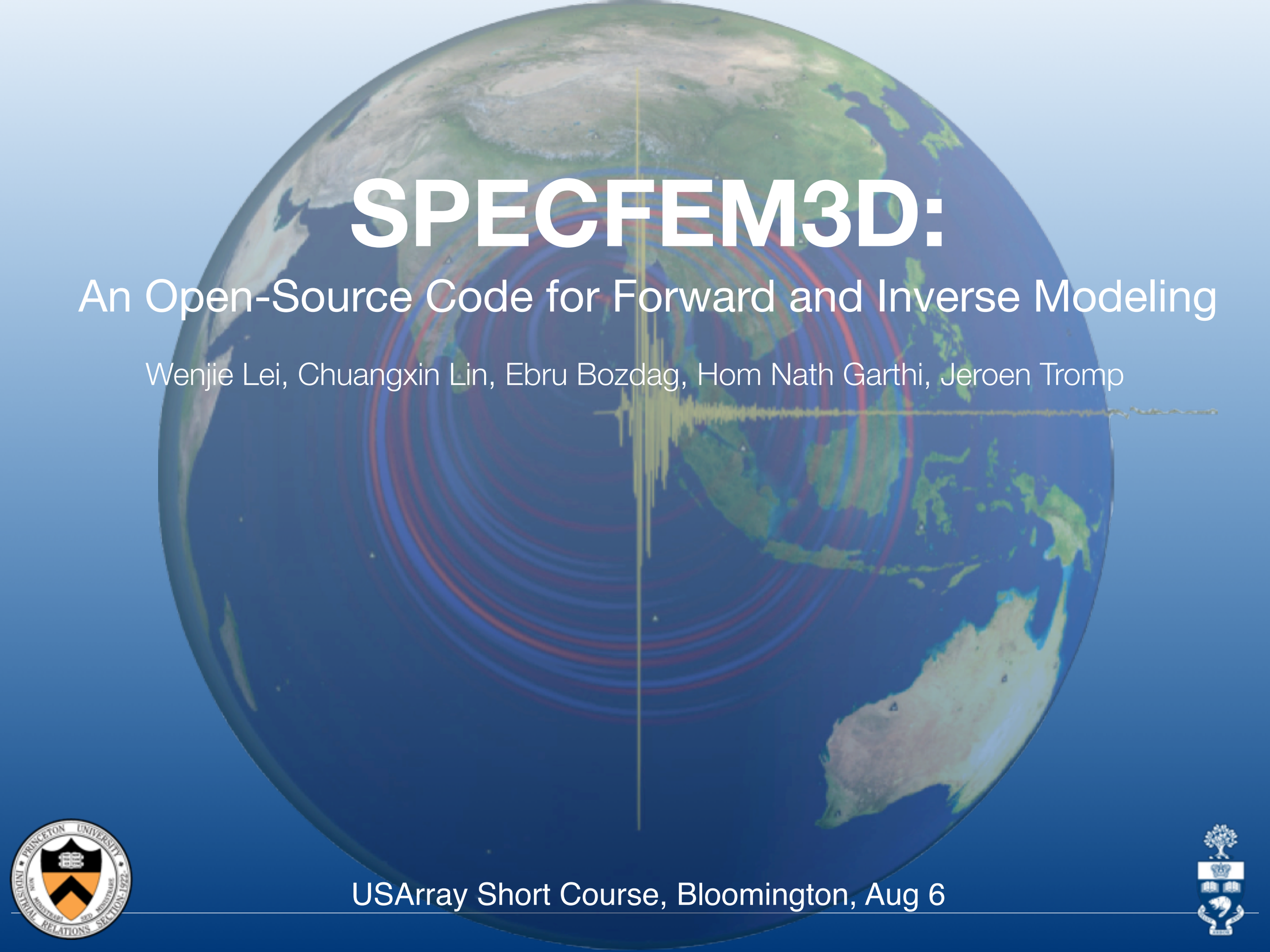
Open a terminal on your desktop. Make a new directory and change to that directory. Then type in terminal:

git clone https://github.com/wjlei1990/USArray2015_SPECFEM

SPECFEM3D:

An Open-Source Code for Forward and Inverse Modeling

Wenjie Lei, Chuangxin Lin, Ebru Bozdag, Hom Nath Garthi, Jeroen Tromp



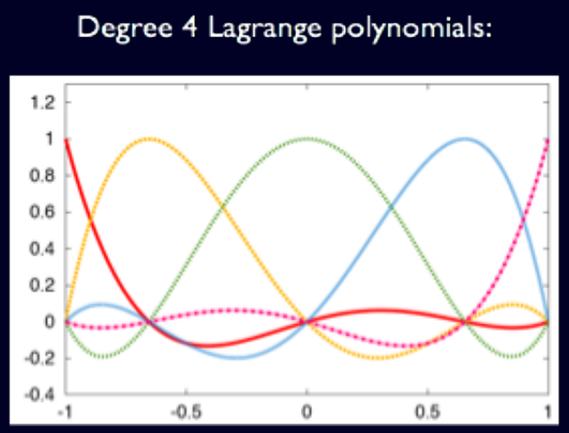
USArray Short Course, Bloomington, Aug 6

Outline

1. Introduction

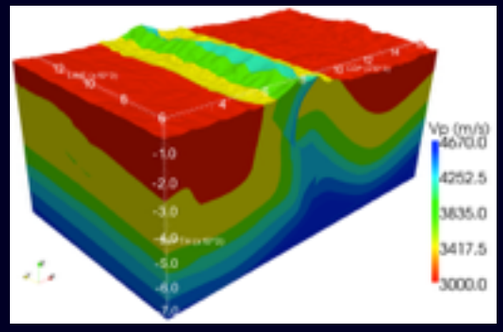
2. Basic Theory:

Spectral Element Method

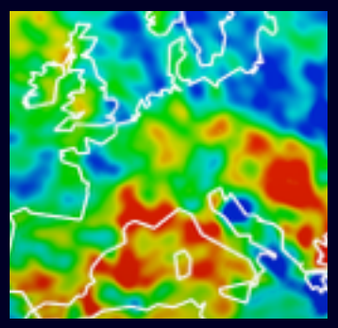


3. Forward Modeling:

SPECFEM2D&3D



4. Adjoint Tomography



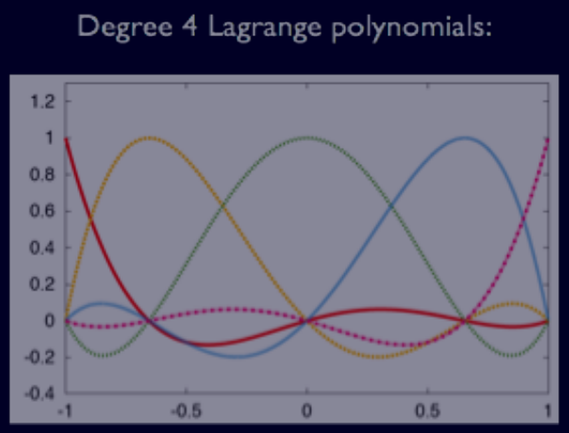
5. Summary

Outline

1. Introduction

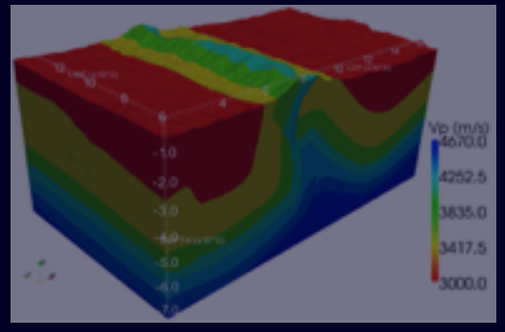
2. Basic Theory:

Spectral Element Method

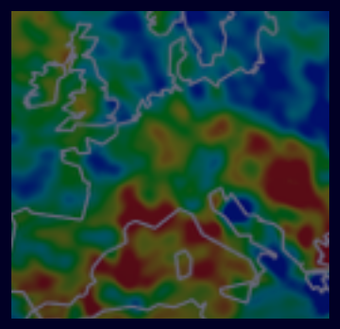


3.

SPECFEM2D



4.



5. Summary

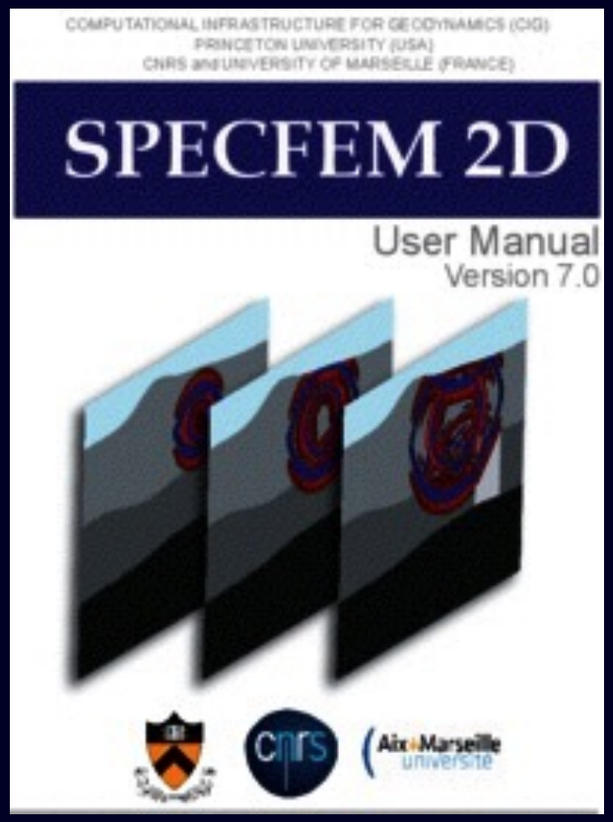
SPECFEM: 2D, 3D_Cartesian & 3D_Globe

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

github SOCIAL CODING

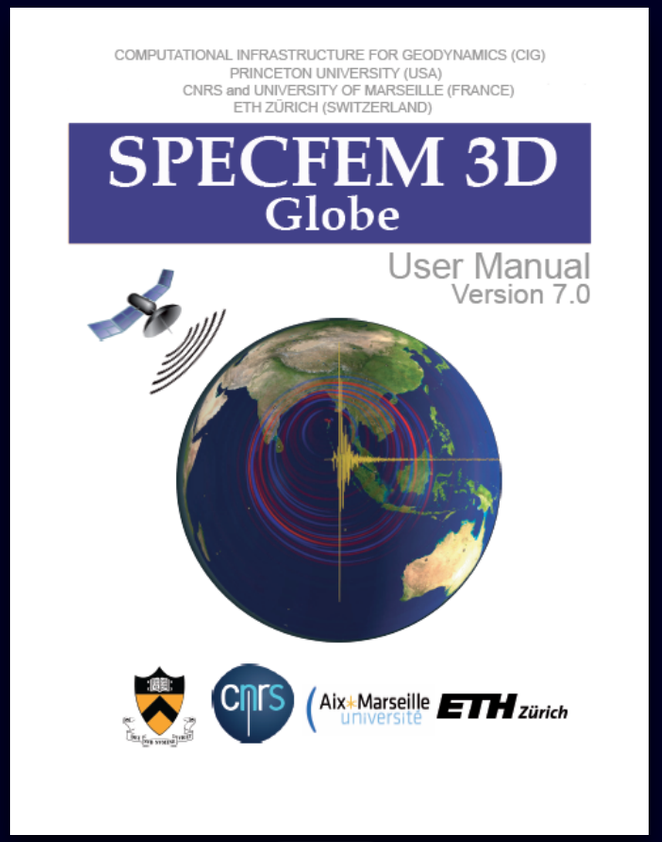
Komatitsch & Tromp 2002

2D Version



3D Cartesian

3D Globe



SPECFEM: 2D, 3D_Cartesian & 3D_Globe

SPECFEM2D:

simulates forward and adjoint seismic wave propagation in **two-dimensional** acoustic, (an)elastic, poroelastic or coupled acoustic-(an)elastic-poroelastic media, with Convolution **PML** absorbing conditions.

SPECFEM3D:

Cartesian simulates acoustic (fluid), elastic (solid), coupled acoustic/elastic, poroelastic or seismic wave propagation in any type of conforming mesh of **hexahedra**. It can, for instance, model seismic waves propagating in **sedimentary basins**.

SPECFEM3D_GLOBE:

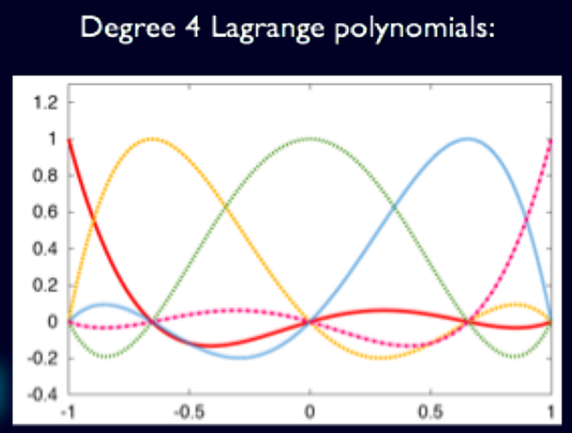
simulates **global** and **regional** (**continental-scale**) seismic wave propagation.

Outline

1. Introduction

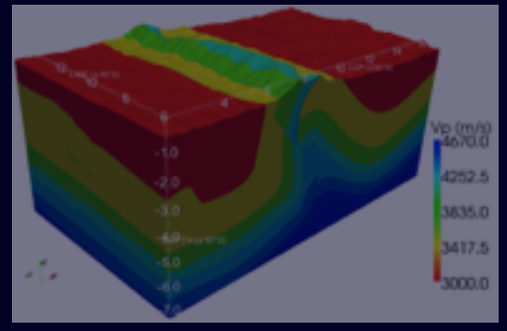
2. Basic Theory:

Spectral Element Method

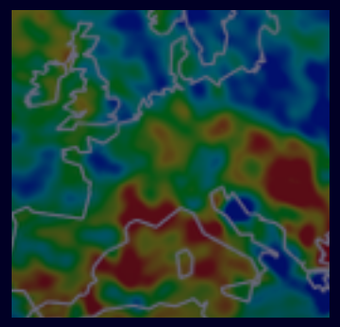


3. Forward Modeling:

SPECFEM2D&3D



4. Adjoint Tomography



5. Summary

Seismic Wave Propagation

Equation of motion: $$\rho\,\partial_t^2\mathbf{s} - \boldsymbol{\nabla}\cdot\mathbf{T} = \mathbf{f}$$

Boundary condition: $$\hat{\mathbf{n}}\cdot\mathbf{T} = \mathbf{0}$$

Hook's Law: $$T = c : \varepsilon$$

Initial conditions: $$\mathbf{s}(\mathbf{x},0) = \mathbf{0}$$
$$\partial_t\mathbf{s}(\mathbf{x},0) = \mathbf{0}$$

Earthquake source: $$\mathbf{f} = -\mathbf{M}\cdot\boldsymbol{\nabla}\delta(\mathbf{x}-\mathbf{x}_s)S(t)$$

Numerical Method

Strong Form

$$L(u) = 0$$

Weak Form

$$\int w\, L(\tilde{u})\, \mathrm{d}\Omega = 0$$

Finite Difference Method

$$\int w\, L(\tilde{u})\, \mathrm{d}\Omega = 0$$

$$\tilde{u} = u_\alpha \quad \text{and} \quad w = \delta_{\alpha\beta}$$

$$L(u_\beta) = 0$$

Solving strong form on discrete points

Finite Element Method

FEM

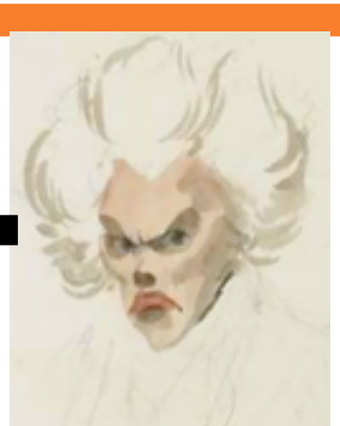
$$\int w\, L(\tilde{u})\, \mathrm{d}\Omega = 0$$

$$\tilde{u} = \sum_{\alpha=1}^{n} N_\alpha u_\alpha \quad \text{and} \quad w = N_\alpha$$

N_α : Interpolation function

$$\int N_\alpha\, L \left(\sum_{\beta=1}^{n} N_\beta u_\beta \right) \mathrm{d}\Omega = 0$$

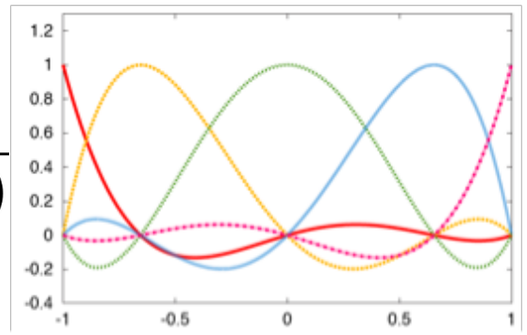




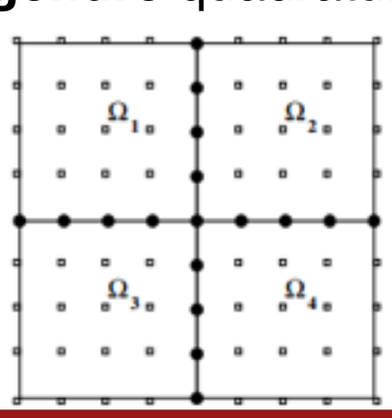
Spectral FEM

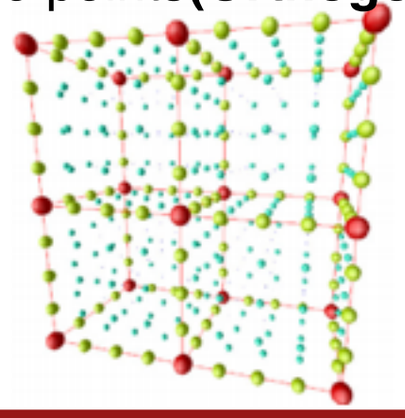
1. use **Lagrange polynomials** as interpolation function

$$N_\alpha(\xi) = \prod_{\substack{\beta=1 \\ \beta \neq \alpha}}^{n} \frac{(\xi - \xi_\beta)}{(\xi_\alpha - \xi_\beta)}$$



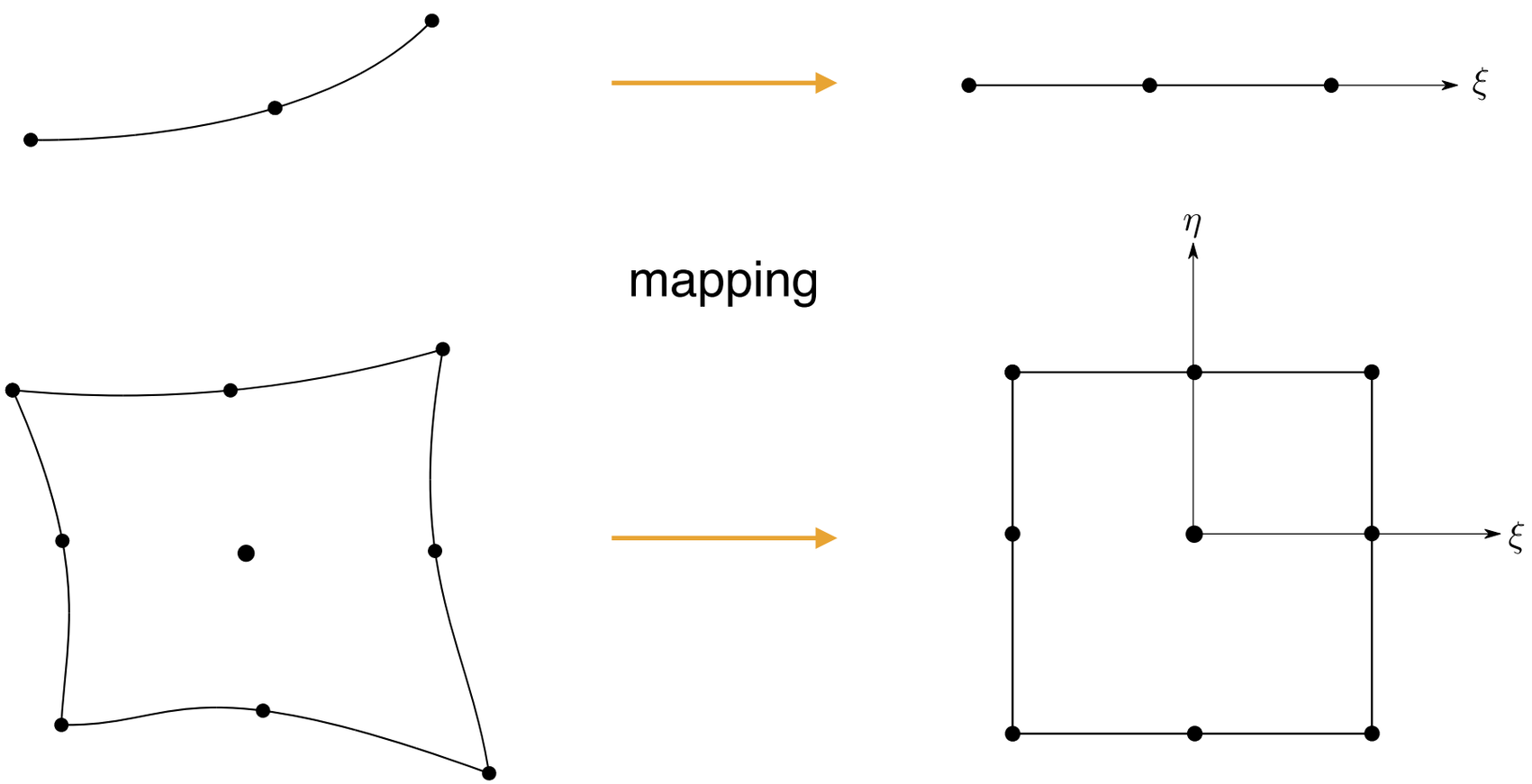
2. use nodes located at **Gauss-Lobattto-Legendre** quadrature points**(Orthogonality)**





Finite Element Method

Integration on the original domain is often difficult, so….



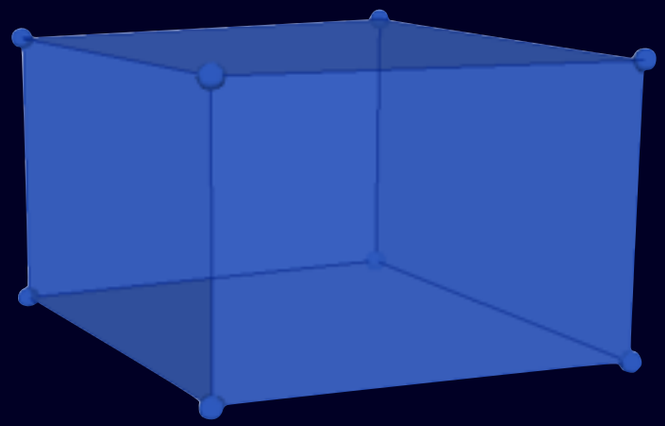
$$(x, y) \rightarrow (\varepsilon, \eta)$$

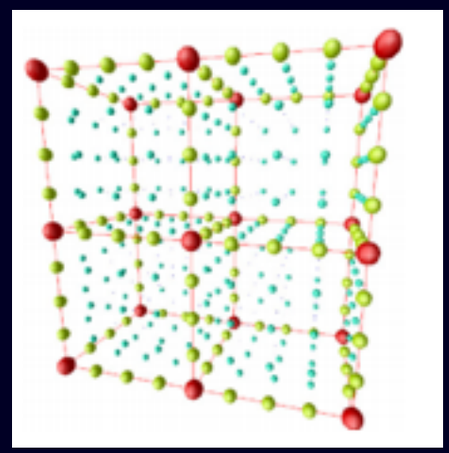
$$dxdy = | J | d\varepsilon d\eta$$

Summary of Spectral-Element Method

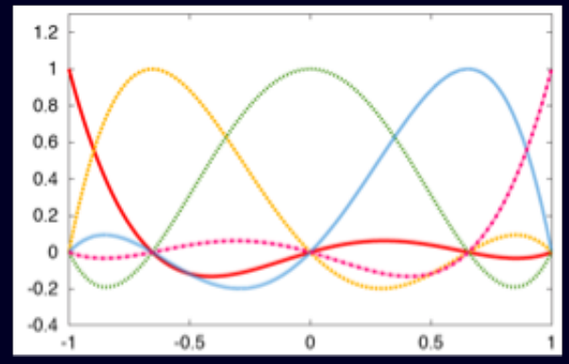
Finite-elements:

- hexahedral elements





- Gauss-Lobatto-Legrendre quadrature



- diagonal mass matrix

- explicit time-marching scheme

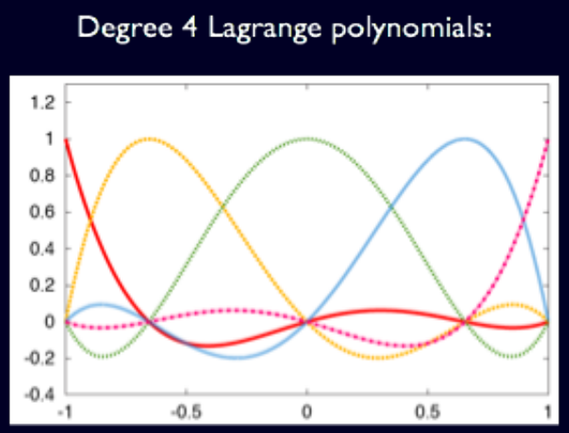
$$\mathsf{M}\ddot{\mathsf{U}} = -\mathsf{K}\mathsf{U} + \mathsf{F}$$

Outline

1. Introduction

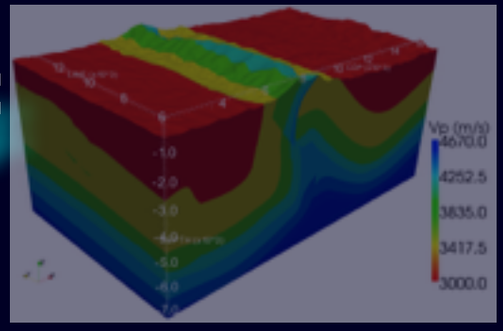
2. Basic Theory:

Spectral Element Method

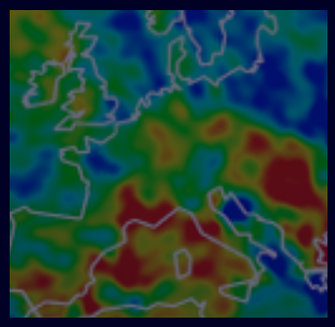


3. Forward Modeling:

SPECFEM2D&3D

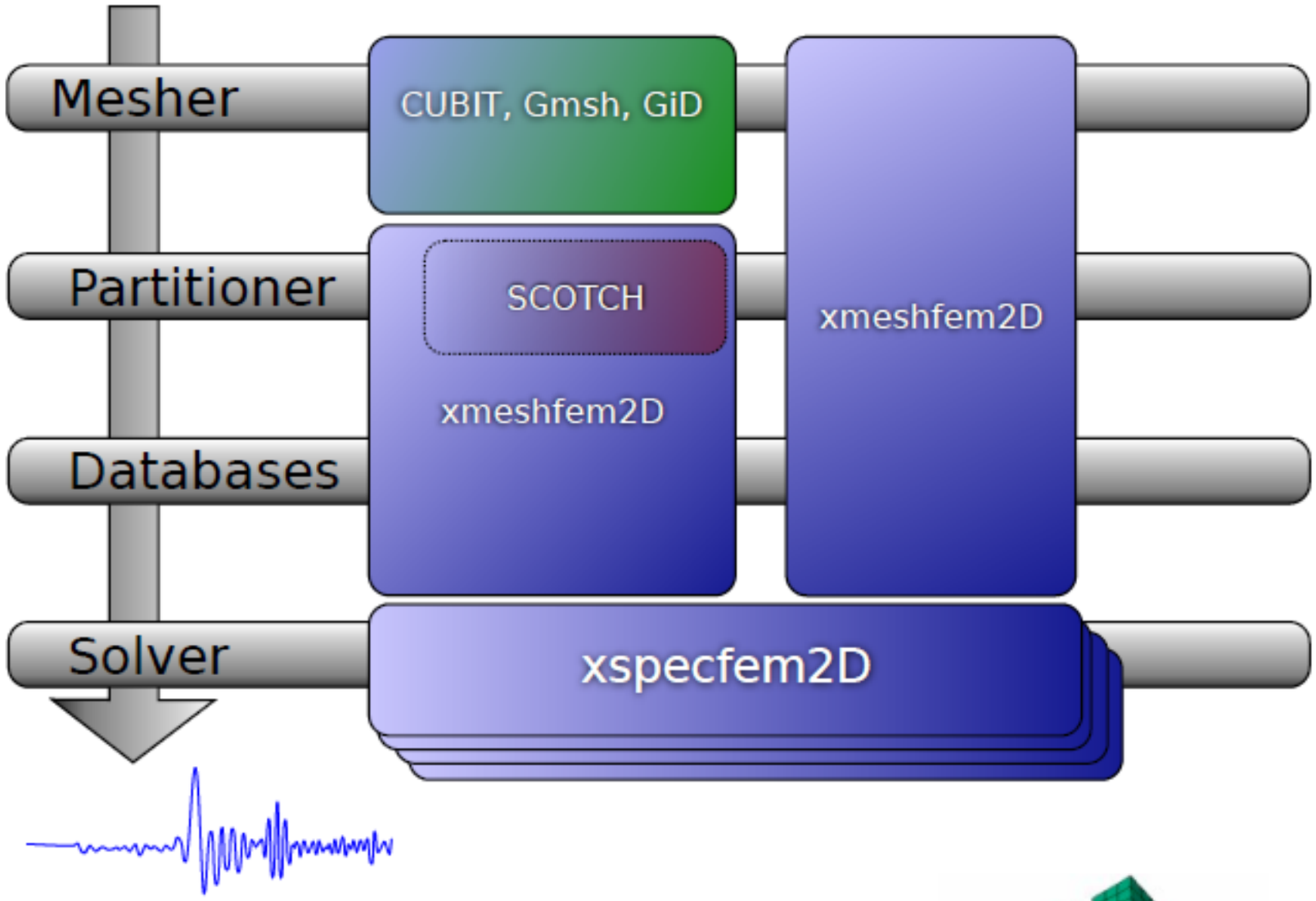


4. Adjoint Tomography



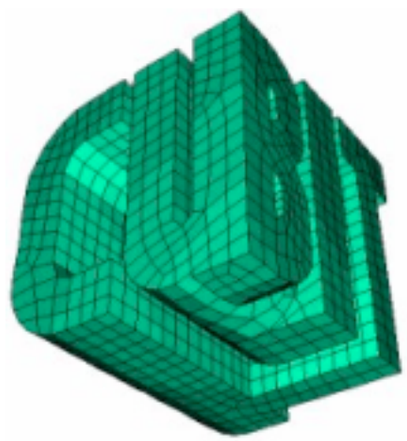
5. Summary

Forward simulation Workflow

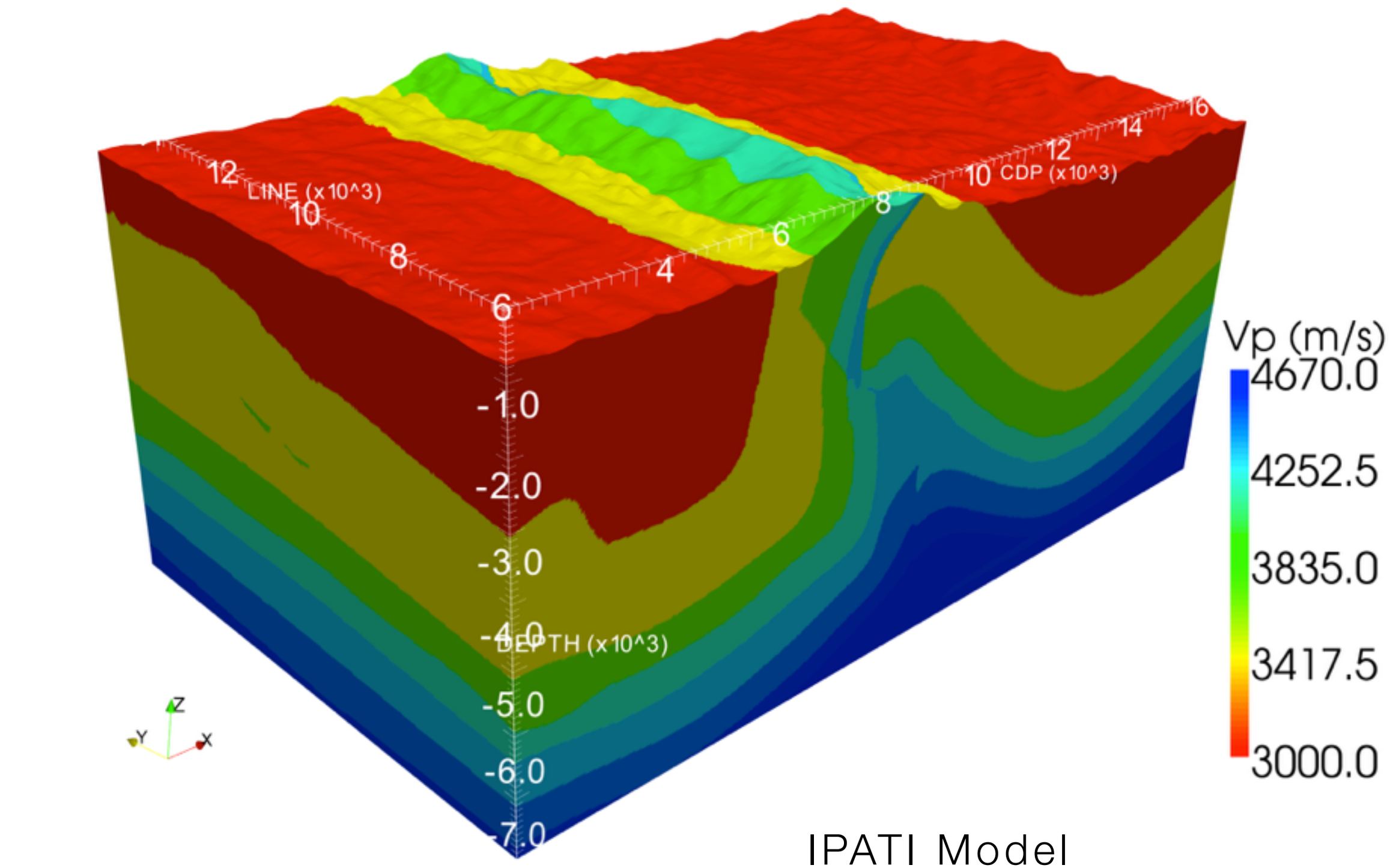


External Mesher: Cubit, Gmsh

Internal Mesher: xmeshfem2d



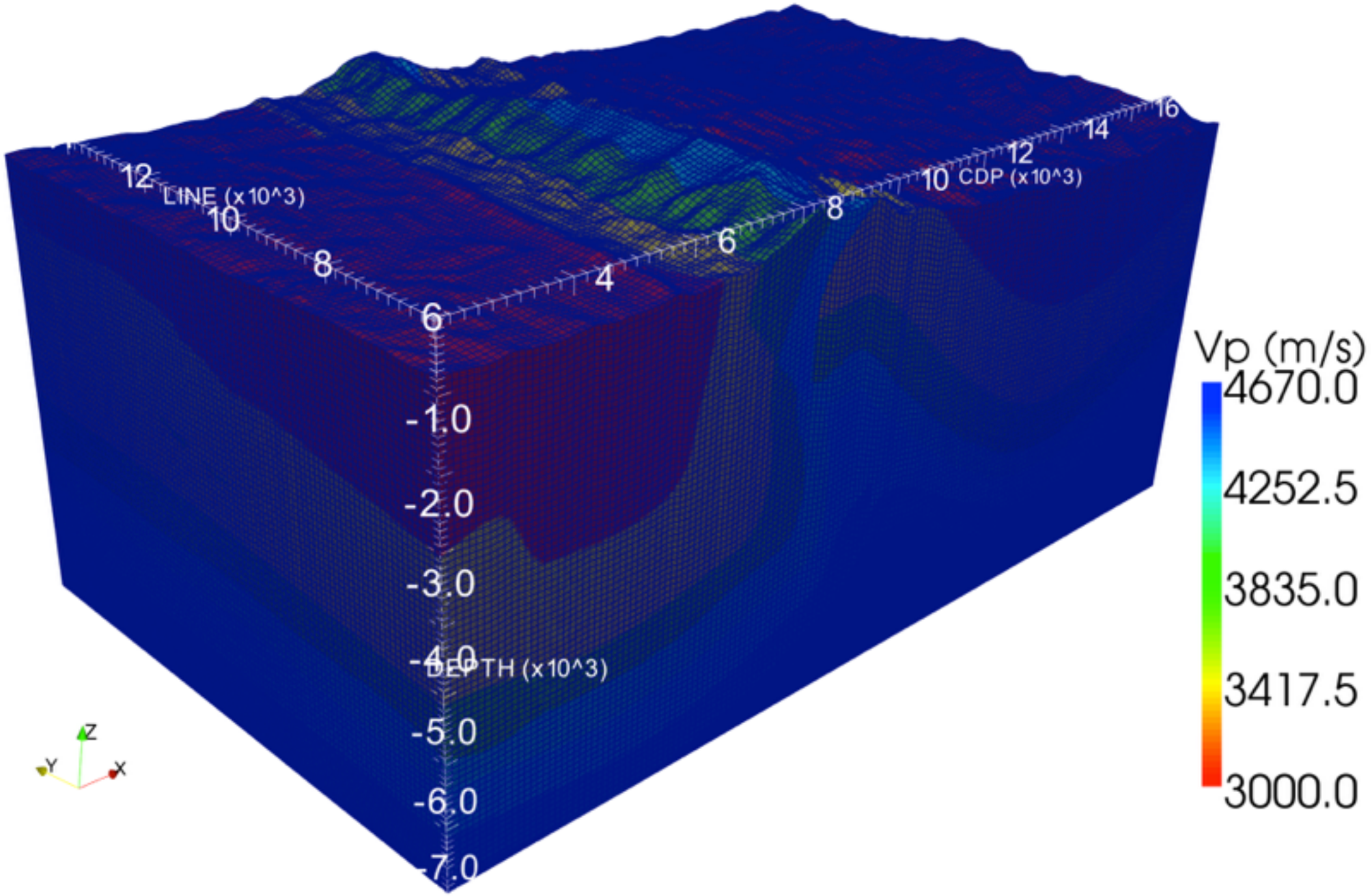
Mesh: example



IPATI Model

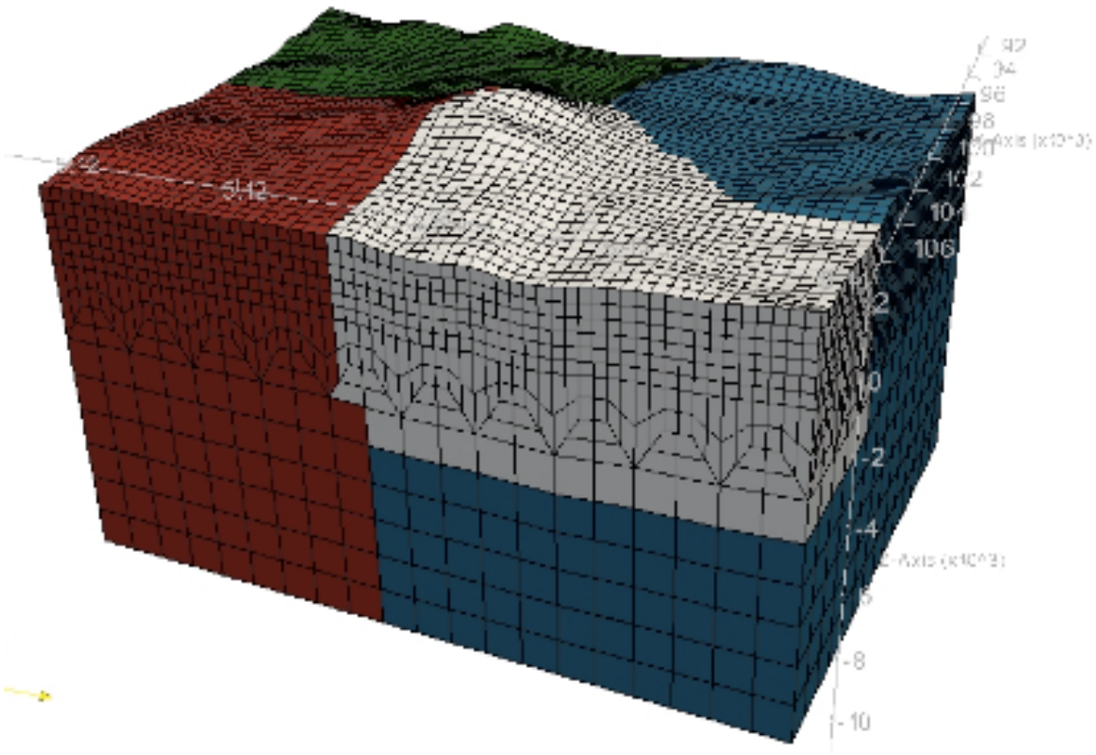
Yang Luo

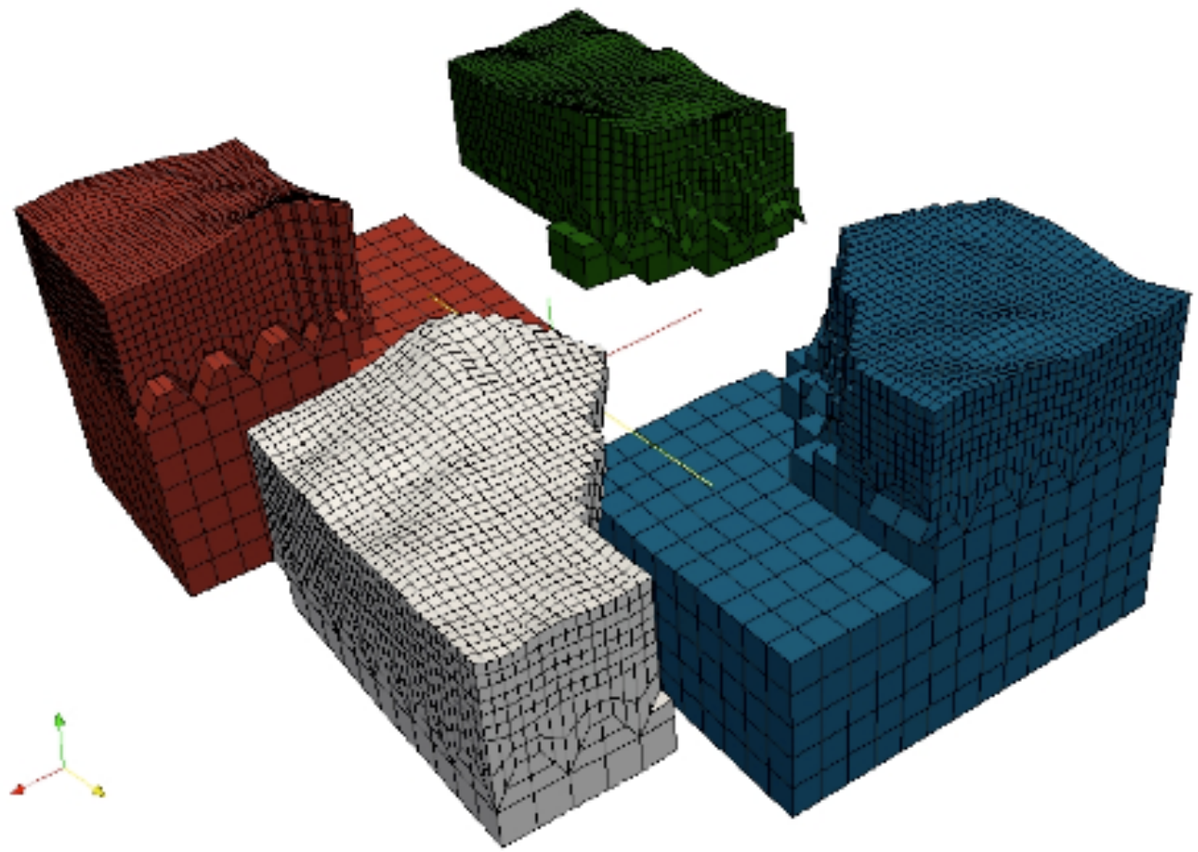
Mesh example



Yang Luo

Mesh: mesh and Partition

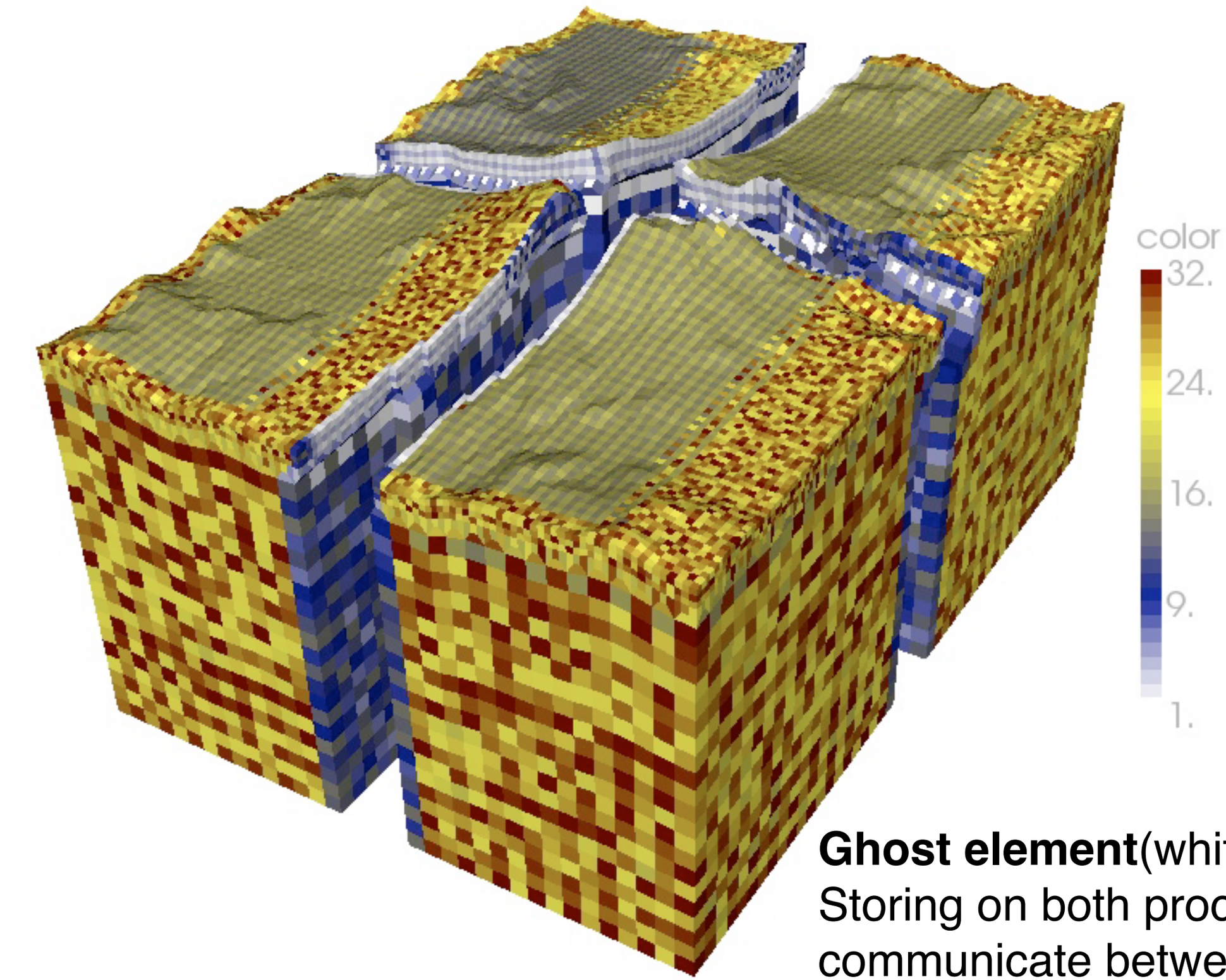




Mesh: Partition and Parallel running(MPI buffer)

Ghost element(white color):
Storing on both processors and communicate between each other

Mesh: Partition and Parallel running(MPI buffer)



Ghost element(white color):
Storing on both processors and communicate between each other

Stability and accuracy condition

Mesh size:

element size <= minimum wave length / 2

(the minimum wavelength is usually decided by S wave if it is a elastic problem)

Time marching step:

The parameter `DT` sets the length of each time step in seconds. The value of this parameter is crucial for the stability of the spectral-element simulation. Your time step `DT` will depend on the minimum ratio between the distance h of neighboring mesh points and the wave speeds v defined in your model. The condition for the time step Δt is:

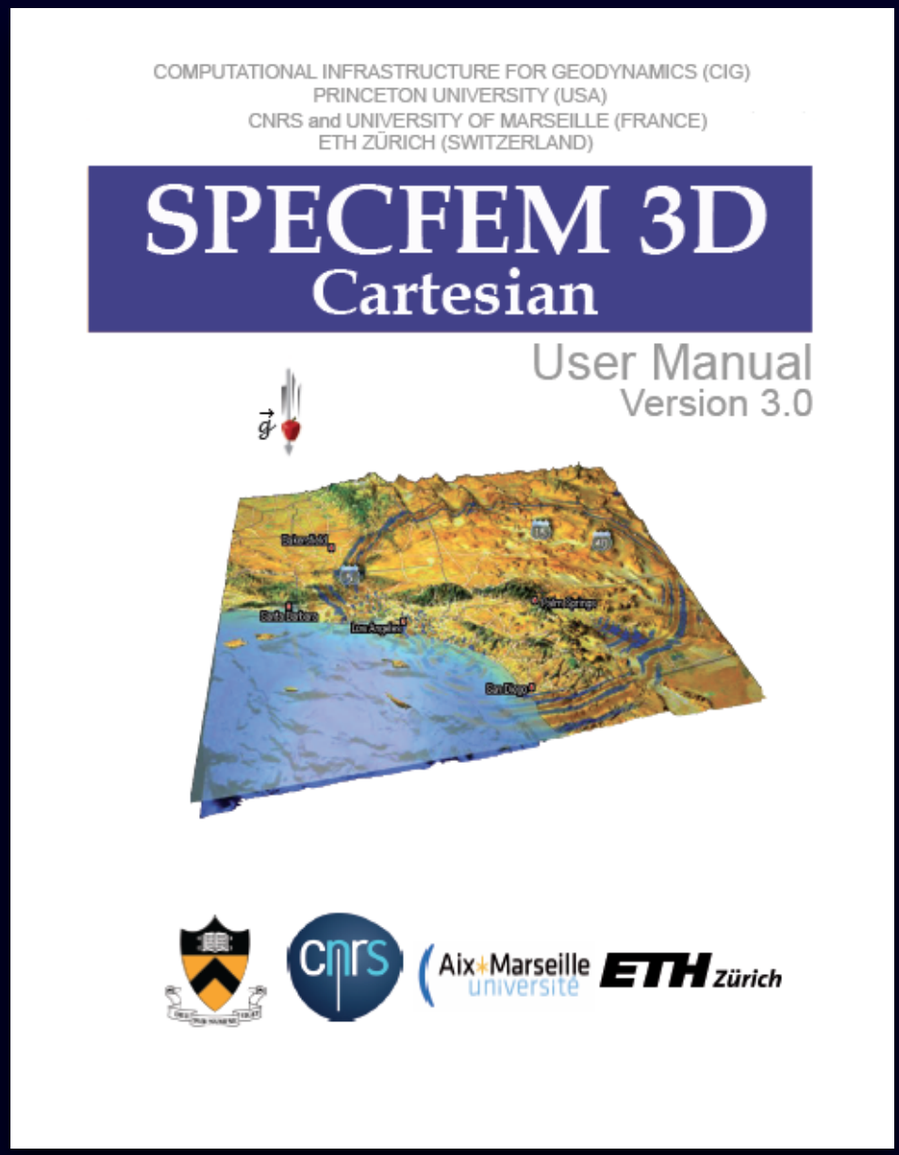
$$\Delta t < C \ \min_{\Omega}(\ h/v \ )$$

where C is the so-called Courant number and Ω denotes the model volume. The distance h depends on the mesh element size and the number of GLL points `NGLL` specified in the main constants file `constants.h` located in the `src/shared/` subdirectory. The wave speed v is determined based on your model's P- (or S-) wave speed values.

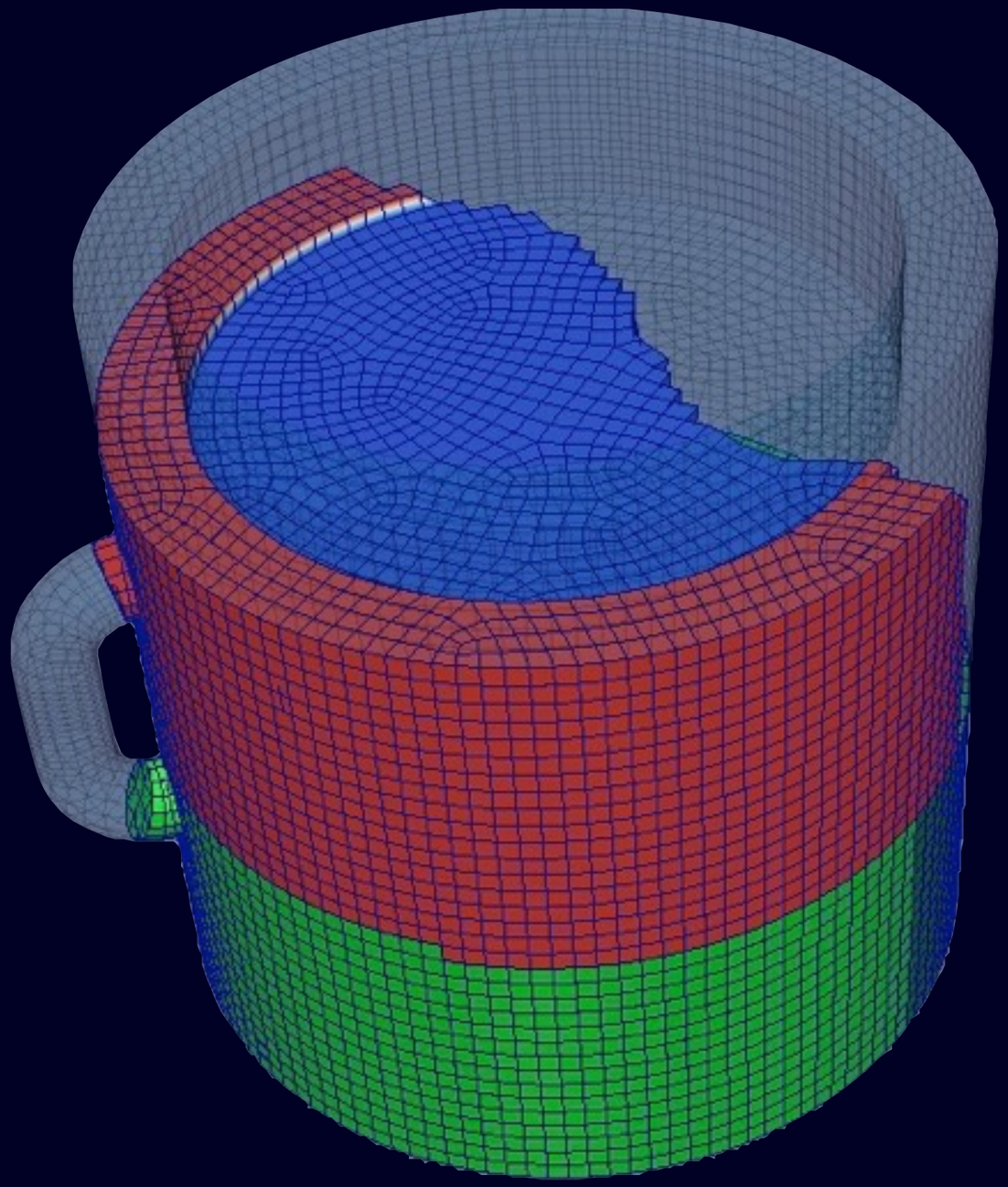
Open Source Community Software

SPECFEM3D

- Unstructured meshes (CUBIT)
- Load-balanced mesh partitioning
- Acoustic & elastic coupling, poroelasticity
- Anisotropy
- Attenuation
- Adjoint kernels
- GPU capable

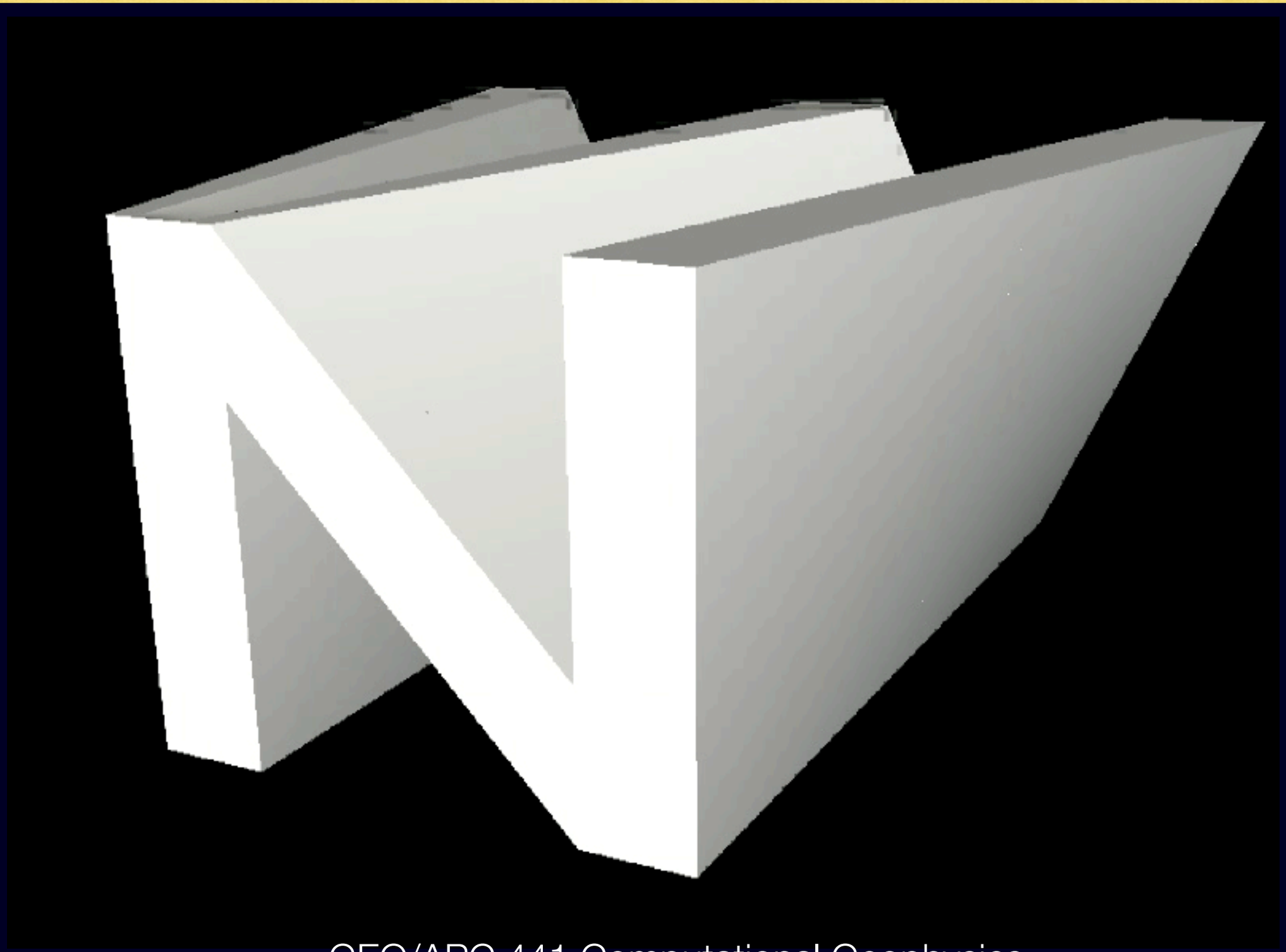


Solver: Coffee cup simulation



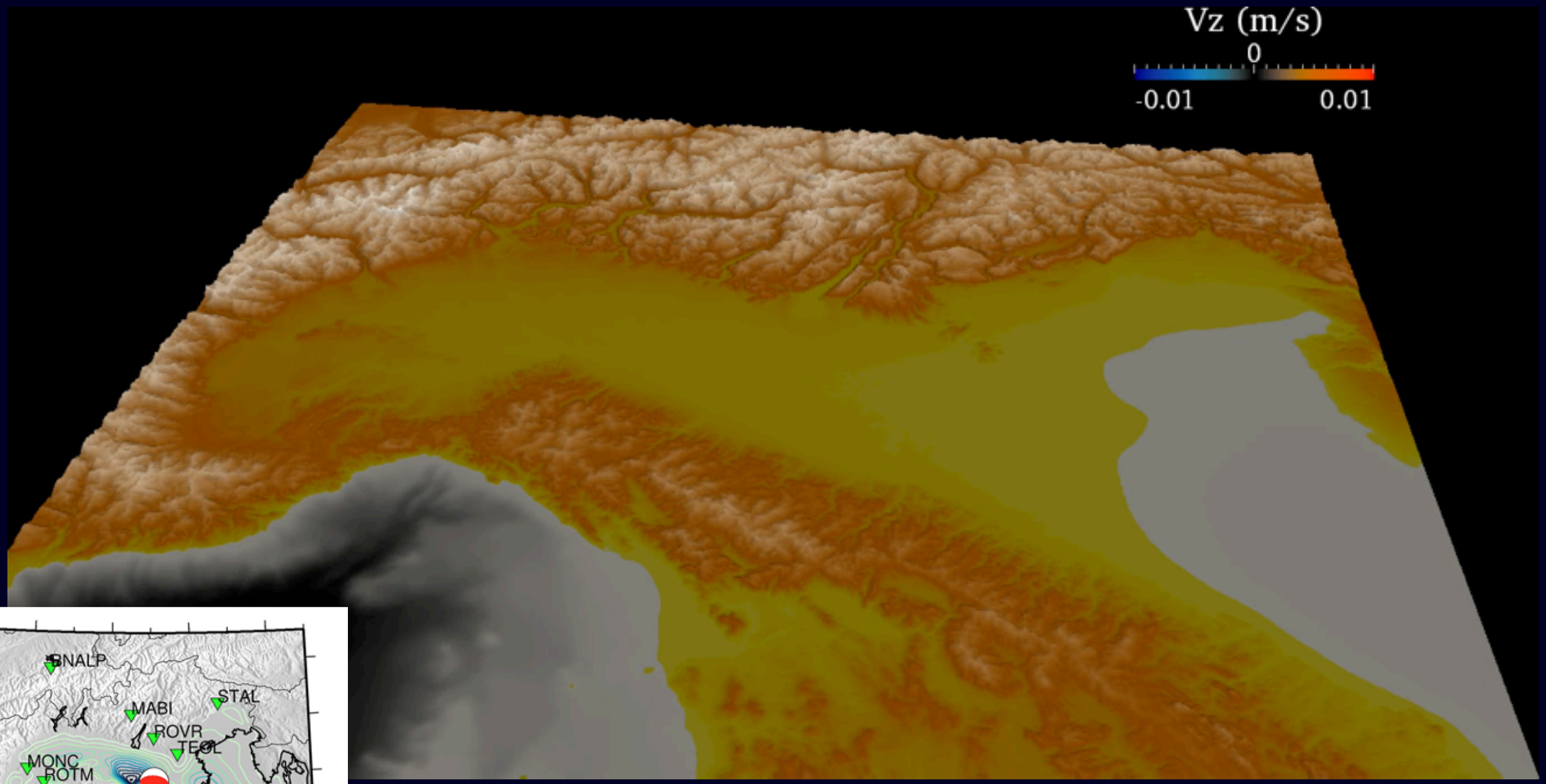
SPECFEM3D “Sesame”

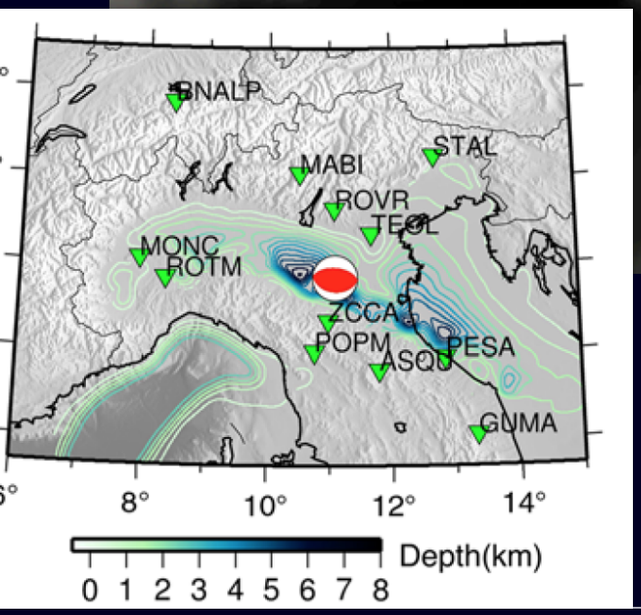
Solver: SPECFEM & Cubit in Education



GEO/APC 441 Computational Geophysics

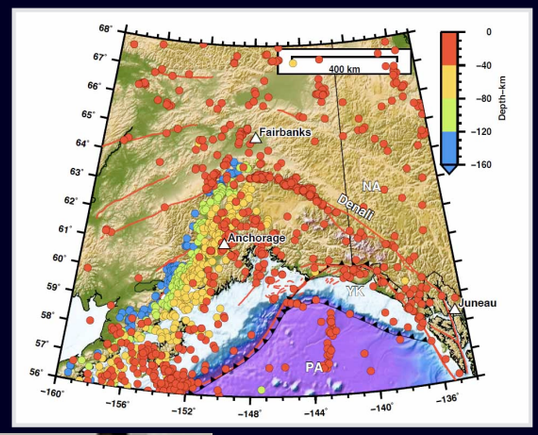
Solver: Basin analysis: Northern Italy

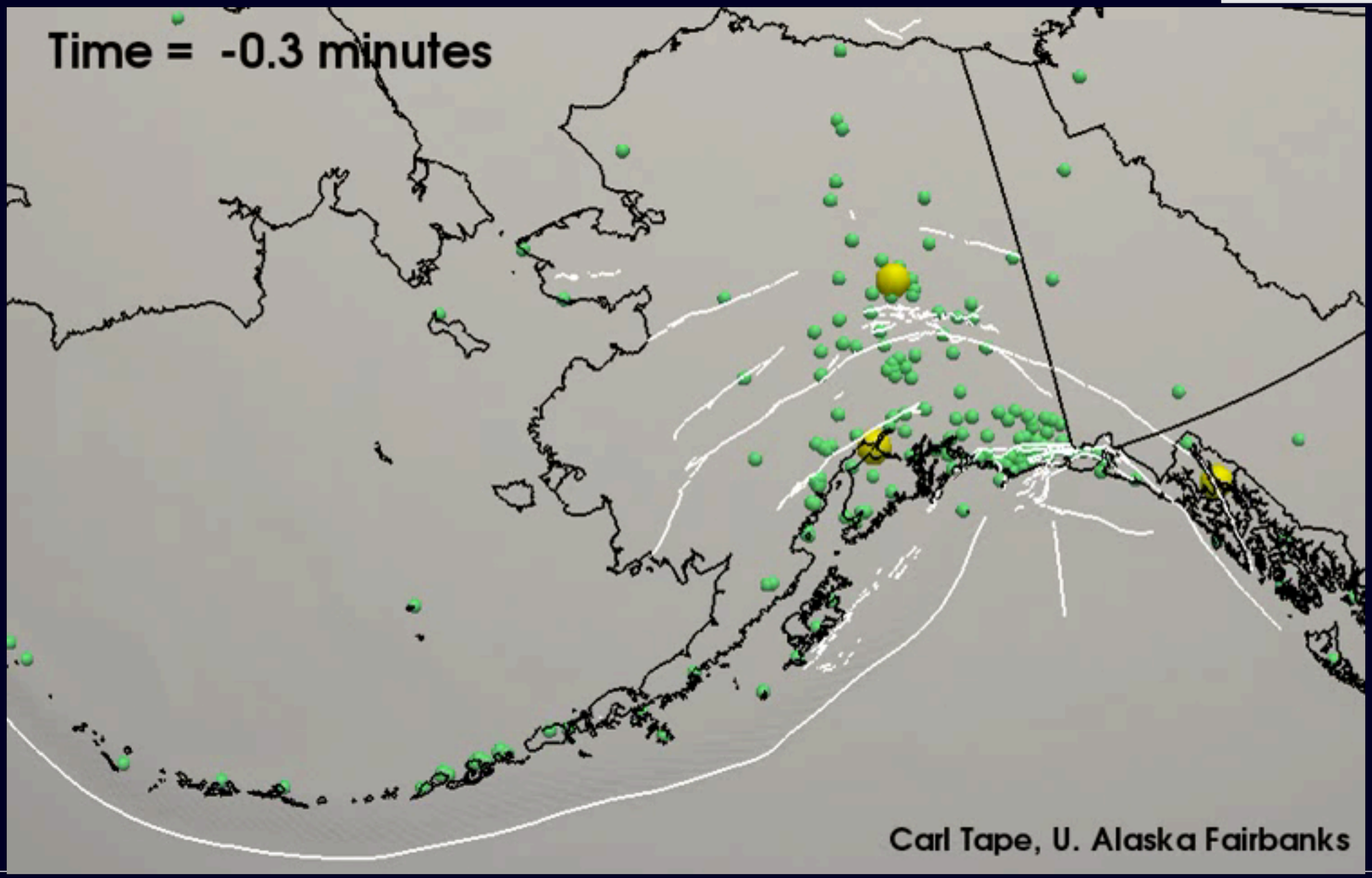




Solver: Finite-source simulations

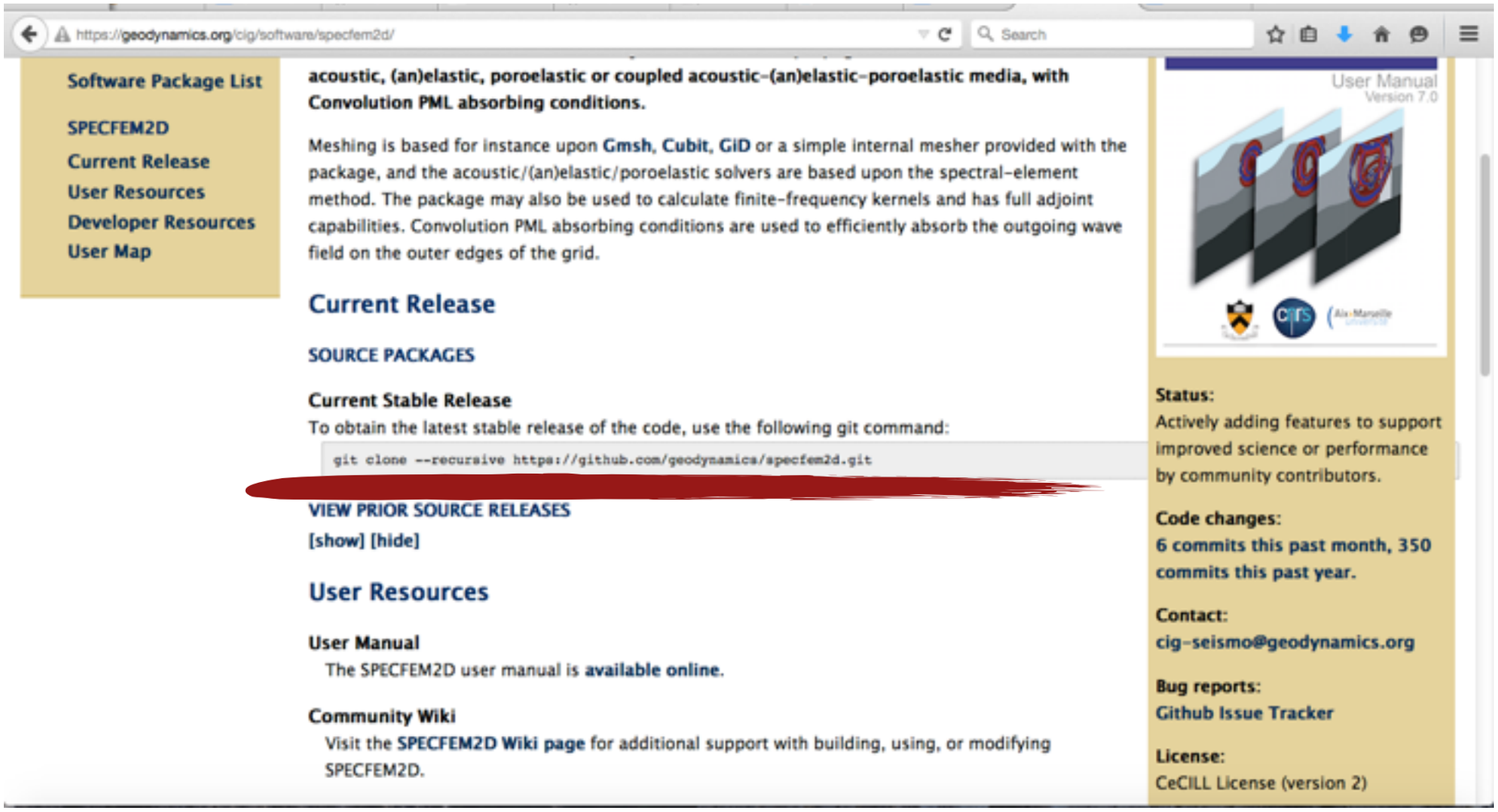
1964 Alaska Earthquake





Specfem2d running session(on your desktop)

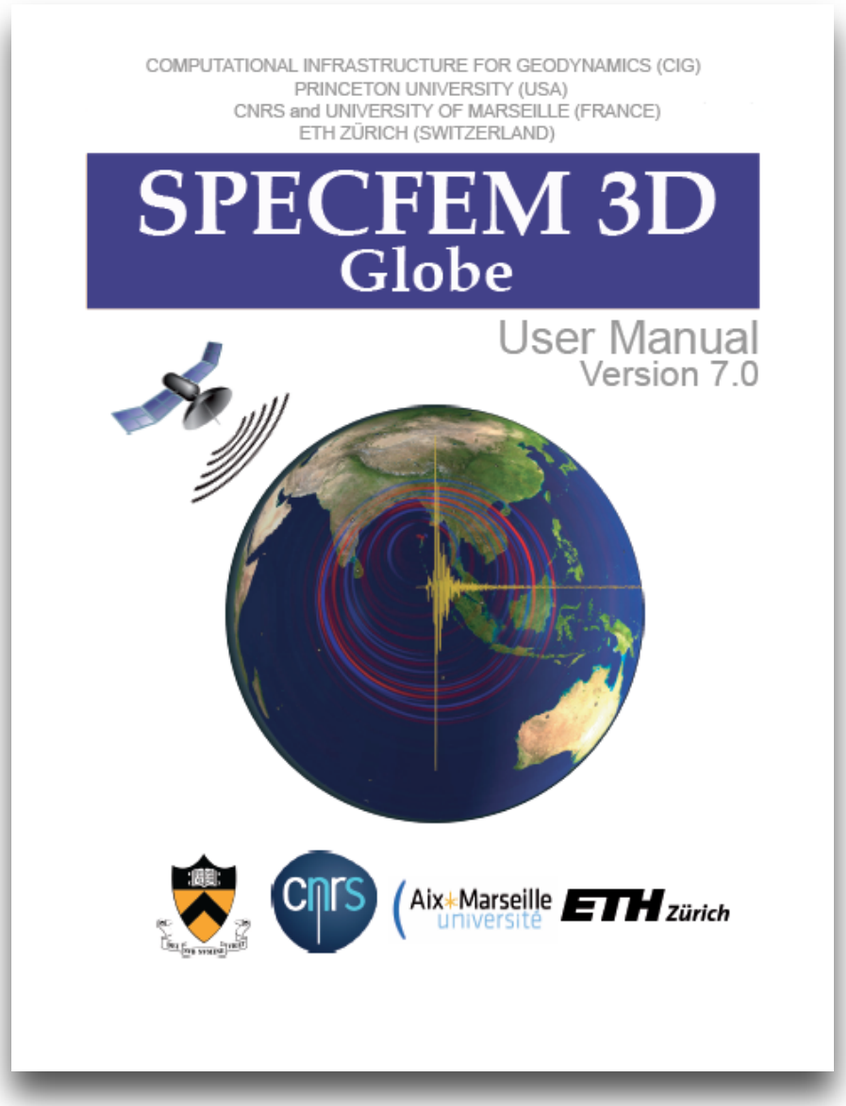
1. Open a terminal and make a **new directory** at your desktop where you want to put the specfem2d code;
2. Google “specfem2d” and find the CIG webpage
3. find the instructions to download the code;



4. Change directory(command: cd) to the your directory(you just made) and type in the git command:

git clone --recursive https://github.com/geodynamics/specfem2d.git

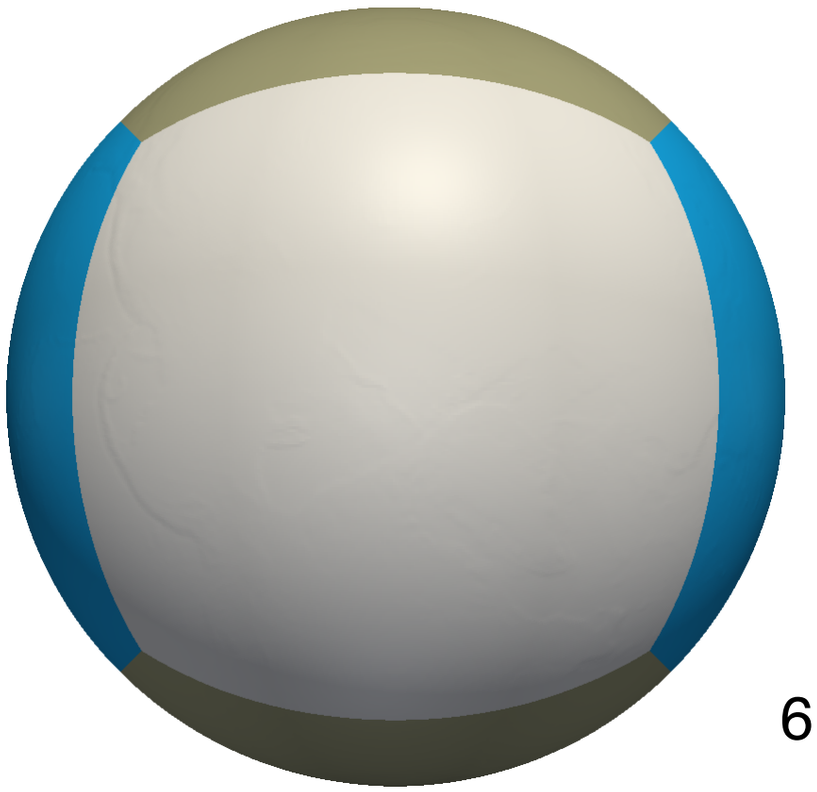
Specfem3D Globe



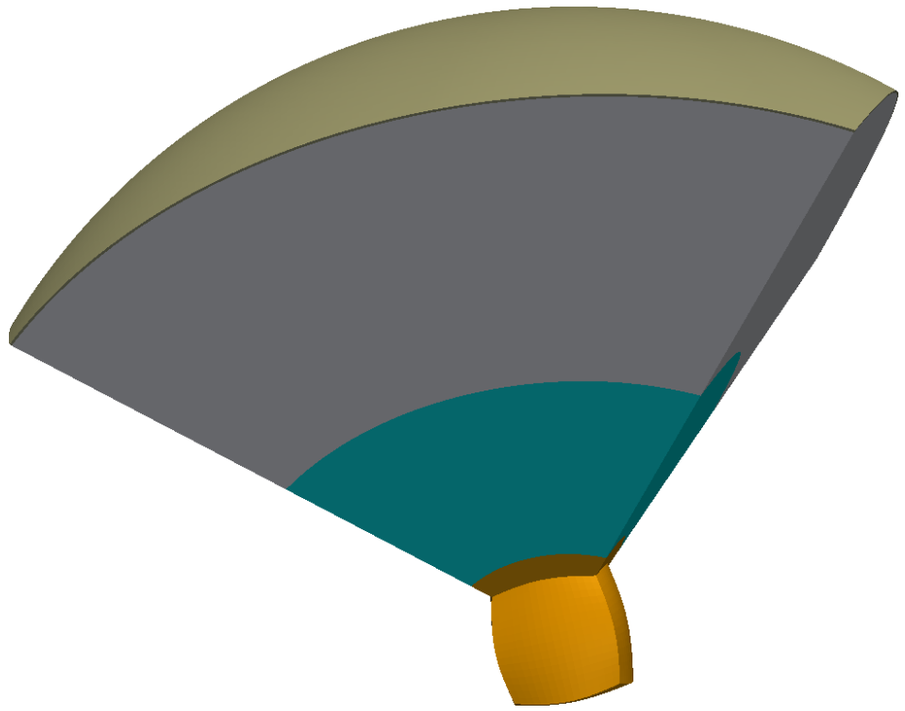
Continental/global simulation

- Attenuation
- Ellipticity
- Rotation
- Gravity
- Ocean
- Topography
- Anisotropy
- Adjoint kernels
- GPU capable

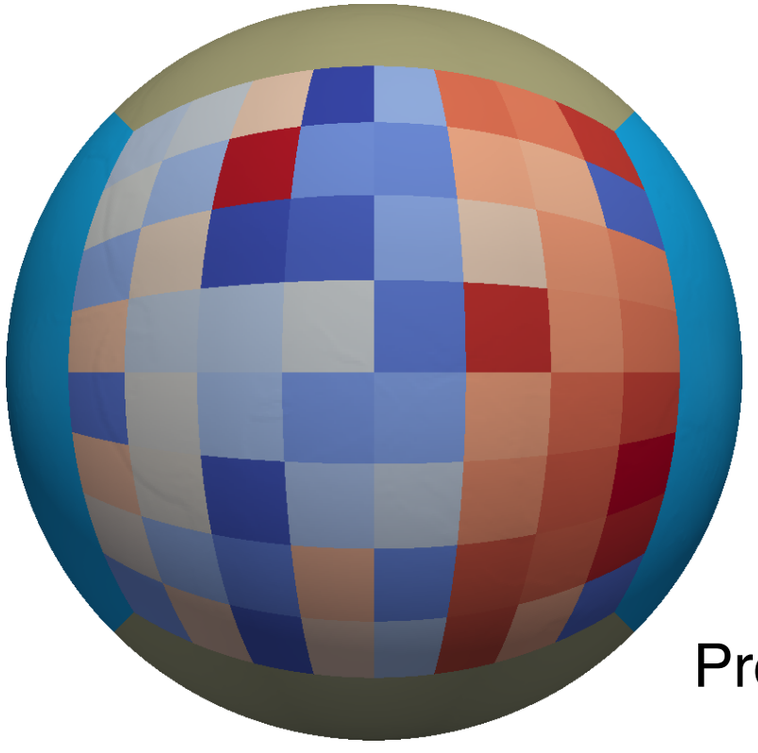
Global Mesher: mesh & partition



6 Chunks

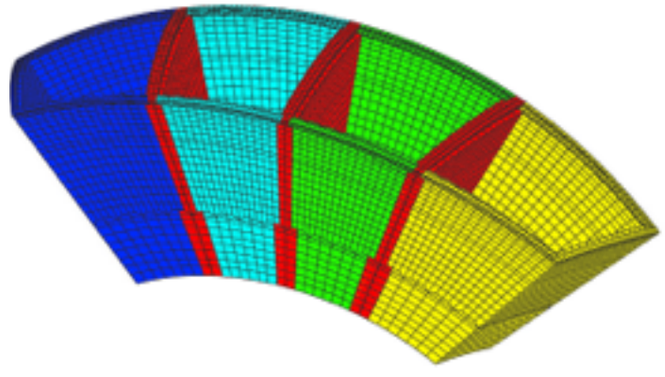


Chunks Partition



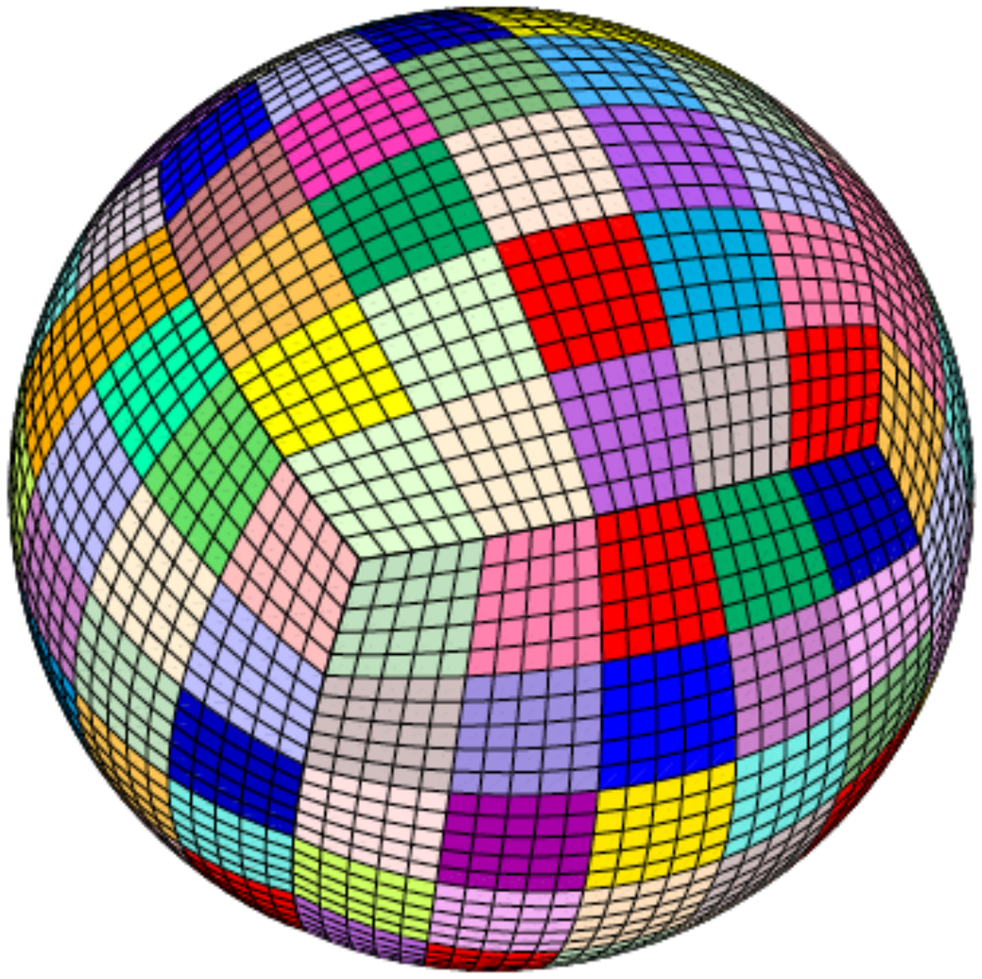
Processor distr

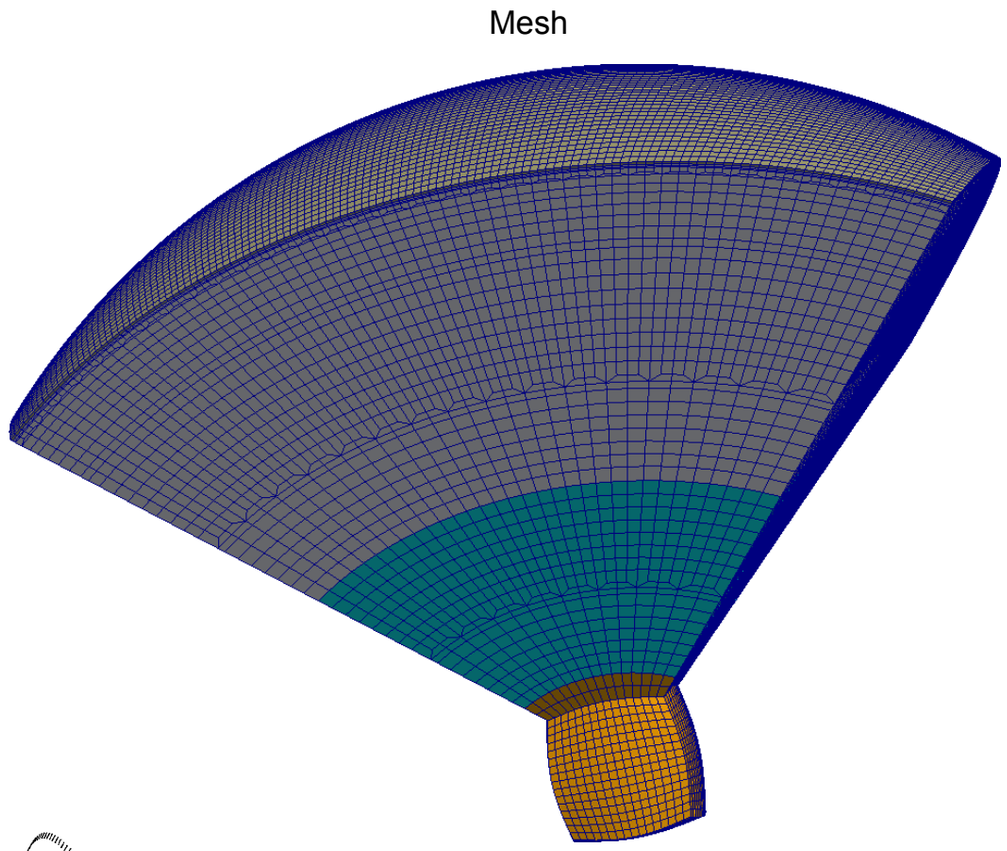
“French Fry”



Ghost element(For MPI)

Global Mesher: mesh & partition





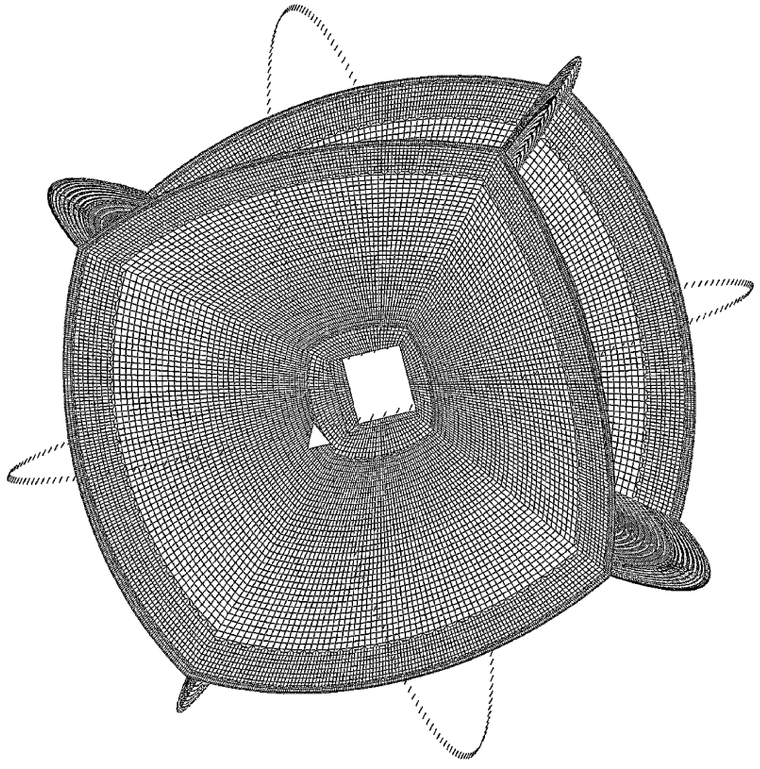
In the example above, using:

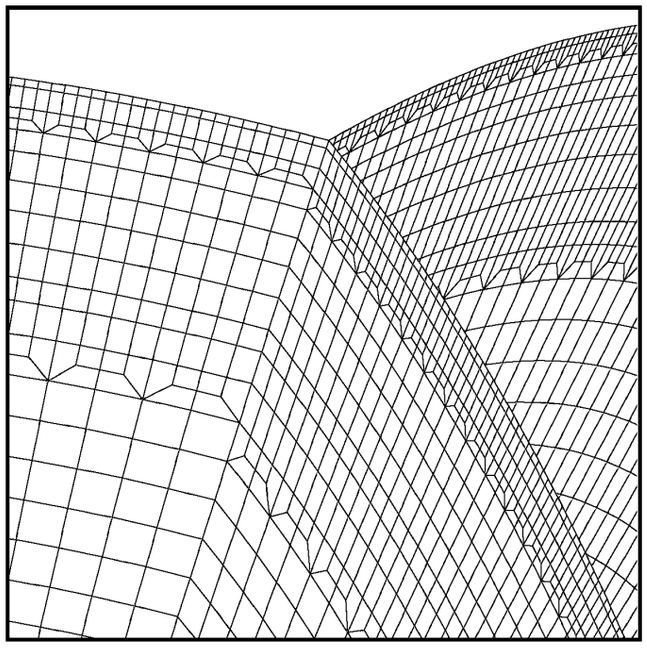
NEX_XI = 25 NEX_ETA = 25

NPROC_XI = 5 NPROC_ETA = 5

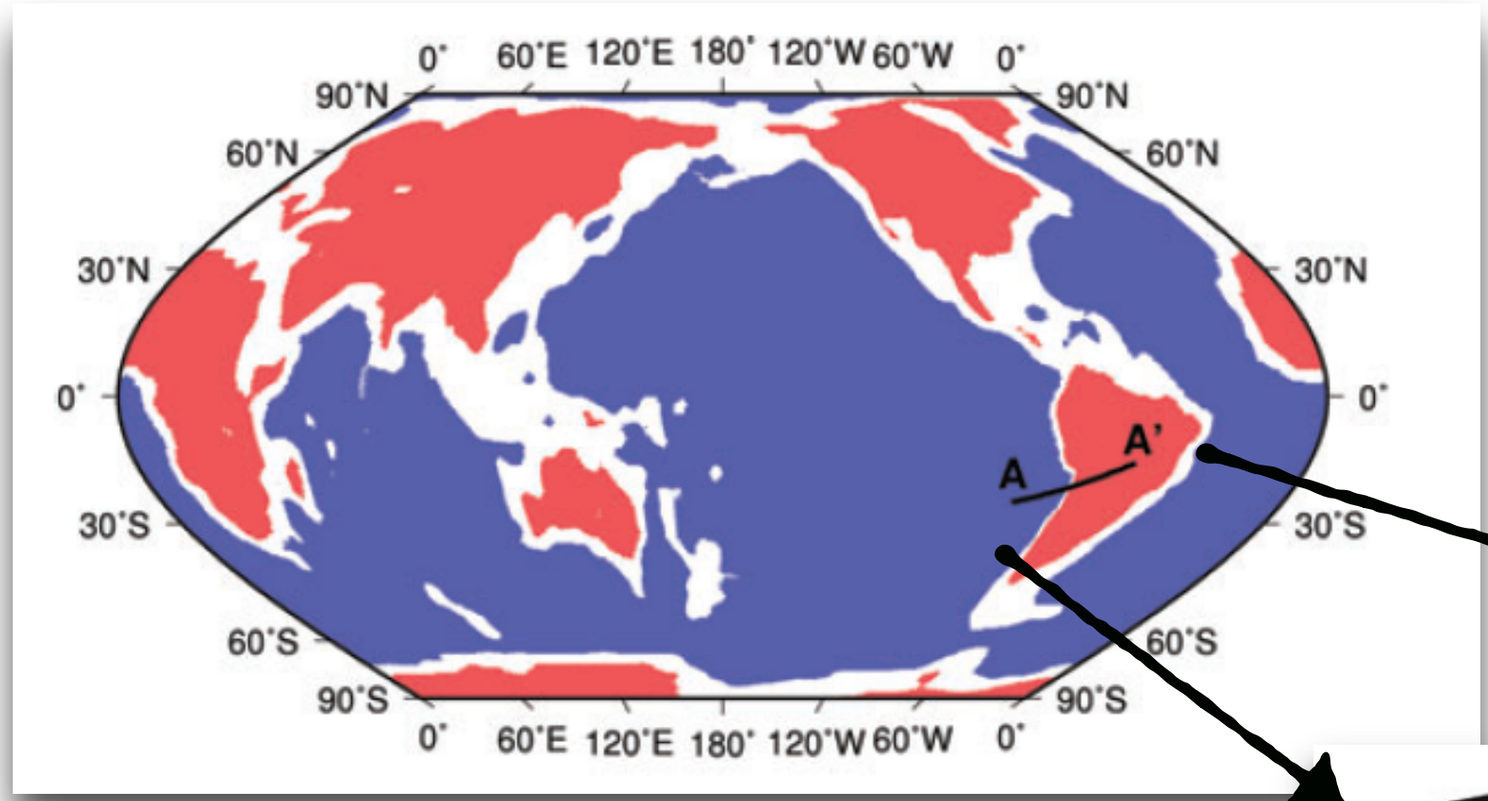
NCHUNKS = 6

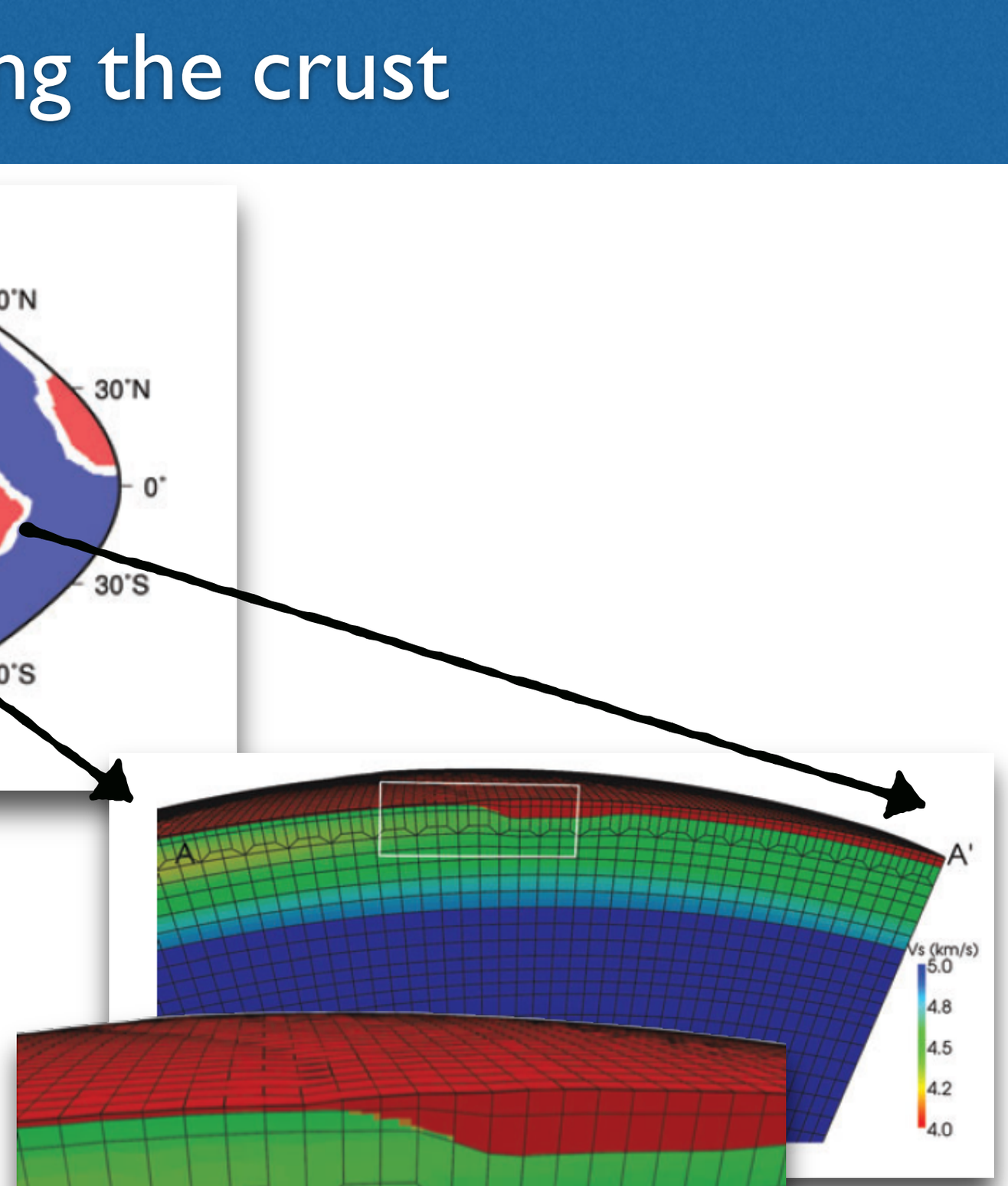
Number of processors?





Global Mesher: Meshing the crust





1) Moho is honored;
2) Crust Option at setup/constant.h.in:
   Crust1.0/ Crust2.0

lithosphere & asthenosphere

Global Seismic Wave Propagation

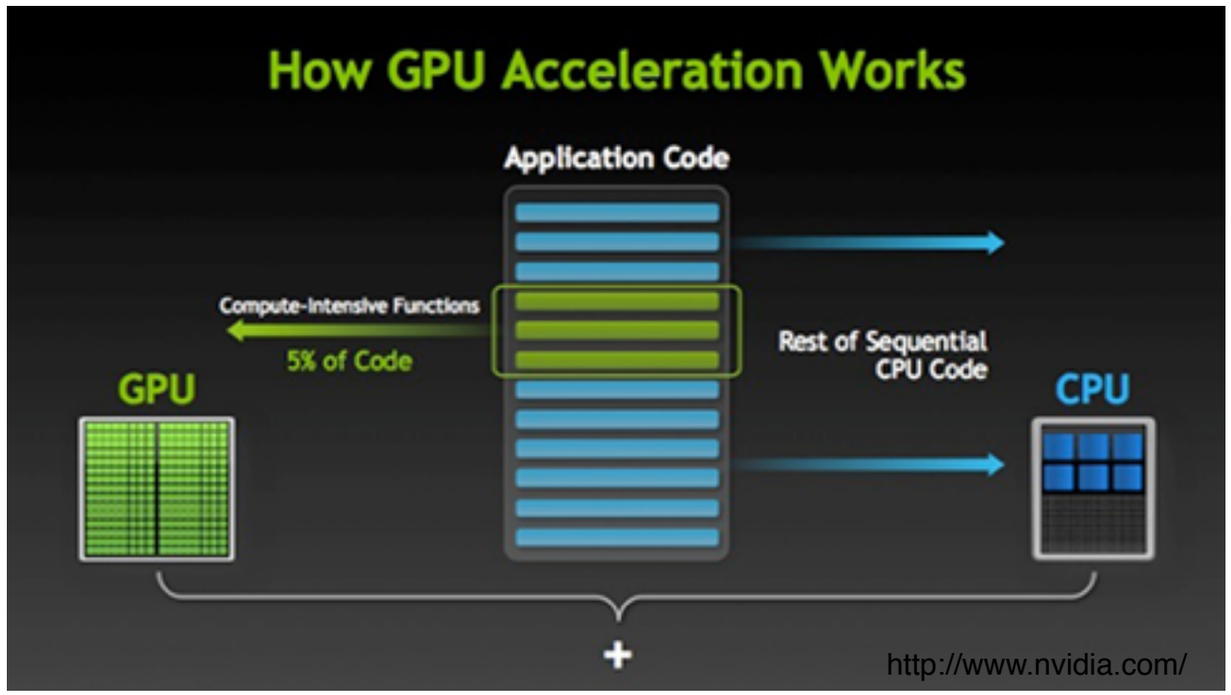
April 1, 2014, NEAR COAST OF NORTHERN CHILE
Mw = 8.1 Depth = 21.9km

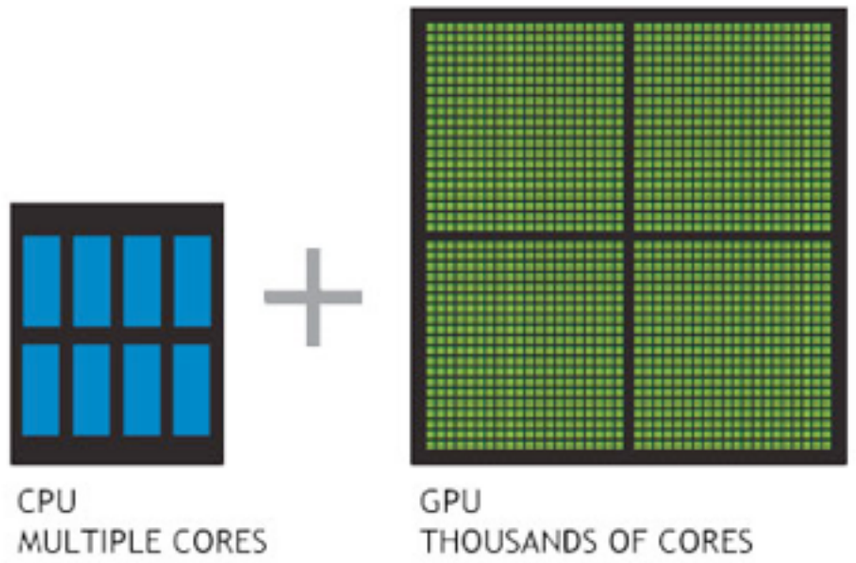
ShakeMovie GLOBAL
Princeton University's Near Real Time Global Seismicity Portal

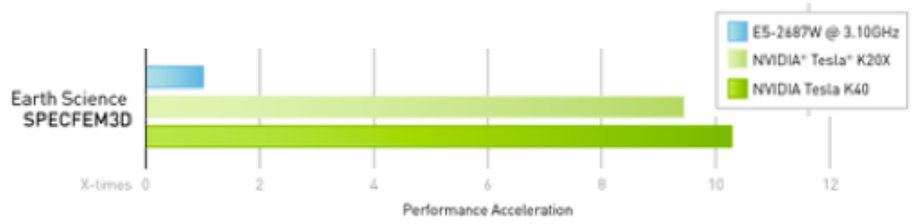
Recent Development

- GPU
- Adaptive Seismic Data Format(ASDF)
- Specfem3d_Mars
- Undo attenuation(adjoint simulation)

GPU Capability



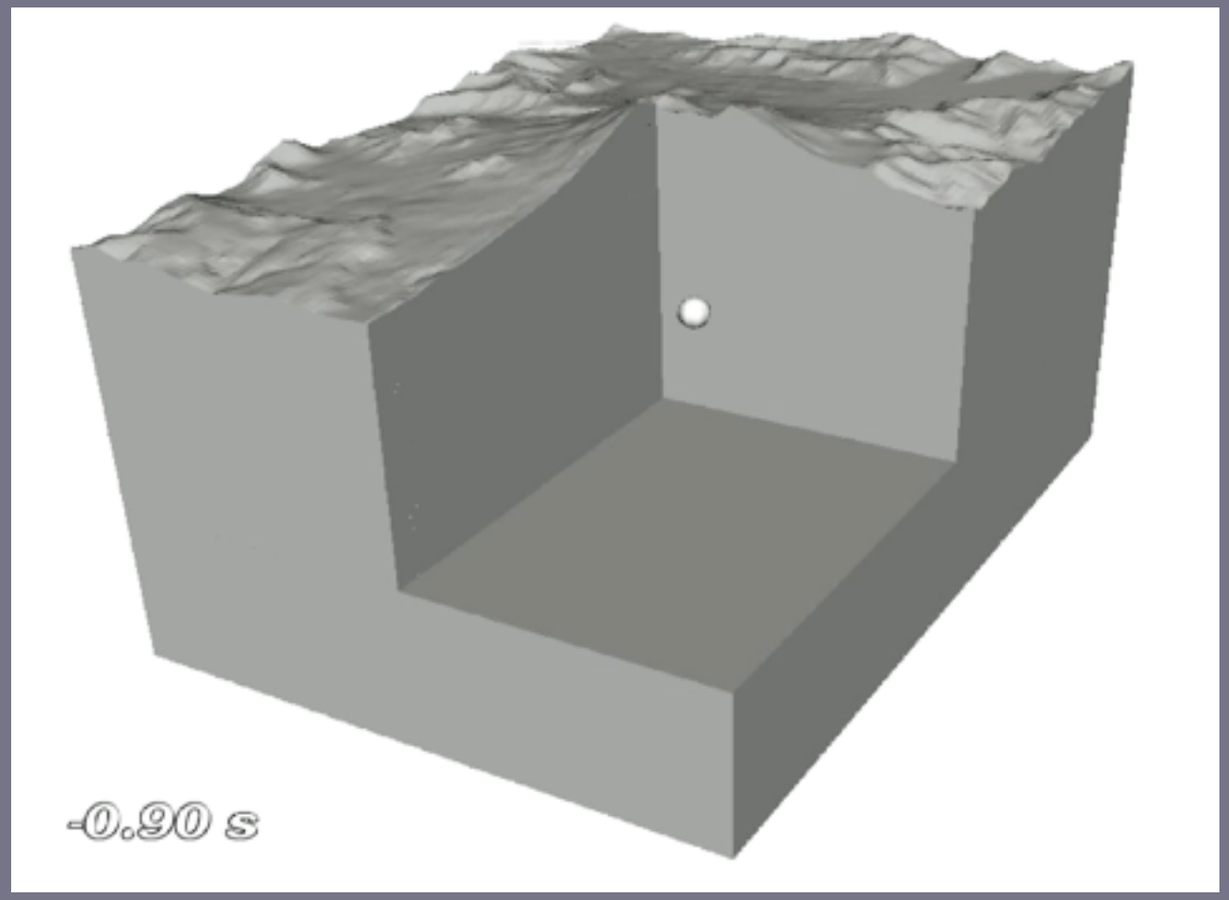




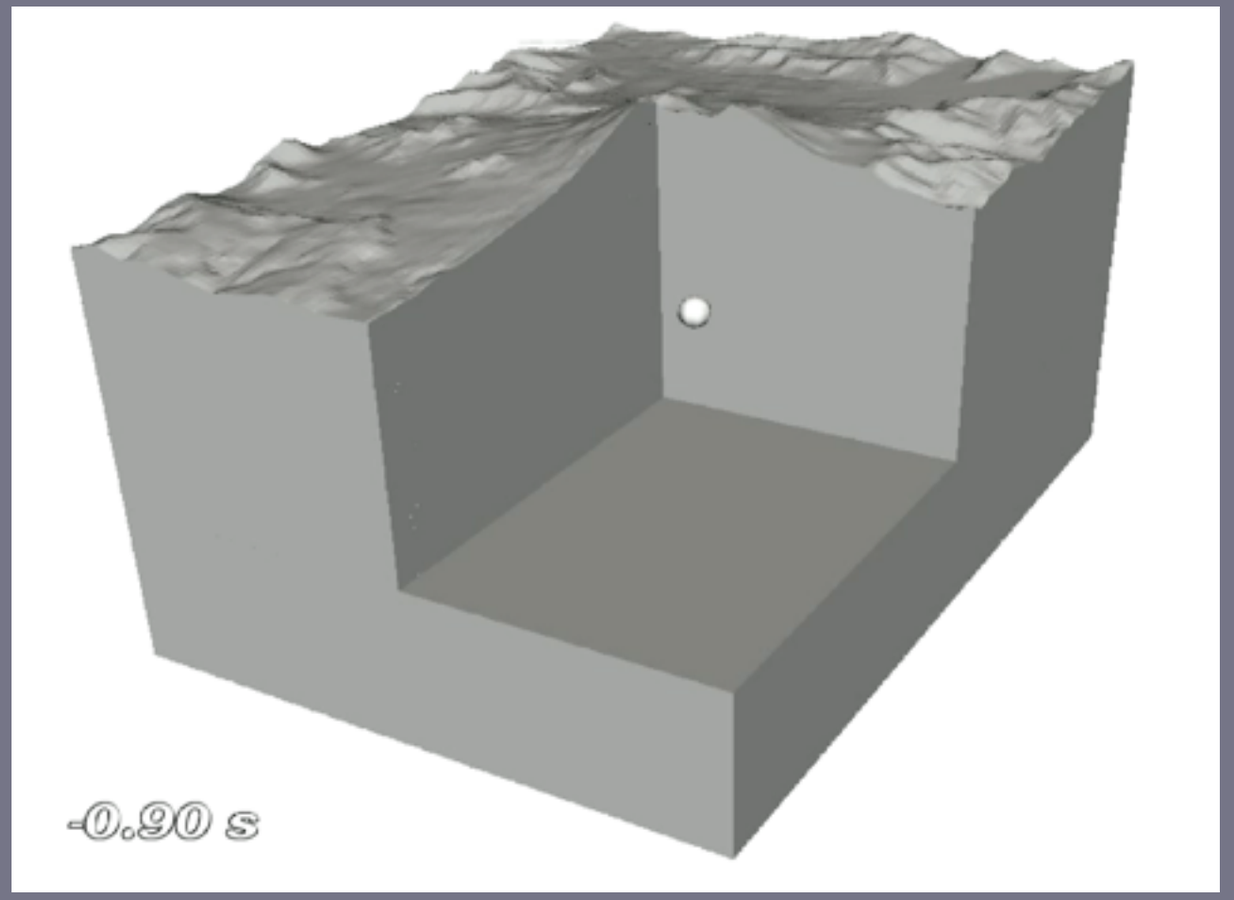
www.olcf.ornl.gov

Architecture:	Cray XK7
Nodes:	18,688 AMD Opterons
GPUs:	18,688 K20X Keplers

GPU Capability



GPU



CPU

Adaptive Seismic Data Format(ASDF)

- **To solve the I/O bottleneck**

  4,200 events * (12,000 observed sac + 12,000 synthetic sac)
  = 100,800,000 single sac files

  or

  4,200 events * (1 observed asdf + 1 synthetic asdf)
  = 8,400 single asdf files

- **Parallel I/O**

  MPI or multi-processing(internal support)

- **Obspy support**

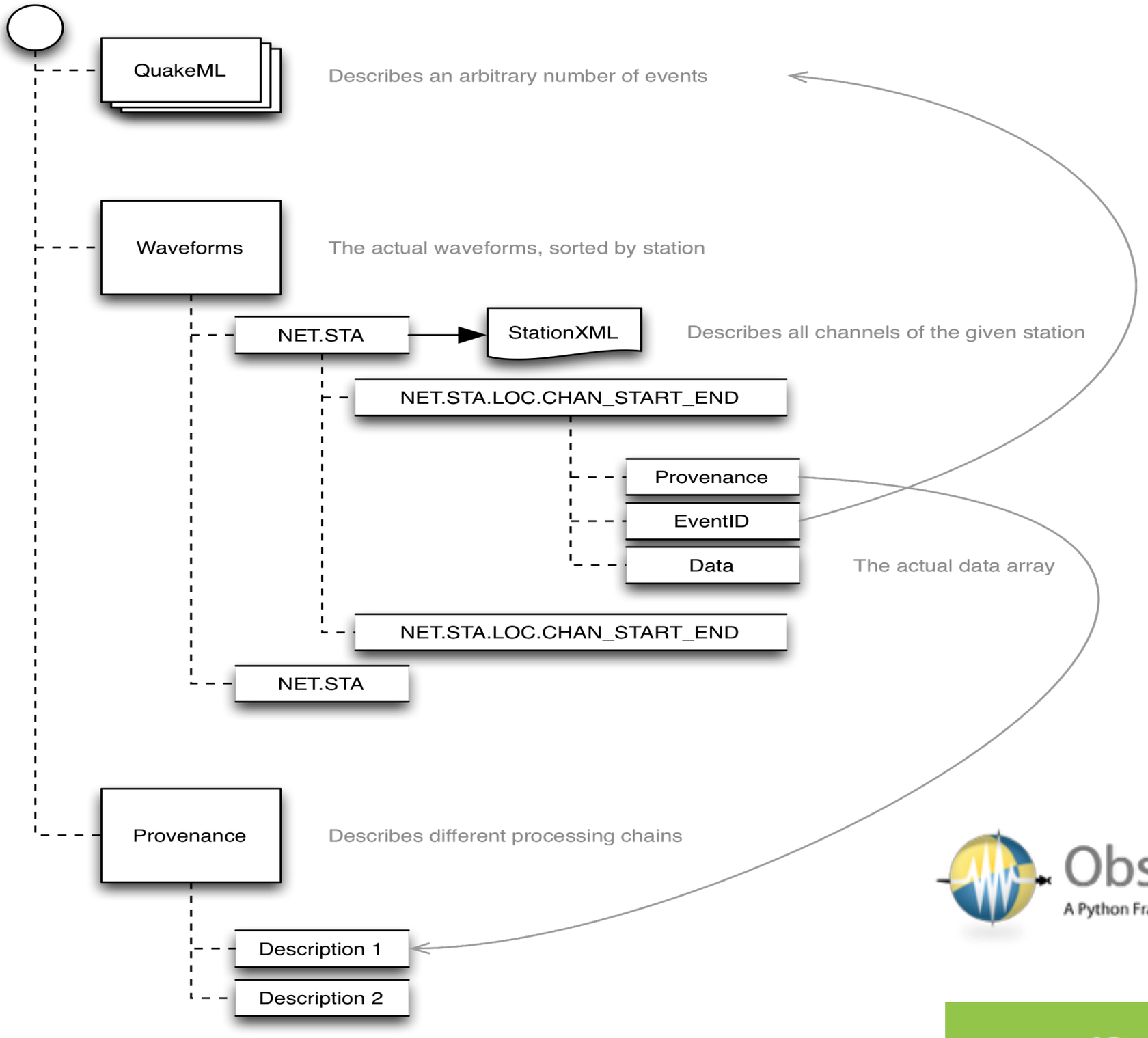
  working seamlessly with obspy

- **Multiple language interface**

  python, fortran, C/C++

Hosted on github: https://github.com/SeismicData

Adaptive Seismic Data Format(ASDF)



Documentation

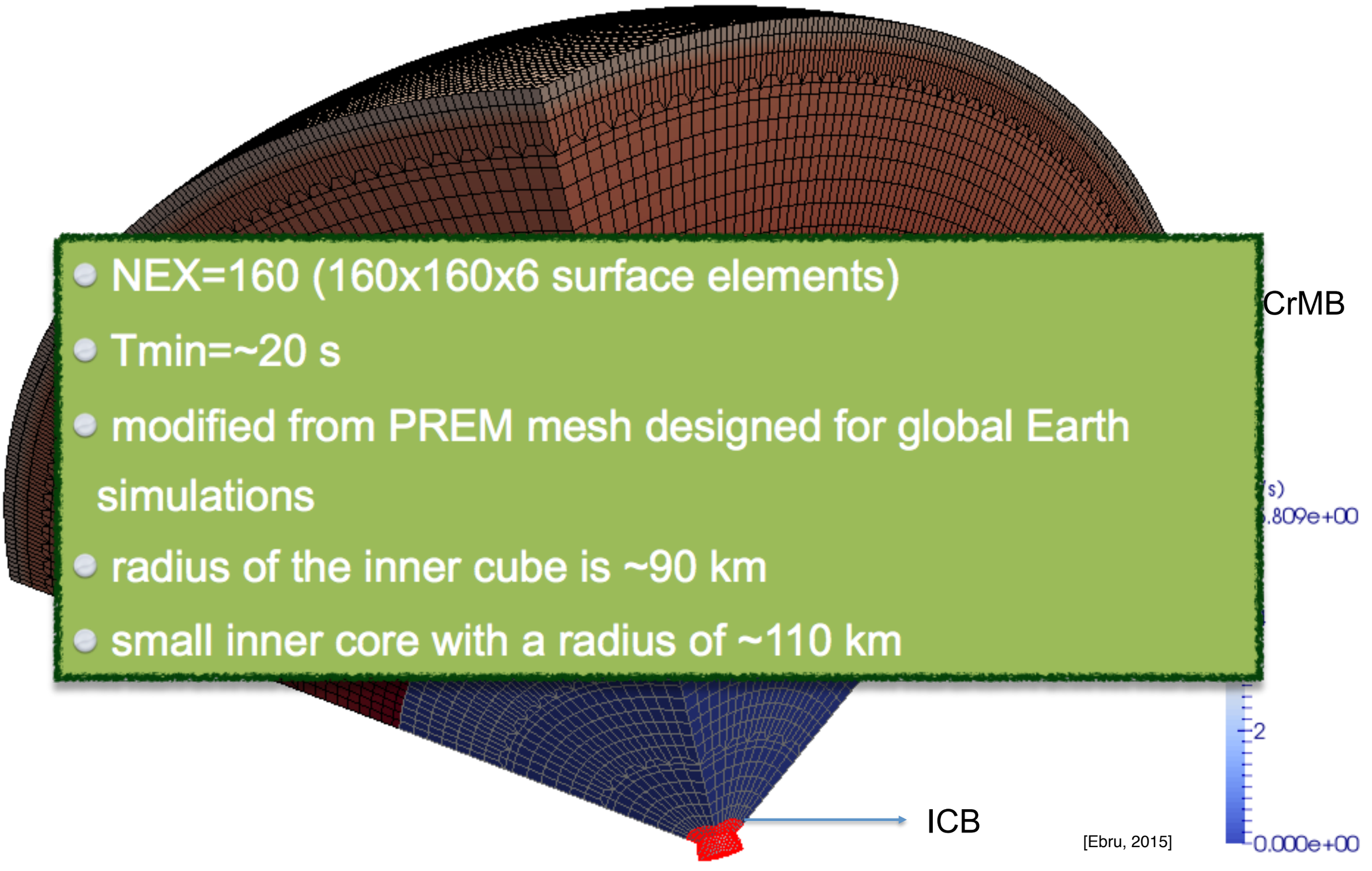
Hosted on github: https://github.com/SeismicData

ObsPy
A Python Framework for Seismology

pyasdf 0.1.x

SEIS-PROV

Specfem3d_Mars



- NEX=160 (160x160x6 surface elements)
- Tmin=~20 s
- modified from PREM mesh designed for global Earth simulations
- radius of the inner cube is ~90 km
- small inner core with a radius of ~110 km

Specfem3d Globe running session(on the cluster)

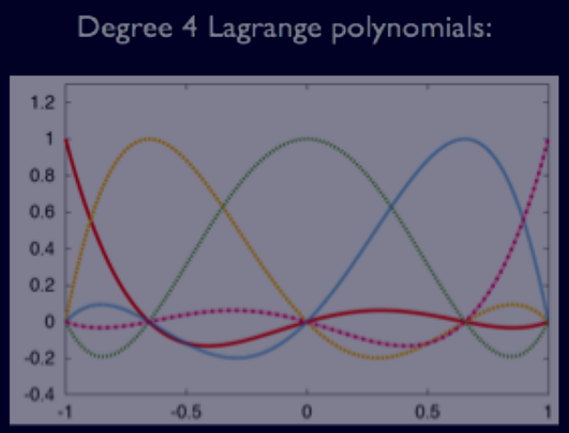
Please follow the instructions on the quick guide(downloaded from github)

Outline

1. Introduction

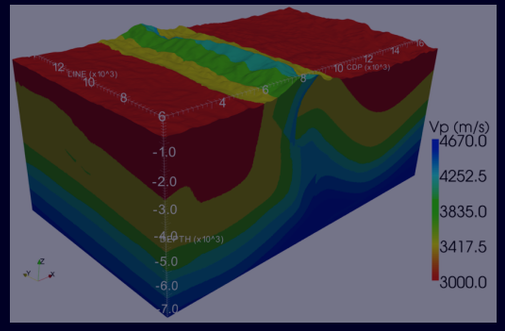
2. Basic Theory:

Spectral Element Method

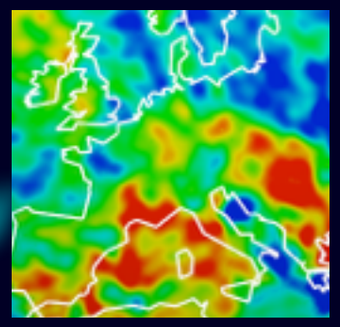


3.

SPECFEM2D



4. Adjoint Tomography



5. Summary

Adjoint Method

$$\chi\ (\mathbf{m}) = \frac{1}{2}\sum_{r=1}^{N_r}\int_0^T w_r(t)||\mathbf{s}(\mathbf{x}_r,t;\mathbf{m})-\mathbf{d}(\mathbf{x}_r,t)||^2\,\mathrm{d}t,$$

$$\delta\chi\ = \int_V [K_\rho(\mathbf{x})\,\delta\ln\rho(\mathbf{x}) + K_\mu(\mathbf{x})\,\delta\ln\mu(\mathbf{x}) + K_\kappa(\mathbf{x})\,\delta\ln\kappa(\mathbf{x})]\,\mathrm{d}^3\mathbf{x},$$

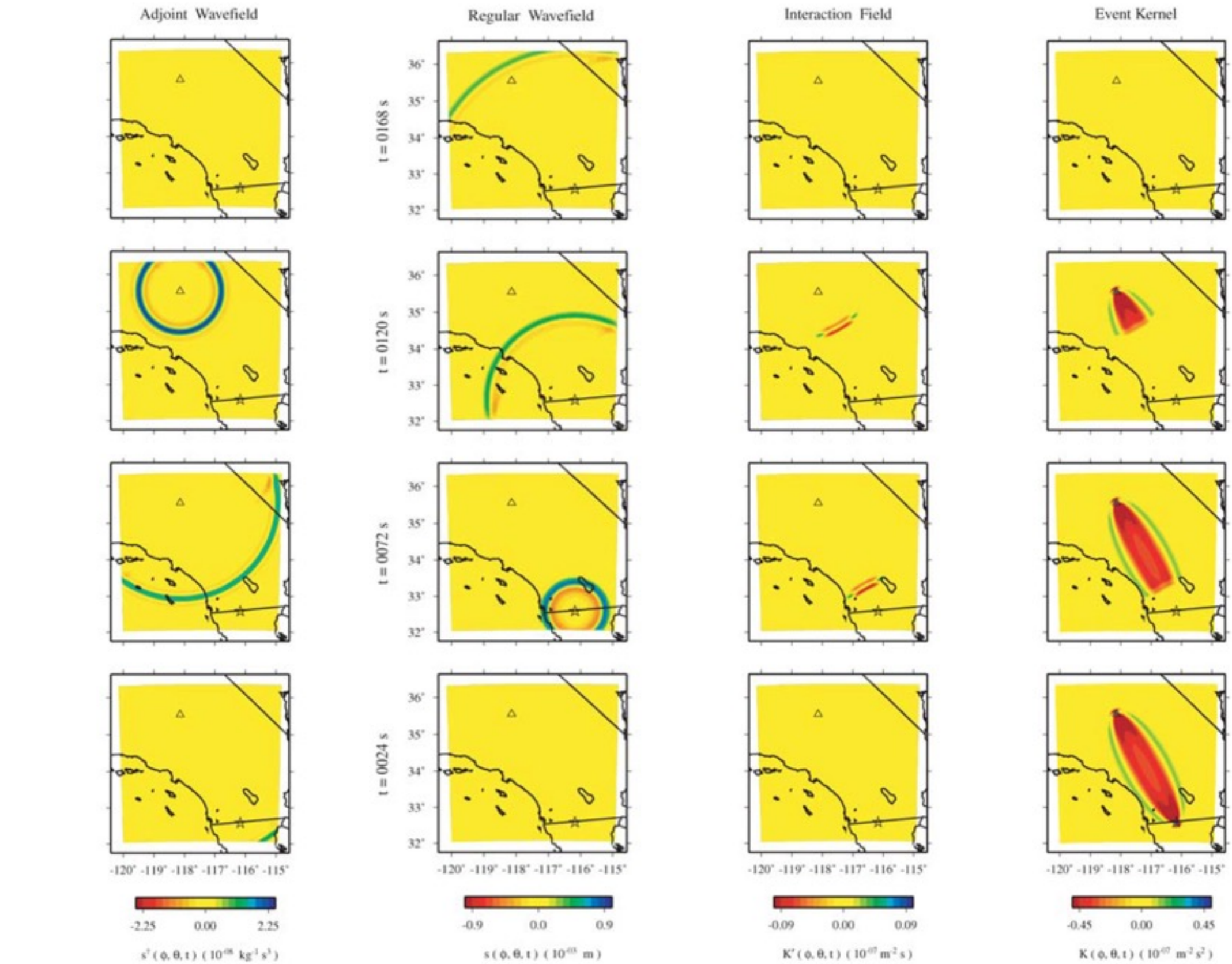
$$K_\rho(\mathbf{x}) = \int_0^T \rho(\mathbf{x})\,\partial_t\mathbf{s}^\dagger(\mathbf{x},T-t)\cdot\partial_t\mathbf{s}(\mathbf{x},t)\,\mathrm{d}t,$$

$$K_\mu(\mathbf{x}) = -\int_0^T 2\mu(\mathbf{x})\,\mathbf{D}^\dagger(\mathbf{x},T-t){:}\,\mathbf{D}(\mathbf{x},t)\,\mathrm{d}t,$$

$$K_\kappa(\mathbf{x}) = -\int_0^T \kappa(\mathbf{x})\,[\nabla\cdot\mathbf{s}^\dagger(\mathbf{x},T-t)]\,[\nabla\cdot\mathbf{s}(\mathbf{x},t)]\,\mathrm{d}t,$$

$$\mathbf{f}_1^\dagger(\mathbf{x},t) = \sum_{r=1}^{N_r} w_r(t)[\mathbf{s}(\mathbf{x}_r,T-t)-\mathbf{d}(\mathbf{x}_r,T-t)]\,\delta(\mathbf{x}-\mathbf{x}_r),$$

Adjoint Simulation

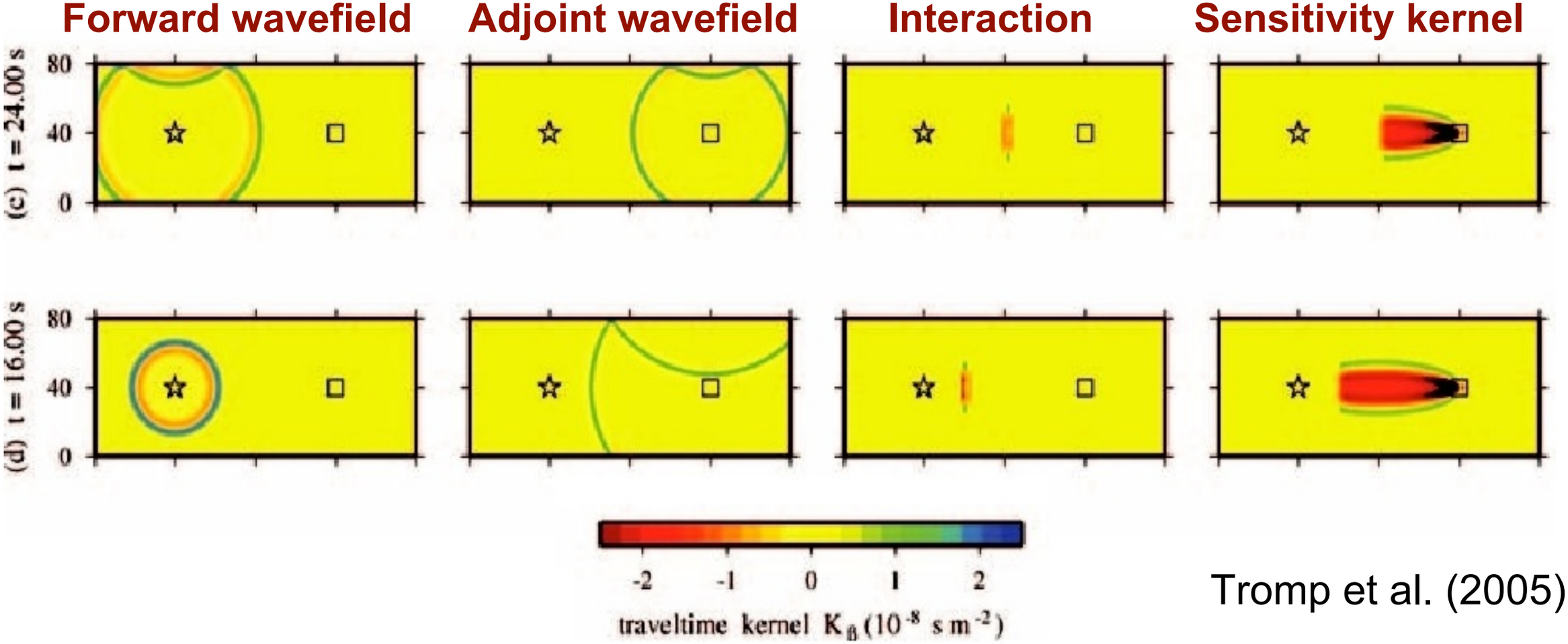


Adjoint tomography

$$\delta\chi = \int_V [K_\rho(\mathbf{x})\delta \ln \rho(\mathbf{x}) + K_{c_{jklm}}(\mathbf{x})\delta \ln c_{jklm}(\mathbf{x})]\, d^3\mathbf{x}$$

$$K_\rho(\mathbf{x}) = -\int_0^T \rho(\mathbf{x})\mathbf{s}^\dagger(\mathbf{x}, T-t)\cdot\partial_t^2\mathbf{s}(\mathbf{x}, t)\, dt$$

$$K_{c_{jklm}}(\mathbf{x}) = -\int_0^T \epsilon_{jk}^\dagger(\mathbf{x}, T-t)c_{jklm}(\mathbf{x})\epsilon_{lm}(\mathbf{x}, t)\, dt$$



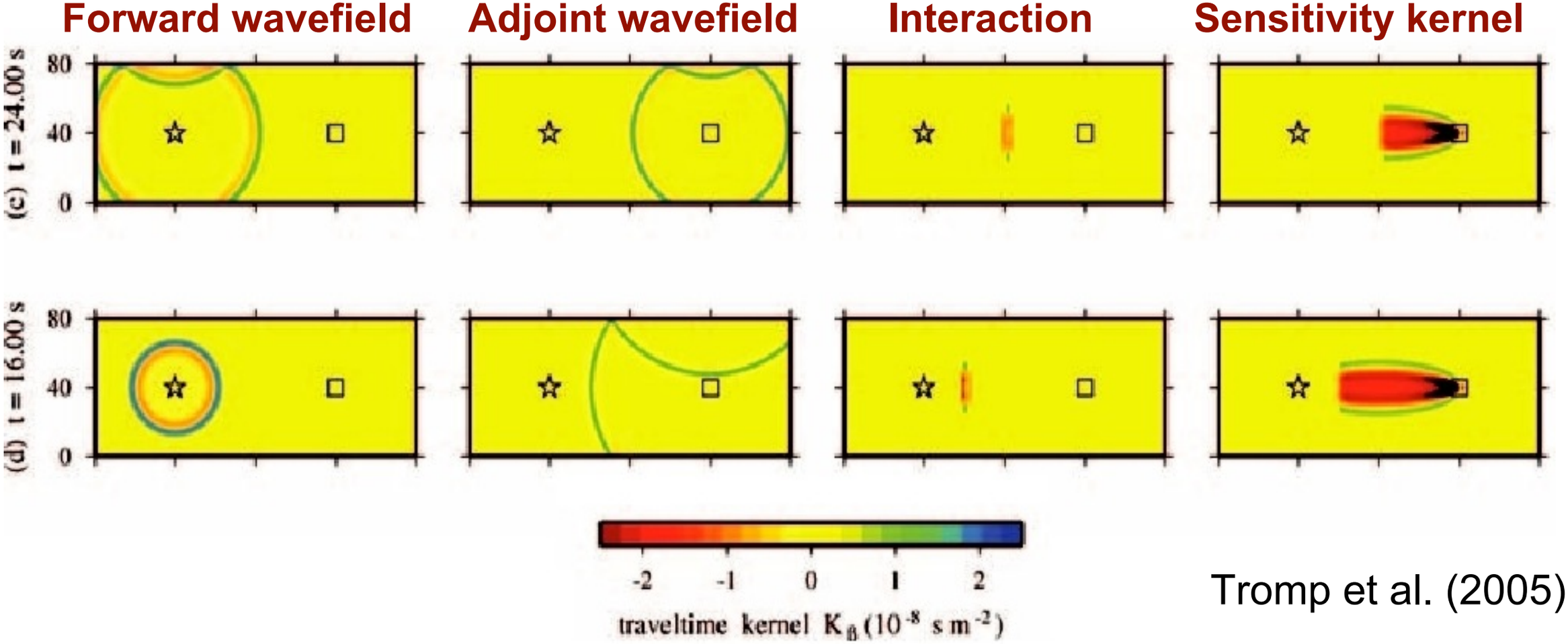
Tromp et al. (2005)

Adjoint tomography

$$\delta\chi = \int_V [K_\rho(\mathbf{x})\delta \ln \rho(\mathbf{x}) + K_{c_{jklm}}(\mathbf{x})\delta \ln c_{jklm}(\mathbf{x})]\, d^3\mathbf{x}$$

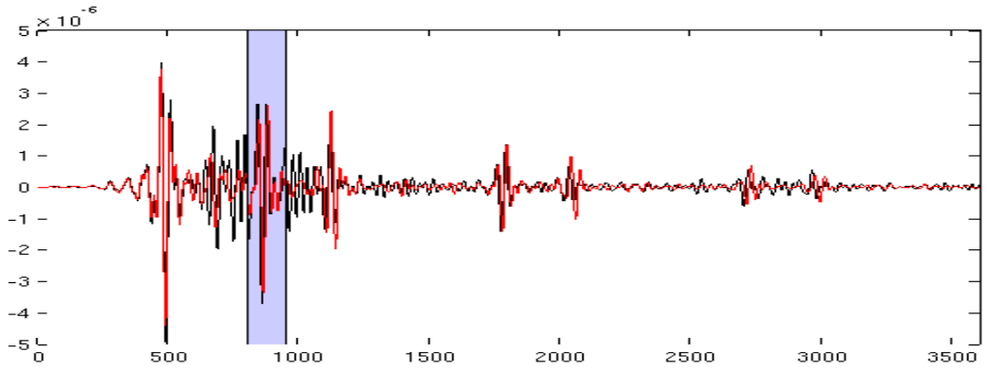
What is undo-attenuation?

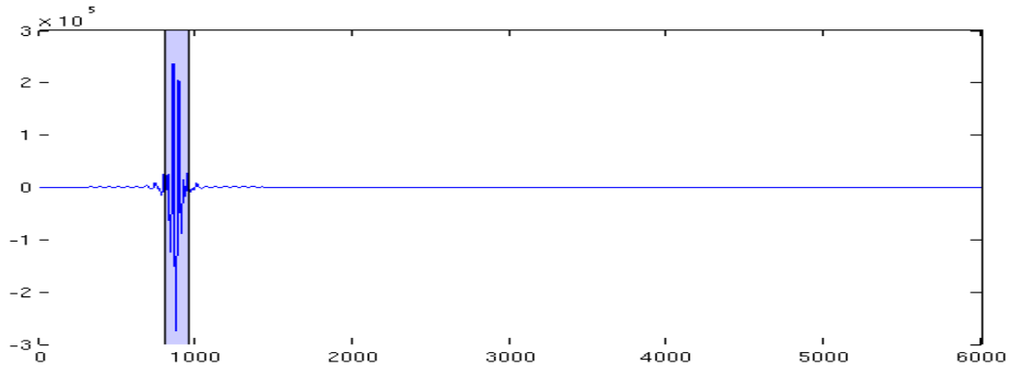
$$K_{c_{jklm}}(\mathbf{x}) = -\int_0^T \epsilon^\dagger_{jk}(\mathbf{x}, T-t) c_{jklm}(\mathbf{x}) \epsilon_{lm}(\mathbf{x}, t)\, dt$$

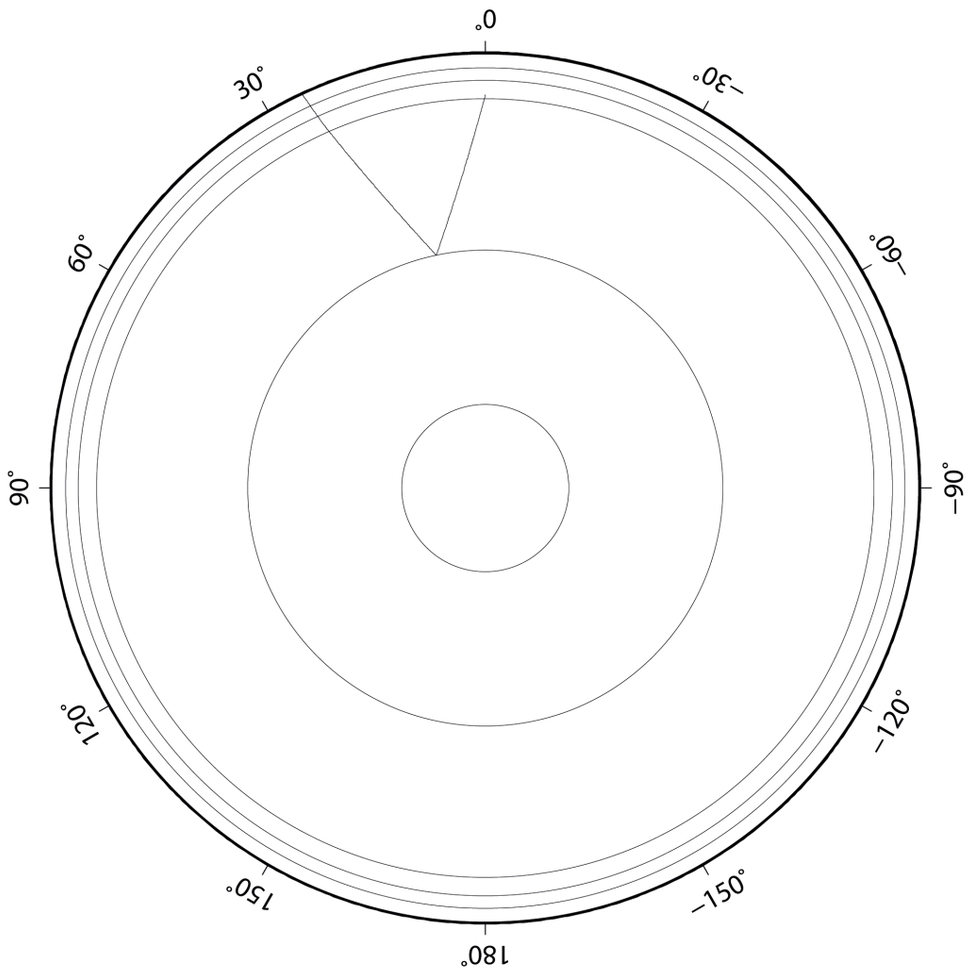


Tromp et al. (2005)

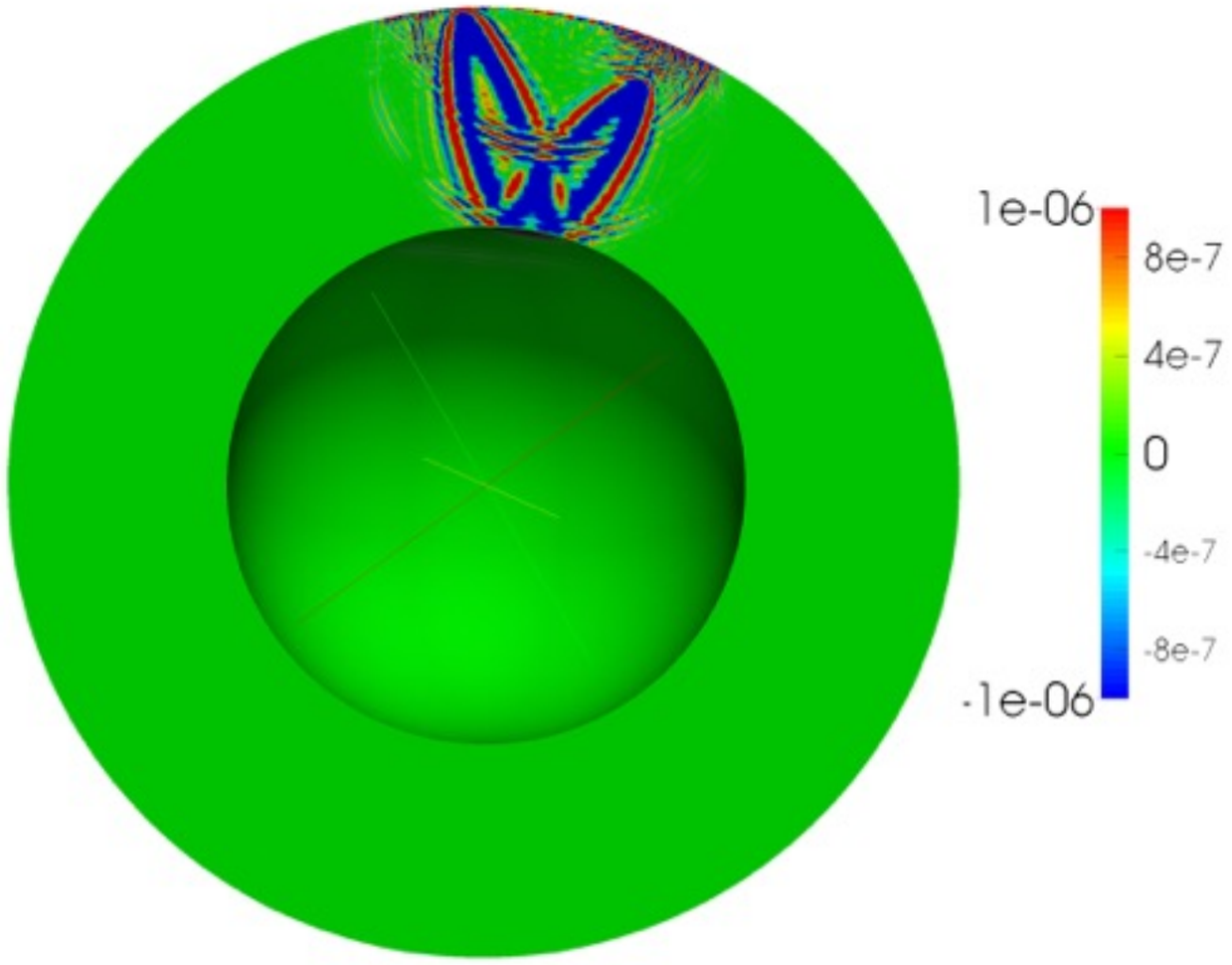
Traveltime Kernel from ScS phase





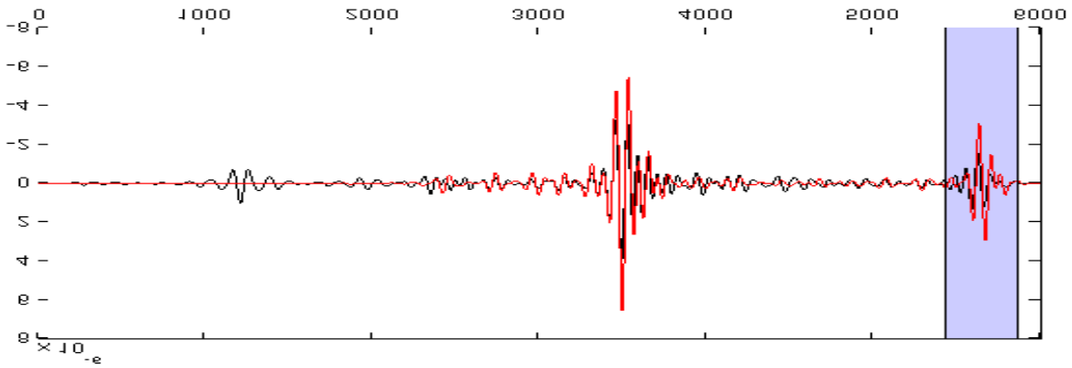


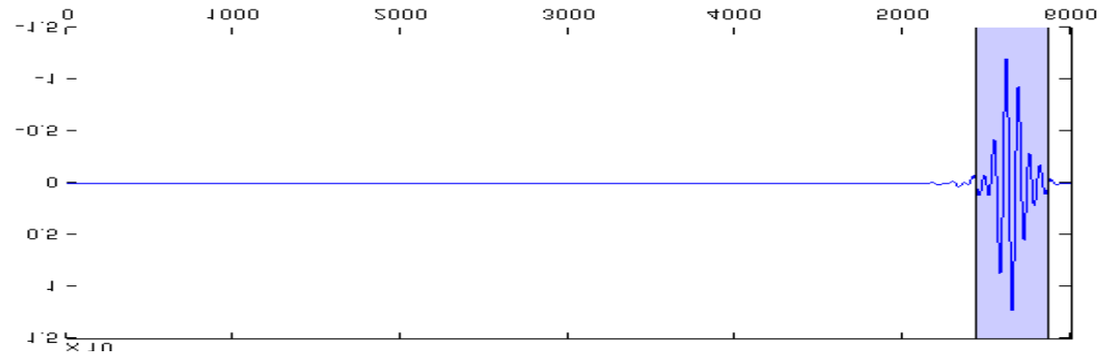
The raypath of the ScS phase

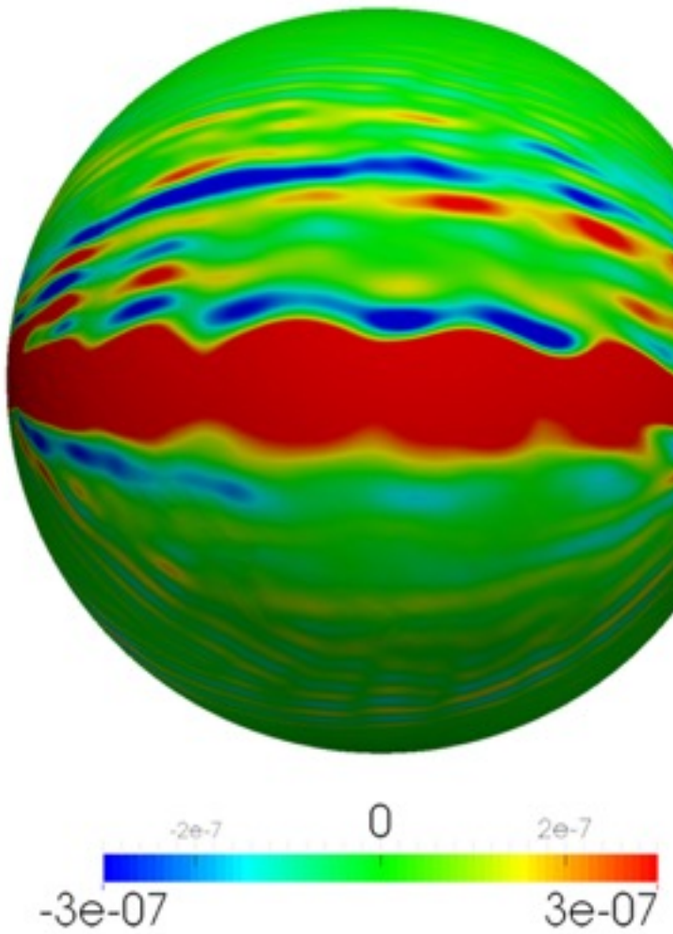


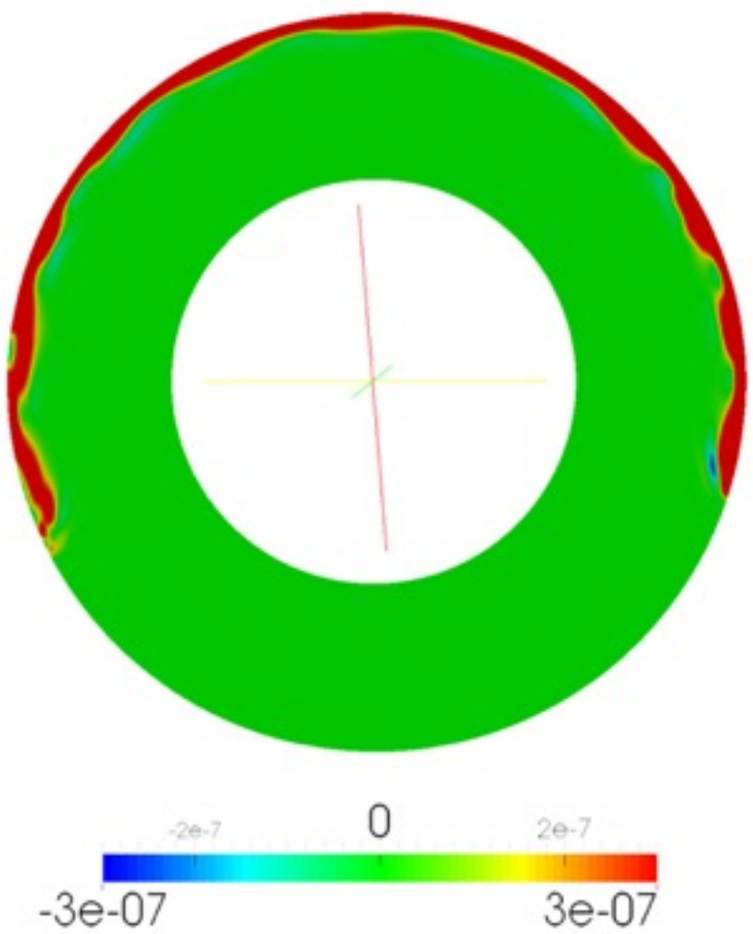
The β_v Kernel from the ScS phase

Traveltime Kernels from Major-arc surface waves









The β_v Kernel from major-arc Love waves

Summary and Questions