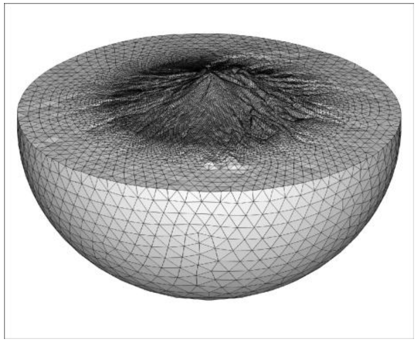
Computational Seismology: An Introduction

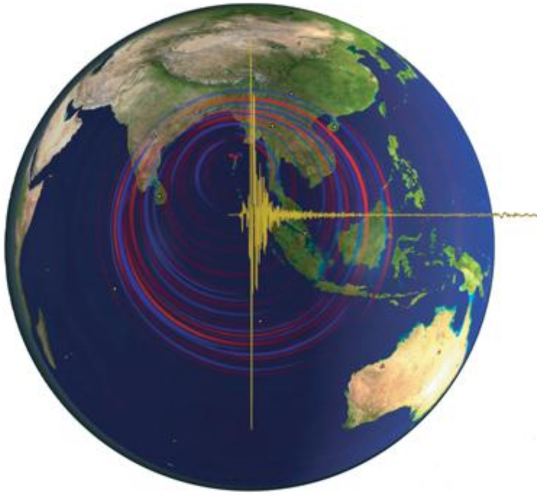


Aim of lecture:

- Understand why we need numerical methods to understand our world
- Learn about various numerical methods (finite differences, pseudospectal methods, finite (spectral) elements) and understand their similarities, differences, and domains of applications
- Learn how to replace simple partial differential equations by their numerical approximation
- Apply the numerical methods to the elastic wave equation
- Turn a numerical algorithm into a computer program (using Matlab, Fortran, or Python)

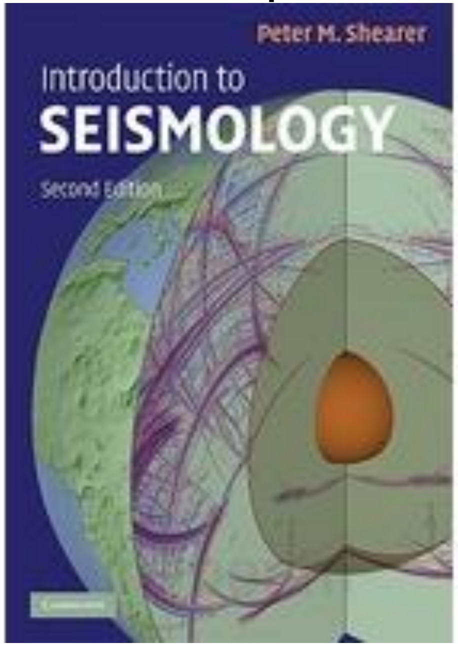
Structure of Course

- Introduction and Motivation
  - The need for synthetic seismograms
  - Other methodologies for simple models
  - 3D heterogeneous models
- Finite differences
  - Basic definition
  - Explicit and implicit methods
- High-order finite differences
  - Taylor weights
  - Truncated Fourier operators
- Pseudospectral methods
  - Derivatives in the Fourier domain
- Finite-elements (low order)
  - Basis functions
  - Weak form of pde‘s
  - FE approximation of wave equation
- Spectral elements
  - Chebyshev and Legendre basis functions
  - SE for wave equation



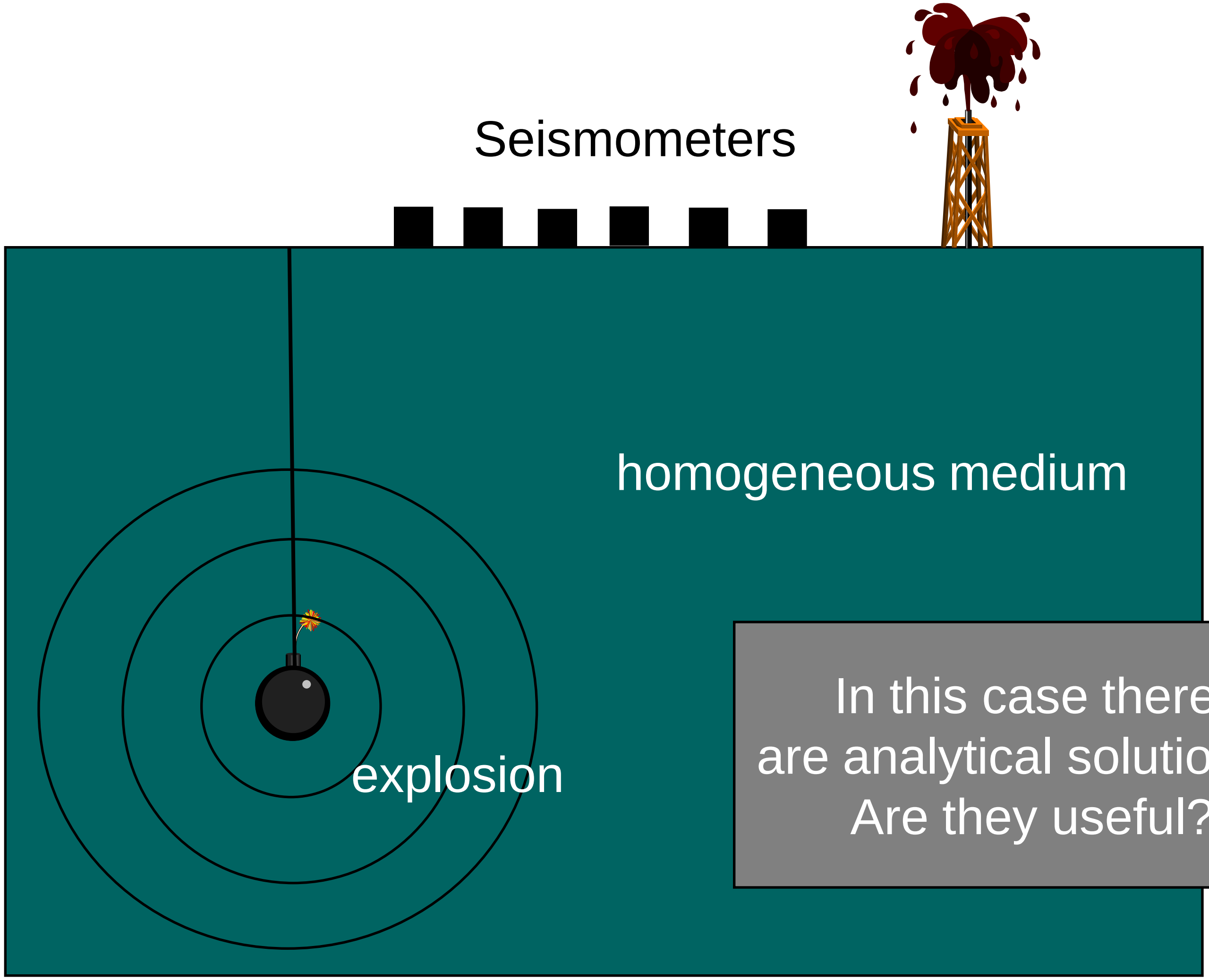
Literature

- Lecture notes (ppt) www.geophysik.uni-muenchen.de/Members/igel
- Presentations and books in SPICE archive www.spice-rtn.org
- Any readable book on numerical methods (lots of open manuscripts downloadable, eg http://samizdat.mines.edu/)
- Shearer: Introduction to Seismology (2nd edition, 2009,Chapter 3.7-3.9)
- Aki and Richards, Quantitative Seismology (1st edition, 1980)
- Mozco: The Finite-Difference Method for Seismologists. An Introduction. (pdf available at spice-rtn.org)



Why numerical methods?

Example: seismic wave propagation



In this case there
are analytical solutions?
Are they useful?

Analytical solution for a double couple point source

Ground displacement

$$\begin{aligned}
\boldsymbol{u}(\boldsymbol{x},t) = {} & \frac{1}{4\pi\rho}\boldsymbol{A}^{N}\frac{1}{r^4}\int_{r/v_P}^{r/v_S}\tau M_0(t-\tau)\,d\tau \\
& + \frac{1}{4\pi\rho v_P^2}\boldsymbol{A}^{IP}\frac{1}{r^2}M_0(t-r/v_P) \\
& + \frac{1}{4\pi\rho v_S^2}\boldsymbol{A}^{IS}\frac{1}{r^2}M_0(t-r/v_S) \\
& + \frac{1}{4\pi\rho v_P^3}\boldsymbol{A}^{FP}\frac{1}{r}\dot{M}_0(t-r/v_P) \\
& + \frac{1}{4\pi\rho v_S^3}\boldsymbol{A}^{FS}\frac{1}{r}\dot{M}_0(t-r/v_S).
\end{aligned}$$

Near field term contains the static deformation

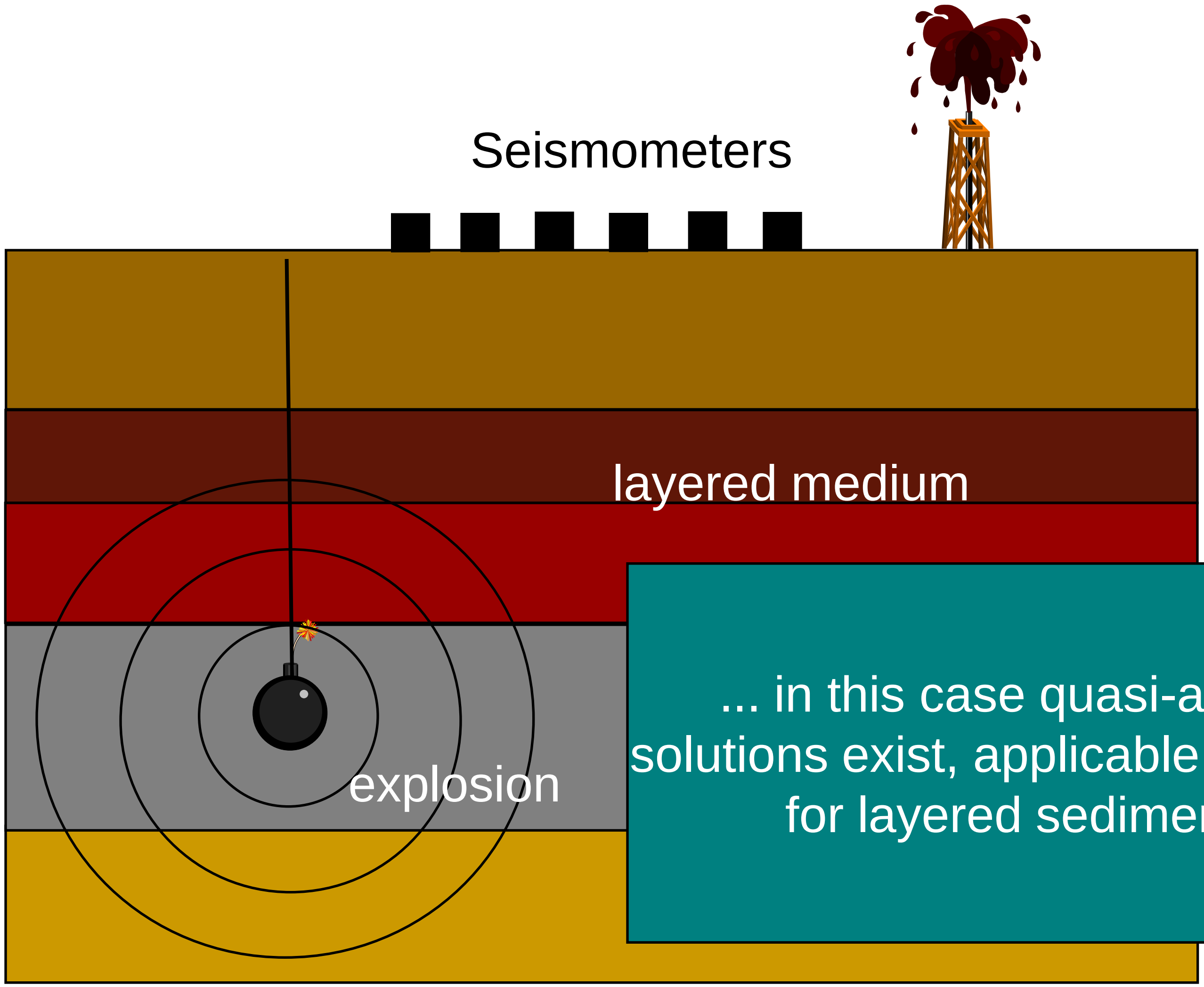
Intermediate terms

Far field terms: the main ingredient for source inversion, ray theory, etc.

Aki and Richards (2002)

… pretty complicated for such a simple problem, no way to do anything analytical in 2D or 3D!!!!

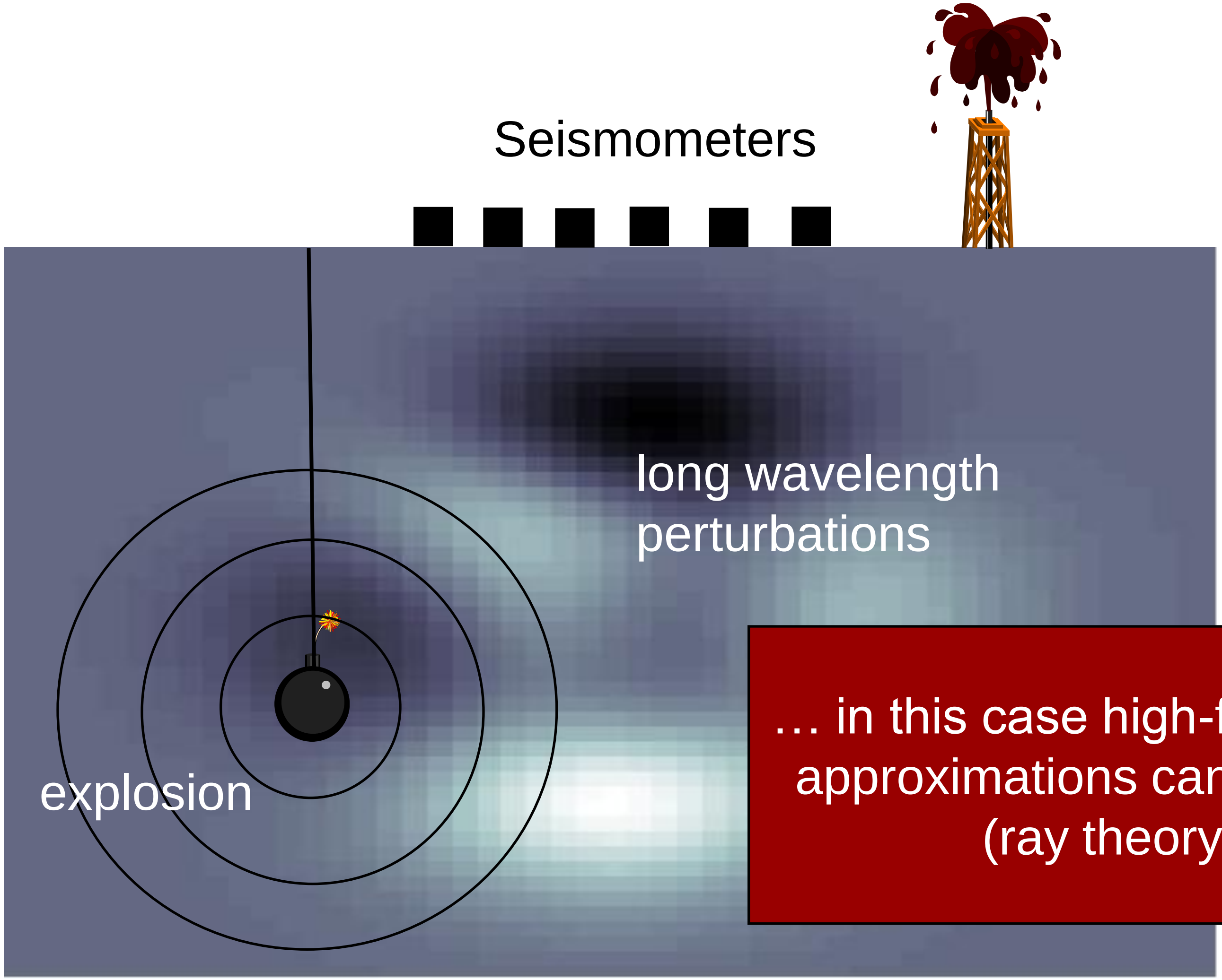
Example: seismic wave propagation



... in this case quasi-analytical solutions exist, applicable for example for layered sediments ...

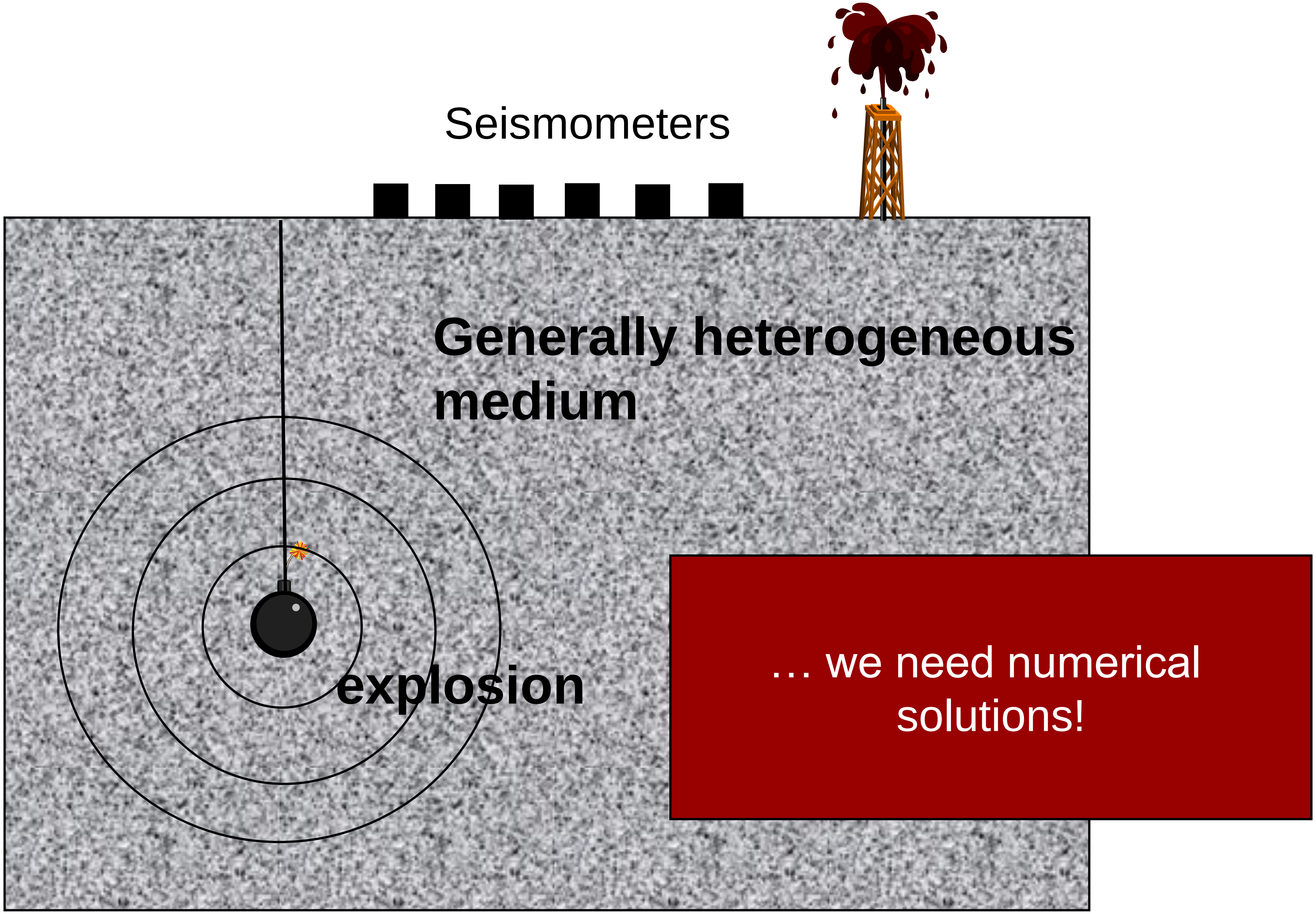
Why numerical methods?

Example: seismic wave propagation



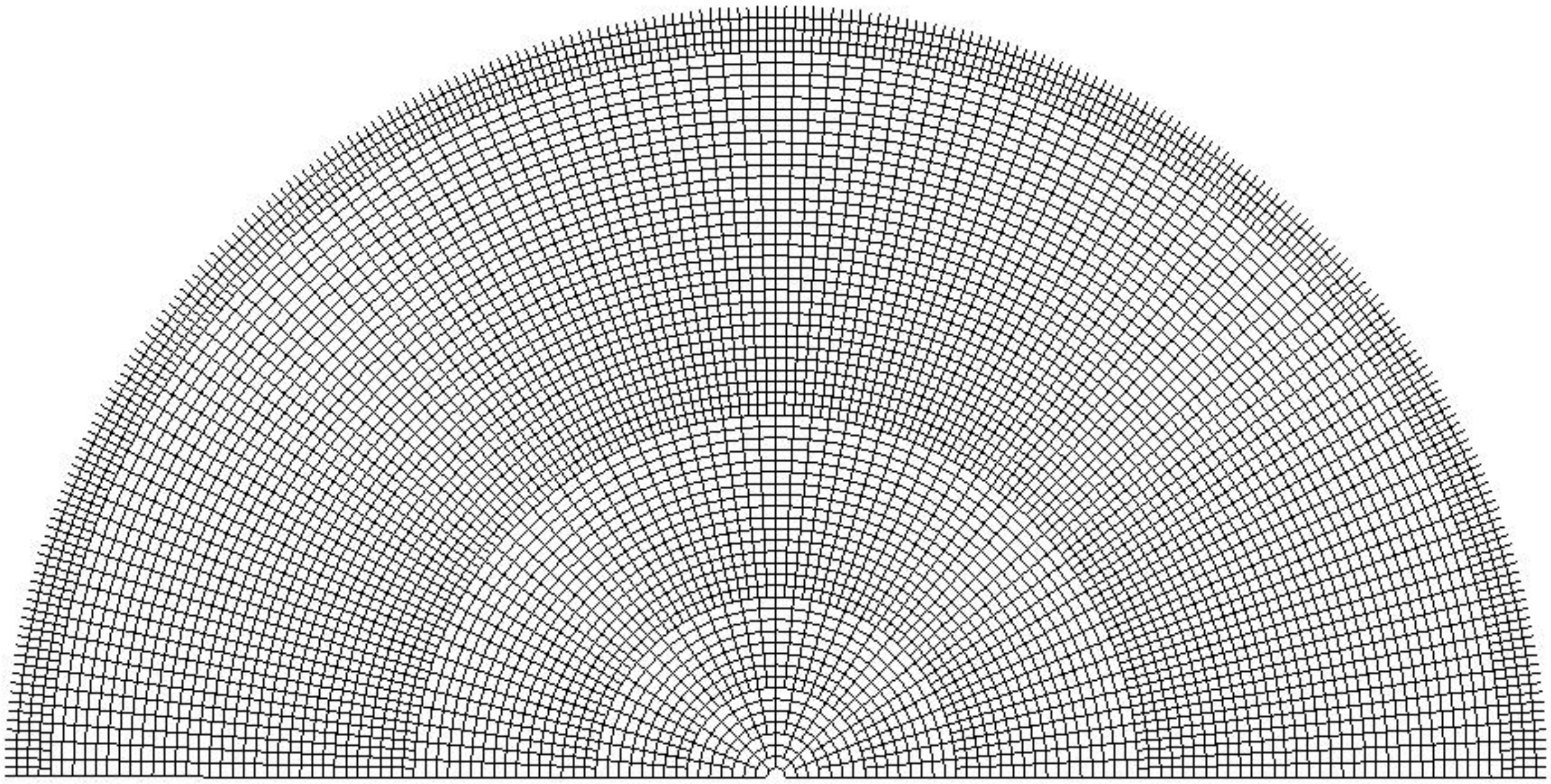
… in this case high-frequency approximations can be used (ray theory)

Example: seismic wave propagation



Applications in Geophysics

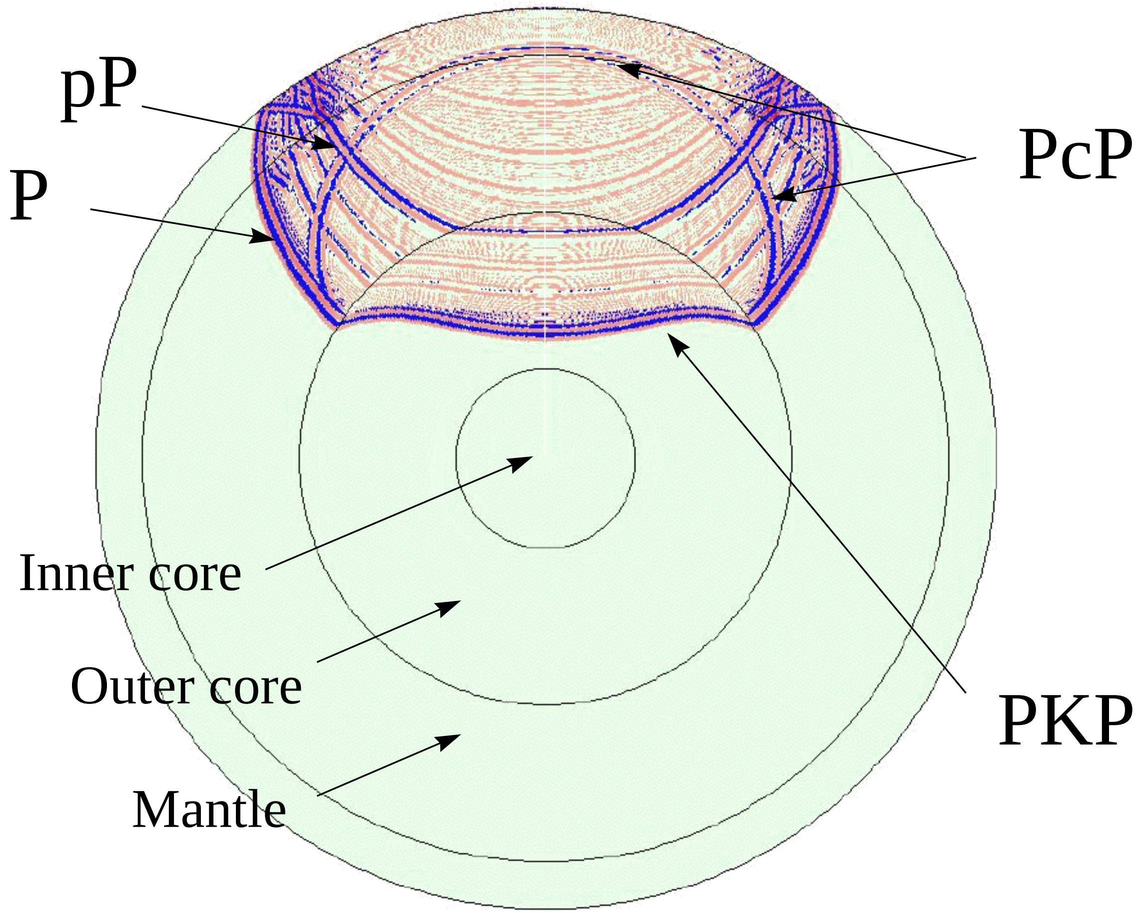
global seismology – spherical coordinates – axisymmetry
- computational grids – spatial discretization – regular/irregular grids



finite differences – multidomain method

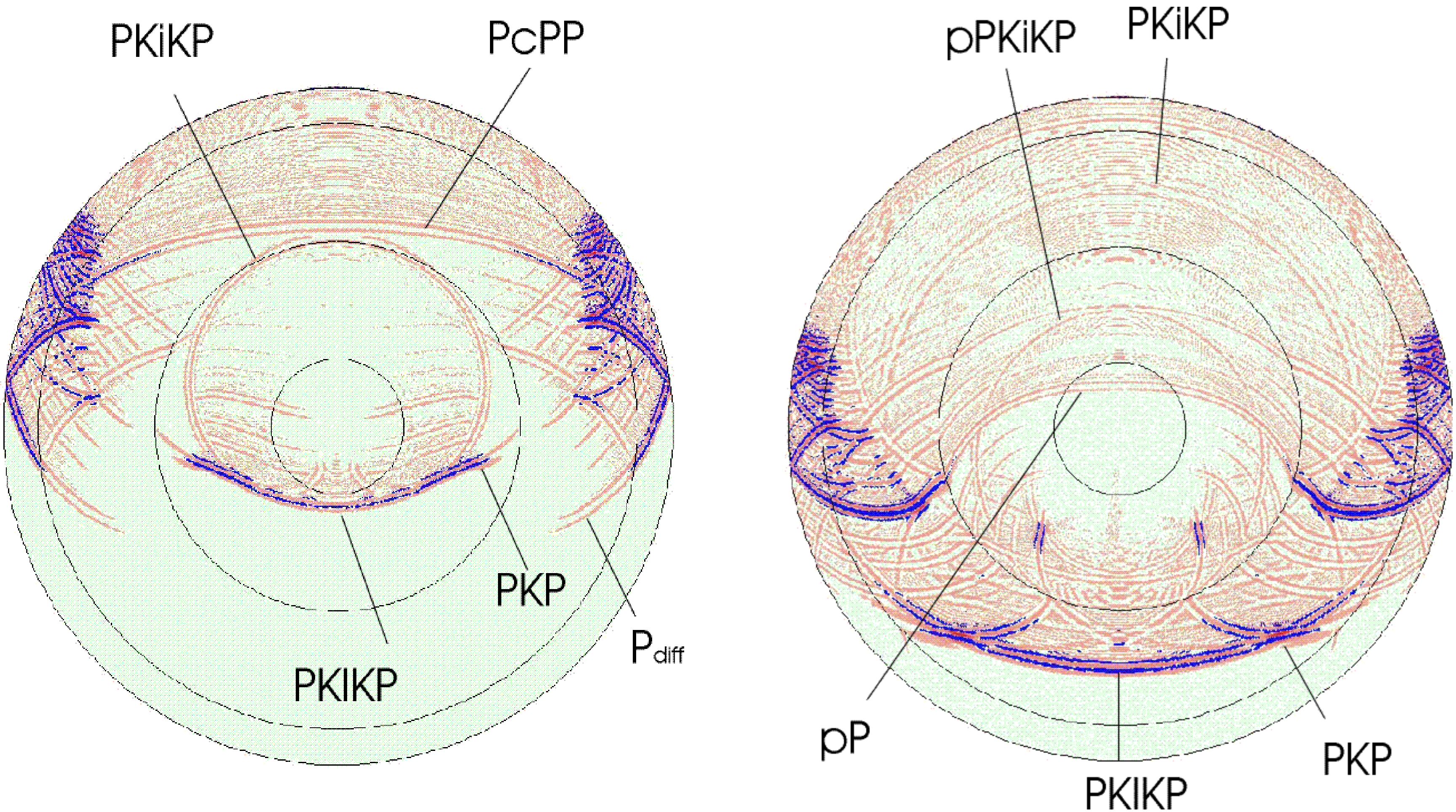
Global wave propagation

global seismology – spherical coordinates - axisymmetry



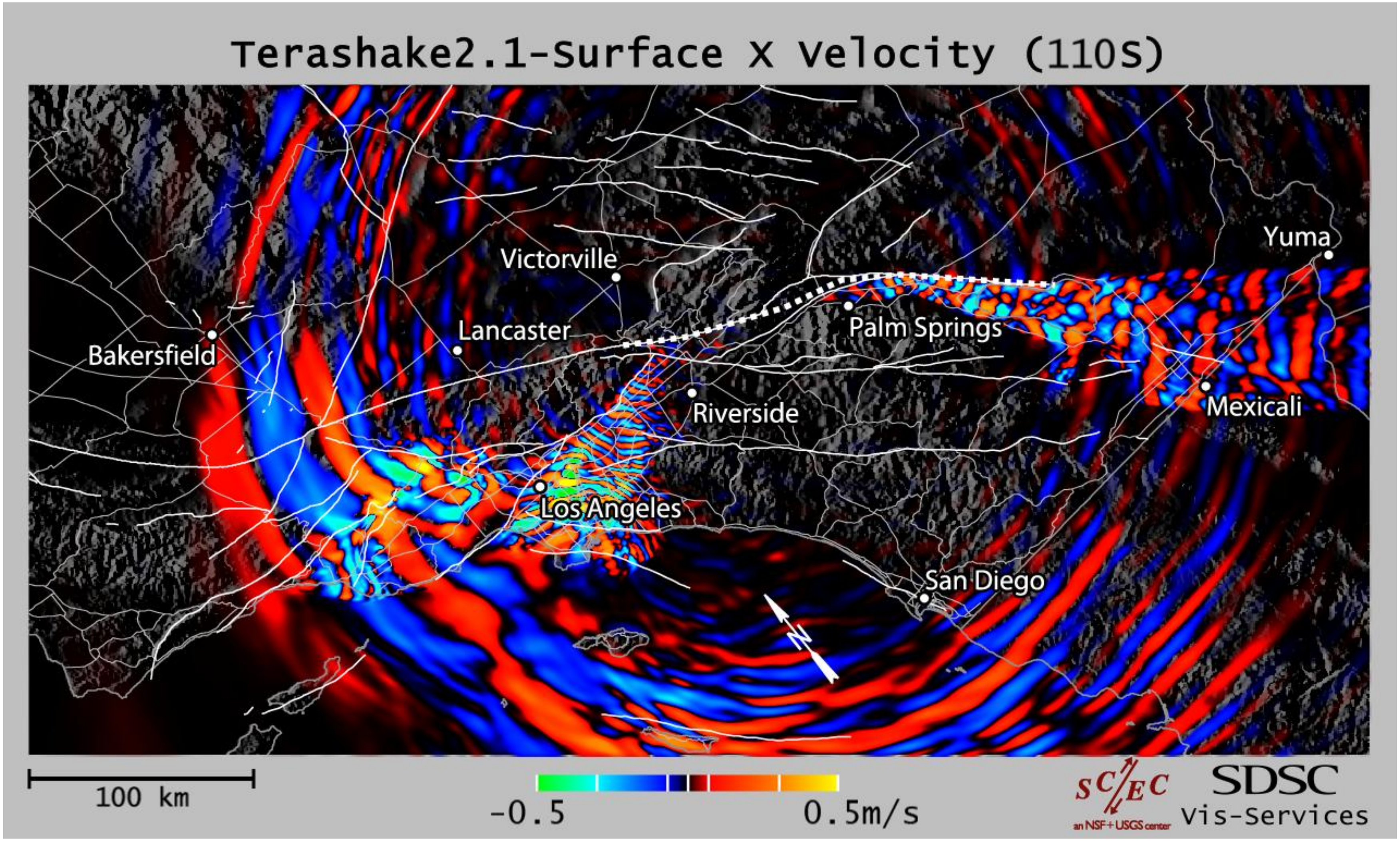
finite differences – multidomain method

global seismology – spherical coordinates - axisymmetry



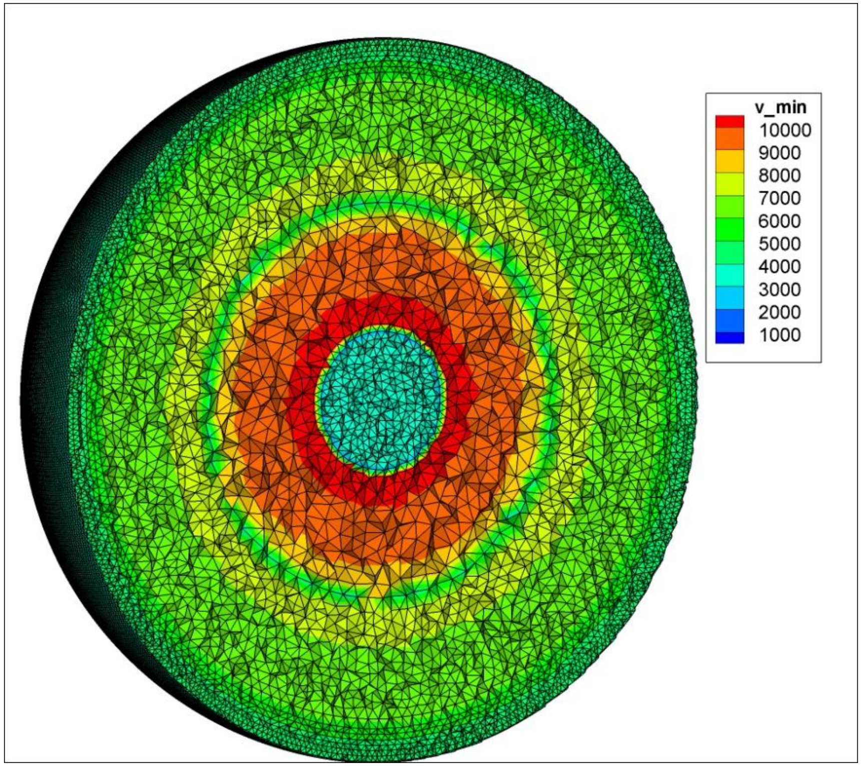
finite differences – multidomain method

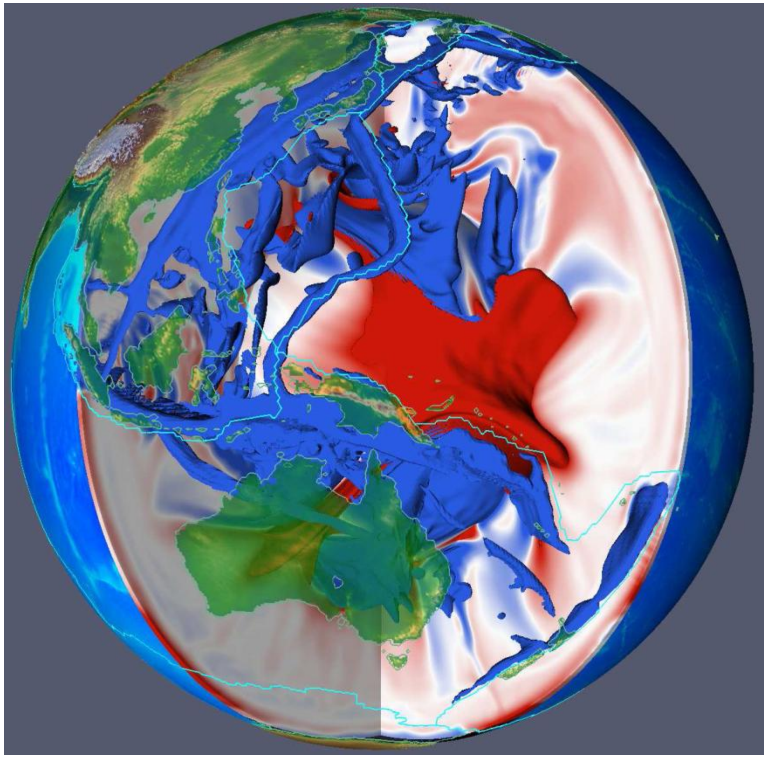
Earthquake Scenarios



visservices.sdsc.edu

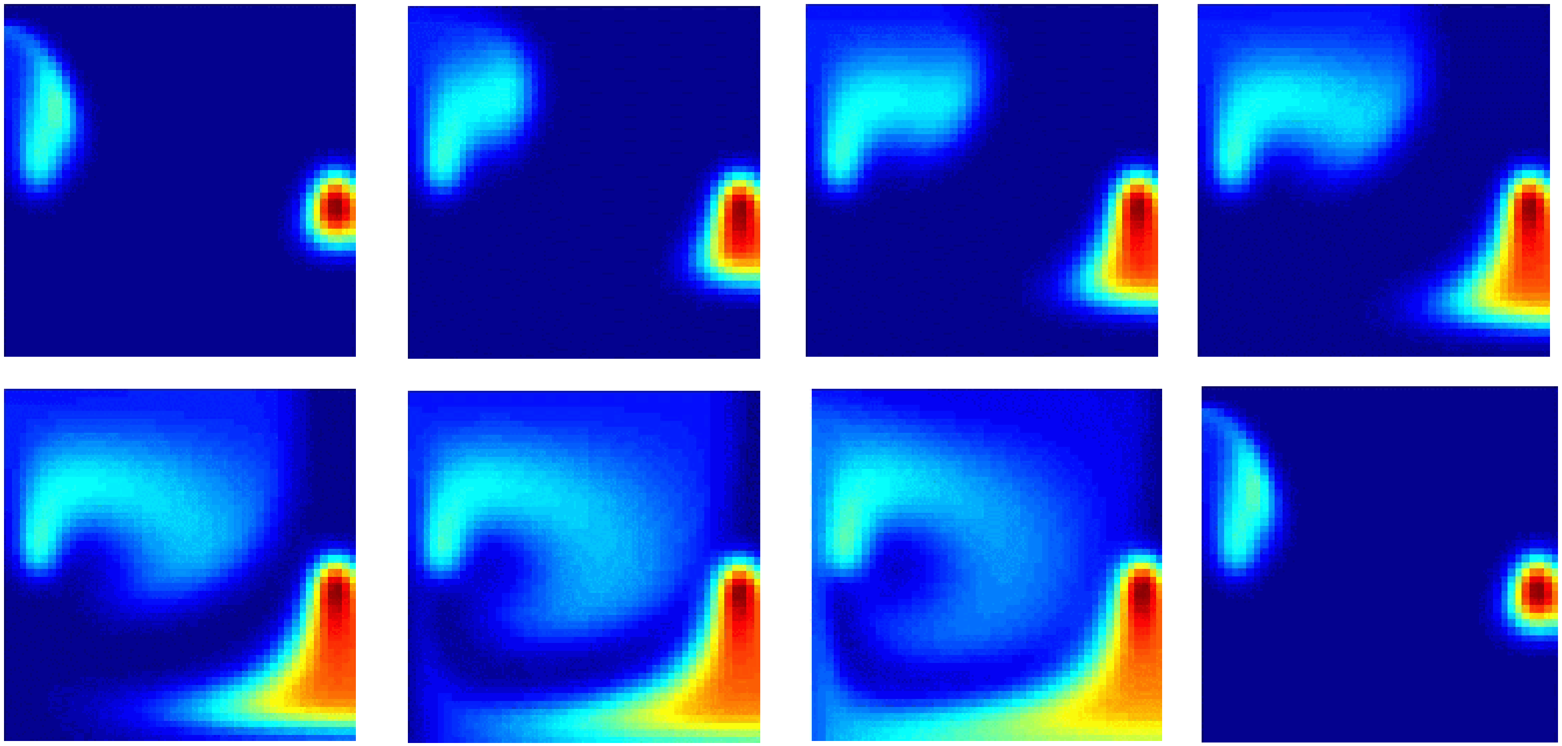
Seismology and Geodynamics





Ocean Mixing of Isotopes

isotope mixing in the oceans
Stommel-gyre
input of isotopes near the boundaries (e.g. rivers)



diffusion – reaction – advection equation

Computational grids and memory

Example: seismic wave propagation, 2-D case

grid size:	1000x1000
number of grid points:	10^6
parameters/grid point:	elastic parameters (3), displacement (2), stress (3) at 2 different times -> 16
Bytes/number:	8
required memory:	16 x 8 x 10^6 x 1.3 x 10^8

You can do this on a standard PC!

130 Mbyte memory (RAM)

… in 3D …

Example: seismic wave propagation, 3-D case

grid size:	1000x1000x1000
number of grid points:	10^9
parameters/grid point:	elastic parameters (3), displacement (3), stress (6) at 2 different times -> 24
Bytes/number:	8

These are no longer grand challenges but rather our standard applications on supercomputers

required memory: $24 \times 8 \times 10^9 \times 1.9 \times 10^{11}$

190 Gbyte memory (RAM)

Discretizing Earth

... this would mean

...we could discretize our planet with volumes of the size

$$4/3\ \pi(6371\text{km})^3 / 10^9 \approx 1000\text{km}^3$$

with an representative cube side length of 10km. Assuming that we can sample a wave with 20 points per wavelength we could achieve a dominant period T of

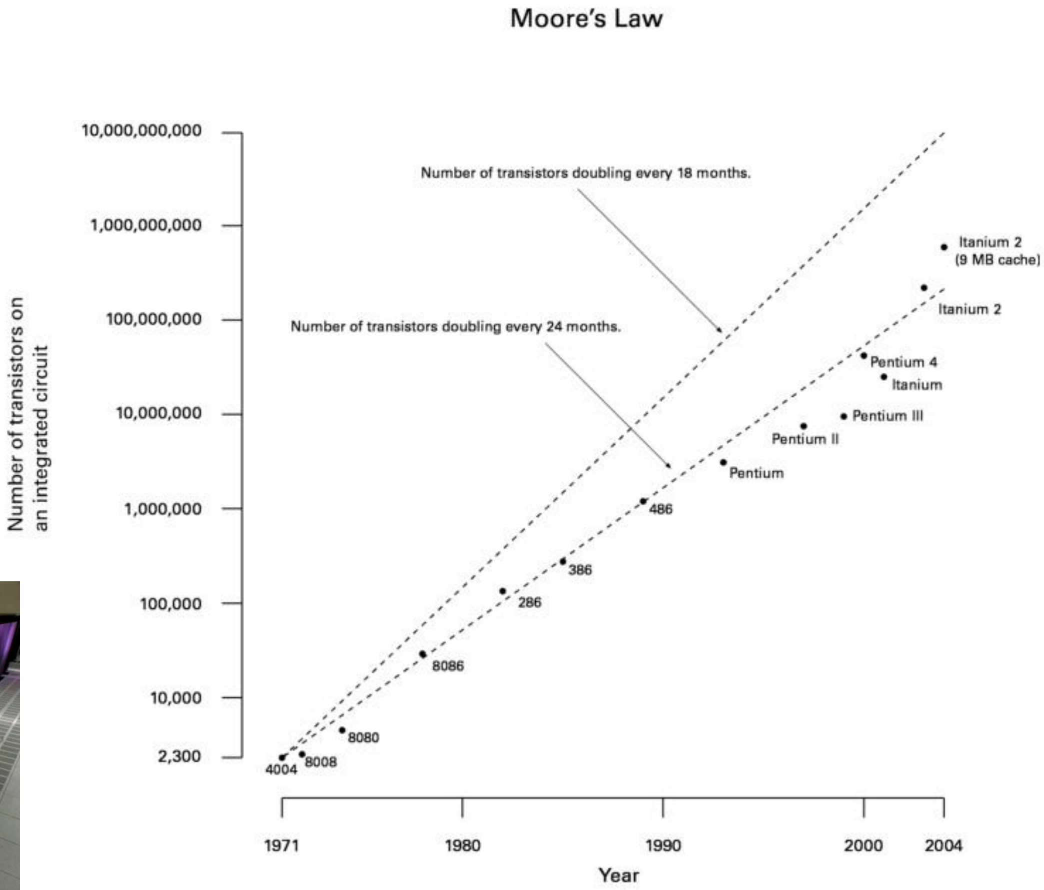
$$T = \lambda / c = 20\text{s}$$

for global wave propagation!

Moore‘s Law – Peak performance

1960: 1 MFlops
1970: 10MFlops
1980: 100MFlops
1990: 1 GFlops
1998: 1 TFlops
2008: 1 Pflops
20??: 1 EFlops

Roadrunner @ Los Alamos



Parallel Computations

What are parallel computations

Example: Hooke's Law
stress-strain relation

$$\boxed{\begin{gathered}\sigma_{xx} = \lambda\,(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}) + 2\mu\varepsilon_{xx} \\ \sigma_{ij}, \varepsilon_{ij} \Rightarrow f(x, y, z, t) \\ \lambda, \mu \Rightarrow f(x, y, z)\end{gathered}}$$

These equations hold at each point in time at all points in space

-> Parallelism

Loops

... in serial Fortran (F77) ...

at some time *t*

```
for i=1,nx
  for j=1,nz
  sxx(i,j)=lam(i,j)*(exx(i,j)+eyy(i,j)+ezz(i,j))+2*mu(i,j)*exx(i,j)
  enddo
enddo
```

add-multiplies are carried out sequentially

Programming Models

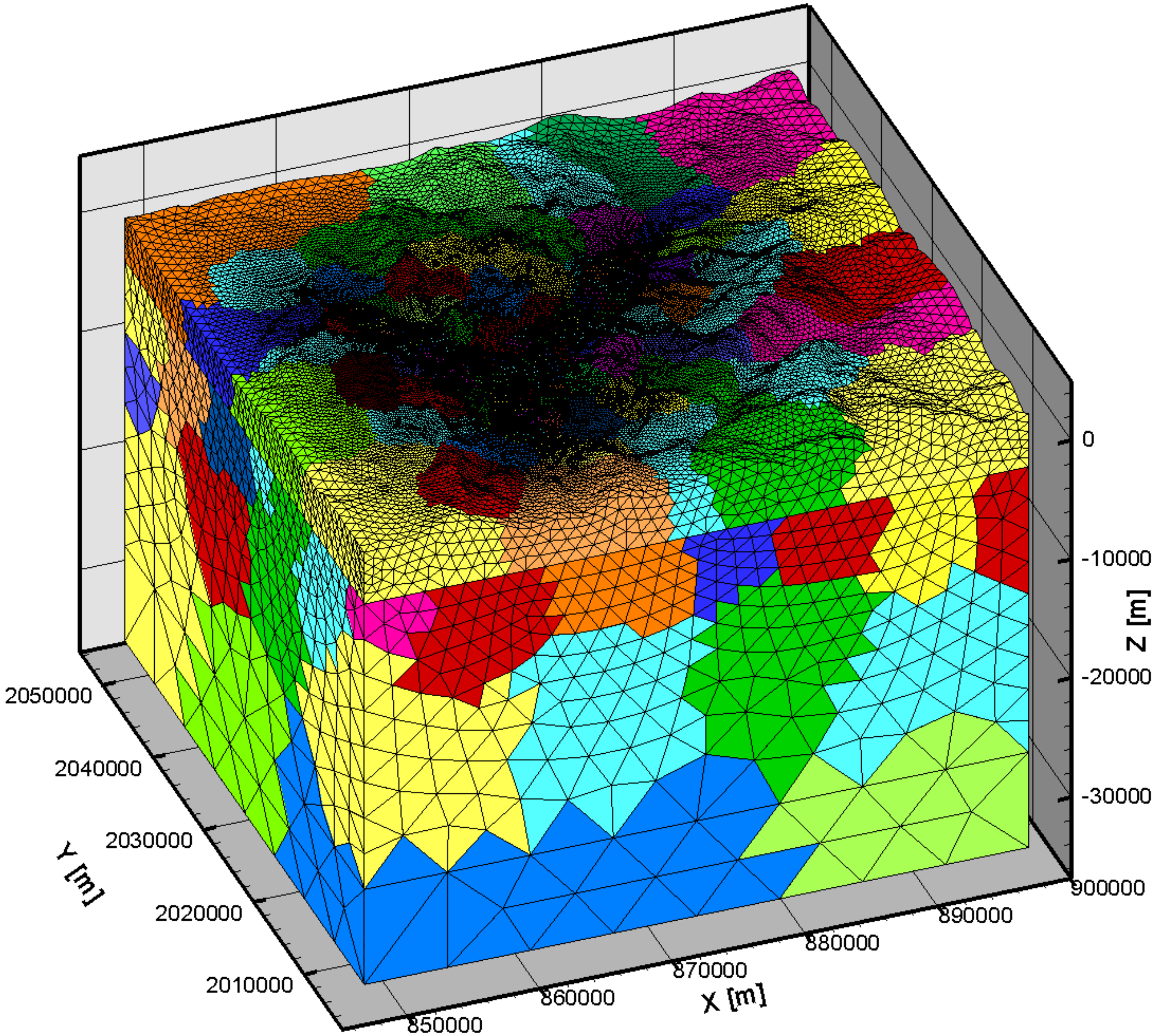
... in parallel Fortran (F90/95/03/05) ...
array syntax

```
sxx = lam*(exx+eyy+ezz) + 2*mu*exx
```

On parallel hardware each matrix is distributed on n processors. In our example no communication between processors is necessary. We expect, that the computation time reduces by a factor 1/n.

Today the most common parallel programming model is the Message Passing (MPI) concept, but ….
www.mpi-forum.org

Domain decomposition - Load balancing



Macro- vs. microscopic description

Macroscopic description:

The universe is considered a continuum. Physical processes are described using partial differential equations. The described quantities (e.g. density, pressure, temperature) are really averaged over a certain volume.

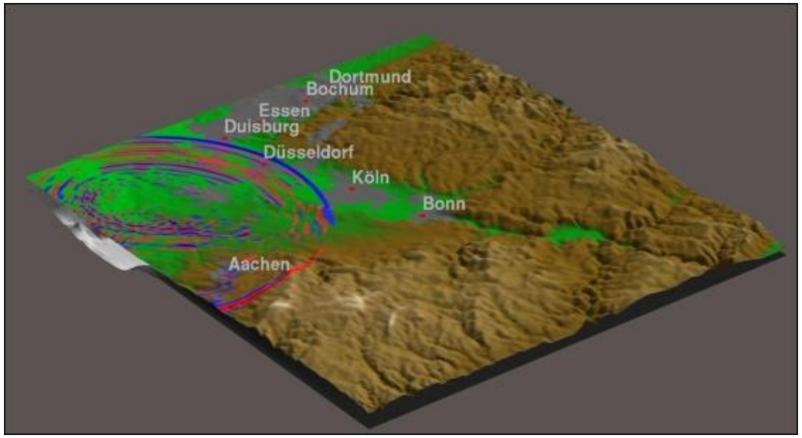
Microscopic description:

If we decrease the scale length or we deal with strong discontinous phenomena we arrive at the discrete world (molecules, minerals, atoms, gas particles). If we are interested in phenomena at this scale we have to take into account the details of the interaction between particles.

Macro- vs. microscopic description

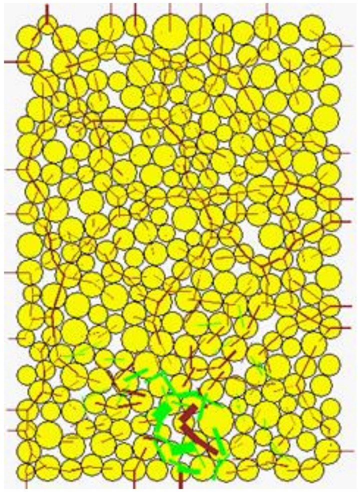
Macroscopic

- elastic wave equation
- Maxwell equations
- convection
- flow processes



Microscopic

- ruptures (e.g. earthquakes)
- waves in complex media
- tectonic processes
- gases
- flow in porous media



Partial Differential Equations in Geophysics

conservation equations

$$\boxed{\partial_t \rho + \partial_j (v_j \rho) = 0}$$

mass

$$\boxed{\partial_t (v_j \rho) + \partial_j (\rho\, v_i v_j - \sigma_{ij}) = f_i}$$

momentum

$$\boxed{f_i = s_i + g_i}$$

gravitation (g)
und sources (s)

Partial Differential Equations in Geophysics

gravitation

$$g_i = -\partial_i \Phi$$

gravitational field

$$\Delta\Phi = -\rho\, 4\pi\, G$$

$$\Delta = (\partial_x^2 + \partial_y^2 + \partial_z^2)$$

gravitational potential
Poisson equation

still missing: forces in the medium

->stress-strain relation

stress and strain

$$\sigma_{ij} = \theta_{ij} + c_{ijkl}\partial_l u_k$$

prestress and incremental stress

$$\varepsilon_{ij} = \frac{1}{2}(\partial_j u_i + \partial_i u_j + \partial_i u_m \partial_j u_m)$$

nonlinear stress-strain relation

$$\varepsilon_{ij} = \frac{1}{2}(\partial_j u_i + \partial_i u_j)$$

… linearized ...

Towards the elastic wave equation

special case: $v \Rightarrow 0$

small velocities

$$\boxed{\partial_t (v_j \rho) + \partial_j (\rho\, v_i v_j - \sigma_{ij}) = f_i}$$

$$\boxed{v_i \rightarrow 0 \Rightarrow \rho v_i v_j \approx 0}$$

We will only consider problems in the low-velocity regime.

Special pde's

hyperbolic differential equations
e.g. the acoustic wave equation

$$\boxed{\frac{1}{K}\partial_t^2 p - \partial_{x_i}\frac{1}{\rho}\partial_{x_i} p = -s}$$

K compression
s source term

parabolic differential equations
e.g. diffusion equation

$$\boxed{\partial_t T = D\partial_i^2 T}$$

T temperature
D thermal diffusivity

elliptical differential equations
z.B. static elasticity

$$\boxed{\partial_{x_i}^2 U(x) = F(x)}$$

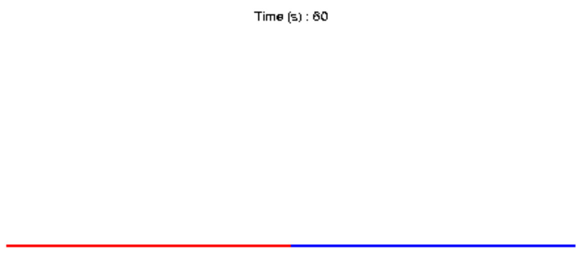
$$U = \partial_m u_m$$

$$F = \partial_m f_m / K$$

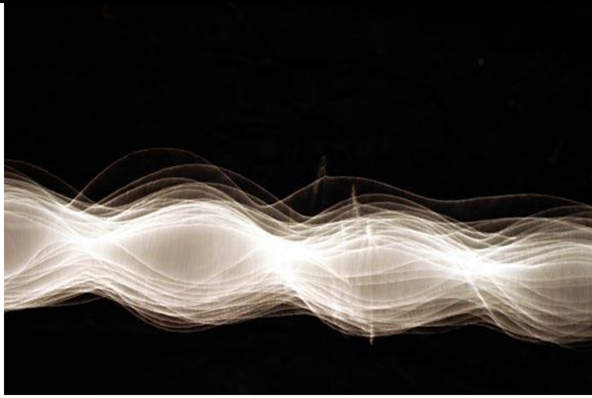
u displacement
f sources

Our Goal

- Approximate the wave equation with a discrete scheme that can be solved numerically in a computer
- Develop the algorithms for the 1-D wave equation and investigate their behavior
- Understand the limitations and pitfalls of numerical solutions to pde‘s
  - Courant criterion
  - Numerical anisotropy
  - Stability
  - Numerical dispersion
  - Benchmarking



The 1-D wave equation – the vibrating guitar string



$$\rho \partial_t^2 u(x,t) = \mu \partial_x^2 u(x,t) + f(x,t)$$

$$u\big|_{x=0} = u\big|_{x=L} = 0$$

$$u\big|_{x=0} = \partial_x u\big|_{x=L} = 0$$

u	displacement
ρ	density
μ	shear modulus
f	force term

Summary

Numerical method play an increasingly important role in all domains of geophysics.

The development of hardware architecture allows an efficient calculation of large scale problems through parallelization.

Most of the dynamic processes in geophysics can be described with time-dependent partial differential equations.

The main problem will be to find ways to determine how best to solve these equations with numerical methods.