Numerical methods

Motivation

Specific methods:

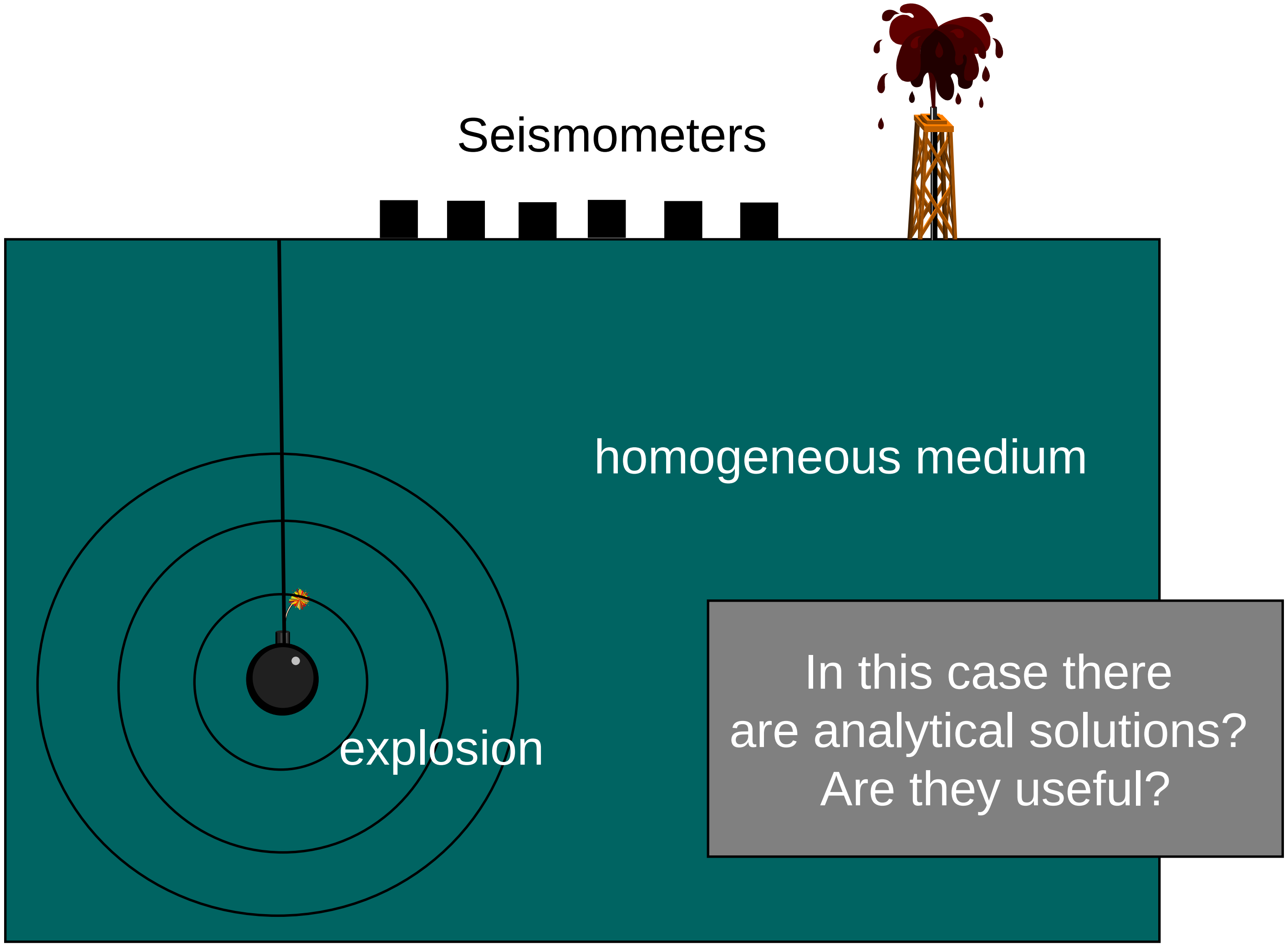
- Finite differences
- High-order FD methods
- Pseudospectral methods
- Finite elements
- Finite volumes

Applications

- Wave propagation
- Rupture problems
- Volcano seismology
- Global wave propagation
- Earthquake scenarios

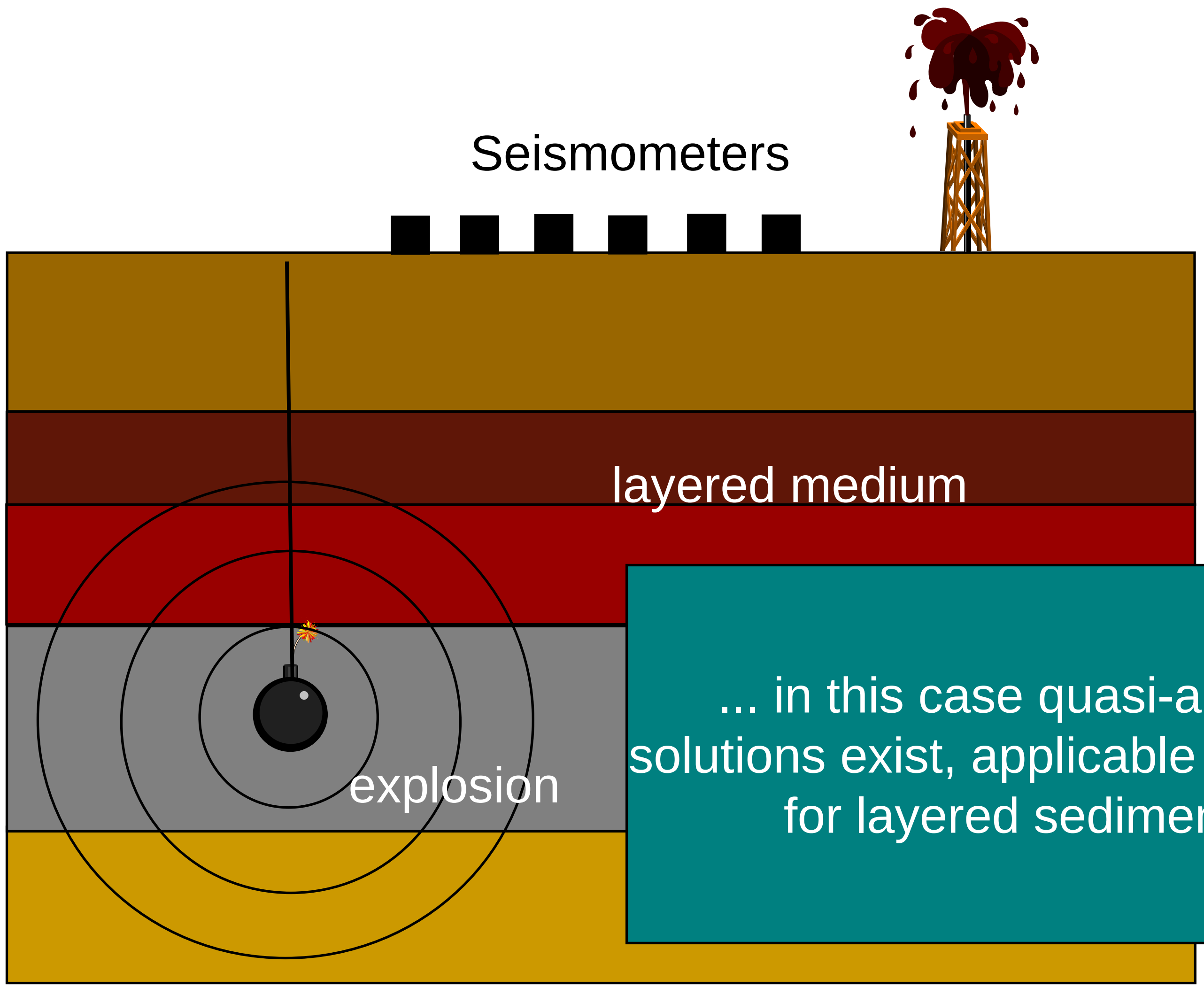
Why numerical methods?

Example: seismic wave propagation



Why numerical methods?

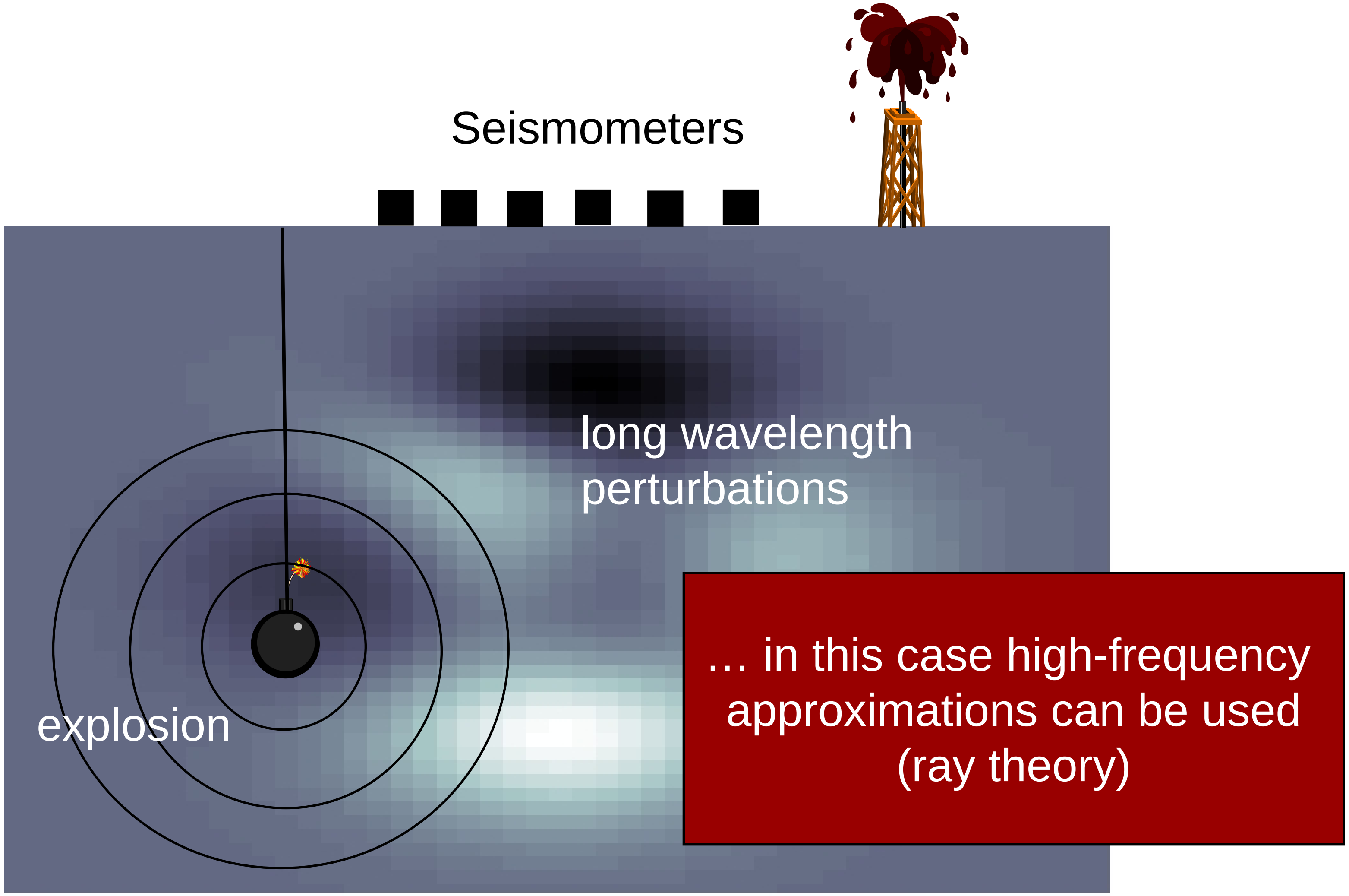
Example: seismic wave propagation



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

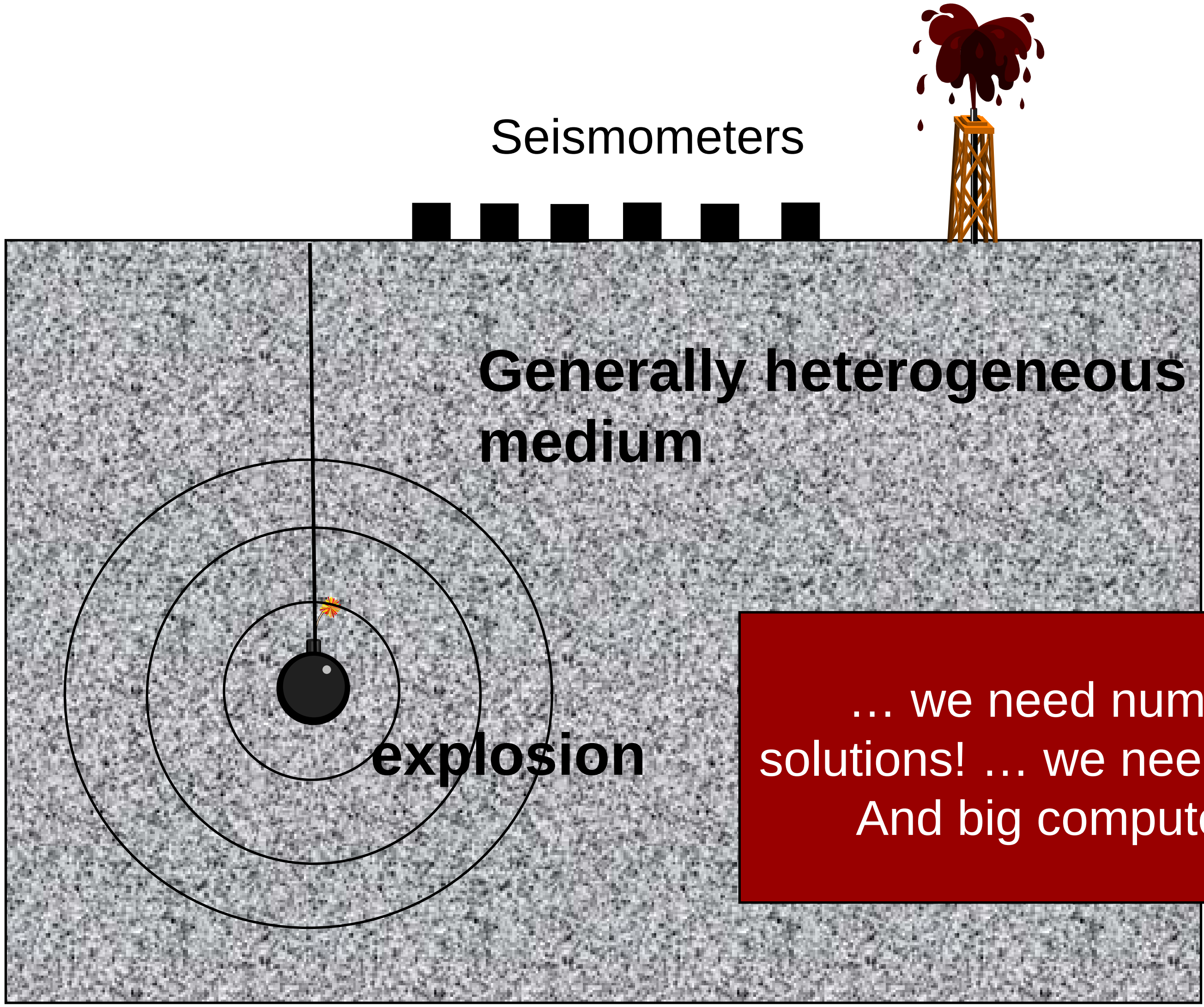
Why numerical methods?

Example: seismic wave propagation

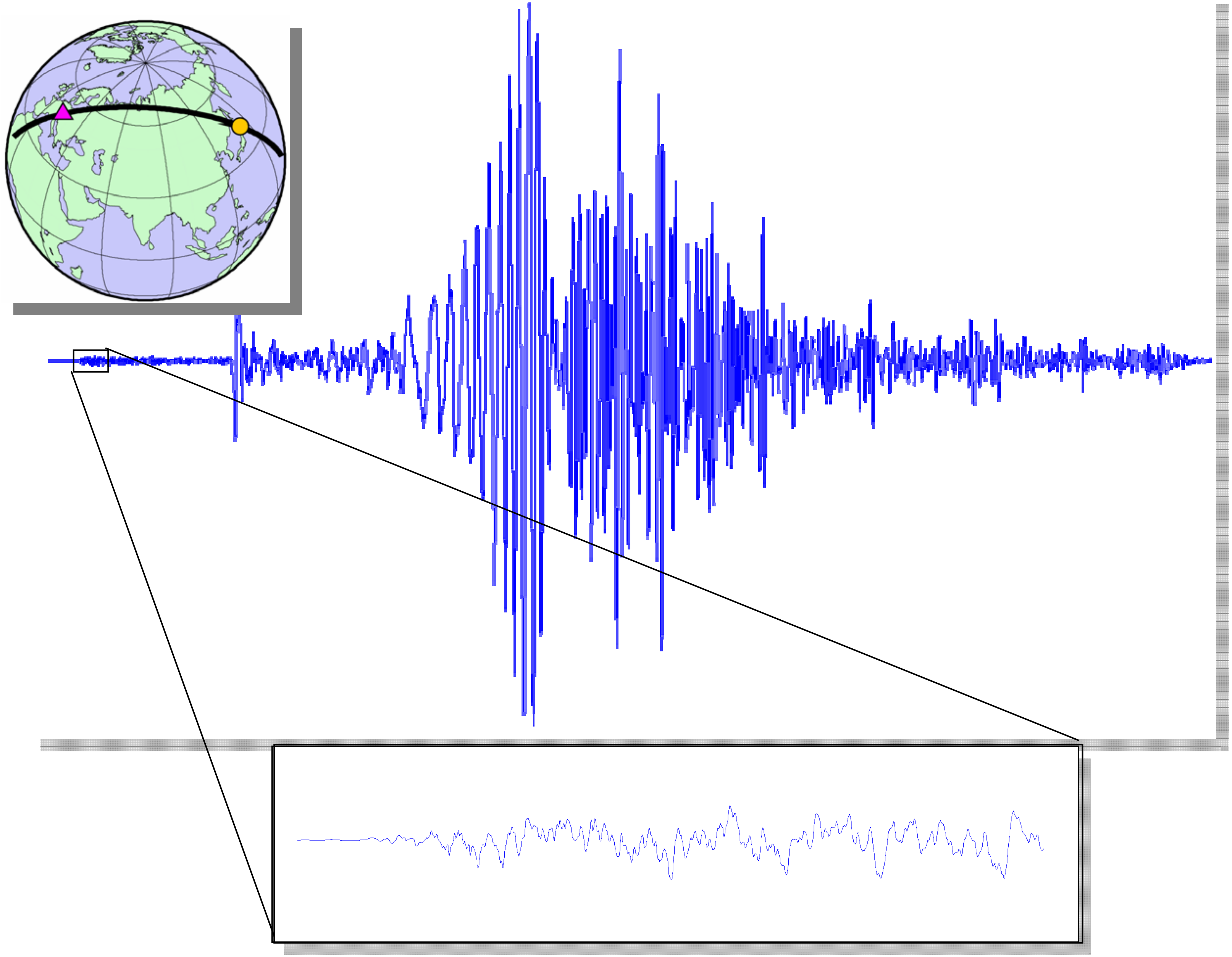


Why numerical methods

Example: seismic wave propagation



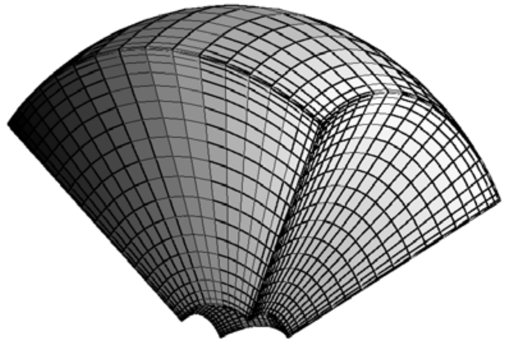
… we need numerical solutions! … we need grids! … And big computers …

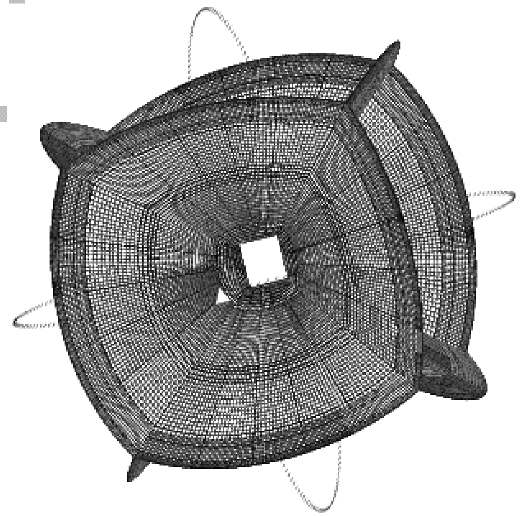


Spatial Scales and Memory

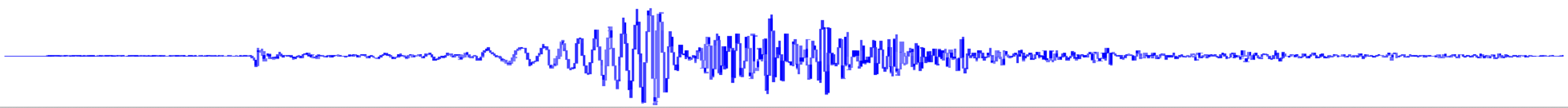
(back of the envelope)

Highest frequency:	**0.1 Hz**
Shortest wavelength:	20 km (crust)
Shortest wavelength:	50 km (mantle)
Grid points per wavelength:	5
Grid spacing:	2000 m (crust)
Grid spacing:	5000 m (mantle)

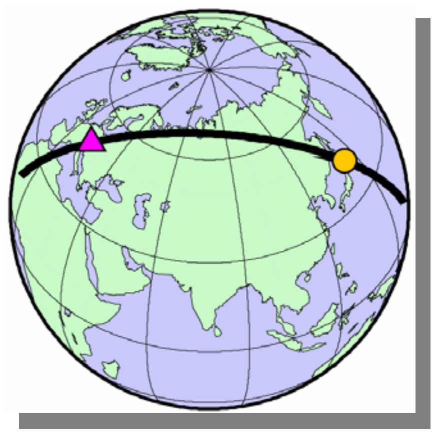


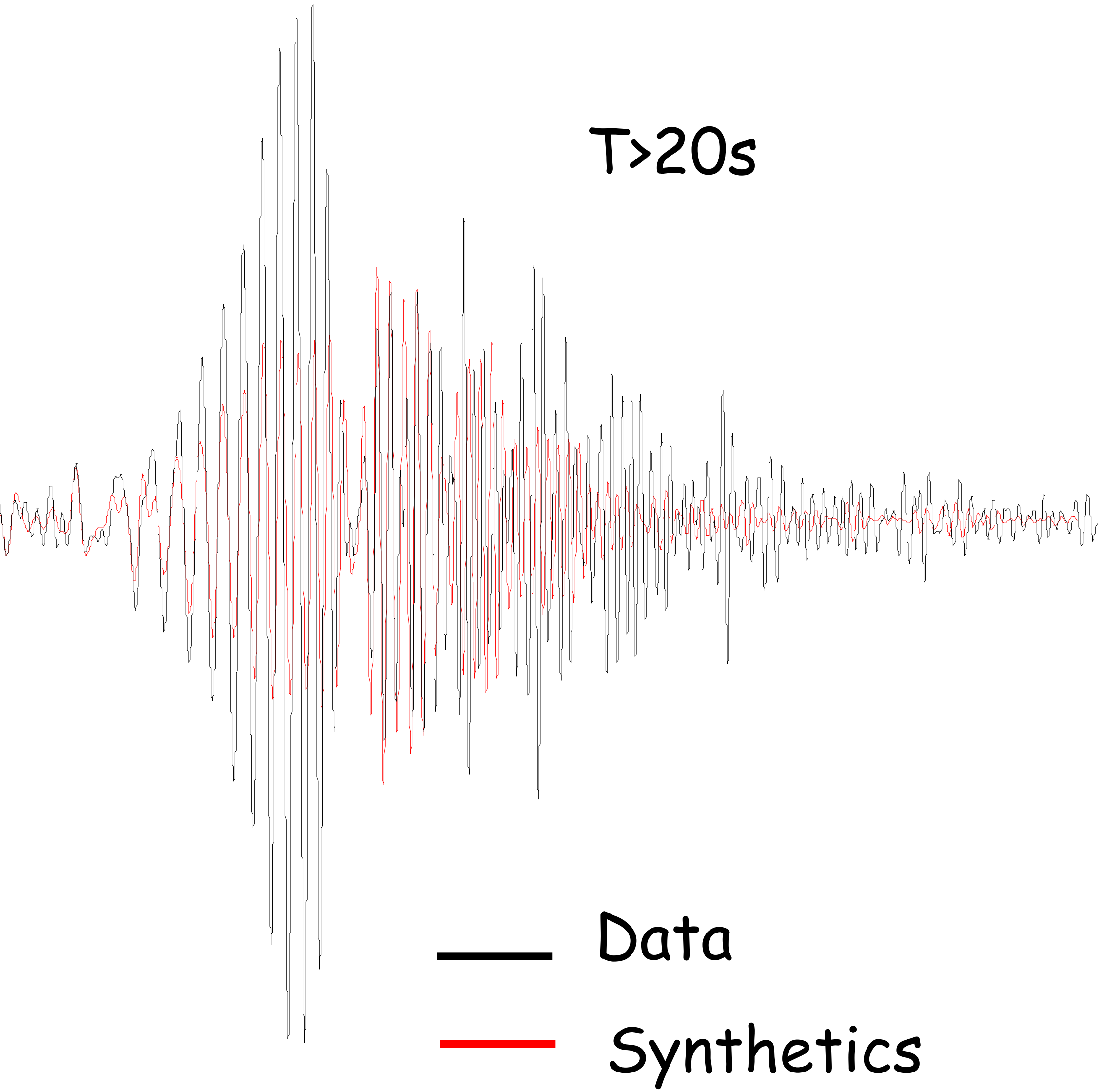


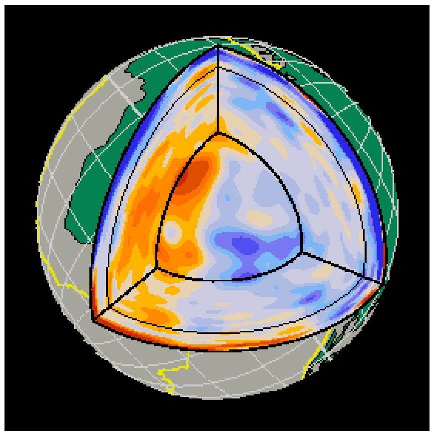
Required grid points: O(10^9)
Required memory: O(100 GBytes)



Data fitting - Inversion







Partial Differential Equations in Geophysics

$$\partial_t^2 p = c^2 \Delta p + s$$

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

P	pressure
c	acoustic wave speed
s	sources

The acoustic wave equation

- seismology
- acoustics
- oceanography
- meteorology

$$\partial_t C = k\Delta C - \mathbf{v} \bullet \nabla C - RC + p$$

C	tracer concentration
k	diffusivity
v	flow velocity
R	reactivity
p	sources

Diffusion, advection, Reaction

- geodynamics
- oceanography
- meteorology
- geochemistry
- sedimentology
- geophysical fluid dynamics

Numerical methods: properties

Finite differences →

- time-dependent PDEs
- seismic wave propagation
- geophysical fluid dynamics
- Maxwell's equations
- Ground penetrating radar

-> robust, simple concept, easy to parallelize, regular grids, explicit method

Finite elements →

- static and time-dependent PDEs
- seismic wave propagation
- geophysical fluid dynamics
- all problems

-> implicit approach, matrix inversion, well founded, irregular grids, more complex algorithms, engineering problems

Finite volumes →

- time-dependent PDEs
- seismic wave propagation
- mainly fluid dynamics

-> robust, simple concept, irregular grids, explicit method

Other Numerical methods:

Particle-based methods →

- lattice gas methods
- molecular dynamics
- granular problems
- fluid flow
- earthquake simulations

-> very heterogeneous problems, nonlinear problems

Boundary element methods →

- problems with boundaries (rupture)
- based on analytical solutions
- only discretization of planes

--> good for problems with special boundary conditions (rupture, cracks, etc)

***Pseudo*spectral methods** →

- orthogonal basis functions, special case of FD
- spectral accuracy of space derivatives
- wave propagation, GPR

-> regular grids, explicit method, problems with strongly heterogeneous media

What is a finite difference?

Common definitions of the derivative of f(x):

$$\partial_x f = \lim_{dx \to 0} \frac{f(x+dx) - f(x)}{dx}$$

$$\partial_x f = \lim_{dx \to 0} \frac{f(x) - f(x-dx)}{dx}$$

$$\partial_x f = \lim_{dx \to 0} \frac{f(x+dx) - f(x-dx)}{2dx}$$

These are all correct definitions in the limit dx->0.

But we want dx to remain **FINITE**

What is a finite difference?

The equivalent ***approximations*** of the derivatives are:

$$\partial_x f^+ \approx \frac{f(x+dx) - f(x)}{dx}$$

forward difference

$$\partial_x f^- \approx \frac{f(x) - f(x-dx)}{dx}$$

backward difference

$$\partial_x f \approx \frac{f(x+dx) - f(x-dx)}{2dx}$$

centered difference

The big question:

How good are the FD approximations?

$$\approx \neq =$$

This leads us to Taylor series....

Our first FD algorithm (ac1d.m) !

$$\partial_t^2 p = c^2 \Delta p + s$$

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

P	pressure
c	acoustic wave speed
s	sources

Problem: Solve the 1D acoustic wave equation using the finite Difference method.

Solution:

$$p(t+dt) = \frac{c^2 dt^2}{dx^2}\left[p(x+dx) - 2p(x) + p(x-dx)\right] + 2p(t) - p(t-dt) + sdt^2$$

Problems: Stability

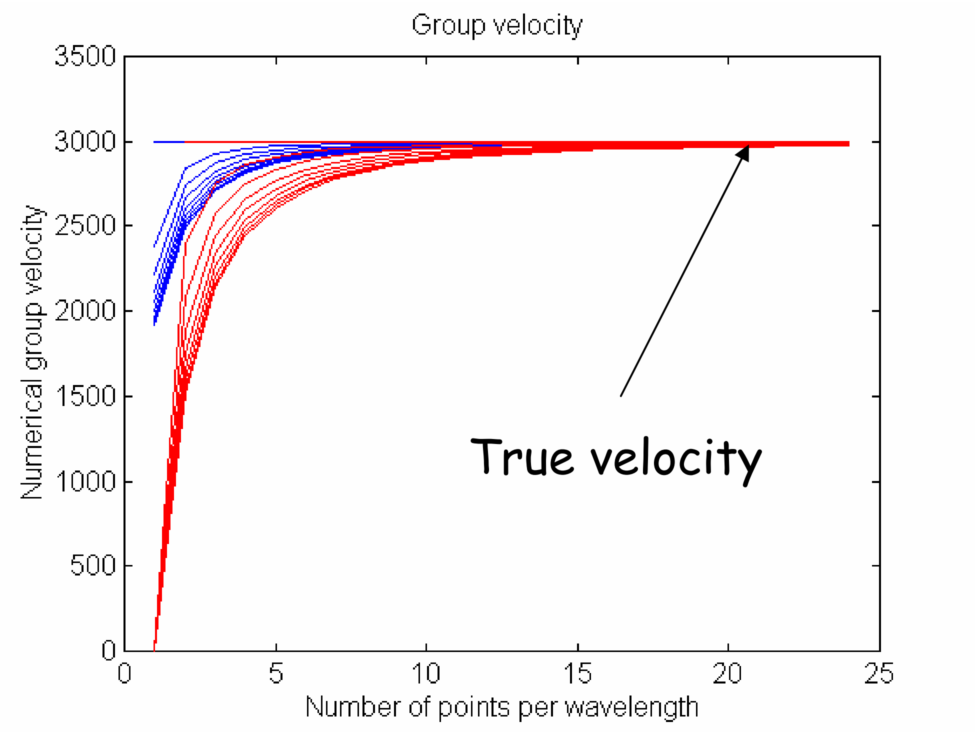
$$p(t+dt) = \frac{c^2 dt^2}{dx^2}\left[p(x+dx) - 2p(x) + p(x-dx)\right] + 2p(t) - p(t-dt) + sdt^2$$

Stability: Careful analysis using harmonic functions shows that a stable numerical calculation is subject to special conditions (conditional stability). This holds for many numerical problems.

$$c\frac{dt}{dx} \le \varepsilon \approx 1$$

Problems: Dispersion

$$p(t+dt) = \frac{c^2 dt^2}{dx^2}\left[p(x+dx) - 2p(x) + p(x-dx)\right] + 2p(t) - p(t-dt) + sdt^2$$



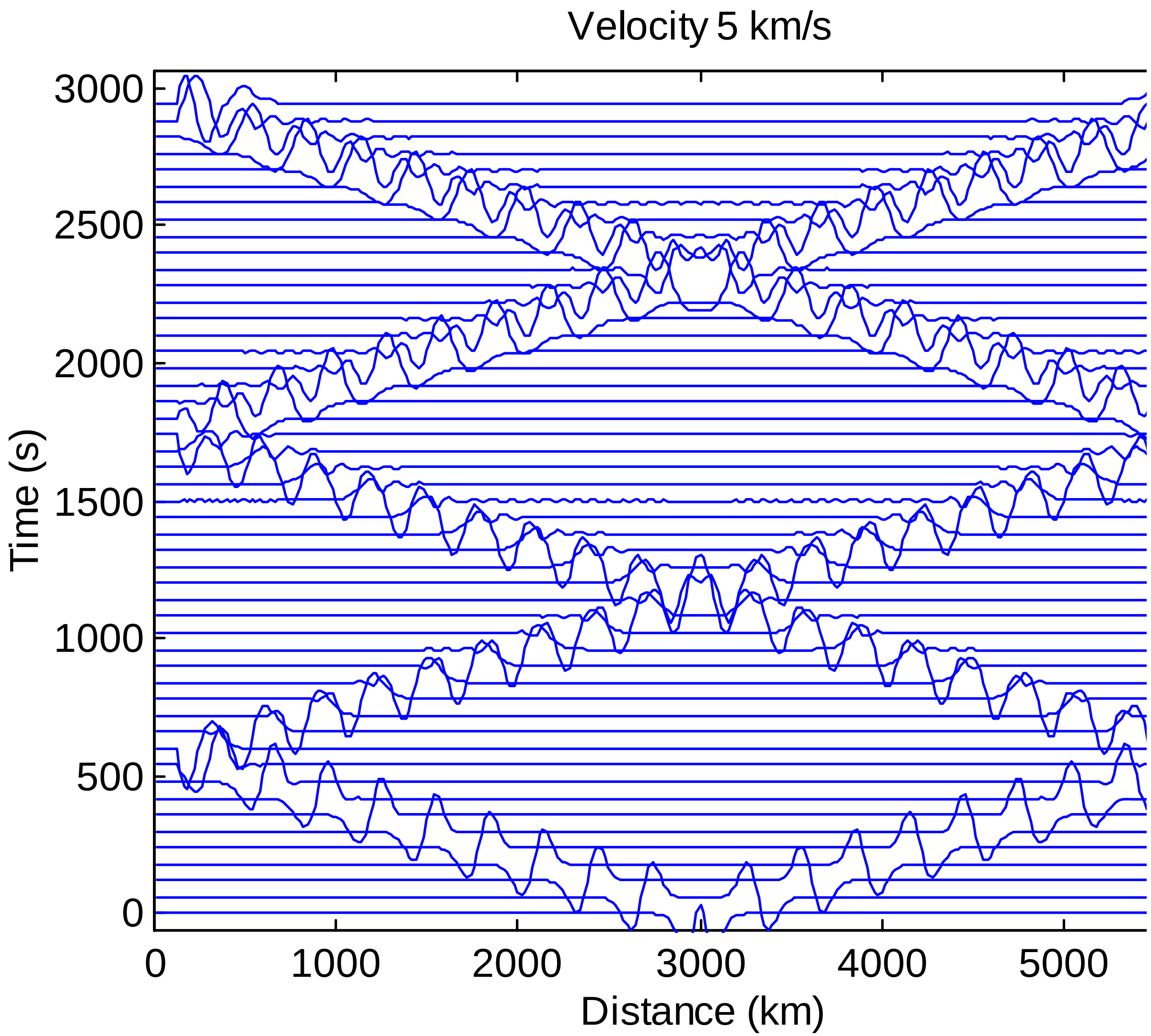
Dispersion: The numerical approximation has artificial dispersion, in other words, the wave speed becomes frequency dependent.
You have to find a frequency bandwidth where this effect is small.
The solution is to use a sufficient number of **grid points per wavelength.**

Our first FD code!

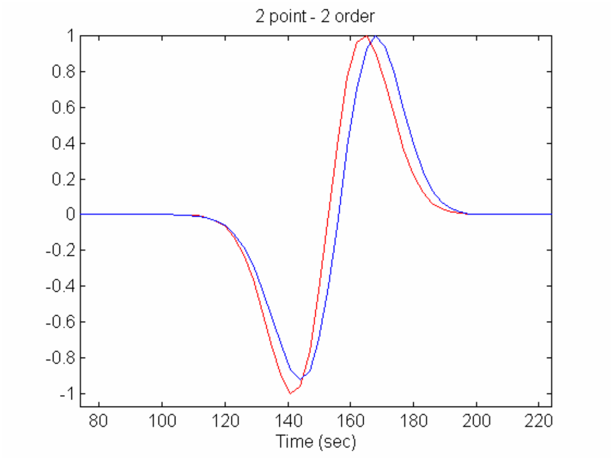
$$p(t+dt)=\frac{c^2 dt^2}{dx^2}\left[p(x+dx)-2p(x)+p(x-dx)\right]$$
$$+2p(t)-p(t-dt)+sdt^2$$

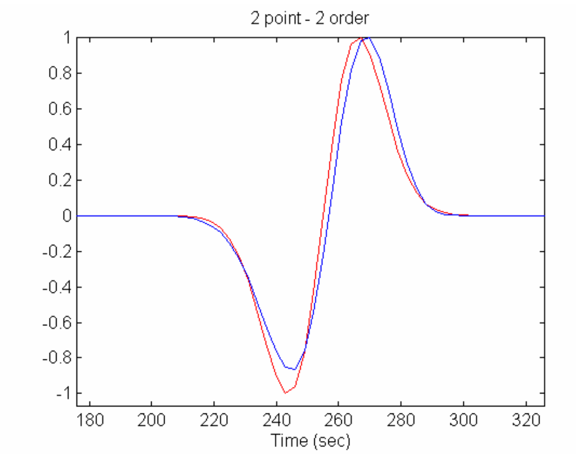
```
% Time stepping

for i=1:nt,

   % FD

   disp(sprintf(' Time step : %i',i));

   for j=2:nx-1
      d2p(j)=(p(j+1)-2*p(j)+p(j-1))/dx^2; % space derivative
   end
   pnew=2*p-pold+d2p*dt^2;                % time extrapolation
   pnew(nx/2)=pnew(nx/2)+src(i)*dt^2;     % add source term
   pold=p;                          % time levels
   p=pnew;
   p(1)=0;              % set boundaries pressure free
   p(nx)=0;

   % Display
   plot(x,p,'b-')
   title(' FD ')
   drawnow

end
```

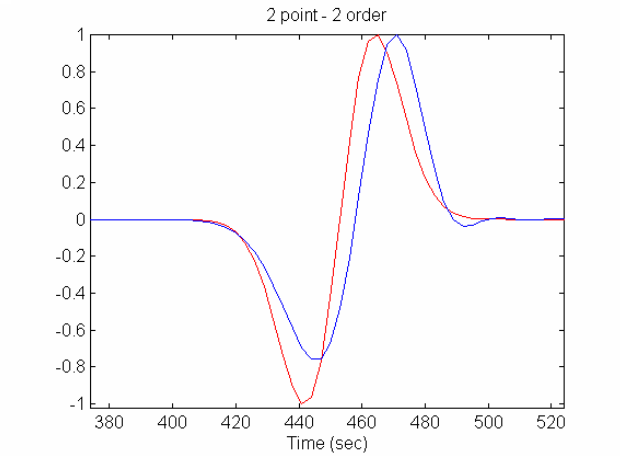
Snapshot Example

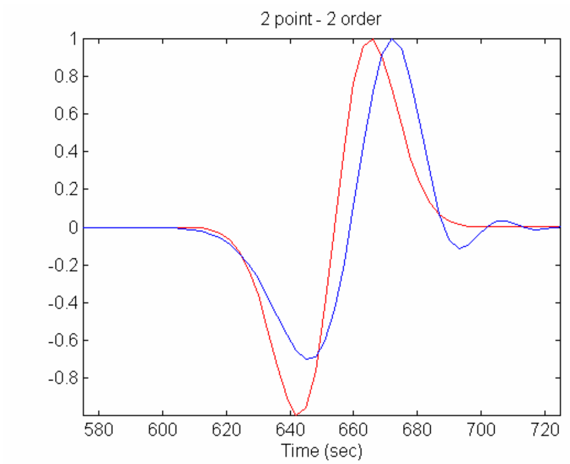


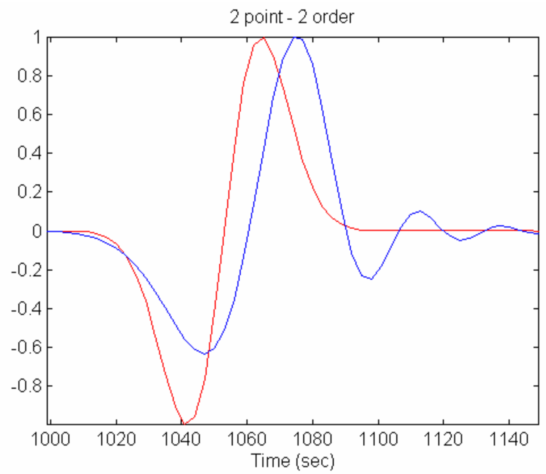
Seismogram Dispersion

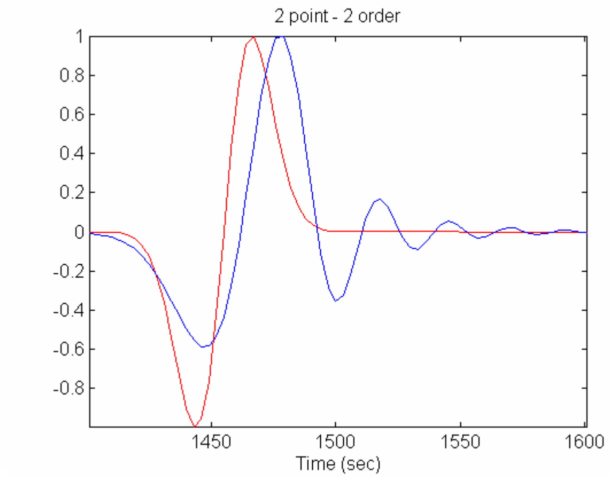












Finite Differences - Summary

- Conceptually the most simple of the numerical methods and can be learned quite quickly
- Depending on the physical problem FD methods are conditionally stable (relation between time and space increment)
- FD methods have difficulties concerning the accurate implementation of boundary conditions (e.g. free surfaces, absorbing boundaries)
- FD methods are usually explicit and therefore very easy to implement and efficient on parallel computers
- FD methods work best on regular, rectangular grids