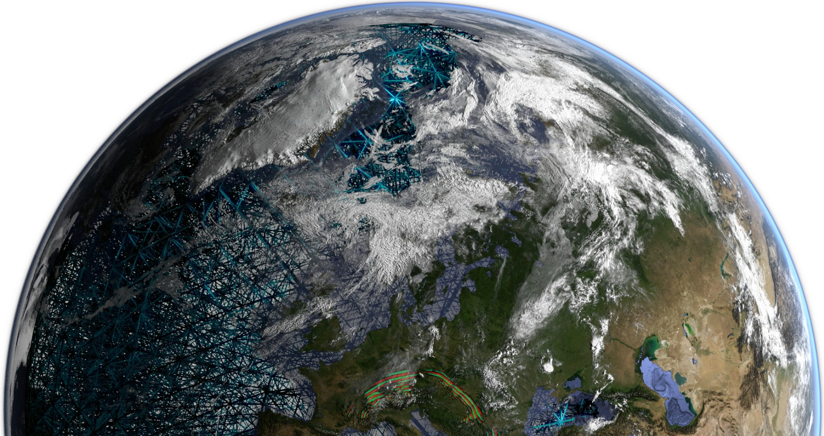
Computational Seismology: Introduction

Heiner Igel

Department of Earth and Environmental Sciences
Ludwig-Maximilians-University Munich

Scope

Introduction



Goals of the course

- Understand methods that allow the calculation of **seismic wavefields in heterogeneous media**
- Prepare you to be able to understand Earth science papers that are based on **3-D wave simulation tools** (e.g., seismic exploration, full waveform imaging, shaking hazard, volcano seismology)
- Know **the dangers, traps, and risks of using simulation tools** (as black boxes -> turning black boxes into white boxes)
- Providing you with basic knowledge about common **numerical methods**:
- Knowing **application domains** of the various methods and guidelines what method works best for various problems
- ... and having fun simulating waves ...

Course structure

- **Introduction**
    - What is computational seismology?
    - When and why do we need numerical maths?
- **Elastic waves in the Earth**
    - What to expect when simulating seismic wave fields?
    - Wave equations
    - Seismic waves in simple media (benchmarks)
    - Seismic sources and radiation patterns
    - Green's functions, linear systems
- **Numerical approximations of the 1 (2, 3) -D wave equation**
    - Finite-difference method
    - Pseudospectral method
    - Spectral-element method
    - Discontinuous Galerkin method
- **Applications in the Earth Sciences**

Who needs Computational Seismology

Many problems rely on the analysis of **elastic wavefields**

- **Global seismology** and tomography of the Earth's interior
- The quantification of **strong ground motion - seismic hazard**
- The understanding of the **earthquake source process**
- The monitoring of **volcanic processes** and the forecasting of eruptions
- **Earthquake early warning** systems
- **Tsunami early warning** systems
- Local, regional, and global **earthquake services**
- Global monitoring of **nuclear tests**
- **Laboratory scale analysis** of seismic events

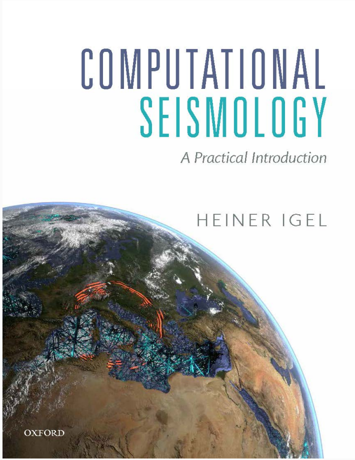
Who needs Computational Seismology (cont'd)

(...)

- Ocean generated **noise measurements** and cross-correlation techniques
- **Planetary seismology**
- **Exploration geophysics**, reservoir scale seismics
- **Geotechnical engineering** (non-destructive testing, small scale tomography
- **Medical applications**, breast cancer detection, reverse acoustics

Literature

- Computational Seismology: A Practical Introduction (Oxford University Press, 2016)
- 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), also as book Cambridge University Press
- Fichtner, Full Seismic Waveform Modelling and Inversion, Springer Verlag, 2010.



What is Computational Seismology?

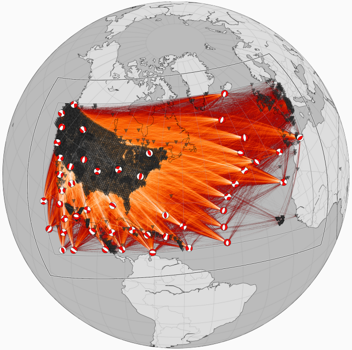
What is Computational Seismology?

We define **computational seismology** such that it **involves the complete solution of the seismic wave propagation (and rupture) problem for arbitrary 3-D models by numerical means.**

What is not covered ...

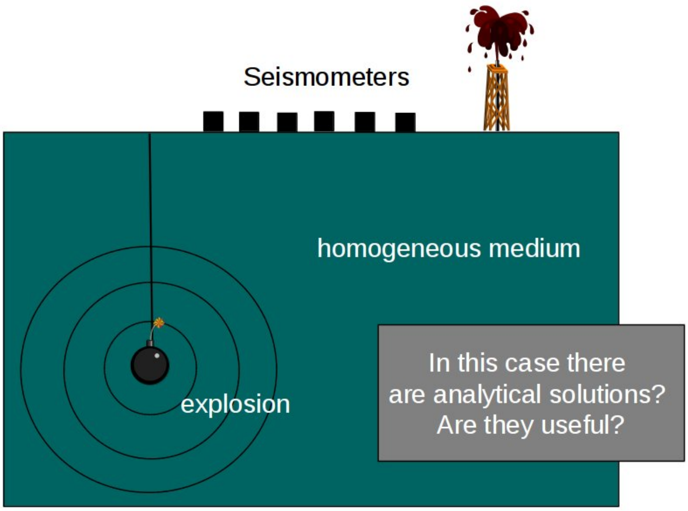
- Ray-theoretical methods
- Quasi-analytical methods (e.g., normal modes, reflectivity method)
- Frequency-domain solutions
- Boundary integral equation methods
- Discrete particle methods

These methods are important for benchmarking numerical solutions!

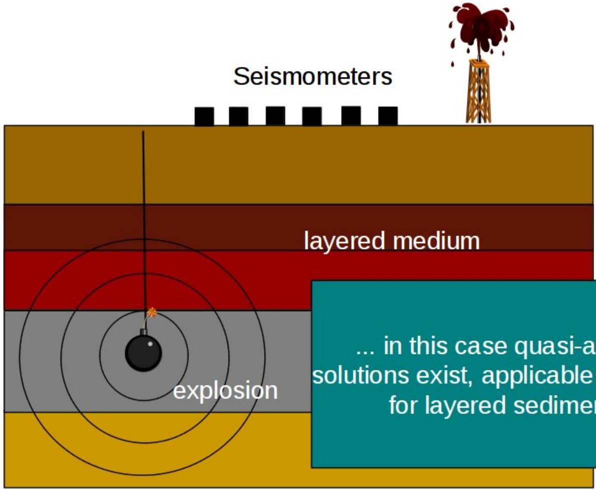


Why numerical methods?

Why numerical methods?

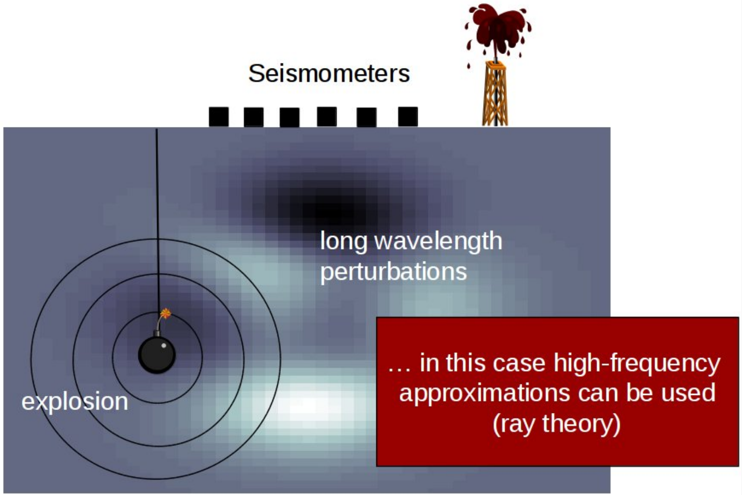


Why numerical methods?

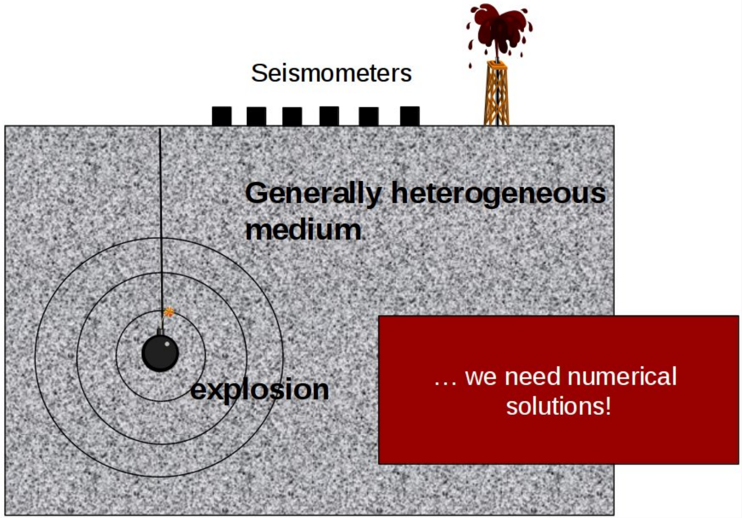


... in this case quasi-analytical solutions exist, applicable for example for layered sediments ...

Why numerical methods?



Why numerical methods?



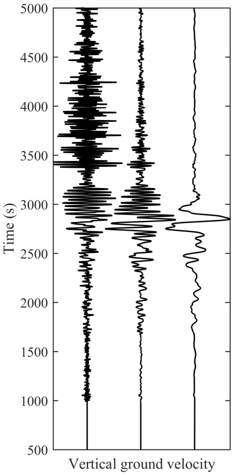
Waves and Computers

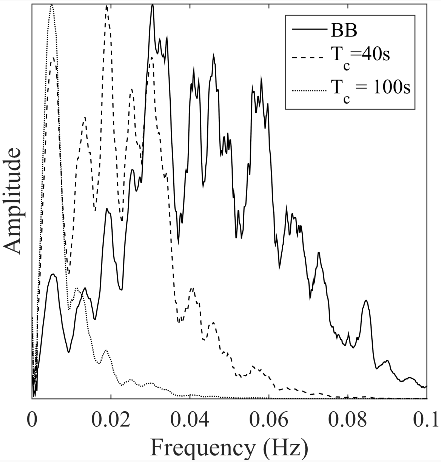
Computational Seismology, Memory, and Compute Power

Numerical solutions necessitate the discretization of Earth models. Estimate how much memory is required to store the Earth model and the required displacement fields.

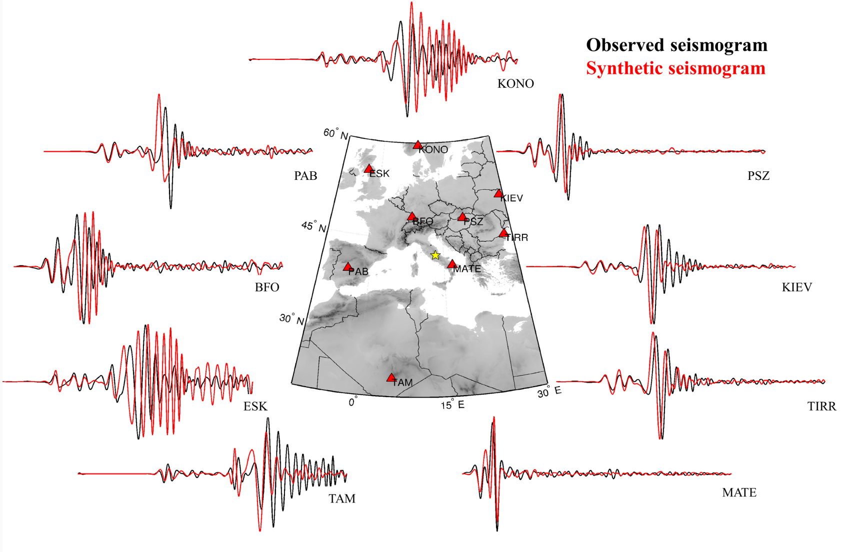
Are we talking laptop or supercomputer?

Seismic Wavefield Observations



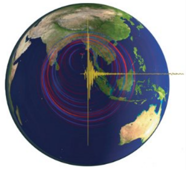


Matching Wavefield Observations



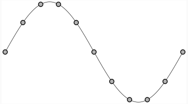
Exercise: Sampling a global seismic wavefield

- The highest frequencies that we observe for global wave fields is 1Hz.
- We assume a homogeneous Earth (radius 6371km).
- P velocity $v_p = 10 km/s$ and the v_p/v_s ratio is $\sqrt{3}$
- We want to use 20 **grid points (cells) per wavelength**
- How many grid cells would you need (assume cubic cells).
- What would be their size?
- How much memory would you need to store one such field (e.g., density in single precision).



You may want to make use of

$$c = \frac{\lambda}{T} = \lambda f = \frac{\omega}{k}$$



Exercise: Solution (Matlab)

```
% Earth volume
```
$v_e = 4/3 * pi * 6371^3$;

```
% smallest velocity (ie, wavelength)
vp=10; vs=vp/sqrt(3);
% Shortest Period
T=10;
% Shortest Wavelength
lam=vs*T;
% Number of points per wavelength and
% required grid spacing
```
$nplambda = 20$;

$dx = lam/nplambda$;

```
% Required number of grid cells
```
$nc = v_e/(dx^3)$;

```
% Memory requirement (TBytes)
```
$mem = nc * 8/1000/1000/1000/1000$;

Results ($@T = 1s$) : 360 TBytes

Results ($@T = 10s$) : 360 GBytes

Results ($@T = 100s$) : 360 MBytes

Computational Seismology, Memory, and Compute Power

1960: 1 MFlops

1970: 10MFlops

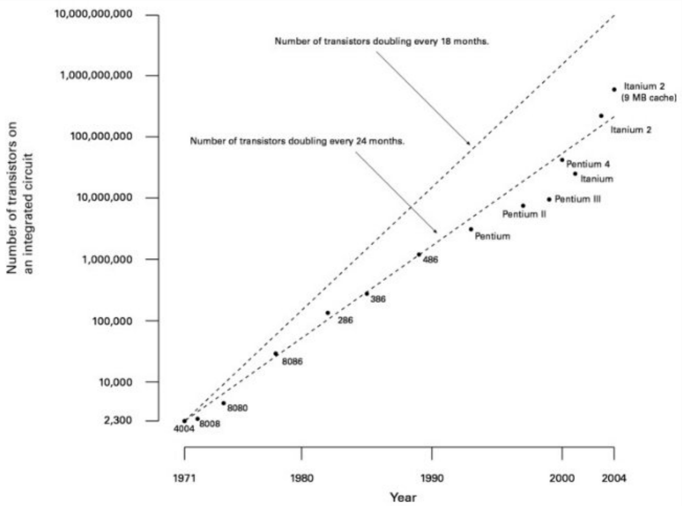
1980: 100MFlops

1990: 1 GFlops

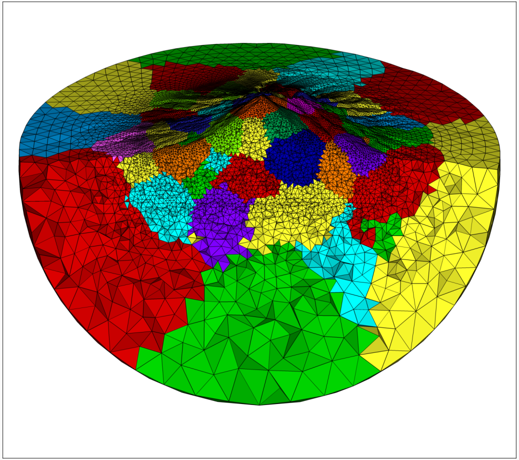
1998: 1 TFlops

2008: 1 Pflops

20??: 1 EFlops

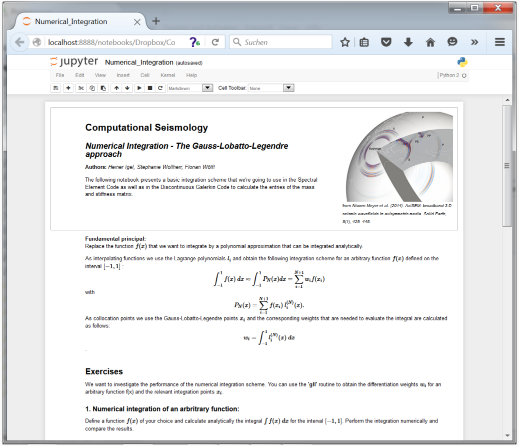


Computational Seismology, Parallel Computing



Computational Seismology, Practical Exercises, Jupyter Notebooks

- *Jupyter notebooks* are interactive documents that work in any browser
- Simple text editing
- Inclusion of graphics
- Equations with Latex
- Executable code cells with Python (or else)
- The coolest thing since ...
- Many examples on: *www.seismo-live.org*



Summary

- Computational wave propagation (as defined here) is turning more and more into a routine tool for many fields of Earth sciences
- There is a zoo of methods and in many cases it is not clear which method works best for a specific problem
- For single researchers (groups, institutions) it is no longer possible to code, implement, maintain an algorithm efficiently
- More and more well engineered community codes become available (e.g., sofi3d, specfem, seissol)
- Community platforms (e.g., verce.eu) are developing facilitating simulation tasks

This course aims at understanding the theory behind these methods and understanding their domains of application.