Computational Geophysics: What is the best strategy for my problem?

IGPP Short Course, September 15 - 20, 2022

Heiner Igel

September 7, 2022

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

Format of Short Course

- **1-1.5 hour blocks** with lectures and pencil/paper or computer exercises
- **Andreas Brotzer** helps with tutorials
- **Questions** by you (always allowed during lecture!). **Interrupt** me!
- Interchange between **paper/pencil** exercises and **Jupyter notebook** exercises

Course Content

- **Day 1 - Morning**
  - Introduction
  - The discrete world, fundamentals of wave propagation, wave equations, analytical solutions, reciprocity, superposition principle, dispersion, linear systems, homogenization
  - Introduction to Github, Jupyter notebooks, Python ecosystem, simple examples

- **Day 1 - Afternoon**
  - Finite-difference method, high-order operators, numerical dispersion, stability, staggered grids
  - 1D and 2D wave propagation, benchmarking, numerical anisotropy, heterogeneous models

Course Content

- **Day 2 - Morning**
  - The Pseudospectral Method, exact interpolation, Fourier transforms, convolutional operators, Fourier and Chebyshev Method
  - Fourier and Chebyshev derivative, wave propagation, comparison with FD

- **Day 2 - Afternoon**
  - Linear Finite-element method, Galerkin principle, weak form, static and dynamic solution, basis functions
  - Static elasticity problem, comparison with relaxation method, 1D elastic wave equation

Course Content

- **Day 3 - Morning**
  - The spectral-element method, Lagrange interpolation, numerical Integration
  - 1D elastic wave equation, heterogeneous models, convergence test
- **Day 3 - Afternoon**
  - Finite Volume and Discontinuous Galerkin Method, numerical fluxes
  - 1D wave equation with Discontinuous Galerkin Method, final discussion, Loose ends

Introduction

Goals of course

- What are some fundamental aspects of computational **wave propagation**?
- Is it a tough or an easy problem as far as **computational resources** are concerned?
- Which **numerical methods** are on the market, basic principles, and domains of application?
- What options do you have to get training (**Jupyter notebooks**, **COURSERA**, etc) ...?

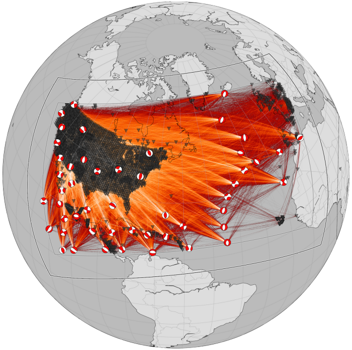
What is Computational Seismology?

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 ... but you can do tomography with ...

- **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!



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

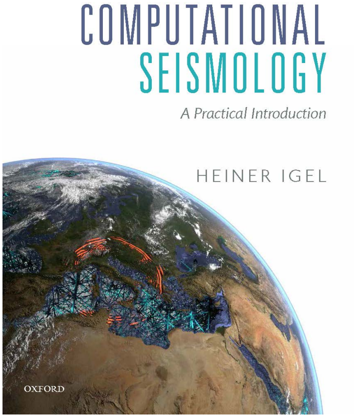
Who needs Computational Seismology (cont'd)

(...)

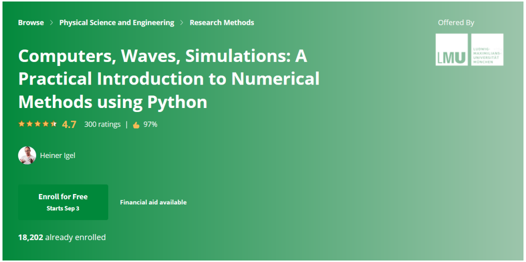
- Ocean generated **noise measurements** and cross-correlation techniques
- **Planetary seismology - Apollo, INSIGHT**
- **Exploration geophysics**, reservoir scale seismics
- **Geotechnical engineering** (non-destructive testing, small scale tomography)
- **Medical applications**, breast cancer detection,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

Literature

- Igel, **Computational Seismology: A Practical Introduction** (Oxford University Press, 2016)
- Shearer, **Introduction to Seismology** (3rd edition, 2019)
- Aki and Richards, **Quantitative Seismology** (1st edition, 1980)
- Mozco, **The Finite-Difference Modelling of Earthquake Motions** (Cambridge University Press)
- Fichtner, **Full Seismic Waveform Modelling and Inversion** (Springer Verlag, 2010).



9-week course in Computational wave propagation on COURSERA (free!)



Covers the finite-difference, pseudospectral, finite-element, and spectral-element method.

Why numerical methods?

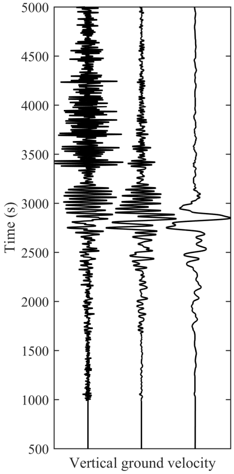
Photo ©Überraschungsbilder via Wikimedia commons

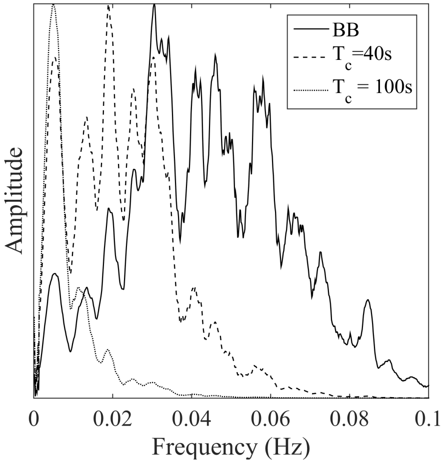
Computational Seismology, Memory, and Compute Power

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?

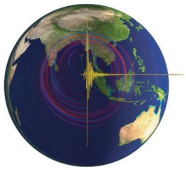
Seismic Wavefield Observations





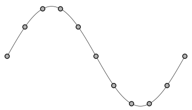
Exercise: Sampling a global seismic wavefield

- 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

```
# Physics

# Earth's Velocity (S waves have smaller wavelength)
c = 10./np.sqrt(3)  # km/s

# Target Frequency
T = 1.  # Hz  (1 Hz is usually the highets frequency of global wave fields)

# Wavelength
lam = c * T   # in km

print(' Wavelength: ', lam, ' km ')
```

```
Wavelength:  5.773502691896258  km
```

```
# Computational method
npts = 20  # number of grid points per wavelength

# Required spatial discretization
dx = lam/npts # in km

# Size of a volume cell
dv = dx**3    # in km**3

print(' Size of Volume Cell',dv, 'km**3 ')
```

```
Size of Volume Cell 0.024056261216234418 km**3
```

```
# Memory requirement (Volume Earth / dv * 8 bytes)
mem = 4./3.*np.pi*6371**3/dv*8

print(' Memory requirement : ', mem/1000/1000/1000/1000, 'TBytes')
```

```
Memory requirement :  360.22452769668104 TBytes
```

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

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

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

Exercise:

```
# Physics

# Earth's Velocity (S waves have smaller wavelength)
c = 10./np.sqrt(3)  # km/s

# Target Frequency
T = 1.  # Hz  (1 Hz is usually the highets frequency of global wave fields)

# Wavelength
lam = c * T   # in km

print(' Wavelength: ', lam, ' km ')
```

```
Wavelength:  5.773502691896258  km
```

```
# Computational method
npts = 20  # number of grid points per wavelength

# Required spatial discretization
dx = lam/npts # in km

# Size of a volume cell
dv = dx**3    # in km**3

print(' Size of Volume Cell',dv, 'km**3 ')
```

```
Size of Volume Cell 0.024056261216234418 km**3
```

```
# Memory requirement (Volume Earth / dv * 8 bytes)
mem = 4./3.*np.pi*6371**3/dv*8

print(' Memory requirement : ', mem/1000/1000/1000/1000, 'TBytes')
```

```
Memory requirement :  360.22452769668104 TBytes
```

Take a few minutes to think of your current or future simulation problem (whatever physical process). What is the scale of the problem? What are the wavelengths involved? What is the size of the physical domain? Laptop or supercomputer?

Computational Seismology, Memory, and Compute Power

1960: 1 MFlops

1970: 10MFlops

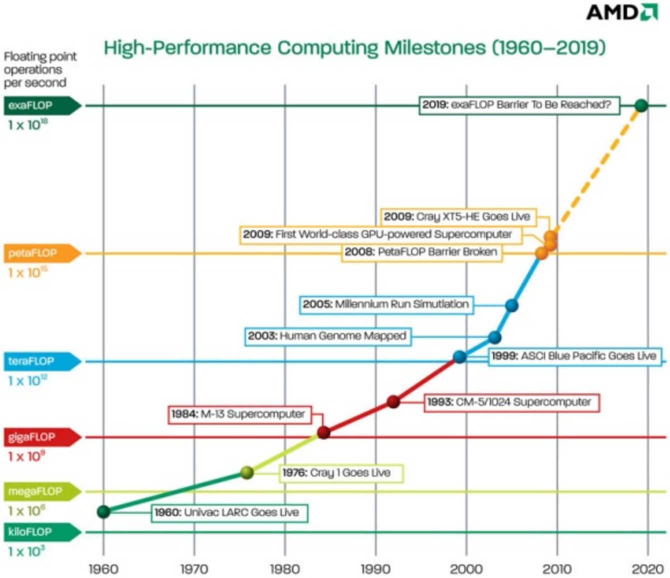
1980: 100MFlops

1990: 1 GFlops

1998: 1 TFlops

2008: 1 Pflops

2021: 1 EFlops

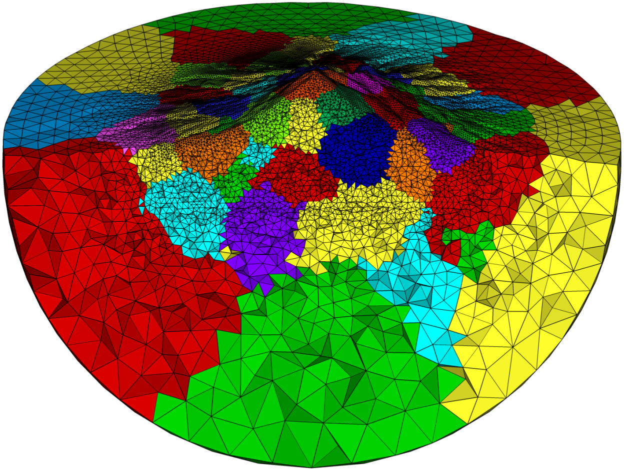


Physics and Parallelism

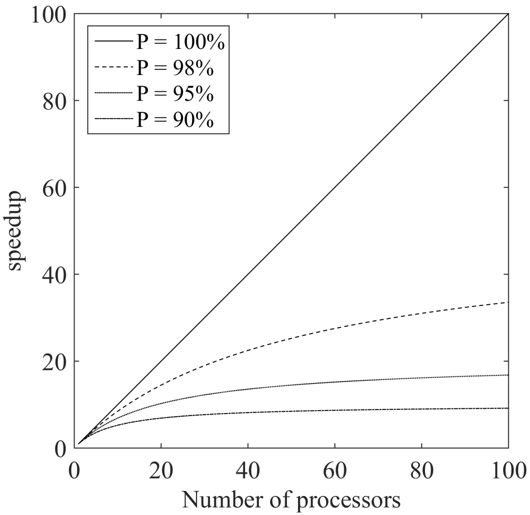
$$p(x, t + dt) = c^2(x)\frac{dt^2}{dx^2}[p(x + dx, t) - 2p(x, t) + p(x - dx, t)] + 2p(x, t) - p(x, t - dt),$$

The (immediate) future $t + dt$ of a physical system—here the pressure $p(x, t + dt)$ at some point in space x—depends (only) on the values in its immediate neighbourhood $p(x \pm dx)$, the presence $p(x, t)$, and the recent past $p(x, t - dt)$.

Parallelism and Load Balancing



To Scale or not to Scale

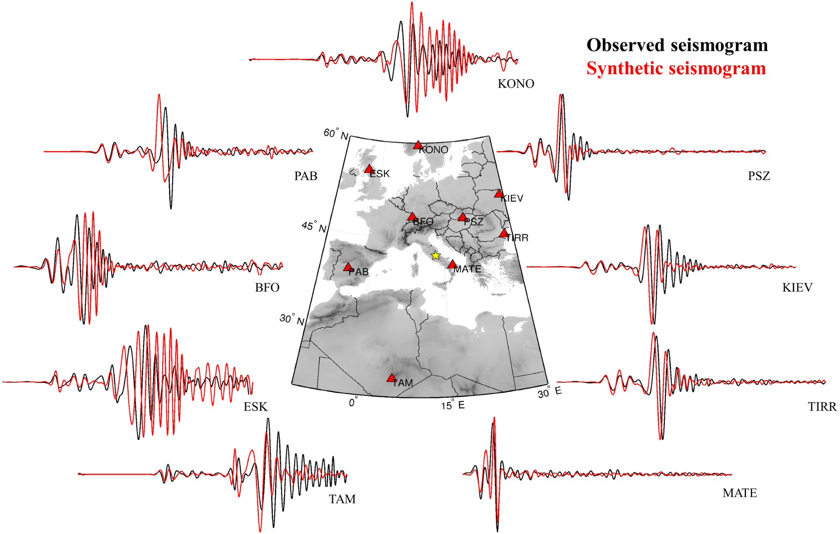


Class exercise:

Define a **frontend computer** (student). Each students writes four numbers on a page. Perform the following tasks:

- **Single-Instruction-Multiple-Data:** The tutor tells all students to multiply each number by 5.
- **Embarrassingly parallel problem:** Add the first three numbers and subtract the fourth.
- **Circular shift operation:** Pass the first number to your right neighbour. If there is none, pass it to the far left neighbour.
- **Global distribute:** The tutor gives 2 numbers to processor 0. Processor 0 distributes these 2 numbers to all processors.
- **Global reduce:** Find the maximum value of all initial 4 numbers of all processors. Processor 0 speaks out the maximum value loudly.

The Ultimate Goal: Matching Wavefield Observations



A Bit of Wave Physics

Acoustic wave equation: no source

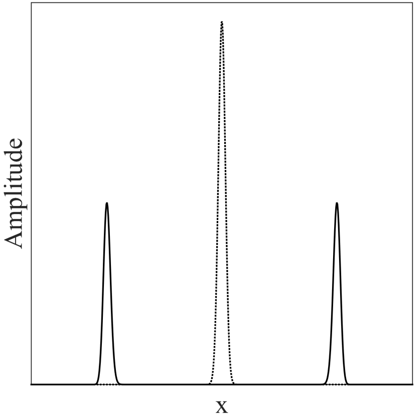
Acoustic wave equation

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

$p \rightarrow p(\mathbf{x}, t)$, pressure
$c \rightarrow c(\mathbf{x})$, velocity
$s \rightarrow s(\mathbf{x}, t)$, source term

Initial conditions

$$p(\mathbf{x}, t = 0) = p_0(\mathbf{x}, t)$$
$$\partial_t p(\mathbf{x}, t = 0) = 0$$



Snapshot of $p(\mathbf{x}, t)$ (solid line) after some time for initial condition $p_0(\mathbf{x}, t)$ (Gaussian, dashed line), 1D case.

Acoustic wave equation: external source

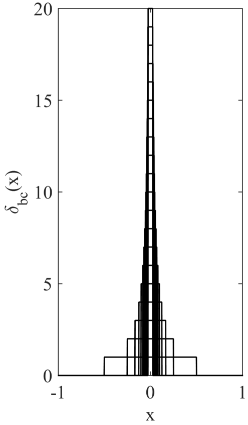
Green's Function G

$$\partial_t^2 G(\mathbf{x}, t; \mathbf{x}_0, t_0) - c^2 \Delta G(\mathbf{x}, t; \mathbf{x}_0, t_0) = \delta(\mathbf{x} - \mathbf{x}_0)\delta(t - t_0)$$

Delta function δ

$$\delta(x) = \begin{cases} \infty & x = 0 \\ 0 & x \neq 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(x)dx = 1 \, , \int_{-\infty}^{\infty} f(x)\delta(x)dx = f(0)$$



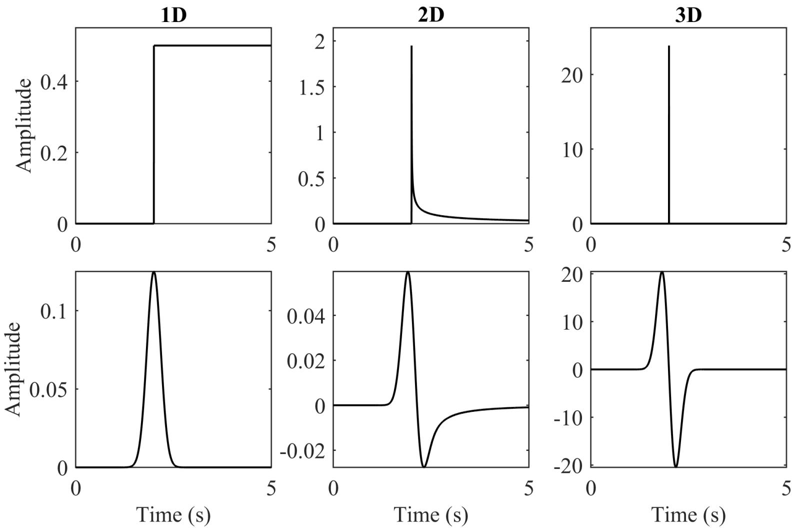
δ-generating function using boxcars.

Acoustic wave equation: analytical solutions

Green's functions for the inhomogeneous acoustic wave equation for all dimensions. $H(t)$ is the Heaviside function.

1D	2D	3D
$\frac{1}{2c}H(t-\frac{\lvert r\rvert}{c})$	$\frac{1}{2\pi c^2}\frac{H(t-\frac{\lvert r\rvert}{c})}{\sqrt{t^2-\frac{r^2}{c^2}}}$	$\frac{1}{4\pi c^2 r}\delta(t-r/c)$
$r=x$	$r=\sqrt{x^2+y^2}$	$r=\sqrt{x^2+y^2+z^2}$

Acoustic wave equation: analytical solutions



The Elastic Wave Equation

Displacement-stress formulation

$$
\begin{aligned}
\rho \partial_t^2 u_i &= \partial_j (\sigma_{ij} + M_{ij}) + f_i \\
\sigma_{ij} &= \lambda \epsilon_{kk} \delta_{ij} \; + \; 2 \; \mu \epsilon_{ij} \\
\epsilon_{kl} &= \frac{1}{2} (\partial_k u_l + \partial_l u_k),
\end{aligned}
$$

Dependencies

$$
\begin{array}{lll}
u_i & \rightarrow & u_i(\mathbf{x}, t) & i & = & 1,2,3 \\
v_i & \rightarrow & v_i(\mathbf{x}, t) & i & = & 1,2,3 \\
\sigma_{ij} & \rightarrow & \sigma_{ij}(\mathbf{x}, t) & i,j & = & 1,2,3 \\
\epsilon_{ij} & \rightarrow & \epsilon_{ij}(\mathbf{x}, t) & i,j & = & 1,2,3 \\
\rho & \rightarrow & \rho(\mathbf{x}) & & & \\
c_{ijkl} & \rightarrow & c_{ijkl}(\mathbf{x}) & i,j,k,l & = & 1,2,3 \\
f_i & \rightarrow & f_i(\mathbf{x}, t) & i & = & 1,2,3 \\
M_{ij} & \rightarrow & M_{ij}(\mathbf{x}, t) & i,j & = & 1,2,3,
\end{array}
$$

1-D elastic wave equation

Shear Motion

$$\rho(x)\partial_t^2 u(x,t) = \partial_x\left[\mu(x)\partial_x u(x,t)\right] \; + \; f(x,t)$$

u displacement

f external force

ρ mass density

μ shear modulus

Velocity - Stress Formulation

Defining velocity v and stress component σ as

$$\begin{aligned} \partial_t u &= v \\ \sigma &= \mu \partial_x u \end{aligned}$$

and assuming space-time dependencies leads to the wave equation

$$\begin{aligned} \rho \partial_t v &= \partial_x \sigma + f \\ \dot{\sigma} &= \mu \partial_x v \end{aligned}$$

Our unknown solution vector is

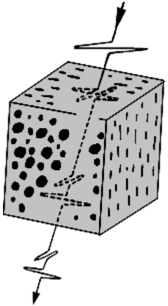
$$\mathbf{q}(x,t) = (v, \sigma)$$

Rheologies

In order of relevance

- Viscoelasticity
- Anisotropy
- Poroelasticity
- Plasticity

Anisotropy

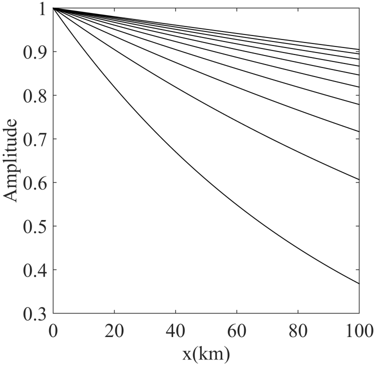


Amplitude decay

$$\frac{1}{Q(\omega)} = -\frac{\Delta E}{2\pi E}$$

$$A(x) = A_0 e^{-\frac{\omega x}{2cQ}}$$

Examples



Anisotropy

Generalized Hooke's Law

$$\sigma_{ij} = c_{ijkl}\,\epsilon_{kl}, \quad i,j,k,l = 1,2,3$$

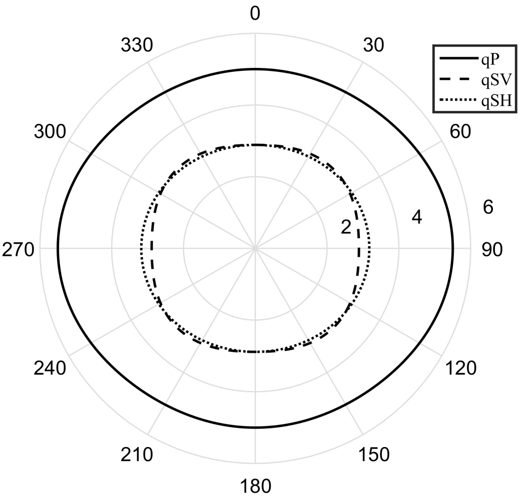
Reduced notation (Kelvin-Voight)

$$c_{pq} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{11} & c_{13} & 0 & 0 & 0 \\ c_{13} & c_{13} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{c_{11}-c_{12}}{2} \end{pmatrix}$$

Velocity variations (Thomson parameters)

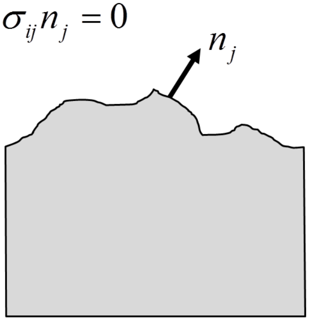
$$\begin{aligned} v_{qP}(\theta) &= v_{P0}\left(1 + \delta\sin^2(\theta)\cos^2(\theta) + \epsilon\sin^4(\theta)\right) \\ v_{qSV}(\theta) &= v_{S0}\left(1 + \frac{v_{P0}^2}{v_{S0}^2}(\epsilon - \delta)\sin^2(\theta)\cos^2(\theta)\right) \\ v_{qSH}(\theta) &= v_{S0}\left(1 + \gamma\sin^2(\theta)\right) \end{aligned} \tag{1}$$

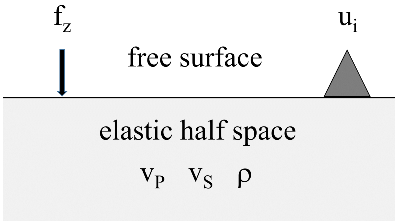
Anisotropic velocities



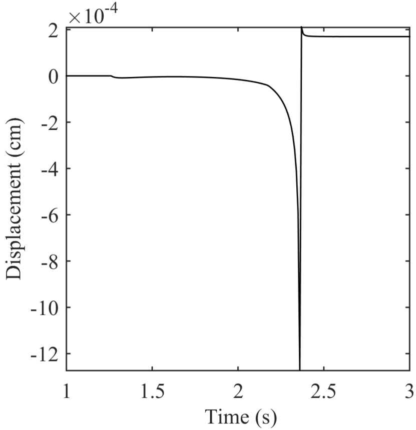
Free Surface Boundary Conditions

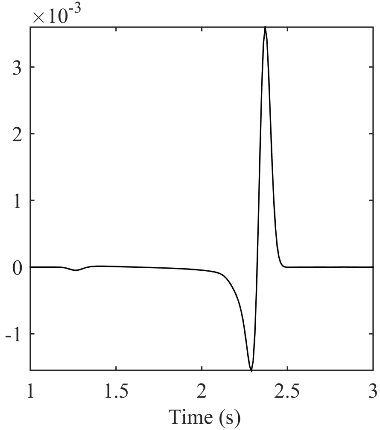
$$t_i = \sigma_{ij} n_j \quad \to \sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$$





Lamb’s Problem

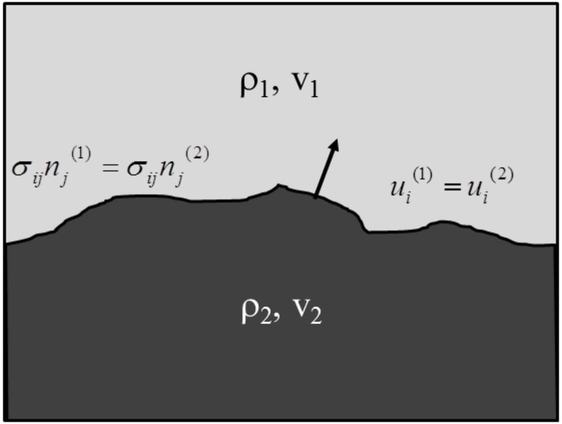




Internal Boundary Conditions

$$\sigma_{ij} n_j^{(1)} = \sigma_{ij} n_j^{(2)}$$
$$u_i^{(1)} = u_i^{(2)}$$

Internal boundary conditions need not be modelled explicitly!



Gradient, Divergence, Curl

Gradient

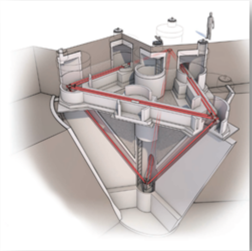
$$\nabla \mathbf{u}(\mathbf{x}, t) = \partial_j u_i(\mathbf{x}, t) \ .$$

Deformation

$$\epsilon_{ij}(\mathbf{x}, t) = \frac{1}{2}(\partial_i u_j(\mathbf{x}, t) + \partial_j u_i(\mathbf{x}, t))$$

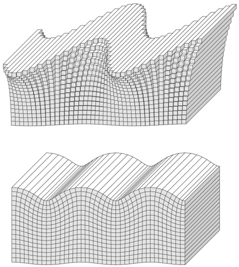
Curl

$$\frac{1}{2}\nabla \times \mathbf{u} = \frac{1}{2}\begin{pmatrix} \partial_y u_z - \partial_z u_y \\ \partial_z u_x - \partial_x u_z \\ \partial_x u_y - \partial_y u_x \end{pmatrix}$$

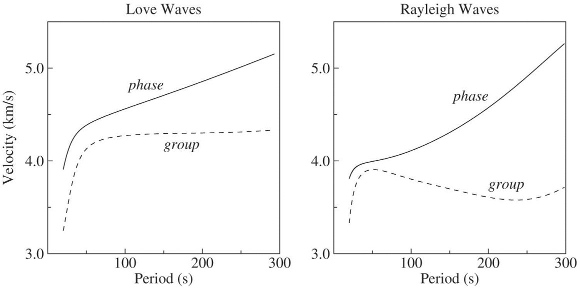


Surface Waves - Dispersion

Ralyeigh and Love Waves

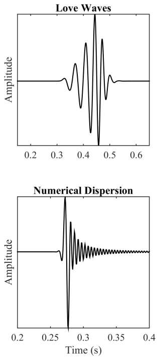


Dispersion Curves



Physical and Numerical Dispersion

Numerical Dispersion



Numerical and physical dispersion can be confused!

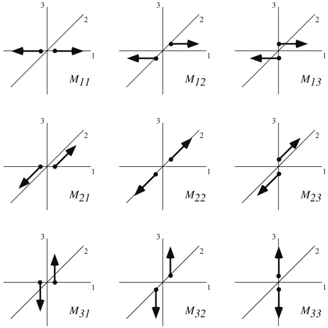
In wave simulations we have to **avoid** numerical dispersion!

The Moment Tensor

Moment Tensor

$$\mathbf{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}$$

$$M_0 = \mu \, A \, d$$

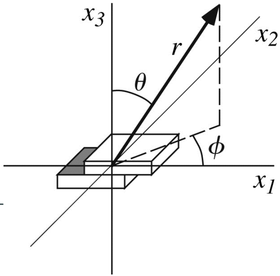


Double Couple Green's function

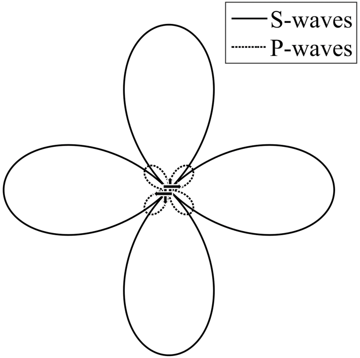
$$\boldsymbol{u}(\boldsymbol{x},t) = \frac{1}{4\pi\rho}\boldsymbol{A}^{N}\frac{1}{r^4}\int\limits_{r/\alpha}^{r/\beta}\tau M_0(t-\tau)d\tau +$$

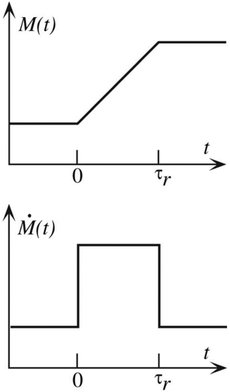
$$+\frac{1}{4\pi\rho\alpha^2}\boldsymbol{A}^{IP}\frac{1}{r^2}M_0(t-\frac{r}{\alpha}) + \frac{1}{4\pi\rho\beta^2}\boldsymbol{A}^{IS}\frac{1}{r^2}M_0(t-\frac{r}{\beta})+$$

$$+\frac{1}{4\pi\rho\alpha^3}\boldsymbol{A}^{FP}\frac{1}{r}\dot{M}_0(t-\frac{r}{\alpha}) + \frac{1}{4\pi\rho\beta^3}\boldsymbol{A}^{FS}\frac{1}{r}\dot{M}_0(t-\frac{r}{\beta})$$

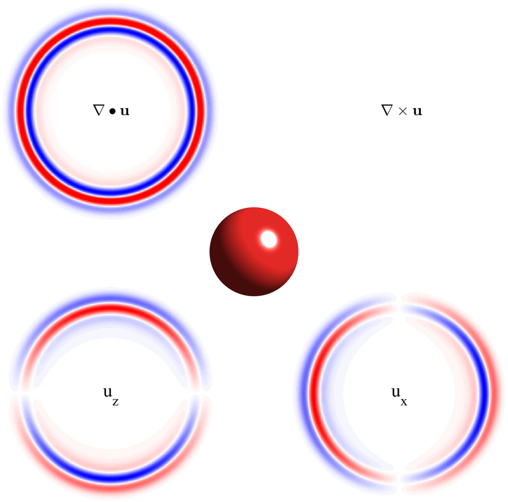


Radiation and Source Time Function

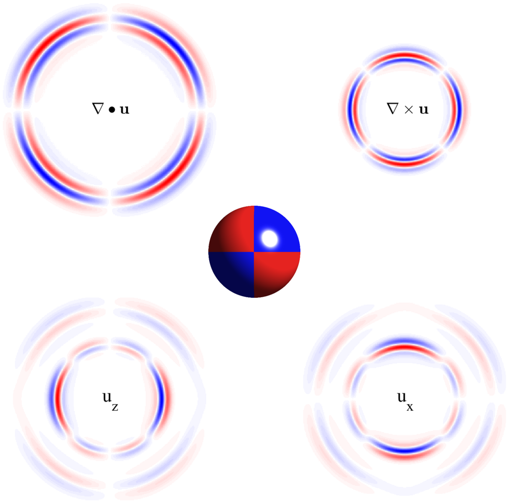




Wavefields from Moment Tensor Sources



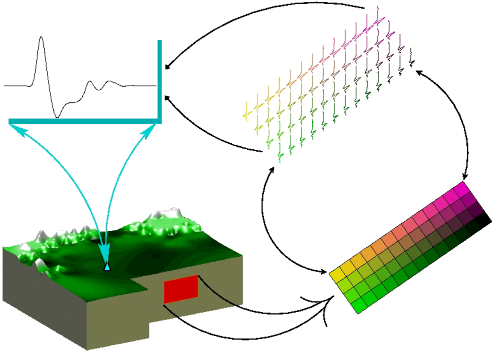
Wavefields from Moment Tensor Sources



Superposition Principle

$$v_l^r(\omega) = \sum_{k=1}^{N} \mathrm{slip}_k \exp[-i\omega t_k(c^{rup})] G_{kl}^r(\omega) S(R,\omega)$$

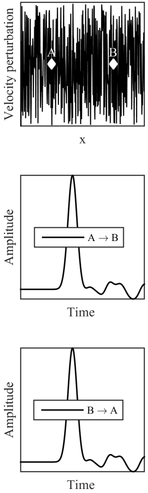
Finite sources can be simulated by summing up over point sources



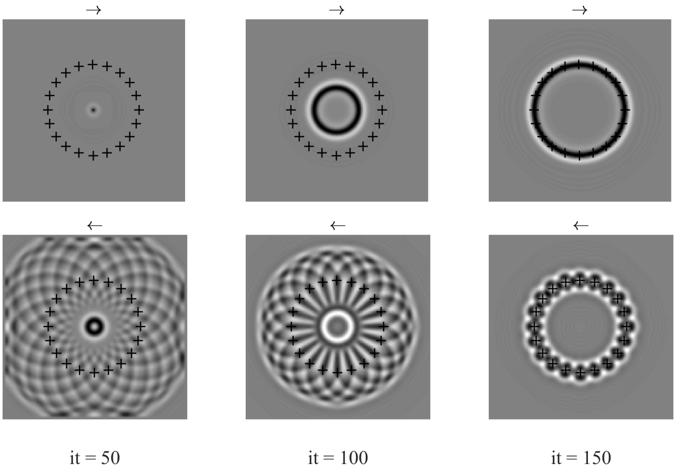
Reciprocity

$$G_{ij}(\boldsymbol{x}, t; \boldsymbol{x}_0, t_0) = G_{ji}(\boldsymbol{x}_0, -t_0; \boldsymbol{x}, -t)$$

The wave equation is symmetric in time. Source and receiver locations can be interchanged. This has dramatic consequences for modeling and inversion!



Time Reversal



Wave Equation as Linear System

Seismogram for arbitrary source s(t) as convolution (exact)

$$p(\mathbf{x}, t) = G(\mathbf{x}, t, \mathbf{x}_0) \otimes s(t)$$

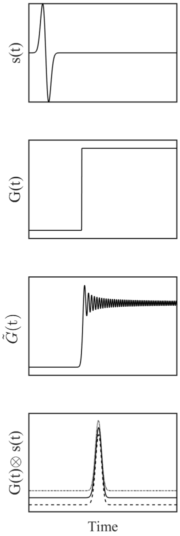
Seismogram for arbitrary source s(t) as convolution (numerical)

$$\tilde{p}(\mathbf{x}, t) = \tilde{G}(\mathbf{x}, t, \mathbf{x}_0) \otimes s(t)$$

Important consequence:

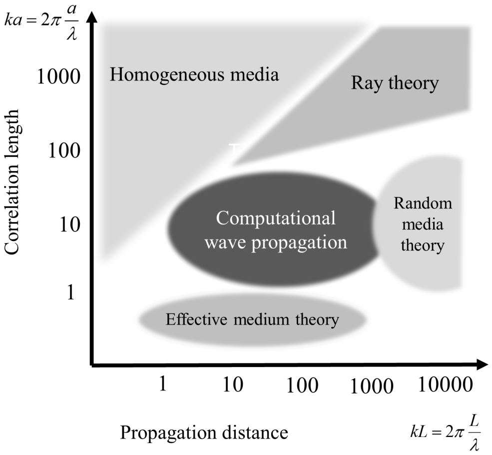
Even if your numerical Green's function $\tilde{G}(\mathbf{x}, t, \mathbf{x}_0)$ is inaccurate, the numerical solution $\tilde{p}(\mathbf{x}, t)$ might be very accurate provided the $s(t)$ is defined in the right frequency band!

Wave Equation as Linear System



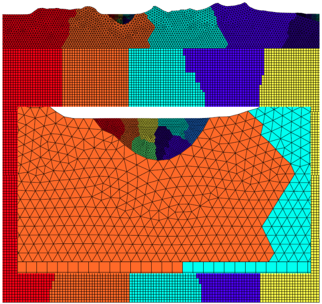
- Accurate Green's functions cannot be calculated numerically
- A numerical solver is a **linear system**
- The convolution theorem applies
- Inaccurate simulations can be filtered afterwards
- Source time functions can be altered afterwards
- ... provided the sampling is good enough ...

Spatial Scales, Scattering, Solution Strategies



- Recorded seismograms are affected by ...
- ... the ratio of seismic wavelength λ and structural wavelength *a* ...
- ... how many wavelengths are propagated ...
- strong scattering when $a \approx \lambda \; \rightarrow$ numerical methods
- ray theory works when $a >> \lambda$
- random medium theory necessary for strong scattering media (and long distances)

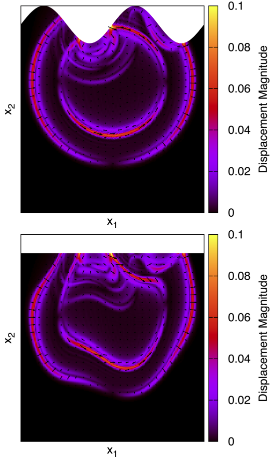
Challenges - Meshing



Human time	Simulation workflow	cpu time
15%	Design	0%
80% (weeks)	Geometry creation, meshing	10%
5%	Solver	90%

- Meshing work flow not well defined
- Still major bottleneck for simulation tasks with complex geometries
- Tetrahedral meshes easier, but ...
- seissol, specfem, salvus?

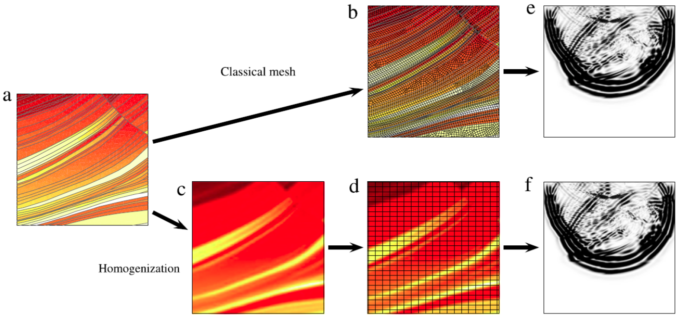
Future Strategies - Alternative Formulations



Al-Attar and Crawford GJI 2016

- Particle relabelling, grid stretching
- Mapping geometrical complexity onto regular grids
- Smart pre-processing rather than meshing?
- Similar concept used in summation-by-parts (SBP) algorithms (SW4)

Future Strategies - Homogenization



- We only *see* low-pass filtered Earth
- So why simulate models with infinite frequencies?
- Homogenisation of discontinuous model
- Renaissance of regular grid methods?

Computational Geophysics Intro - Summary

- We need to understand the **discretization** of our systems
- Do we have many **scales** involved (requires complex meshing)?
- What is the (minimum) **physics** that we need?
- What **dimensionality** is required to answer our questions (1D, 2D, 2.5D, 3D)?
- Is **space-time domain** the right approach or is it better to work in the **Fourier domain**?

Questions?

Computational Geophysics Intro - Comprehension Questions

1. Explain the concept of **2D, 2.5D, and 3D simulations**. What problems can arise for $<$3D simulations when comparing with observations?
2. Explain the concepts of **structured** and **unstructured meshes**. Give examples. Discuss pros and cons.
3. Illustrate the differences between regular meshes in various **coordinate systems: cartesian, cylindrical, spherical**. What are the consequences for simulation problems?
4. What are the basic models for **parallel computers**?
5. What is the most common **model of parallelisation** for seismic wave propagation and why?
6. Find some **current supercomputers** on the internet and extract the main specifications (e.g., number of processors, memory, peak performance, etc.).