

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

EnvSeis Short Course, May 27, 2024

Heiner Igel

May 23, 2024

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

Course Content

- Part I
 - Introduction - **Motivation**
 - **Fundamentals** of wave propagation (**wave equations, analytical solutions, reciprocity, superposition principle, dispersion, homogenization**)
- Part II
 - The **finite-difference** method
 - The **pseudospectral** method
 - Linear **finite-element** method
- Part III
 - The **spectral-element** method
 - The **finite-volume** method
 - The **discontinuous Galerkin** method
 - Outlook

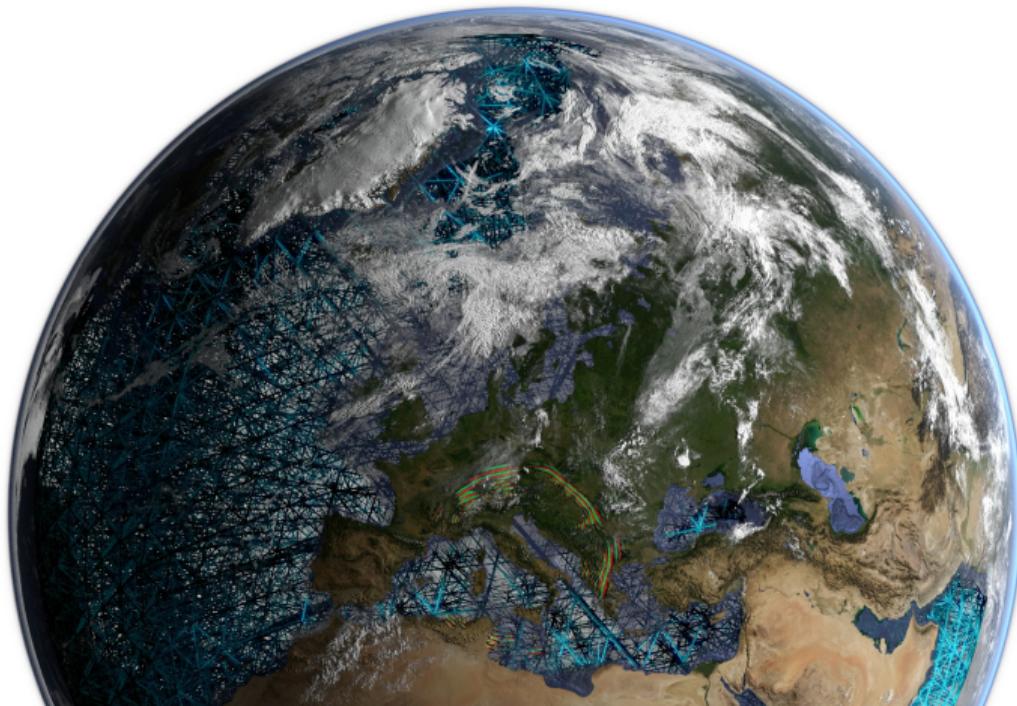
What you get ...

- Basic ideas of **discretization**, **size** of a problem, designing a solution
- The fundamentals of **wave propagation** (what should I see in a wavefield?)
- What is in the **engine** of simulation algorithms?
- What are the major parameters, where are **dangers** and **traps**?
- What is a good strategy **to know my simulation is right**?
- What is the difference between the various approaches, **pros and cons**?
- What are the **simplest programs** to see how things work?

What you do not get ...

- How specific **community solvers** work (e.g., specfem, seissol, salvus, etc.)
- **Classic approaches** like ray tracing or reflectivity methods
- **Codes** for 3D solutions
- Beamforming

Introduction



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

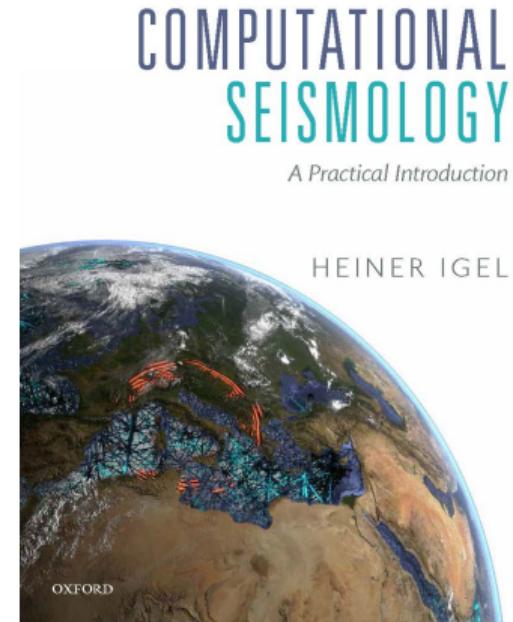
Who needs Computational Seismology (cont'd)

(...)

- Ocean generated **noise measurements** and cross-correlation techniques
- **Environmental seismology**
- **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!)

The screenshot shows a Coursera course page. At the top, there's a breadcrumb navigation: Blättern > Physik und Ingenieurwesen > Forschungsmethoden. The main title is "Computers, Waves, Simulations: A Practical Introduction to Numerical Methods using Python". Below the title, it says "4.7 289 Bewertungen". The instructor is listed as "Heiner Igel". There are two buttons: "Kostenlos anmelden" (Free enrollment) and "Finanzielle Unterstützung verfügbar" (Financial support available). It also mentions "Beginnt am 26. März". At the bottom, it says "16.370 bereits angemeldet". On the right side of the page, the logo of the University of Munich (LMU) is displayed.

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

Why numerical methods?



Photo ©Überraschungsbilder via Wikimedia commons

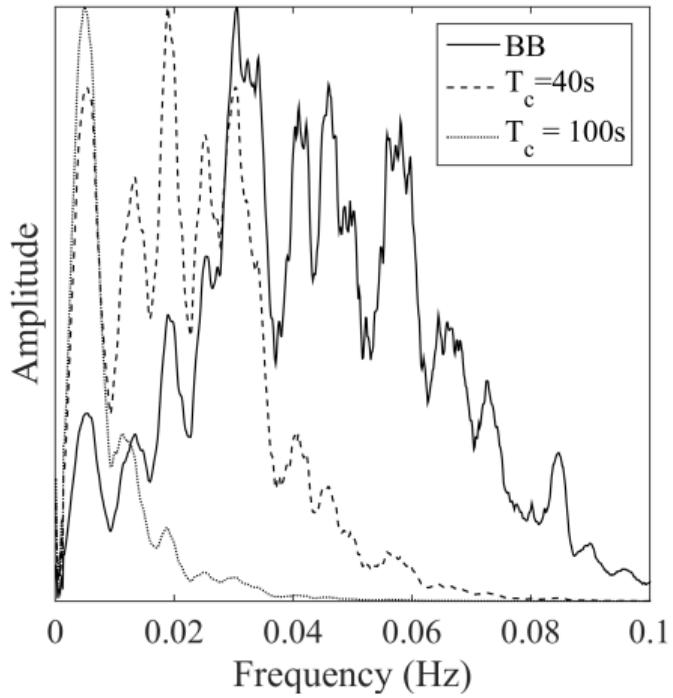
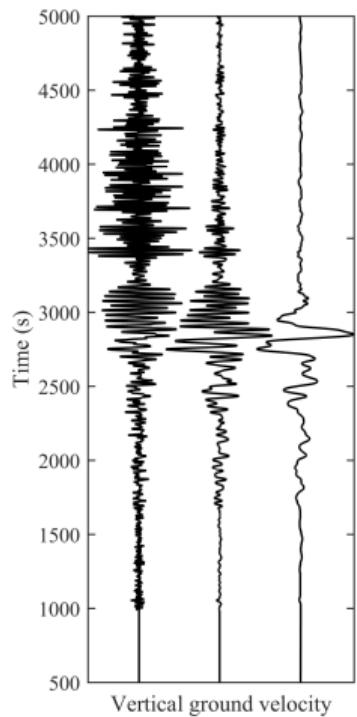
Computational Seismology, Memory, and Compute Power

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?

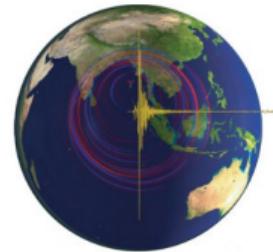


Seismic Wavefield Observations



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 $v_p = 10\text{ km/s}$ 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 highest 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 (@ $T = 1\text{ s}$) : 360
TBytes

Results (@ $T = 10\text{ s}$) : 360
GBytes

Results (@ $T = 100\text{ s}$) : 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 highest 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 **with your neighbor** to think of your current or future simulation problem (whatever physical process). What is the **physical scale** of the problem? **2D or 3D?** What are the **wavelengths** involved?

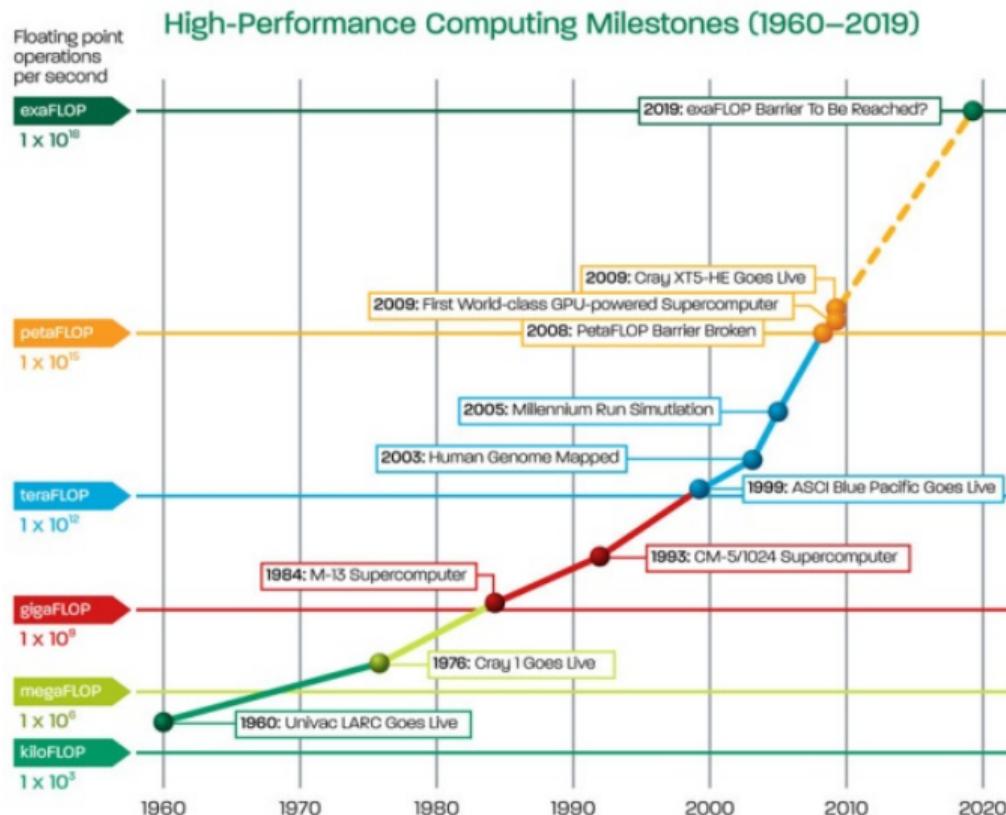
How much **memory** per field (elastic parameters, displacement)? **Laptop or supercomputer?**

*Back of the envelope or Jupyter Notebook on github: 01
Introduction → 02
Laptop_or_Supercomputing*

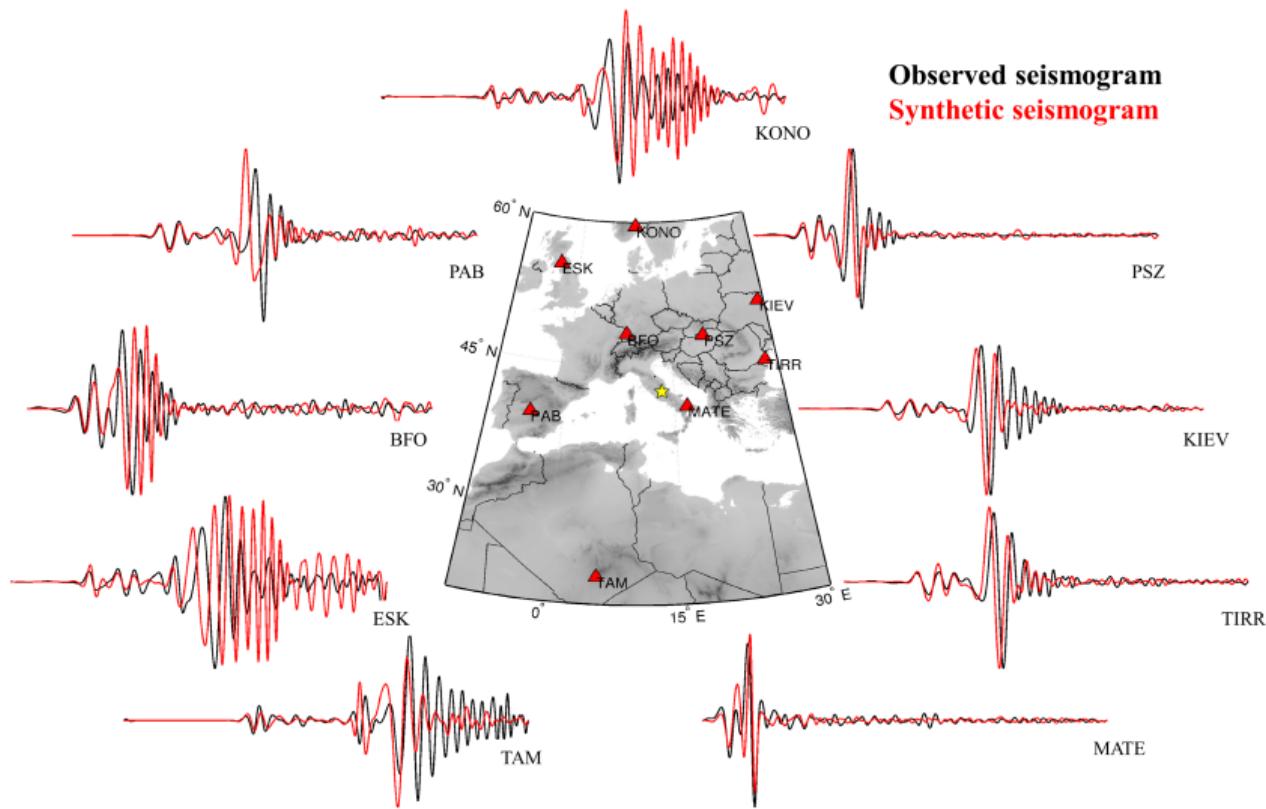
Computational Seismology, Memory, and Compute Power



- 1960: 1 MFlops
- 1970: 10 MFlops
- 1980: 100 MFlops
- 1990: 1 GFlops
- 1998: 1 TFlops
- 2008: 1 Pflops
- 2021: 1 EFlops



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

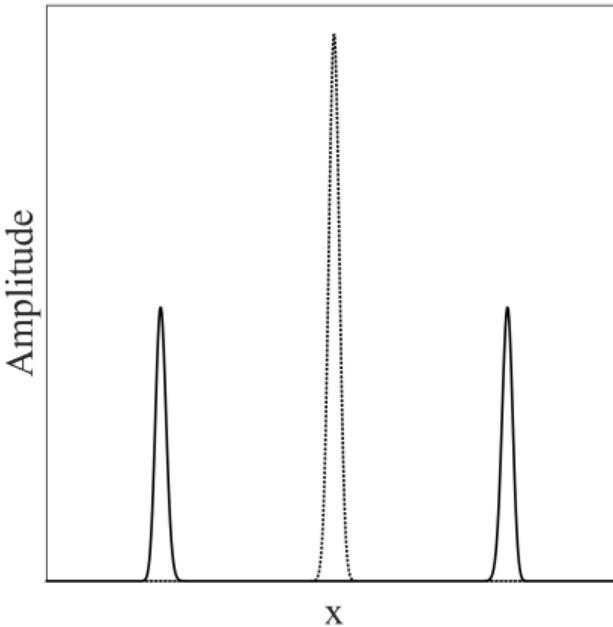
$c \rightarrow c(\mathbf{x})$, velocity

$s \rightarrow s(\mathbf{x}, t)$, 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

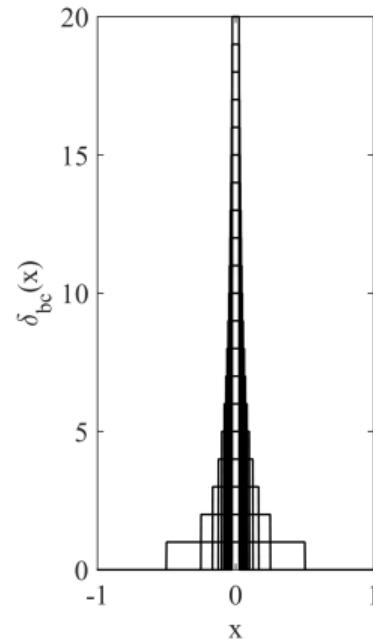
Green's Function G

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

Delta function δ

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

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



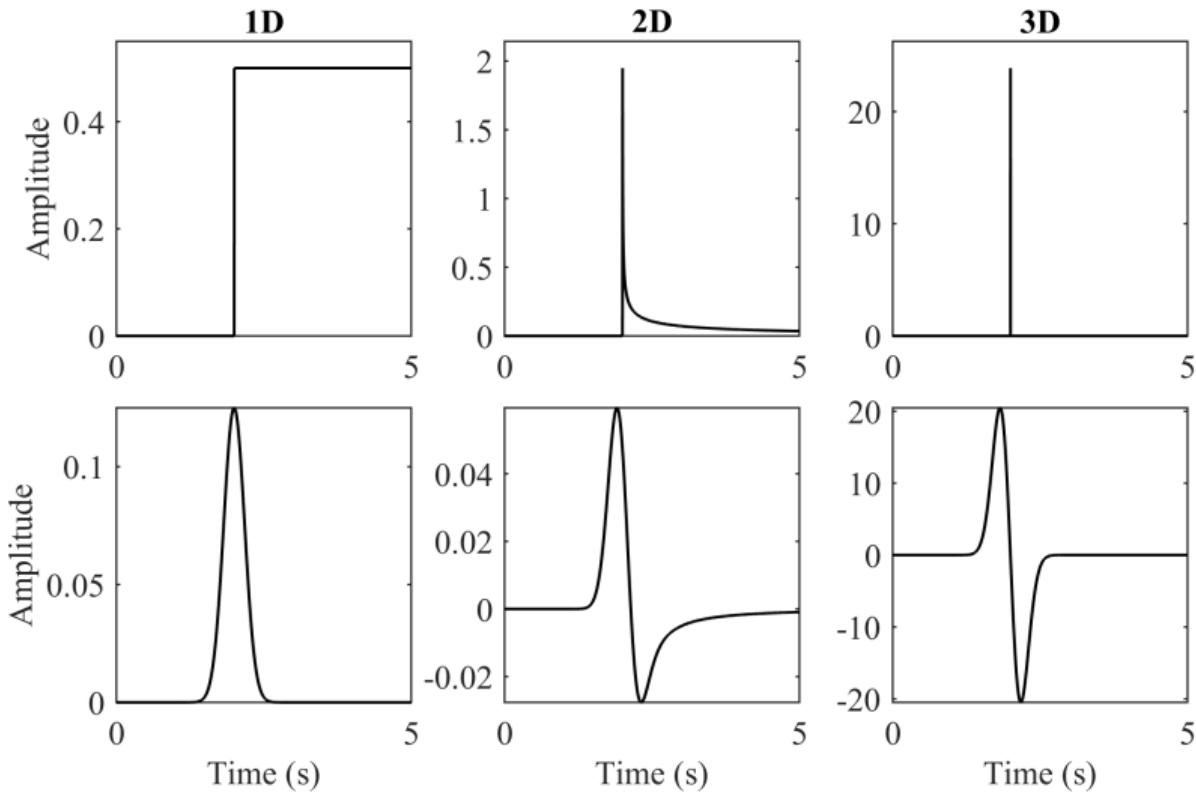
δ -generating function using
boxcars.

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

Acoustic wave equation: analytical solutions



The Elastic Wave Equation

Displacement-stress formulation

$$\rho \partial_t^2 u_i = \partial_j (\sigma_{ij} + M_{ij}) + f_i$$

$$\sigma_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2 \mu \epsilon_{ij}$$

$$\epsilon_{kl} = \frac{1}{2} (\partial_k u_l + \partial_l u_k),$$

Dependencies

u_i	$\rightarrow u_i(\mathbf{x}, t)$	$i = 1, 2, 3$
v_i	$\rightarrow v_i(\mathbf{x}, t)$	$i = 1, 2, 3$
σ_{ij}	$\rightarrow \sigma_{ij}(\mathbf{x}, t)$	$i, j = 1, 2, 3$
ϵ_{ij}	$\rightarrow \epsilon_{ij}(\mathbf{x}, t)$	$i, j = 1, 2, 3$
ρ	$\rightarrow \rho(\mathbf{x})$	
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}	$\rightarrow M_{ij}(\mathbf{x}, t)$	$i, j = 1, 2, 3$

1-D elastic wave equation

Shear Motion

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

u displacement

f external force

ρ mass density

μ shear modulus

Velocity - Stress Formulation

Defining velocity v and stress component σ as

$$\partial_t u = v$$

$$\sigma = \mu \partial_x u$$

and assuming space-time dependencies leads to the wave equation

$$\rho \partial_t v = \partial_x \sigma + f$$

$$\dot{\sigma} = \mu \partial_x v$$

Our unknown solution vector is

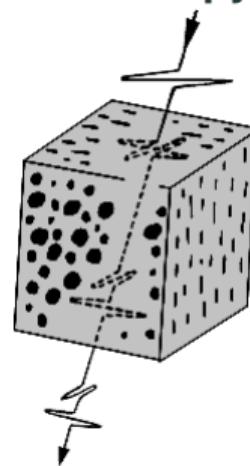
$$\mathbf{q}(x, t) = (v, \sigma)$$

Rheologies

In order of relevance

- Viscoelasticity
- Anisotropy
- Poroelasticity
- Plasticity

Anisotropy



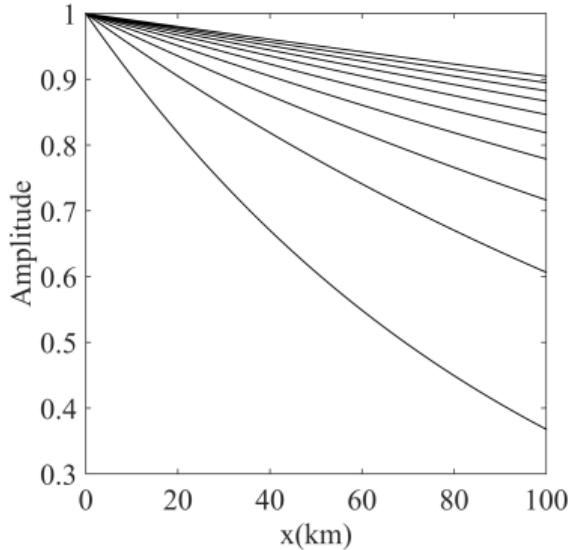
Attenuation

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

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

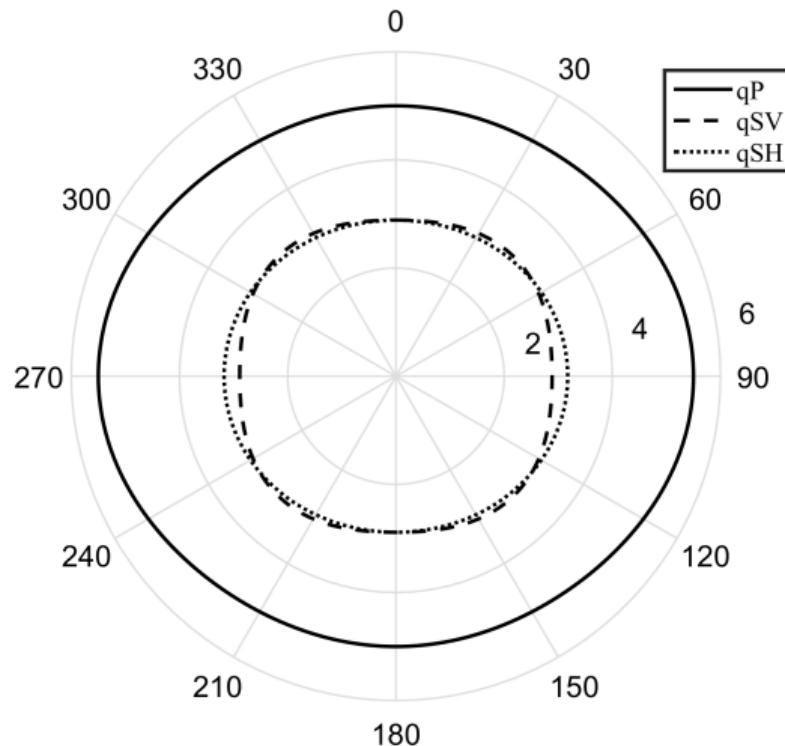
Reduced notation (Kelvin-Voight)

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

Velocity variations (Thomson parameters)

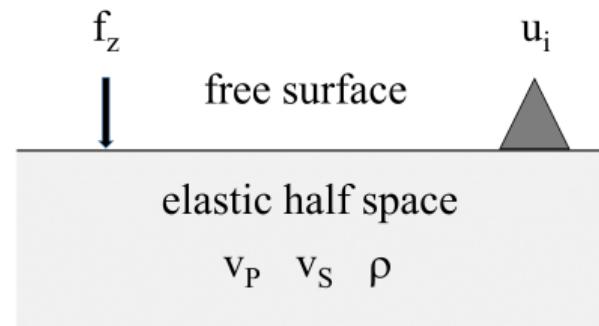
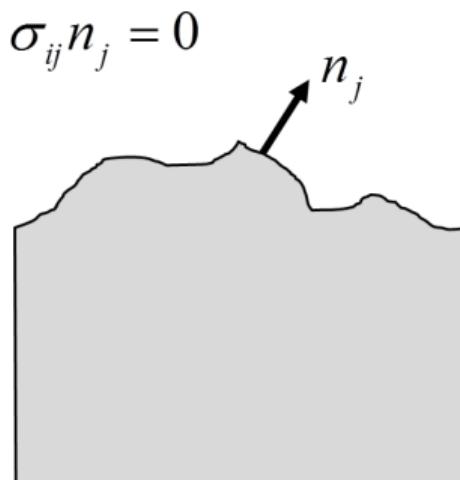
$$\begin{aligned} v_{qP}(\theta) &= v_{P0} \left(1 + \delta \sin^2(\theta) \cos^2(\theta) + \epsilon \sin^4(\theta) \right) \\ v_{qSV}(\theta) &= v_{S0} \left(1 + \frac{v_{P0}^2}{v_{S0}^2} (\epsilon - \delta) \sin^2(\theta) \cos^2(\theta) \right) \\ v_{qSH}(\theta) &= v_{S0} \left(1 + \gamma \sin^2(\theta)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

Anisotropic velocities

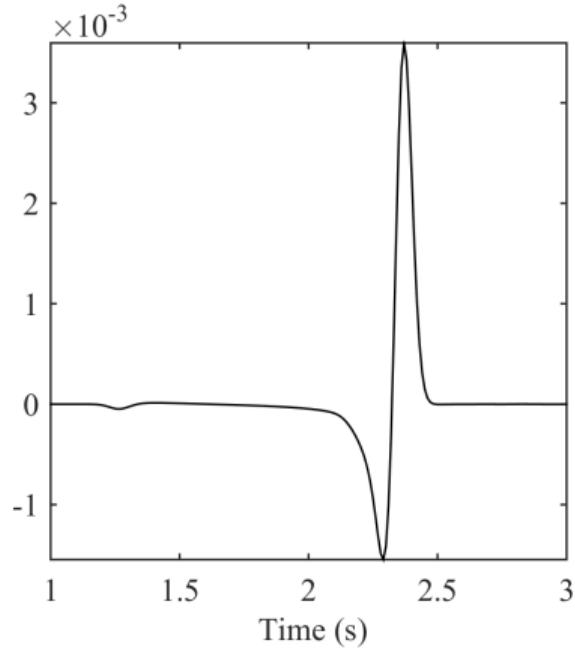
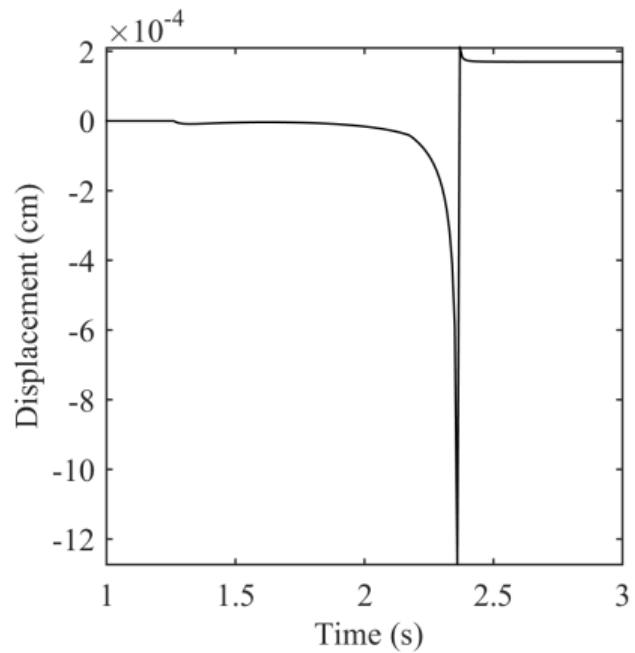


Free Surface Boundary Conditions

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



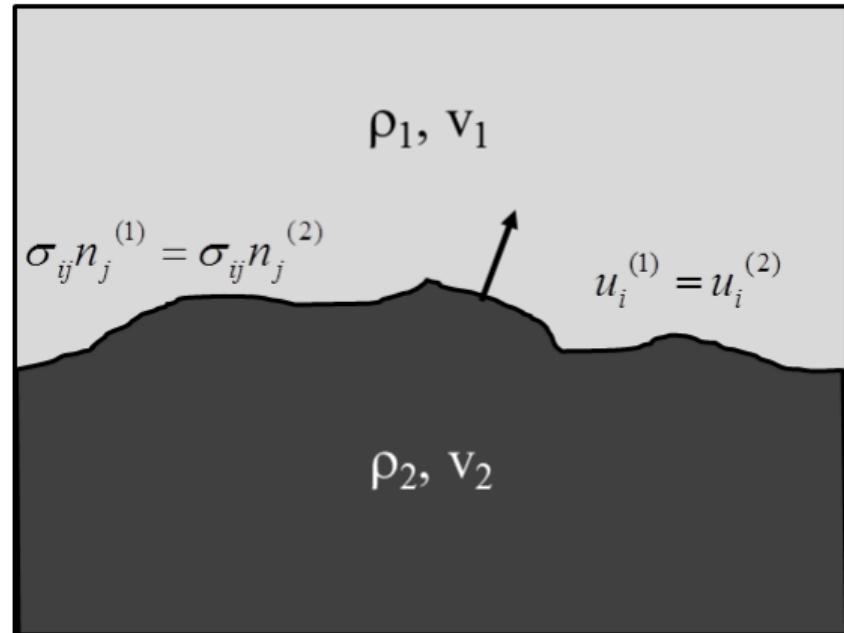
Lamb's Problem



Internal Boundary Conditions

$$\begin{aligned}\sigma_{ij} n_j^{(1)} &= \sigma_{ij} n_j^{(2)} \\ u_i^{(1)} &= u_i^{(2)}\end{aligned}$$

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

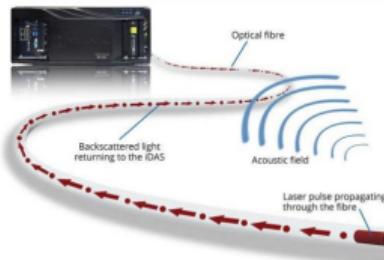
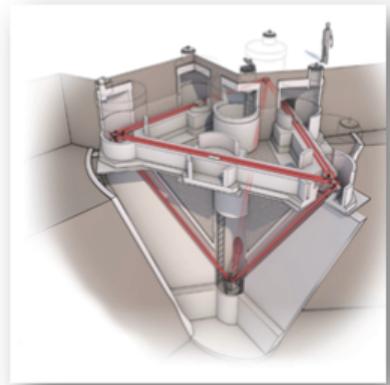
$$\epsilon_{ij}(\mathbf{x}, t) = \frac{1}{2} (\partial_i u_j(\mathbf{x}, t) + \partial_j u_i(\mathbf{x}, t))$$

Curl

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

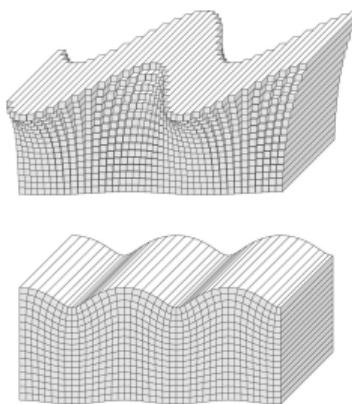
Divergence

$$\nabla \cdot \mathbf{u} = \partial_x u_x + \partial_y u_y + \partial_z u_z$$

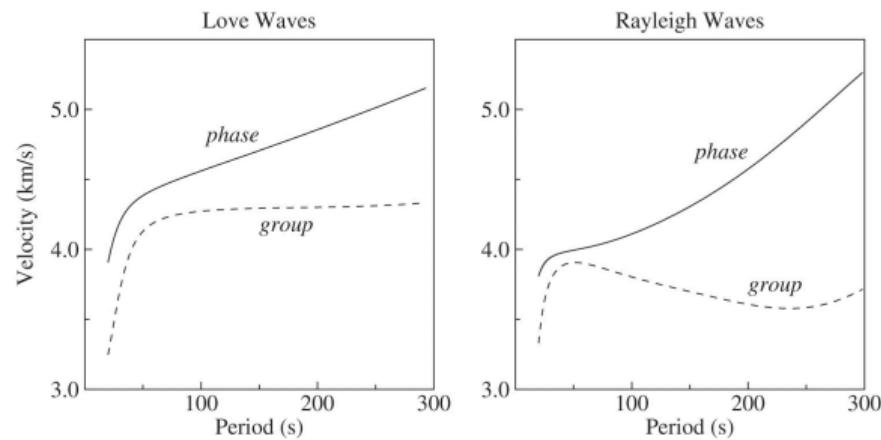


Surface Waves - Dispersion

Ralyeigh and Love Waves

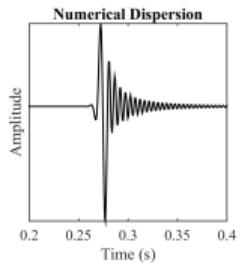
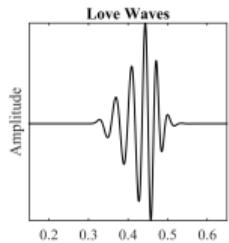


Dispersion Curves



Physical and Numerical Dispersion

Numerical Dispersion



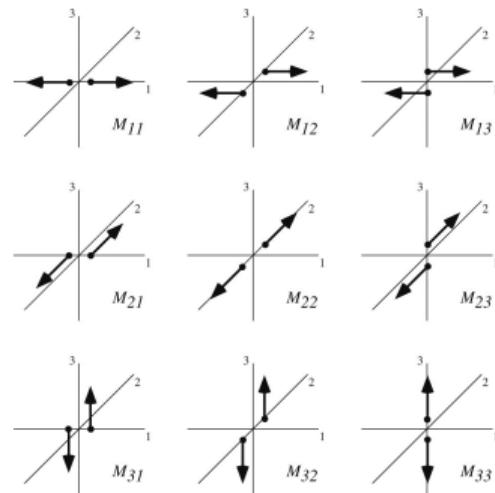
Numerical and physical dispersion can be confused!

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

The Moment Tensor

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

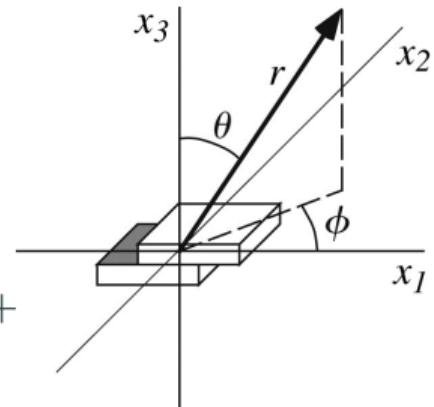
$$M_0 = \mu A d$$



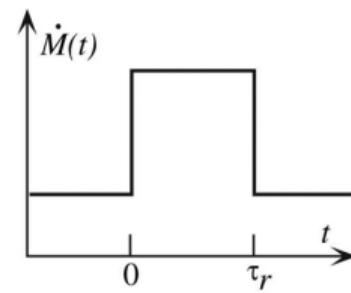
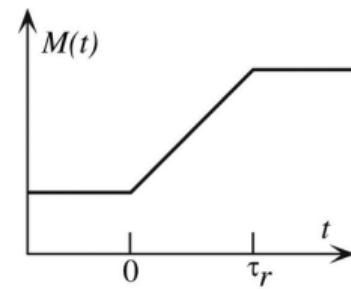
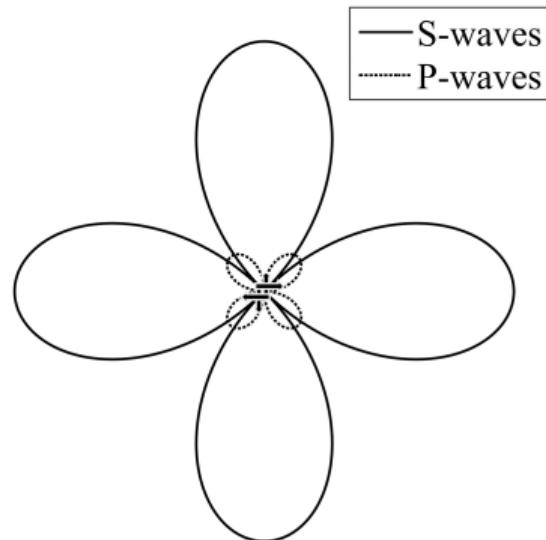
The DC analytical solution

Double Couple Green's function

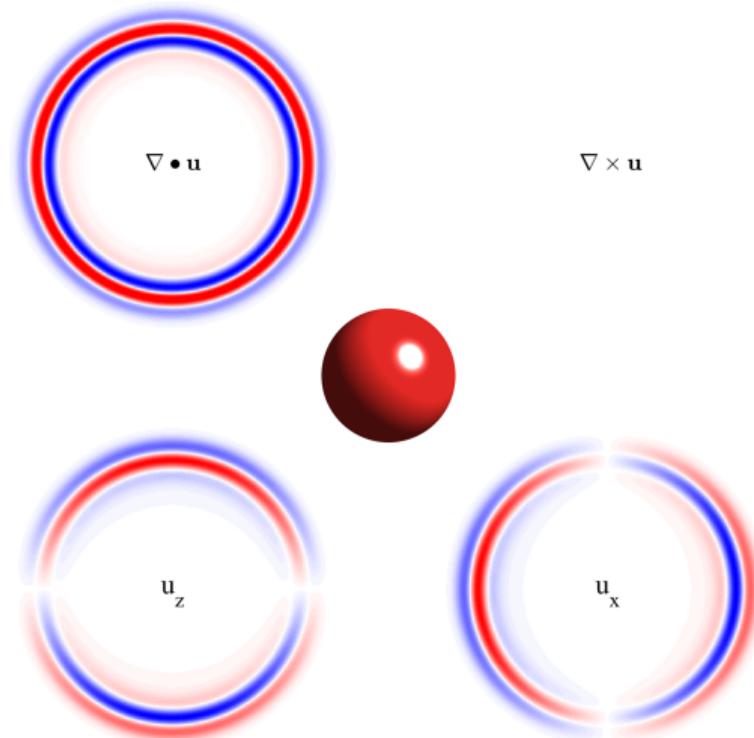
$$\begin{aligned}\mathbf{u}(\mathbf{x}, t) = & \frac{1}{4\pi\rho} \mathbf{A}^N \frac{1}{r^4} \int_{r/\alpha}^{r/\beta} \tau M_0(t - \tau) d\tau + \\ & + \frac{1}{4\pi\rho\alpha^2} \mathbf{A}^{IP} \frac{1}{r^2} M_0(t - \frac{r}{\alpha}) + \frac{1}{4\pi\rho\beta^2} \mathbf{A}^{IS} \frac{1}{r^2} M_0(t -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



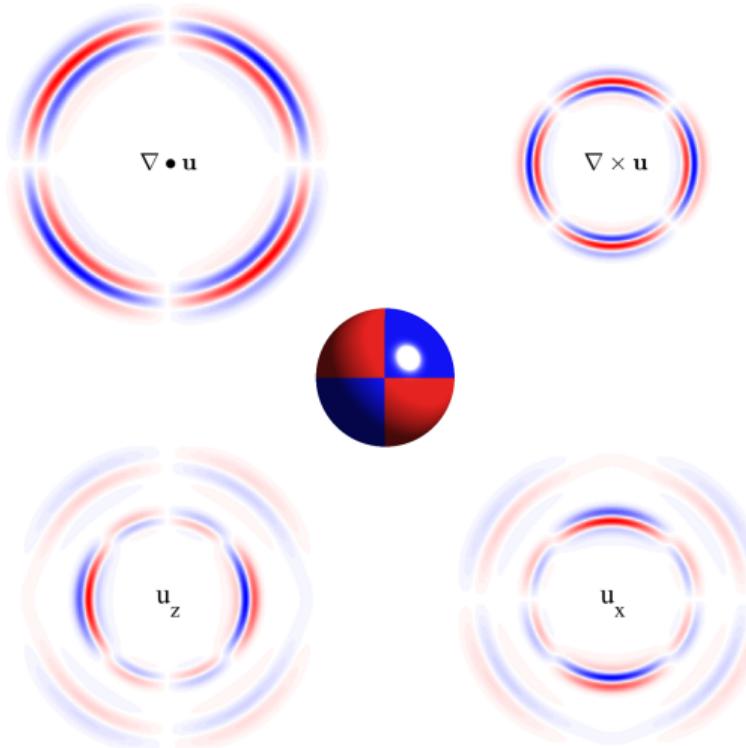
Radiation and Source Time Function



Wavefields from Moment Tensor Sources



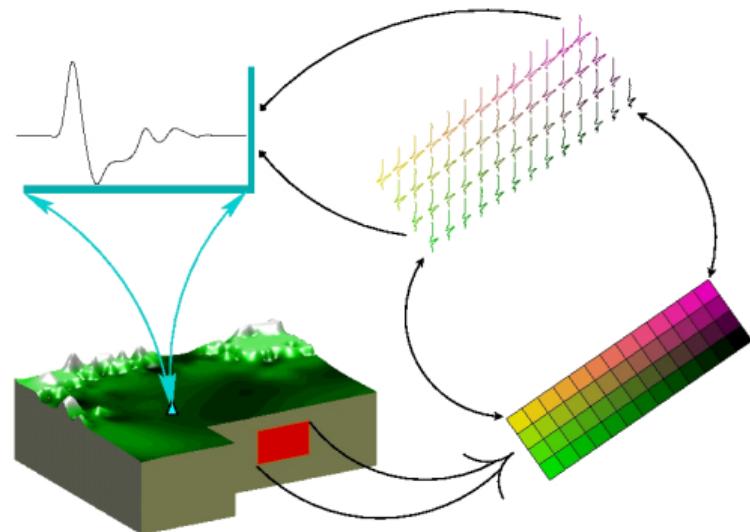
Wavefields from Moment Tensor Sources



Superposition Principle

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

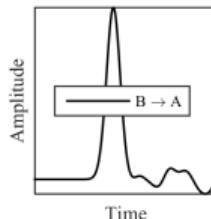
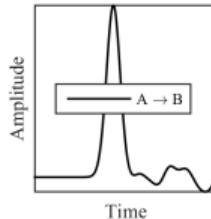
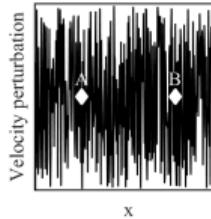
Finite sources can be simulated by summing up over point sources



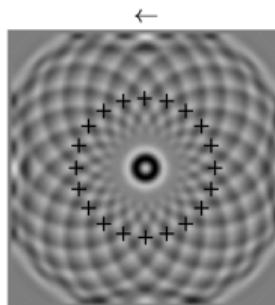
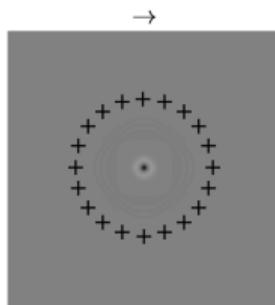
Reciprocity

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

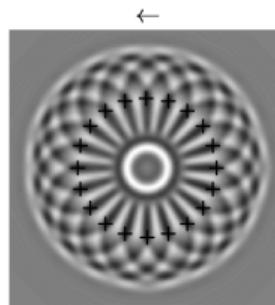
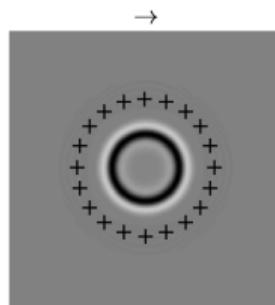
The wave equation is symmetric in time. Source and receiver locations can be interchanged. This has dramatic consequences for modeling and inversion!



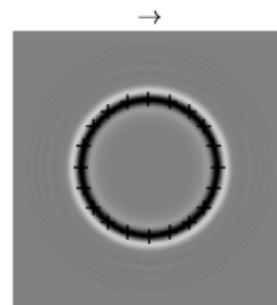
Time Reversal



it = 50

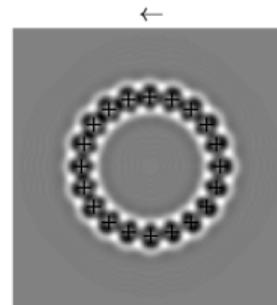
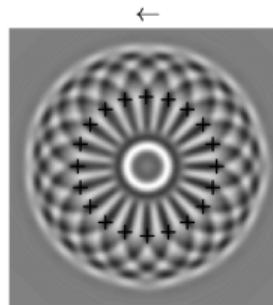
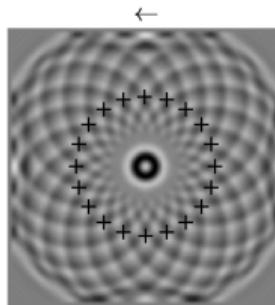
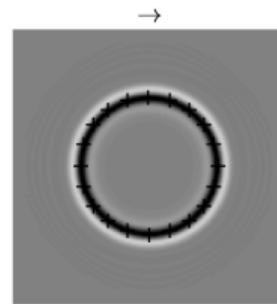
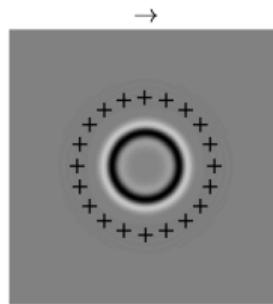
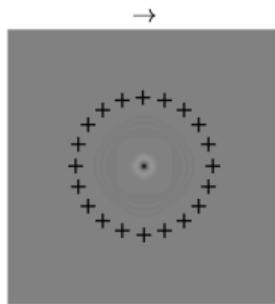


it = 100



it = 150

Time Reversal - Exercise



it = 50

it = 100

it = 150

On github run the Jupyter Notebook 03_time_reversal_reciprocity and change the number of receivers.

Wave Equation as Linear System

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

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

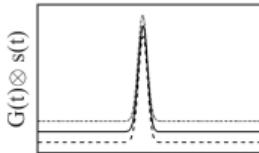
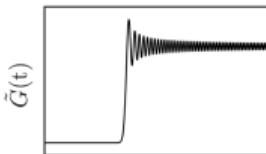
Seismogram for arbitrary source $s(t)$ as convolution (numerical)

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

Important consequence:

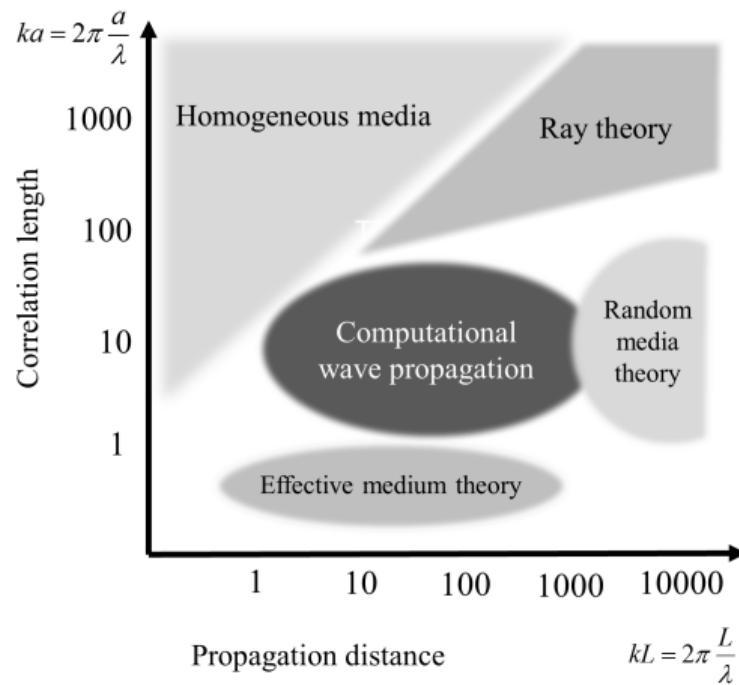
Even if your numerical Green's function $\tilde{G}(\mathbf{x}, t, \mathbf{x}_0)$ is inaccurate, the numerical solution $\tilde{p}(\mathbf{x}, t)$ might be very accurate provided the $s(t)$ is defined in the right frequency band!

Wave Equation as Linear System



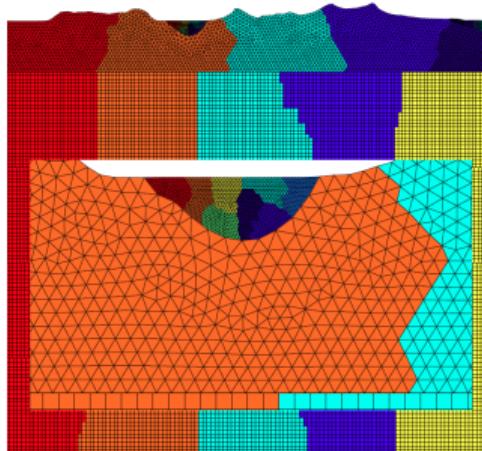
- Accurate Green's functions cannot be calculated numerically
- A numerical solver is a **linear system**
- The convolution theorem applies
- Inaccurate simulations can be filtered afterwards
- Source time functions can be altered afterwards
- ... provided the sampling is good enough ...

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 $a \gg \lambda$
- random medium theory necessary for strong scattering media (and long distances)

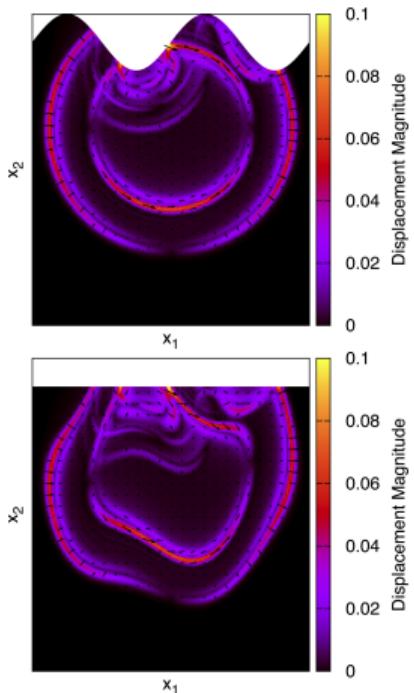
Challenges - Meshing



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

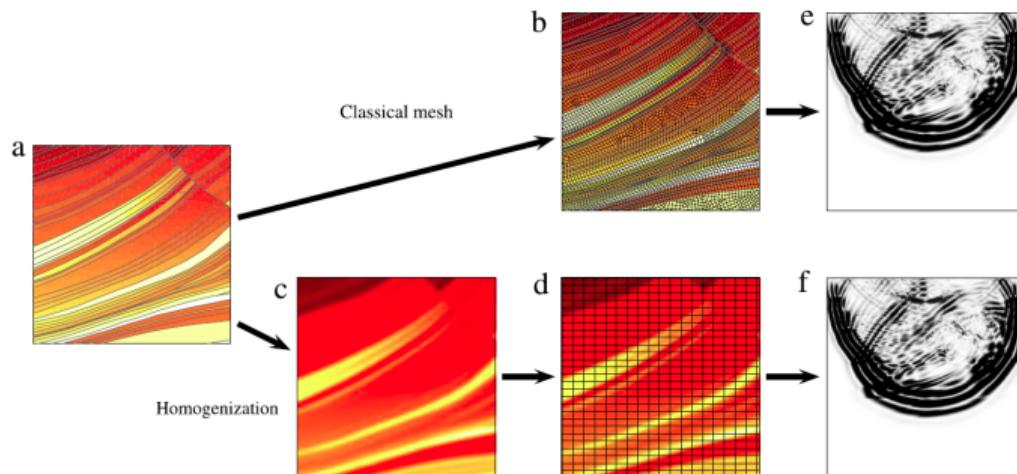
- Meshing work flow not well defined
- Still major bottleneck for simulation tasks with complex geometries
- Tetrahedral meshes easier, but ...
- Salvus?

Future Strategies - Alternative Formulations



- 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 Seismology - Part I

Questions?