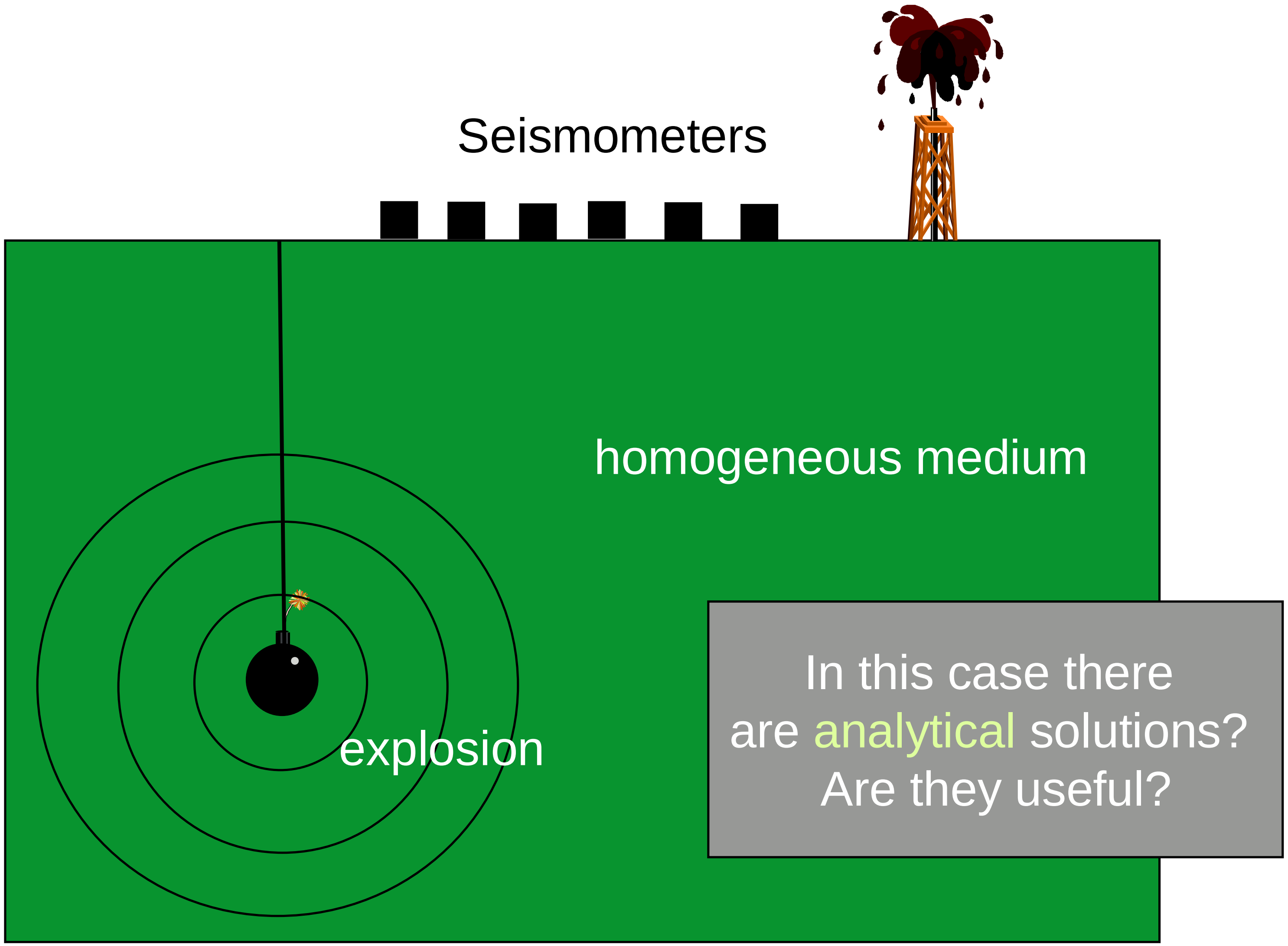
Which one of the finite {differences, elements, volumes} should I use?

Heiner Igel
Department of Earth and Environmental Sciences
LMU Munich

- General Introduction: Why numerical methods?
- Specific methods:
  - Finite differences
  - Finite elements
  - Finite volumes
- Current challenges

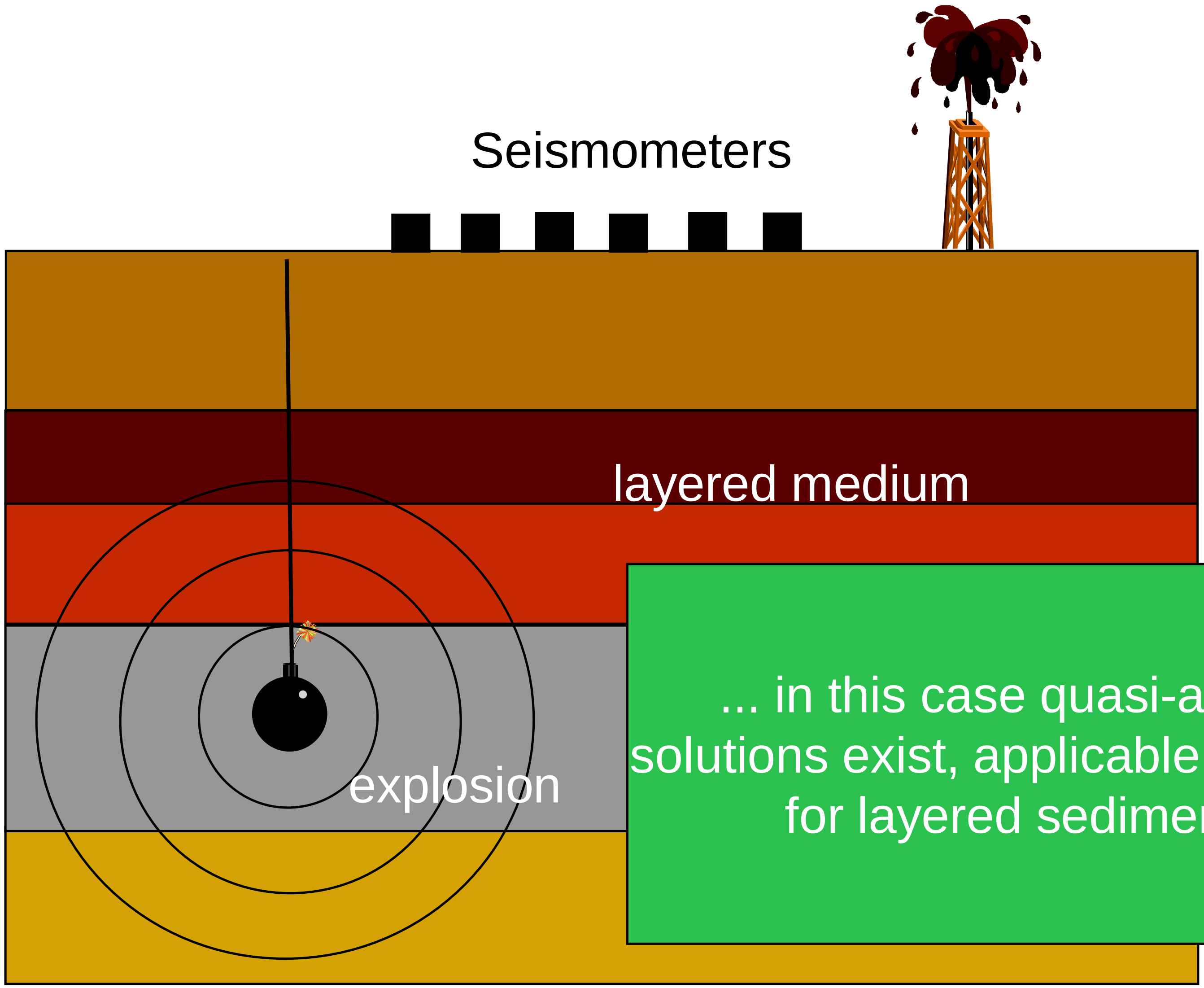
Why numerical methods?

Example: seismic wave propagation



Why numerical methods?

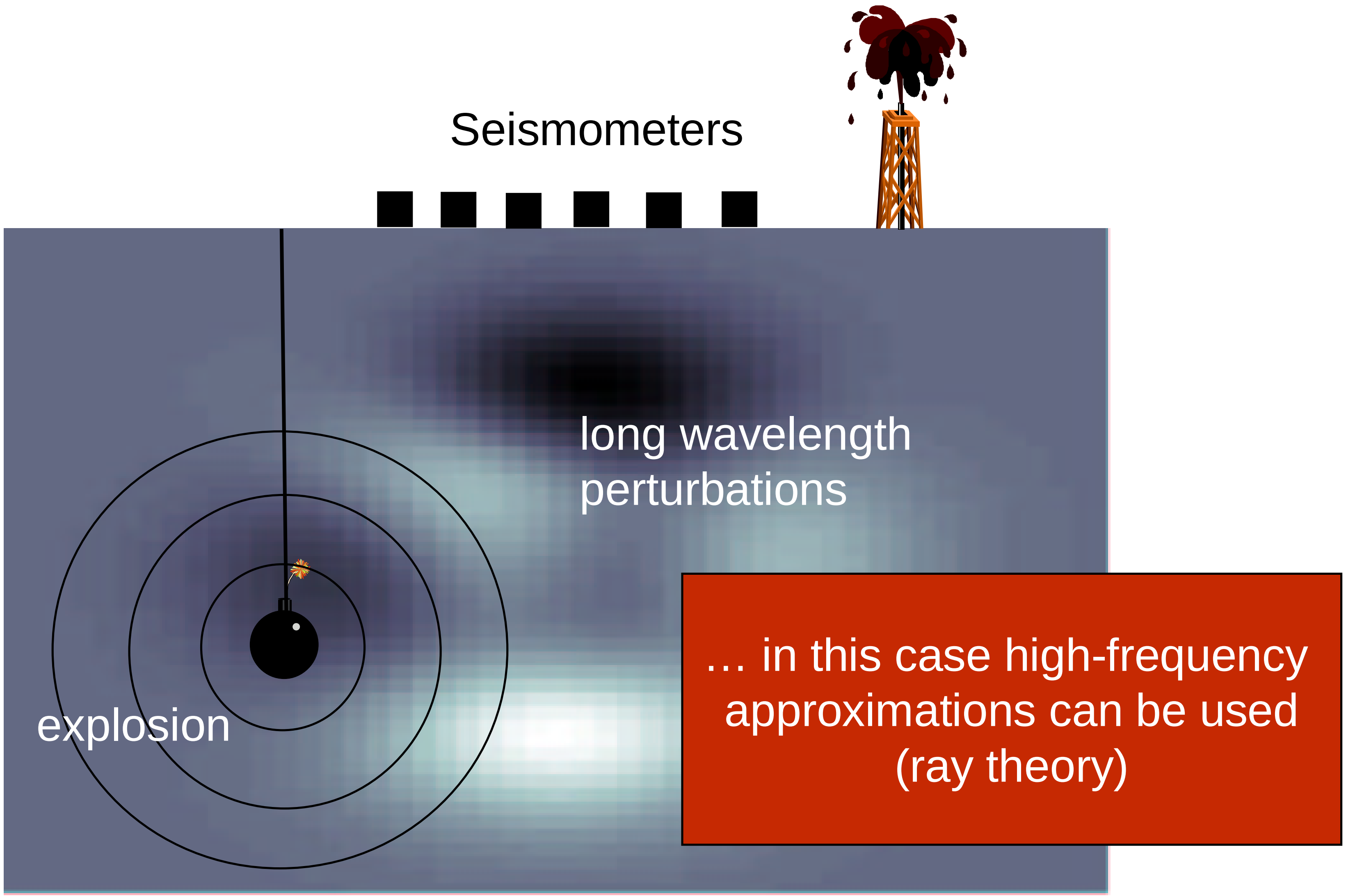
Example: seismic wave propagation



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

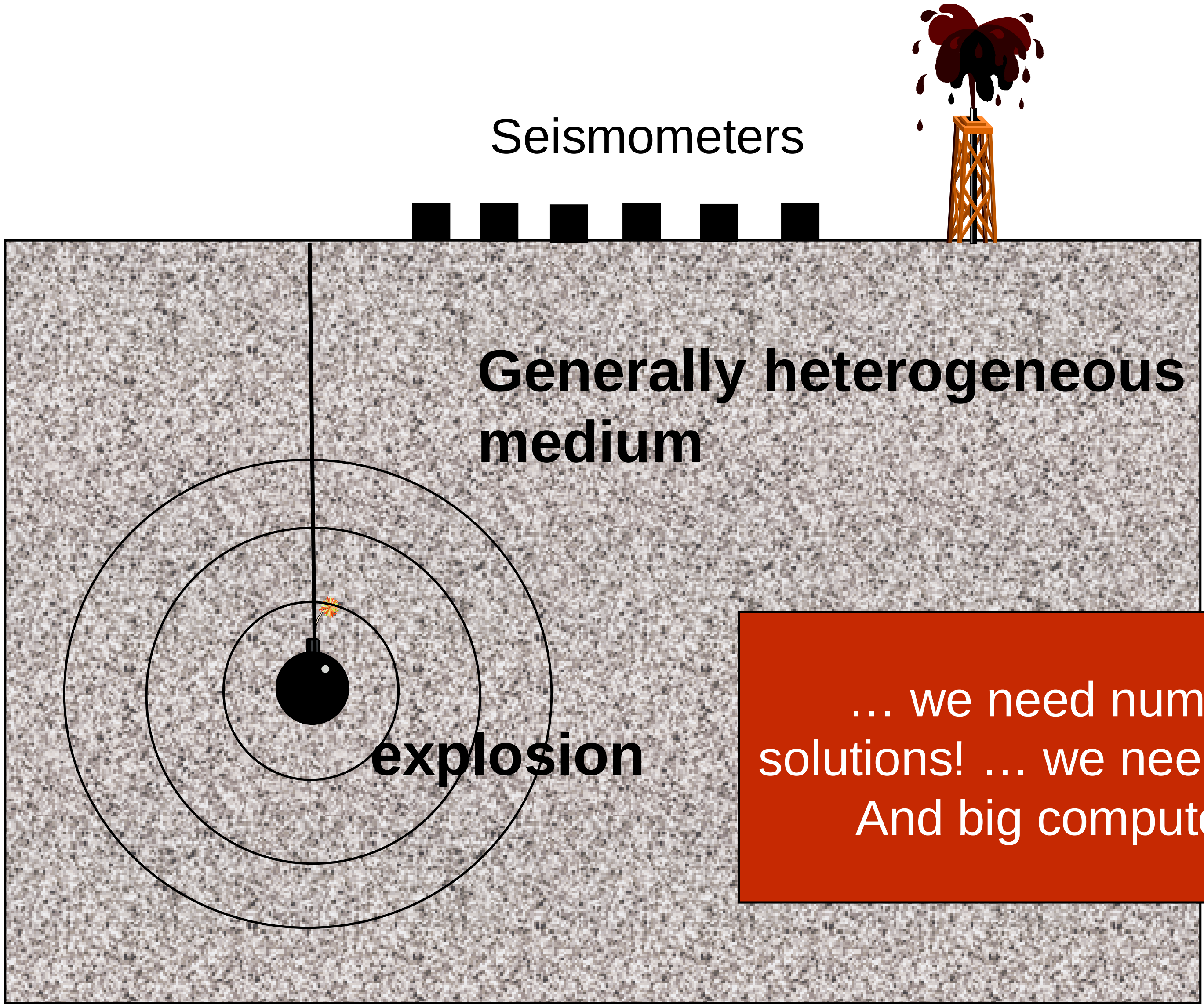
Why numerical methods?

Example: seismic wave propagation



Why numerical methods

Example: seismic wave propagation



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

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: fields of application

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, <u>irregular grids</u>, more complex algorithms, engineering problems

Finite volumes →

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

-> robust, simple concept, <u>irregular grids</u>, 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 in analytical solutions
- only discretization of planes

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

*Pseudo*spectral methods

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

-> regular grids, explicit method, problems with discontinuities

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....

Taylor Series

... that leads to :

$$\frac{f(x+dx)-f(x)}{dx}=\frac{1}{dx}\left[\mathrm{dx}f^{'}(x)+\frac{dx^2}{2!}f^{''}(x)+\frac{dx^3}{3!}f^{'''}(x)+\ldots\right]$$

$$=f^{'}(x)+O(dx)$$

The error of the first derivative using the *forward* formulation is *of order dx*.

Is this the case for other formulations of the derivative? Let’s check!

Taylor Series

... with the *centered* formulation we get:

$$\frac{f(x+dx/2)-f(x-dx/2)}{dx}=\frac{1}{dx}\left[\mathrm{dx}f^{'}(x)+\frac{dx^{3}}{3!}f^{'''}(x)+\ldots\right]$$

$$=f^{'}(x)+O(dx^{2})$$

The error of the first derivative using the centered approximation is *of order* dx^2.

This is an **important** results: it DOES matter which formulation we use. The centered scheme is more accurate!

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

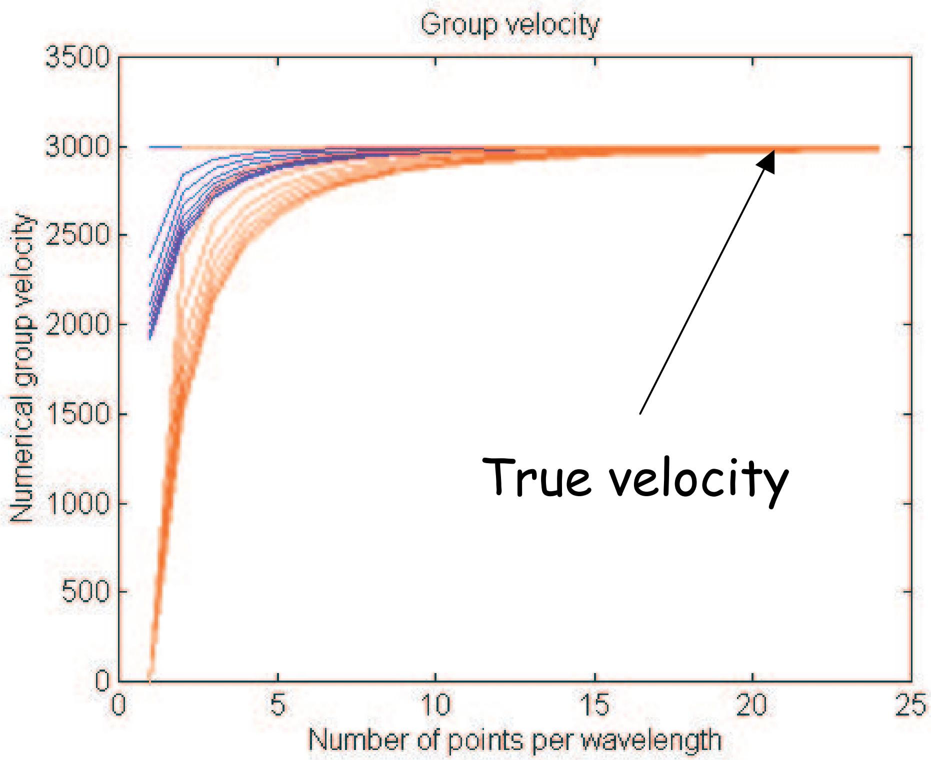
$$p(t+dt)=\frac{c^2dt^2}{dx^2}\left[p(x+dx)-2p(x)+p(x-dx)\right]+2p(t)-p(t-dt)+sdt^2$$

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.

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

Problems: Dispersion

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



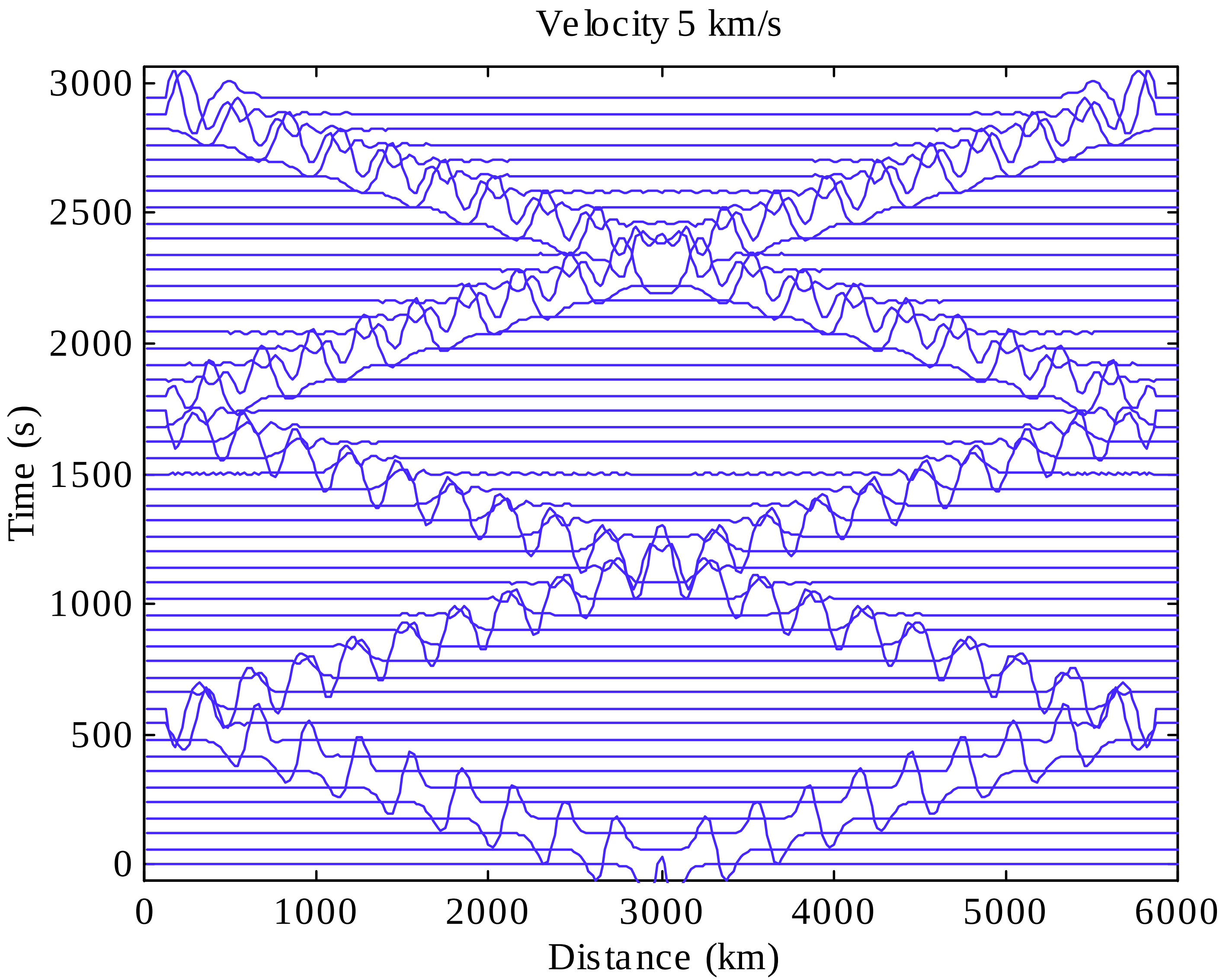
Dispersion: The numerical approximation has artificial dispersion, in other words, the wave speed becomes frequency dependent.
You have to find a frequency bandwidth where this effect is small.
The solution is to use a sufficient number of **grid points per wavelength.**

Our first FD code!

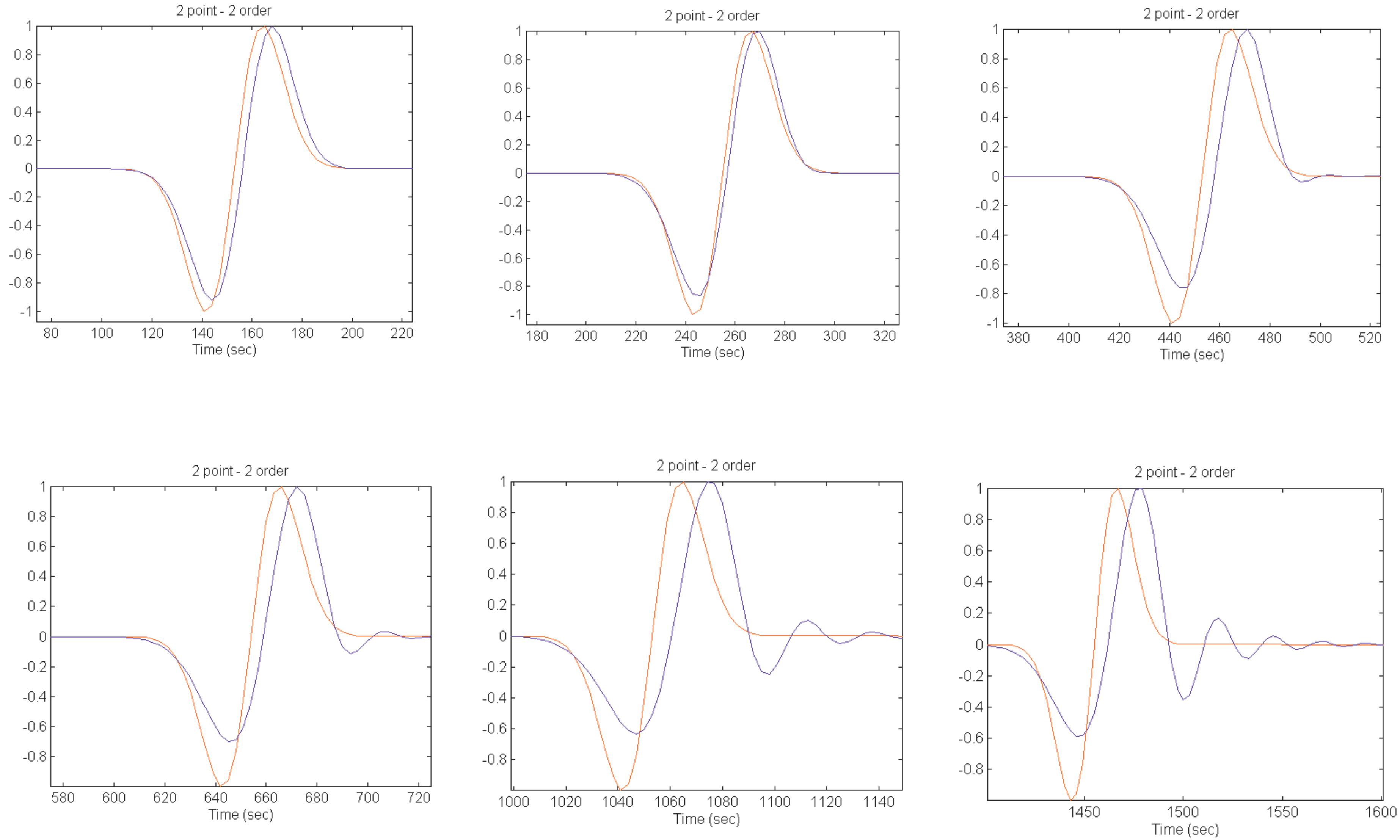
$$p(t+dt) = \frac{c^2 dt^2}{dx^2}\left[p(x+dx) - 2p(x) + p(x-dx)\right] + 2p(t) - p(t-dt) + sdt^2$$

```
% 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

Finite Elements – a definition

Finite elements ...

A general discretization procedure of continuum problems posed by mathematically defined statements

Finite Elements – the concept

How to proceed in FEM analysis:

- Divide structure into pieces (like LEGO)
- Describe behaviour of the physical quantities in each element
- Connect (assemble) the elements at the nodes to form an approximate system of equations for the whole structure
- Solve the system of equations involving unknown quantities at the nodes (e.g. displacements)
- Calculate desired quantities (e.g. strains and stresses) at selected elements

Finite Elements - Why?

FEM allows discretization of bodies with arbitrary shape. Originally designed for problems in static elasticity.

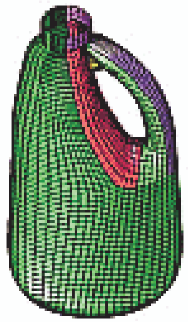
FEM is the most widely applied computer simulation method in engineering.

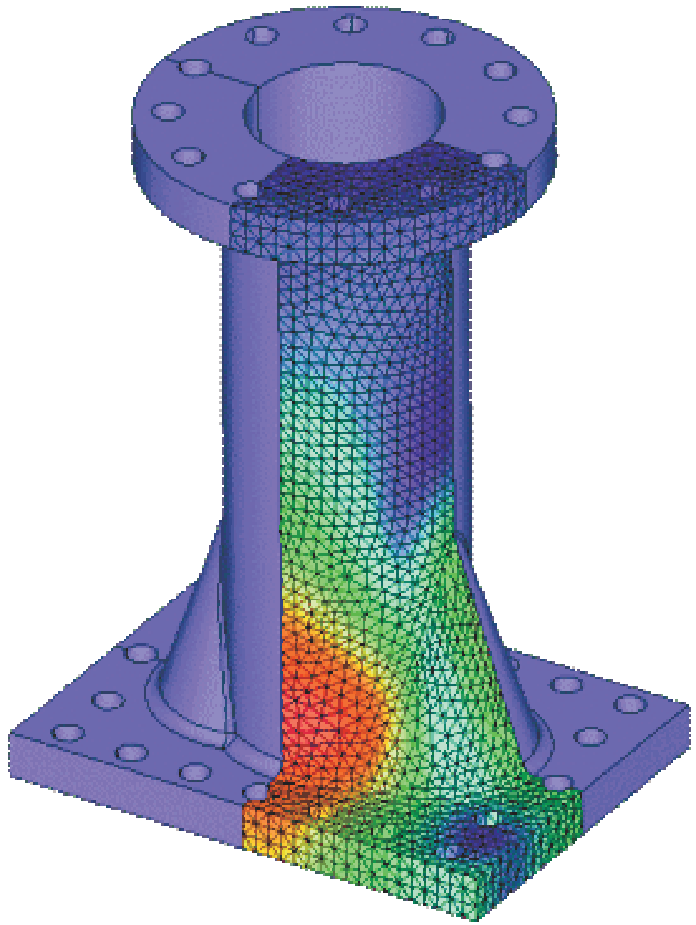
Today spectral elements is an attractive new method with applications in seismology and geophysical fluid dynamics

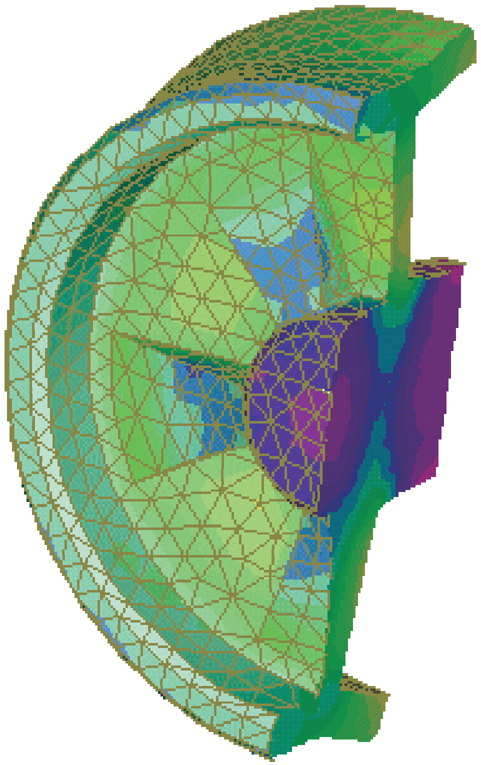
The required grid generation techniques are interfaced with graphical techniques (CAD).

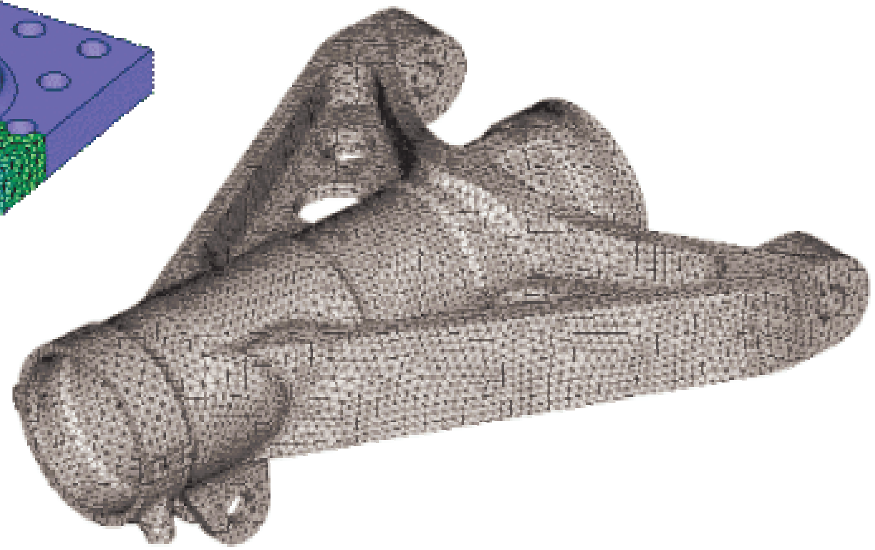
Today a large number of commercial FEM software is available (e.g. ANSYS, SMART, MATLAB, ABACUS, etc.)

Finite Elements - Examples









Finite elements – basic formulation

Let us start with a simple linear system of equations

$$\mathbf{Ax} = \mathbf{b} \qquad | * \mathbf{y}$$

and observe that we can generally multiply both sides of this equation with **y** without changing its solution. Note that **x**,**y** and **b** are vectors and **A** is a matrix.

$$\rightarrow \mathbf{yAx} = \mathbf{yb} \qquad \mathbf{y} \in \Re^{\mathbf{n}}$$

We first look at Poisson's equation

$$-\Delta u(x) = f(x)$$

where u is a scalar field, f is a source term and in 1-D

$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2}$$

Formulation – Poisson's equation

We now multiply this equation with an arbitrary function v(x), (dropping the explicit space dependence)

$$-\Delta u v = f v$$

... and integrate this equation over the whole domain. For reasons of simplicity we define our physical domain *D* in the interval [0, 1].

$$-\int_D \Delta u v = \int_D f v$$

$$-\int_0^1 \Delta u v dx = \int_0^1 f v dx$$

... why are we doing this? ... be patient ...

Partial Integration

... partially integrate the left-hand-side of our equation ...

$$-\int_0^1 \Delta u v dx = \int_0^1 f v dx$$

$$-\int_0^1 (\nabla \bullet \nabla u) v dx = \left[\nabla u v\right]_0^1 + \int_0^1 \nabla v \nabla u dx$$

we assume for now that the derivatives of u at the boundaries vanish so that for our particular problem

$$-\int_0^1 (\nabla \bullet \nabla u) v dx = \int_0^1 \nabla v \nabla u dx$$

... so that we arrive at ...

$$\int_0^1 \nabla u \nabla v dx = \int_0^1 f v dx$$

... with u being the unknown. This is also true for our approximate numerical system

$$\int_0^1 \nabla \tilde{u} \nabla v dx = \int_0^1 f v dx$$

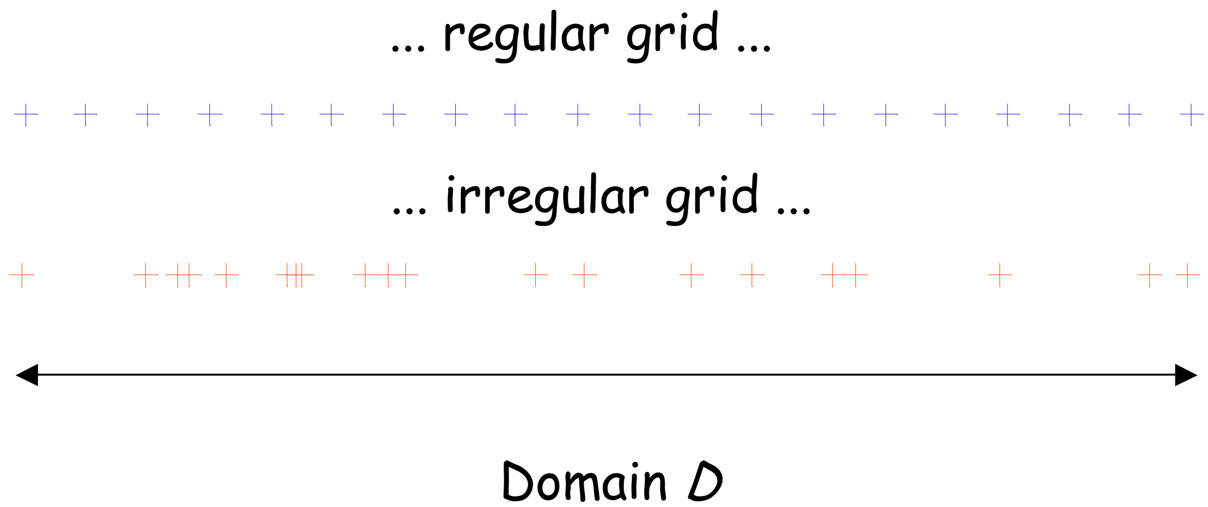
... where ...

$$\tilde{u} = \sum_{i=1}^{N} c_i \varphi_i$$

was our choice of approximating u using basis functions.

Discretization

As we are aiming to find a numerical solution to our problem it is clear we have to discretize the problem somehow. In FE problems – similar to FD – the functional values are known at a discrete set of points.



The key idea in FE analysis is to approximate all functions in terms of basis functions φ, so that

$$u \approx \tilde{u} = \sum_{i=1}^{N} c_i \varphi_i$$

The discrete system

The ingredients:

$$v = \varphi_k$$

$$\tilde{u} = \sum_{i=1}^{N} c_i \varphi_i$$

$$\int_0^1 \nabla \tilde{u} \nabla v dx = \int_0^1 f v dx$$

$$\int_0^1 \nabla \left(\sum_{i=1}^{n} c_i \varphi_i \right) \nabla \varphi_k dx = \int_0^1 f \varphi_k dx$$

... leading to ...

The discrete system

... the coefficients c_k are constants so that for one particular function φ_k this system looks like ...

$$\sum_{i=1}^{n} c_i \int_0^1 \nabla \varphi_i \nabla \varphi_k dx = \int_0^1 f \varphi_k dx$$

... probably not to your surprise this can be written in matrix form

$$b_i A_{ik} = g_k$$

$$A_{ik}^T b_i = g_k$$

The solution

... with the even less surprising solution

$$b_i = \left(A_{ik}^T\right)^{-1} g_k$$

remember that while the b_i's are really the coefficients of the basis functions these are the actual function values at node points i as well because of our particular choice of basis functions.

This become clear further on ...

The basis functions

we are looking for functions φ_i with the following property

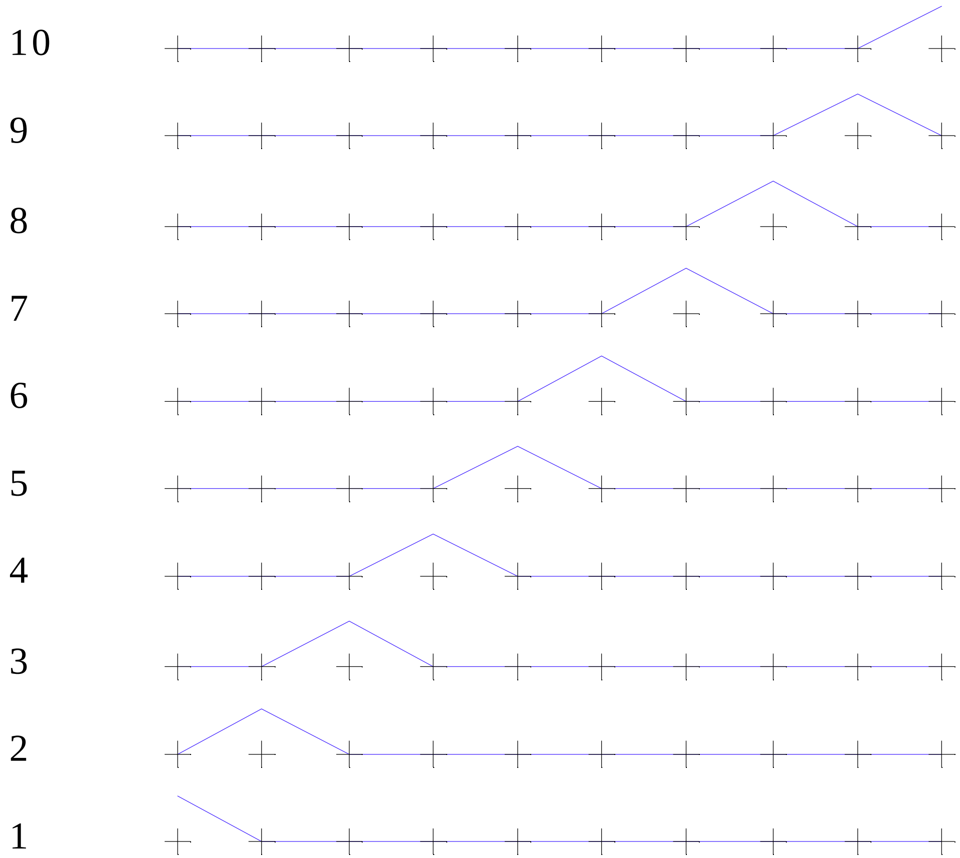
$$\varphi_i(x) = \begin{cases} 1 & for \quad x = x_i \\ 0 & for \quad x = x_j,\ j \neq i \end{cases}$$

... otherwise we are free to choose any function ...

The simplest choice are of course linear functions:

+ grid nodes

blue lines – basis functions φ_i



rectangles: linear elements

With the linear *Ansatz*

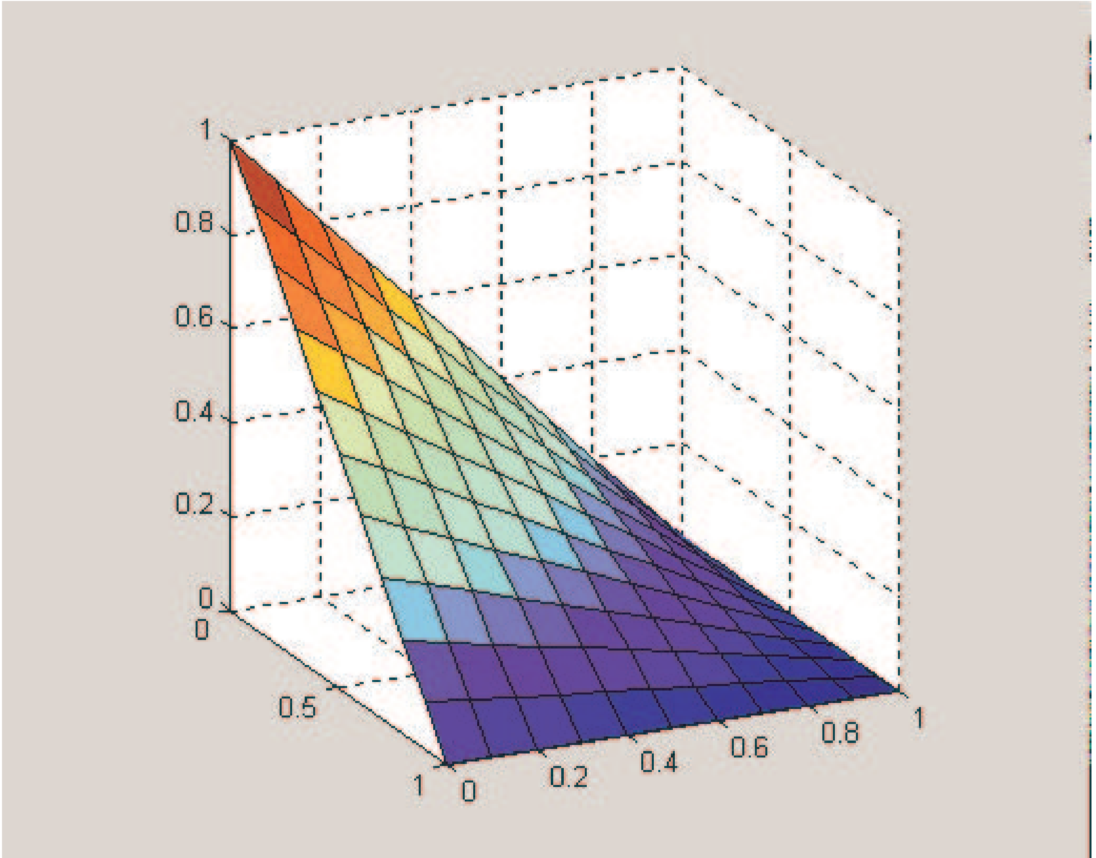
$$u(\xi,\eta) = c_1 + c_2\xi + c_3\eta + c_4\xi\eta$$

we obtain matrix A as

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

and the basis functions

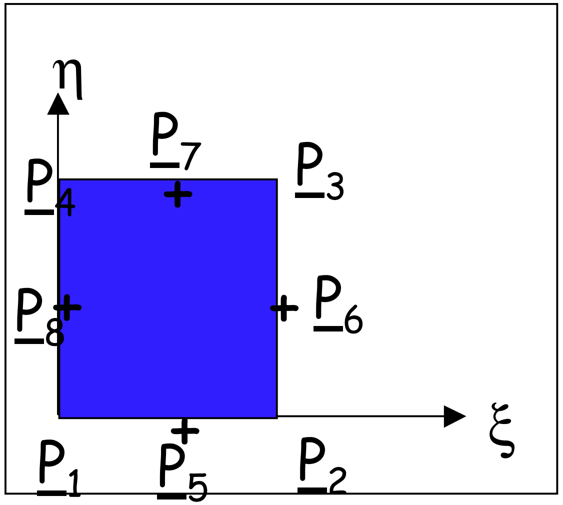
$$N_1(\xi,\eta) = (1-\xi)(1-\eta)$$
$$N_2(\xi,\eta) = \xi(1-\eta)$$
$$N_3(\xi,\eta) = \xi\eta$$
$$N_4(\xi,\eta) = (1-\xi)\eta$$



rectangles: quadratic elements

With the quadratic *Ansatz*

$$u(\xi,\eta) = c_1 + c_2\xi + c_3\eta + c_4\xi^2 + c_5\xi\eta + c_6\eta^2 + c_7\xi^2\eta + c_8\xi\eta^2$$

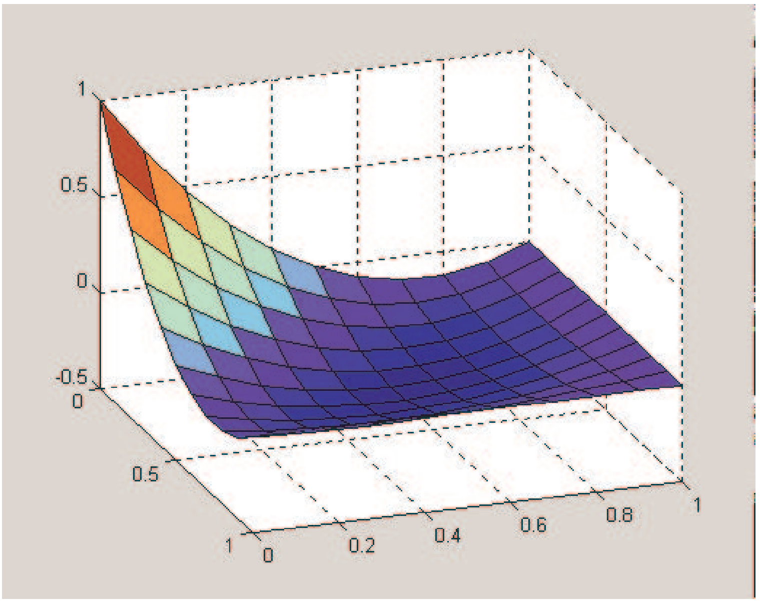


we obtain an 8x8 matrix A ... and a basis function look e.g. like

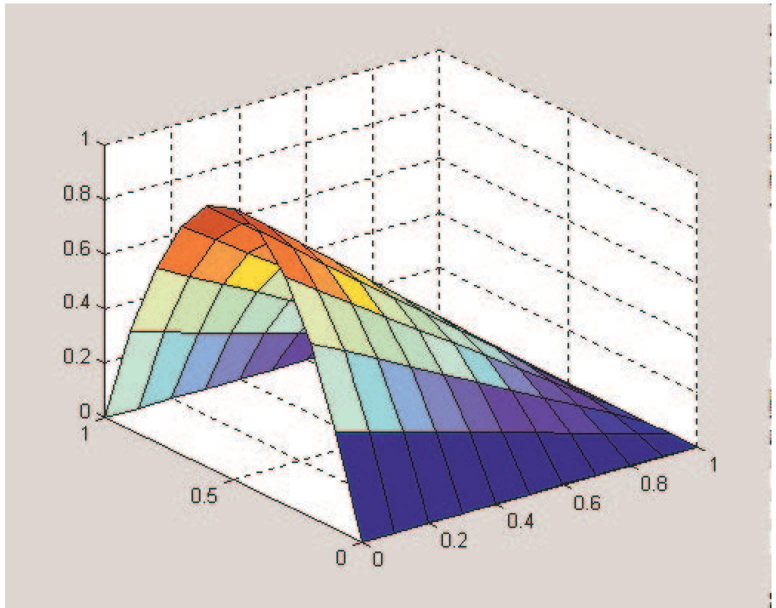
$$N_1(\xi,\eta) = (1-\xi)(1-\eta)(1-2\xi-2\eta)$$

$$N_5(\xi,\eta) = 4\xi(1-\xi)(1-\eta)$$

N_1



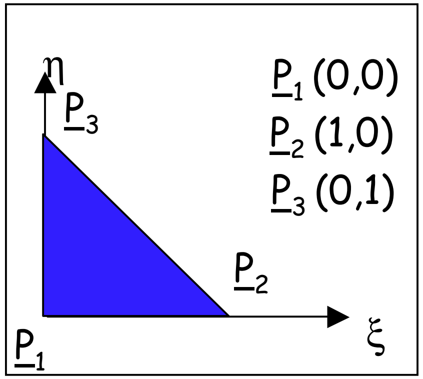
N_2

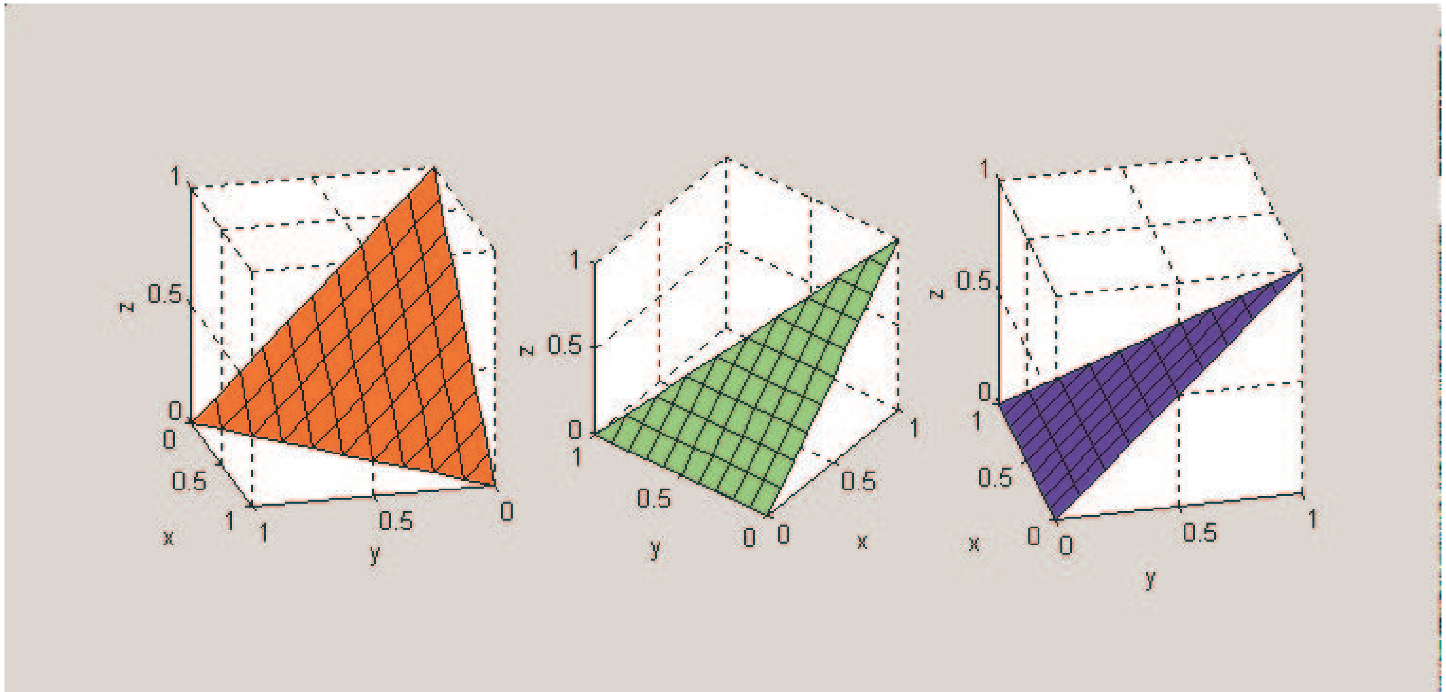


triangles: linear basis functions

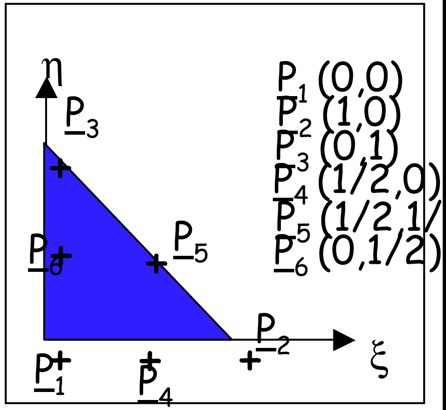
from matrix A we can calculate the linear basis functions for triangles

$$N_1(\xi,\eta) = 1-\xi-\eta$$
$$N_2(\xi,\eta) = \xi$$
$$N_3(\xi,\eta) = \eta$$

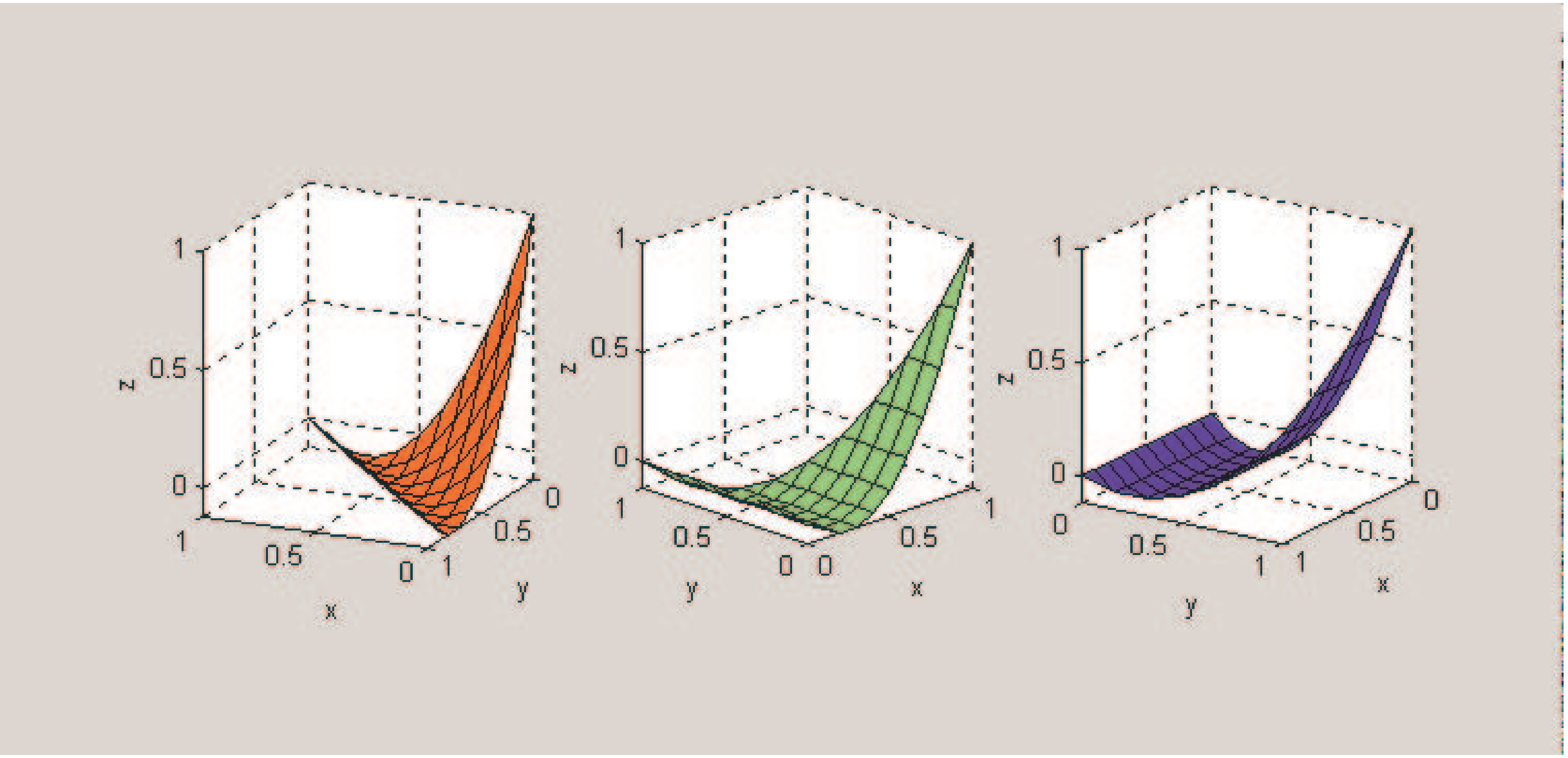




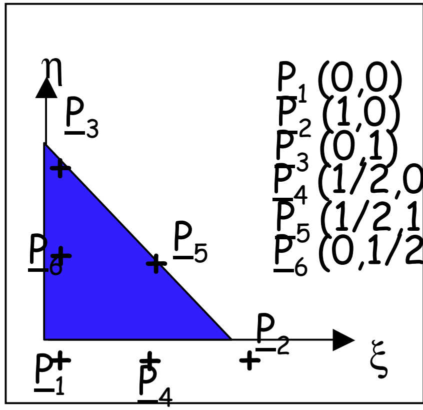
triangles: quadratic basis functions



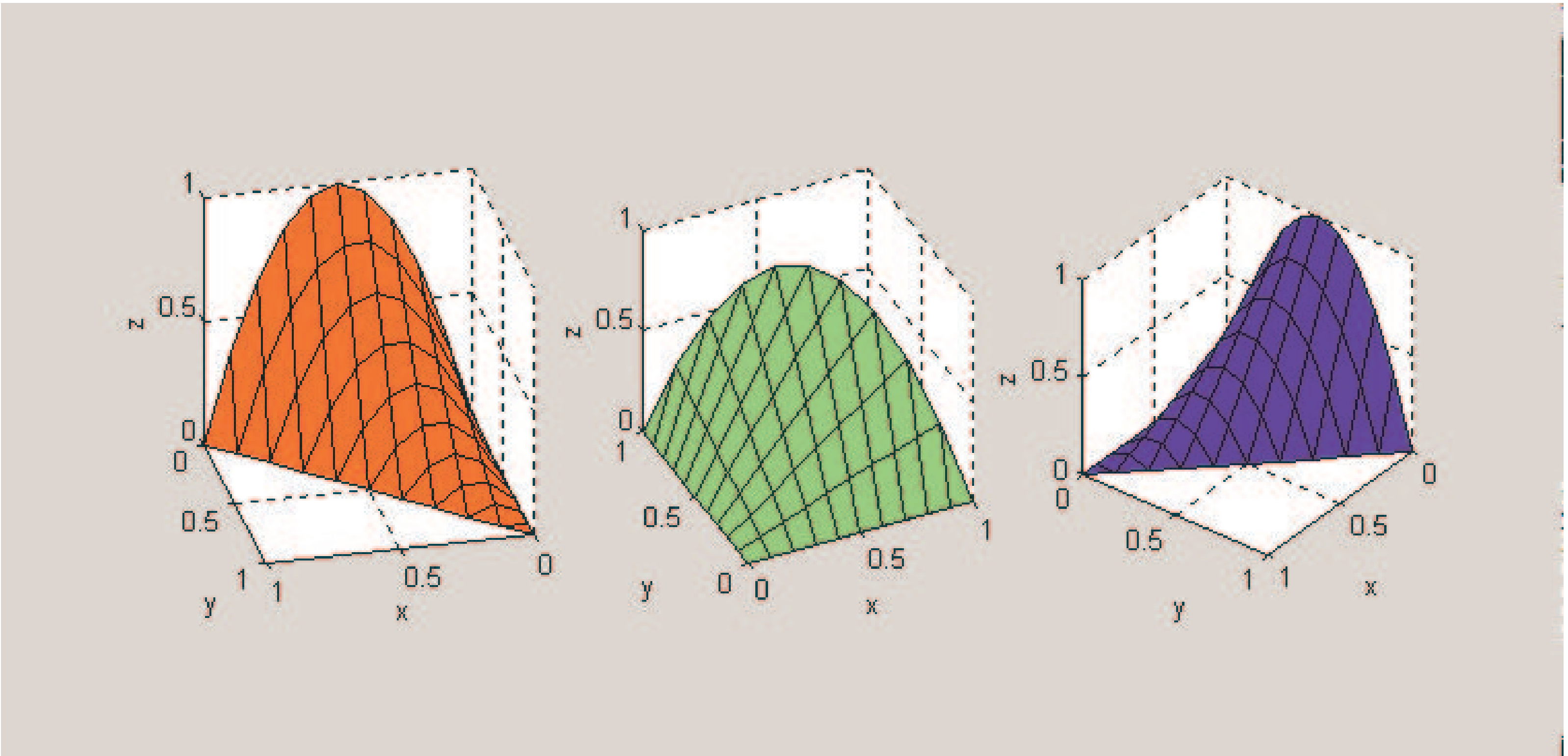
The first three quadratic basis functions ...



triangles: quadratic basis functions



.. and the rest ...



The Acoustic Wave Equation 1-D

How do we solve a time-dependent problem such as the acoustic wave equation?

$$\partial_t^2 u - v^2 \Delta u = f$$

where v is the wave speed.
using the same ideas as before we multiply this equation with an arbitrary function and integrate over the whole domain, e.g. [0,1], and after partial integration

$$\int_0^1 \partial_t^2 u \varphi_j dx - v^2 \int_0^1 \nabla u \nabla \varphi_j dx = \int_0^1 f \varphi_j dx$$

.. we now introduce an approximation for u using our previous basis functions...

The Acoustic Wave Equation 1-D

$$u \approx \tilde{u} = \sum_{i=1}^{N} c_i(t)\varphi_i(x)$$

note that now our coefficients are time-dependent!

... and ...

$$\partial_t^2 u \approx \partial_t^2 \tilde{u} = \partial_t^2 \sum_{i=1}^{N} c_i(t)\varphi_i(x)$$

together we obtain

$$\left[\sum_i \partial_t^2 c_i \int_0^1 \varphi_i \varphi_j dx\right] + v^2 \left[\sum_i c_i \int_0^1 \nabla\varphi_i \nabla\varphi_j dx\right] = \int_0^1 f\varphi_j$$

which we can write as ...

Time extrapolation

$$\left[\sum_i \partial_t^2 c_i \int_0^1 \varphi_i \varphi_j dx\right] + v^2 \left[\sum_i c_i \int_0^1 \nabla\varphi_i \nabla\varphi_j dx\right] = \int_0^1 f\varphi_j$$

M — mass matrix

A — stiffness matrix

b

... in Matrix form ...

$$M^T \ddot{c} + v^2 A^T c = g$$

... remember the coefficients c correspond to the actual values of u at the grid points for the right choice of basis functions ...

How can we solve this time-dependent problem?

FD extrapolation

$$M^T \ddot{c} + v^2 A^T c = g$$

... let us use a finite-difference approximation for the time derivative ...

$$M^T \left(\frac{c_{k+1} - 2c + c_{k-1}}{dt^2} \right) + v^2 A^T c_k = g$$

... leading to the solution at time t_{k+1}:

$$c_{k+1} = \left[(M^T)^{-1} (g - v^2 A^T c_k) \right] dt^2 + 2c_k - c_{k-1}$$

we already know how to calculate the matrix A but
how can we calculate matrix M?

Matrix assembly

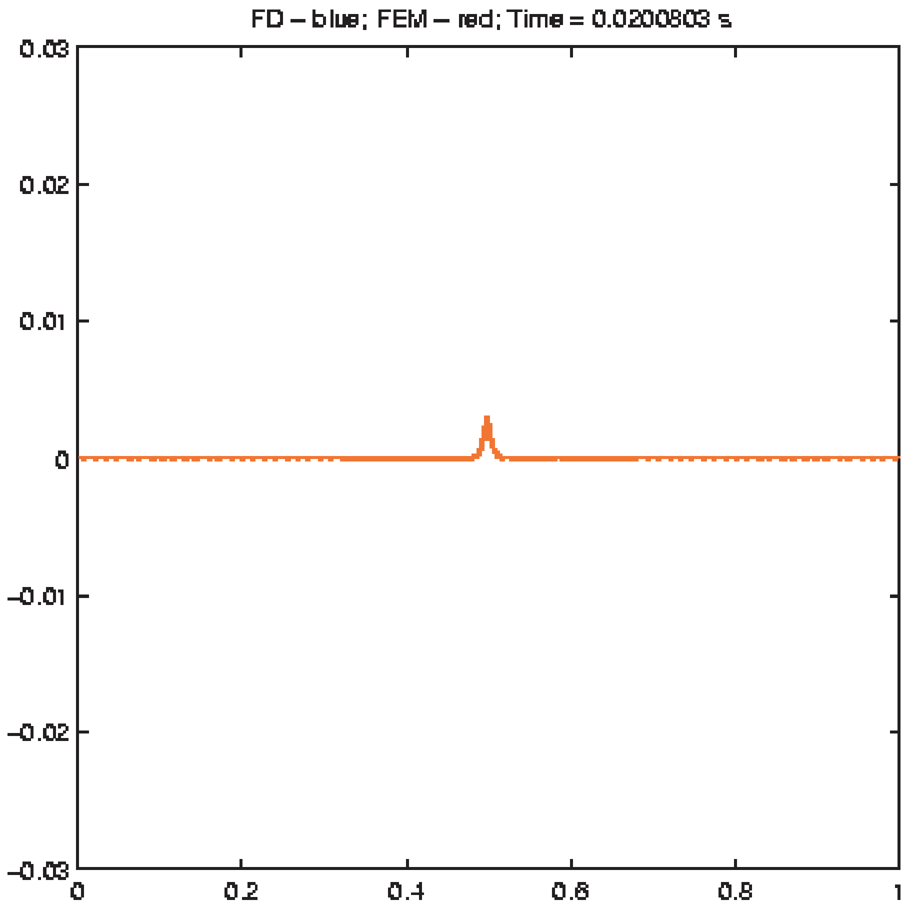
M_{ij}

```
% assemble matrix Mij

M=zeros(nx);

for i=2:nx-1,
   for j=2:nx-1,
      if i==j,
         M(i,j)=h(i-1)/3+h(i)/3;
      elseif j==i+1
         M(i,j)=h(i)/6;
      elseif j==i-1
         M(i,j)=h(i)/6;
      else
         M(i,j)=0;
      end
   end
end
```

A_{ij}

```
% assemble matrix Aij

A=zeros(nx);

for i=2:nx-1,
   for j=2:nx-1,
      if i==j,
         A(i,j)=1/h(i-1)+1/h(i);
      elseif i==j+1
         A(i,j)=-1/h(i-1);
      elseif i+1==j
         A(i,j)=-1/h(i);
      else
         A(i,j)=0;
      end
   end
end
```

Numerical example - regular grid

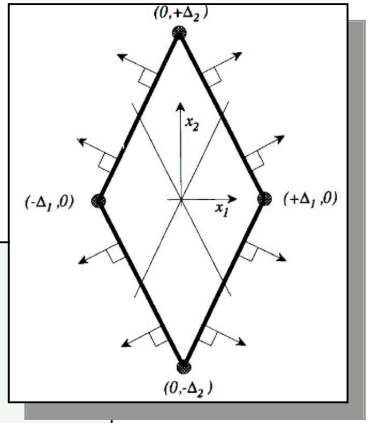


This is a movie obtained with the sample Matlab program: femfd.m

Finite Elements - Summary

- FE solutions are based on the "weak form" of the partial differential equations
- FE methods lead in general to a linear system of equations that has to be solved using matrix inversion techniques (sometimes these matrices can be diagonalized)
- FE methods allow rectangular (hexahedral), or triangular (tetrahedral) elements and are thus more flexible in terms of grid geometry
- The FE method is mathematically and algorithmically more complex than FD
- The FE method is well suited for elasto-static and elasto-dynamic problems (e.g. crustal deformation)

Finite volumes



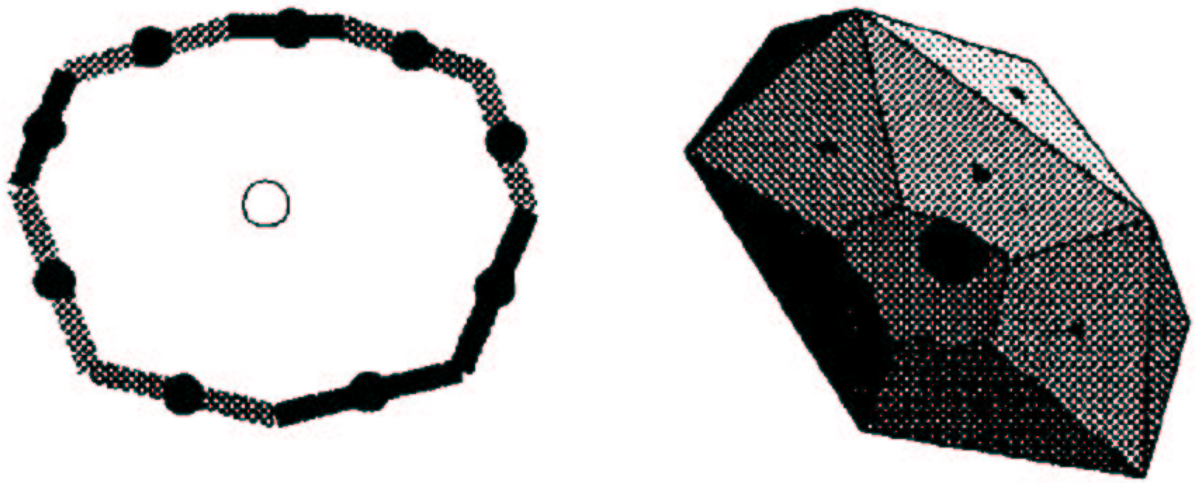
Finite volumes …

A numerical method based on a discrete version of Gauss´ theorem.

- The theoretical basis
- FV for hexagonal and irregular grids

… this part is based on :
Dormy E. and Tarantola A., J. Geophys. Res., 100, 2123-2133, 1995.

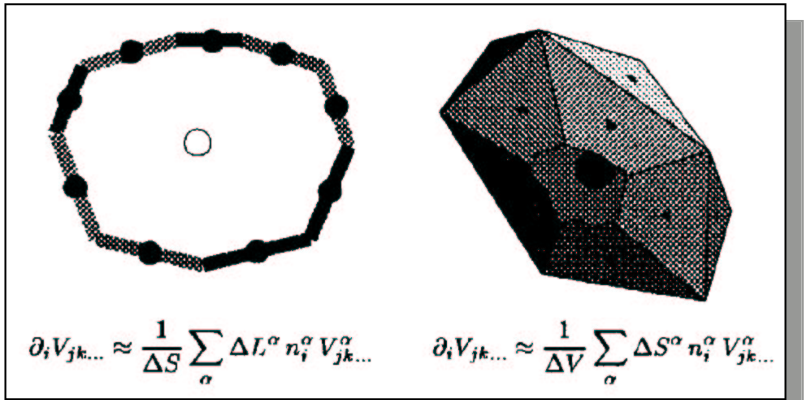
Finite volumes – basic theory



$$\partial_i V_{jk\ldots} \approx \frac{1}{\Delta S}\sum_\alpha \Delta L^\alpha\, n_i^\alpha\, V_{jk\ldots}^\alpha \qquad \partial_i V_{jk\ldots} \approx \frac{1}{\Delta V}\sum_\alpha \Delta S^\alpha\, n_i^\alpha\, V_{jk\ldots}^\alpha$$

… as the figure suggests, the FV method is based on the idea of knowing a 3D field at the sides of a surface surrounding a **finite volume**. Is there a mathematical theorem relating the (vector) fields inside a volume with the values at its surface? …. Yes, it s **Gauss´ theorem** …

Finite volumes – Gauss´ theorem



Gauss´ theorem:
(by the way one of the most important results on mathematical physics)

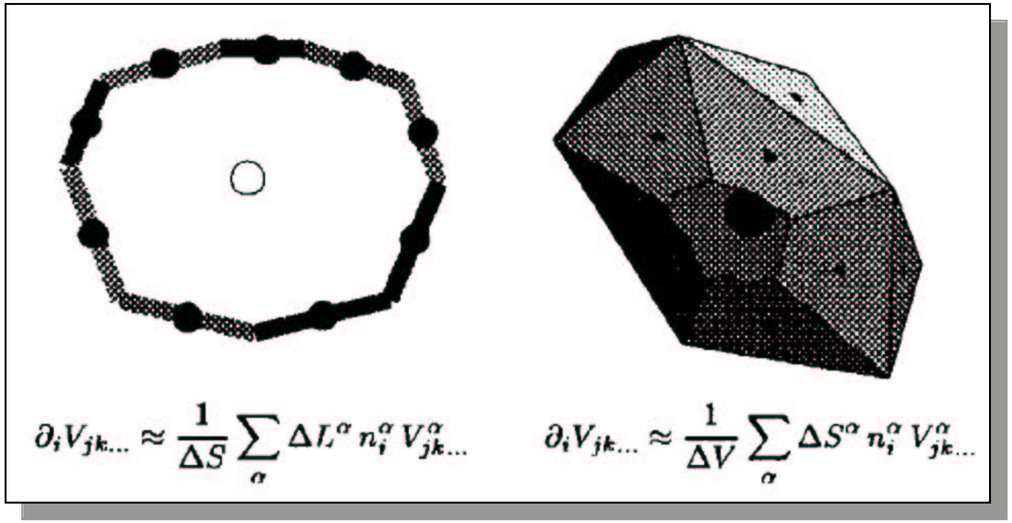
$$\int_V dV \partial_i w_i = \int_S dS n_i w_i$$

S	boundary surrounding V
V	volume inside S
w_i	vector field
n_i	unitary normal to the surface (pointing outwards)

Finite volumes - 3D

We simply need to turn Gauss´ theorem into a discrete version!

Assumption: smoothly varying W_{jk}

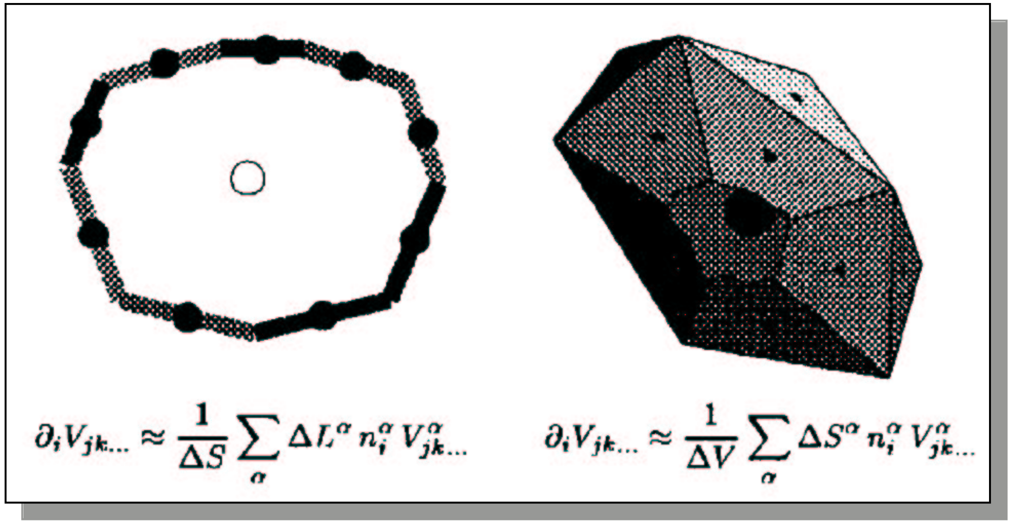


$$\partial_i W_{jk} \approx \frac{1}{\Delta V} \sum_\alpha \Delta S_\alpha n_i^\alpha W_{jk}^\alpha$$

W_{jk}	arbitrary tensor field
ΔV	total volume
ΔS_α	surface segment
n_i	unitary normal to the surface
α	number of surface segments

Finite volumes - 2D

$$\partial_i W_{jk} \approx \frac{1}{\Delta S} \sum_{\alpha} \Delta L_{\alpha} n_i^{\alpha} W_{jk}^{\alpha}$$

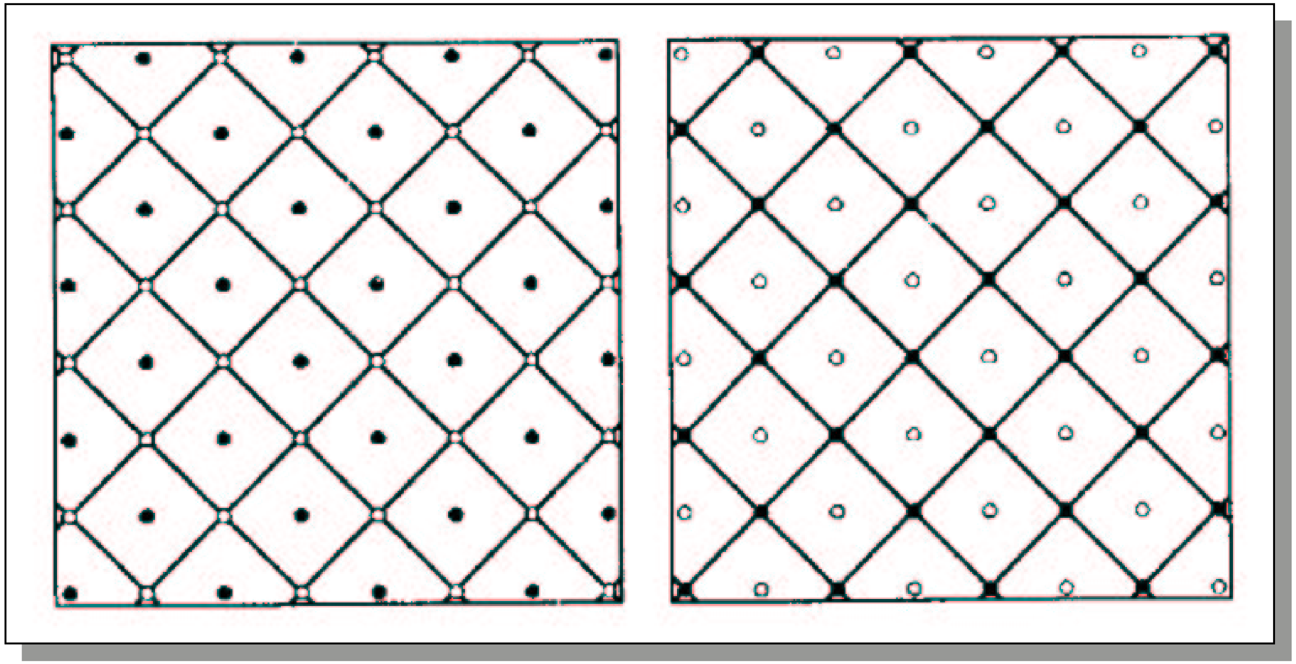


W_{jk}	arbitrary tensor field
ΔS	total surface
ΔL_{α}	boundary segment
n_i	unitary normal to the surface
α	number of surface segments

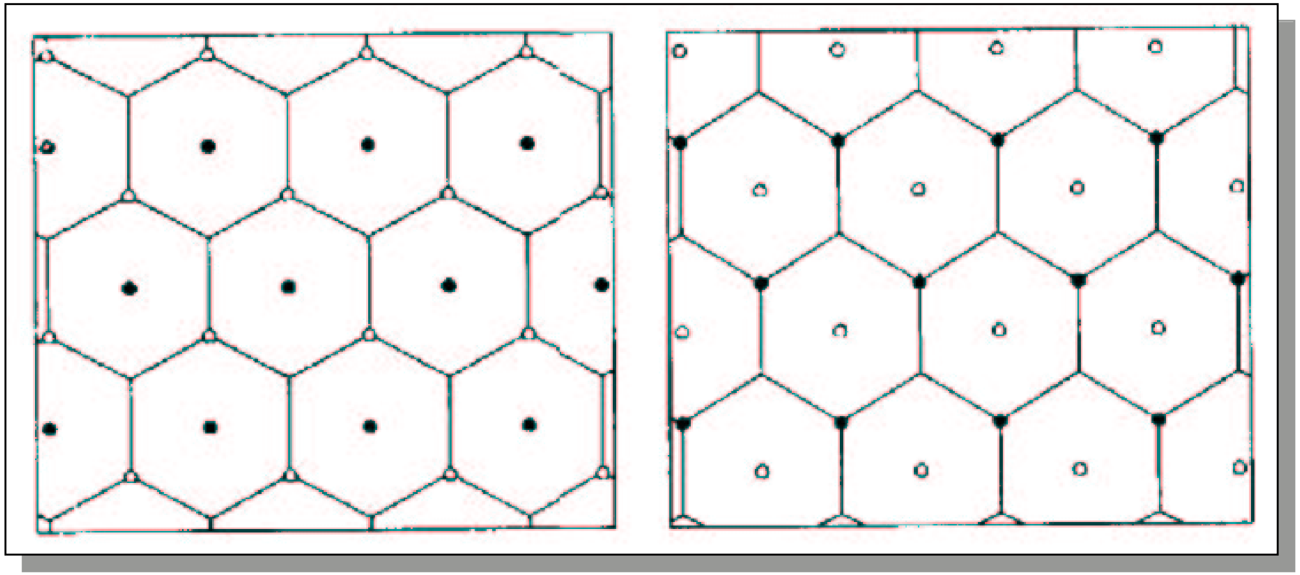
How can we use these ideas to solve p.d.e.´s ?

Finite volumes - space grids

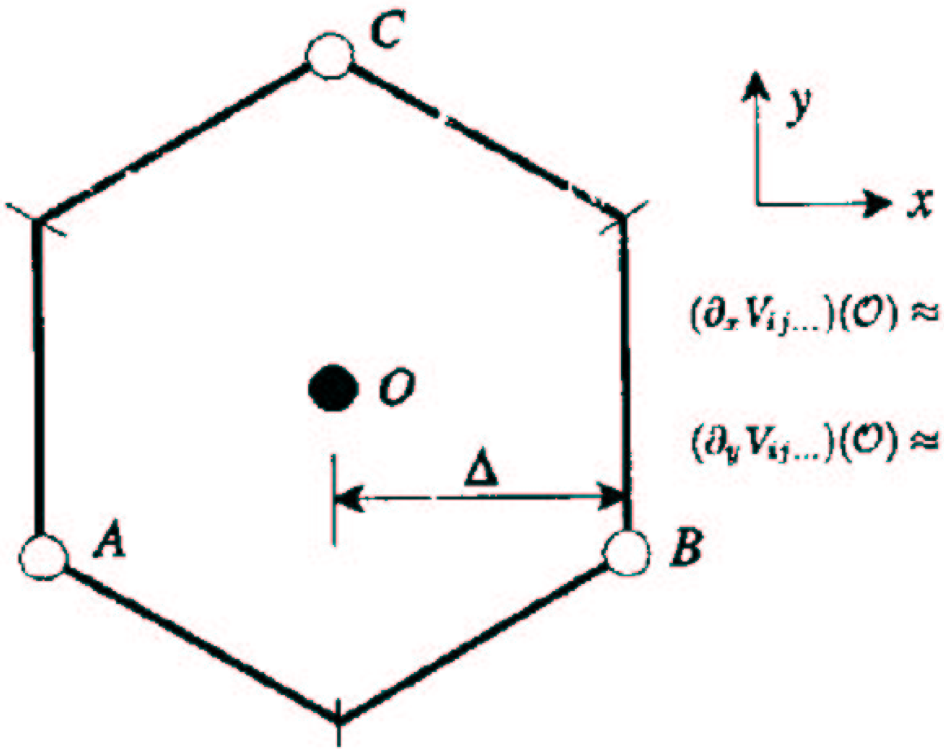
2D Euclidian space
- Lozenges
- staggered grid



2D Euclidian space
- hexagons
- minimal grid

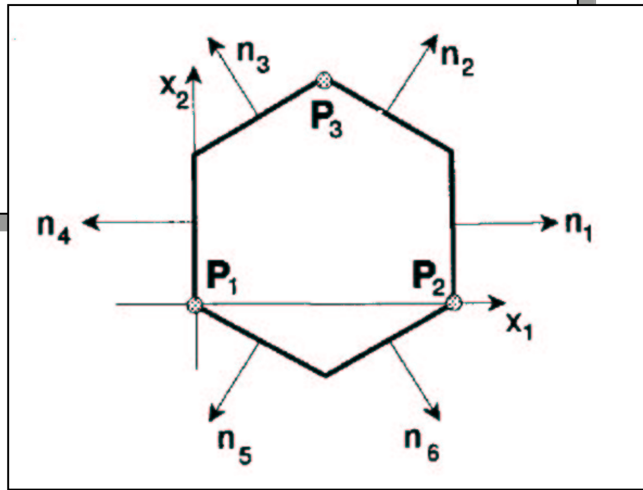


Finite volumes



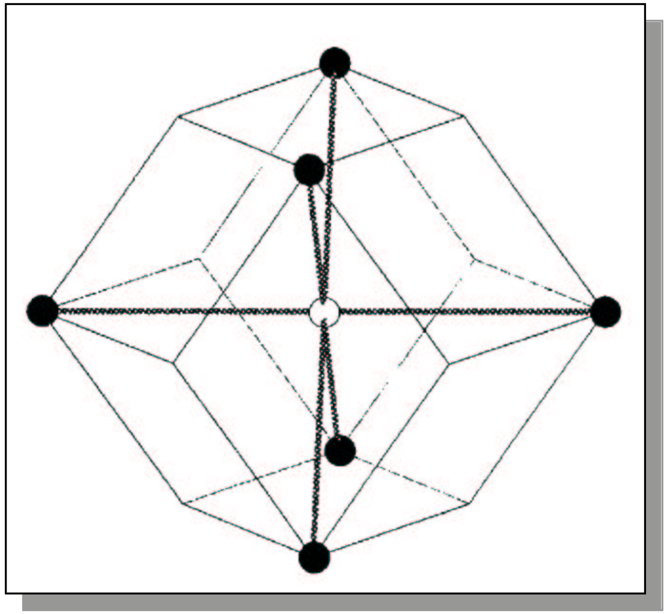
$$(\partial_x V_{ij\ldots})(O) \approx \frac{V_{ij\ldots}(B) - V_{ij\ldots}(A)}{2\Delta}$$

$$(\partial_y V_{ij\ldots})(O) \approx \frac{V_{ij\ldots}(C) - \frac{1}{2}(V_{ij\ldots}(B) + V_{ij\ldots}(A))}{\sqrt{3}\,\Delta}$$

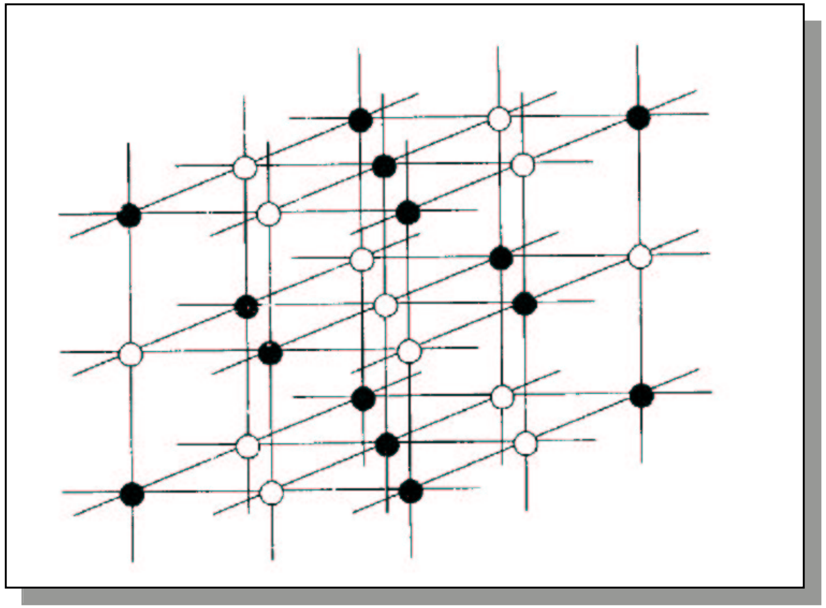


Minimal grid for finite volumes

Finite volumes - space grids



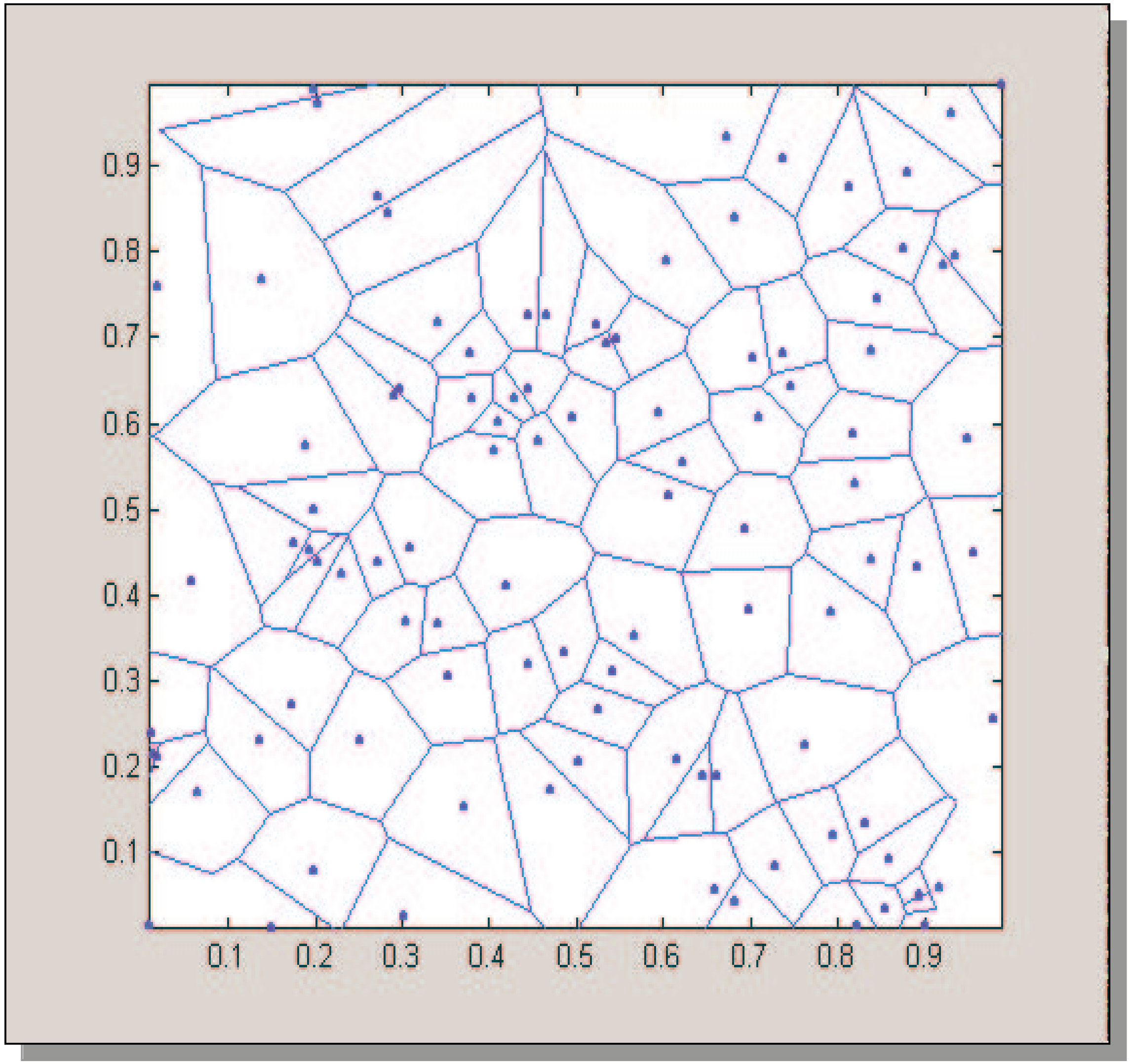
Voronoi cell for FD grid

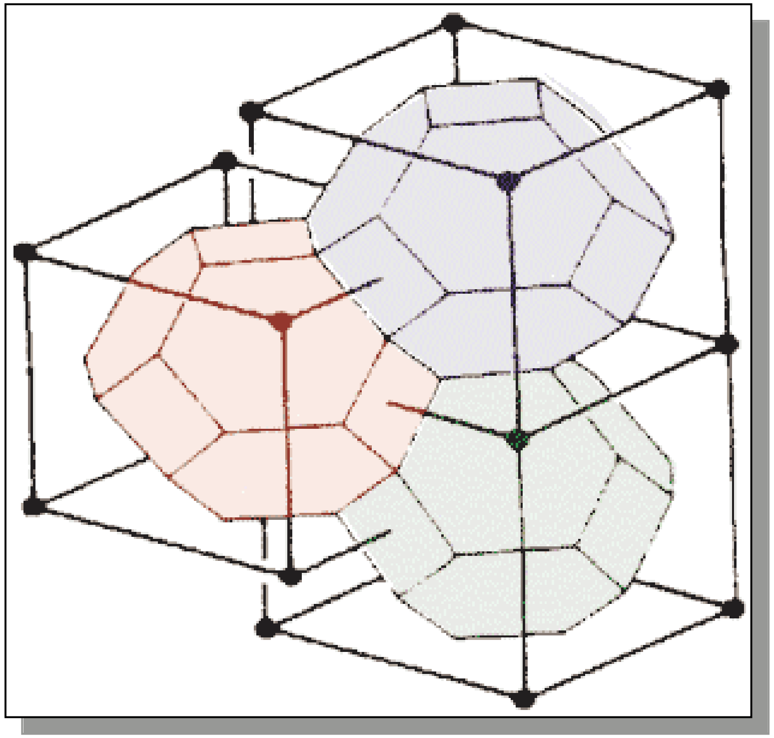


Classic FD grid in 3D

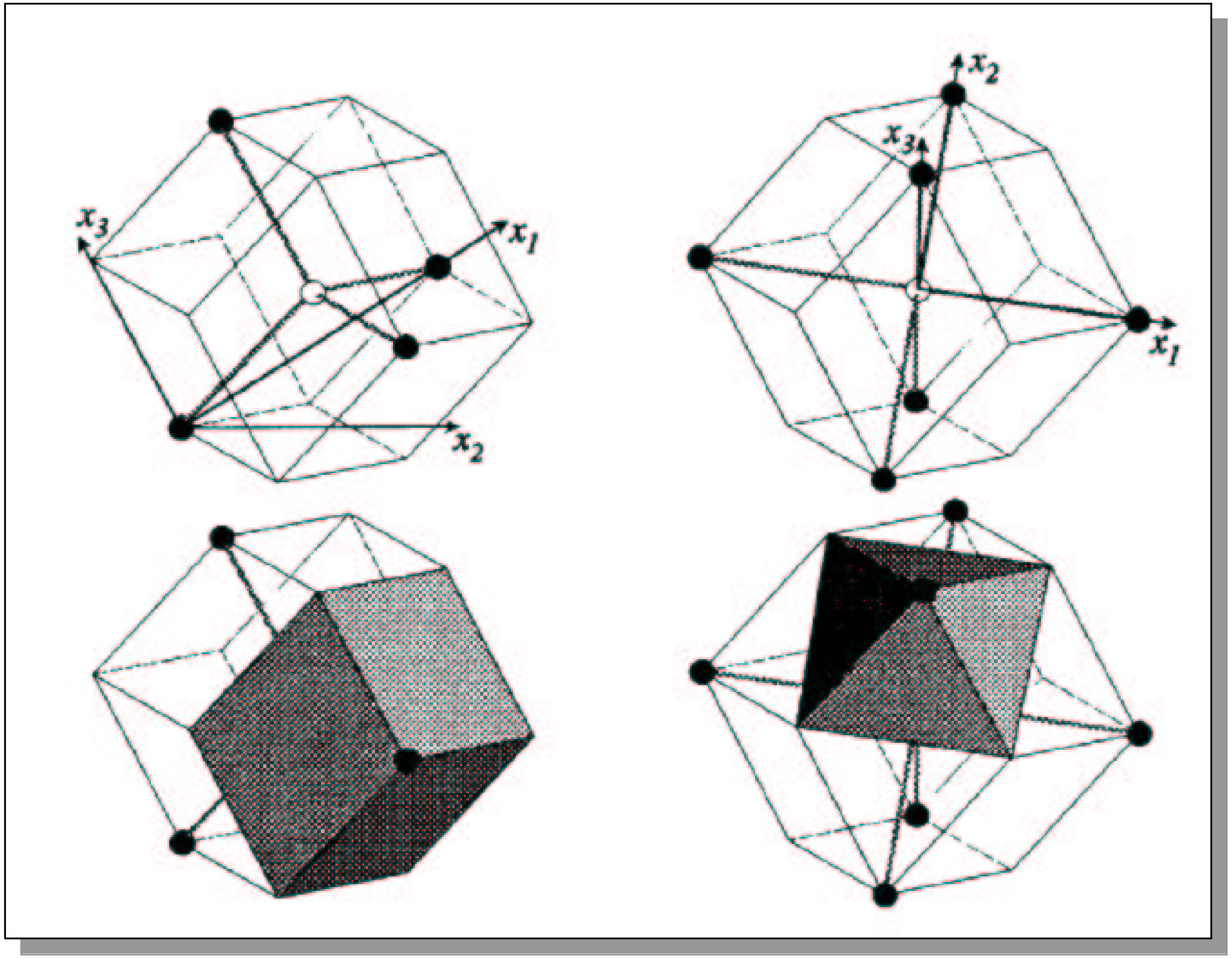
The Voronoi diagrams of an unstructured set of nodes divides the plane into a set of regions, one for each node, such that any point in a particular region is closer to that regions node than to any other.

Voronoi cells

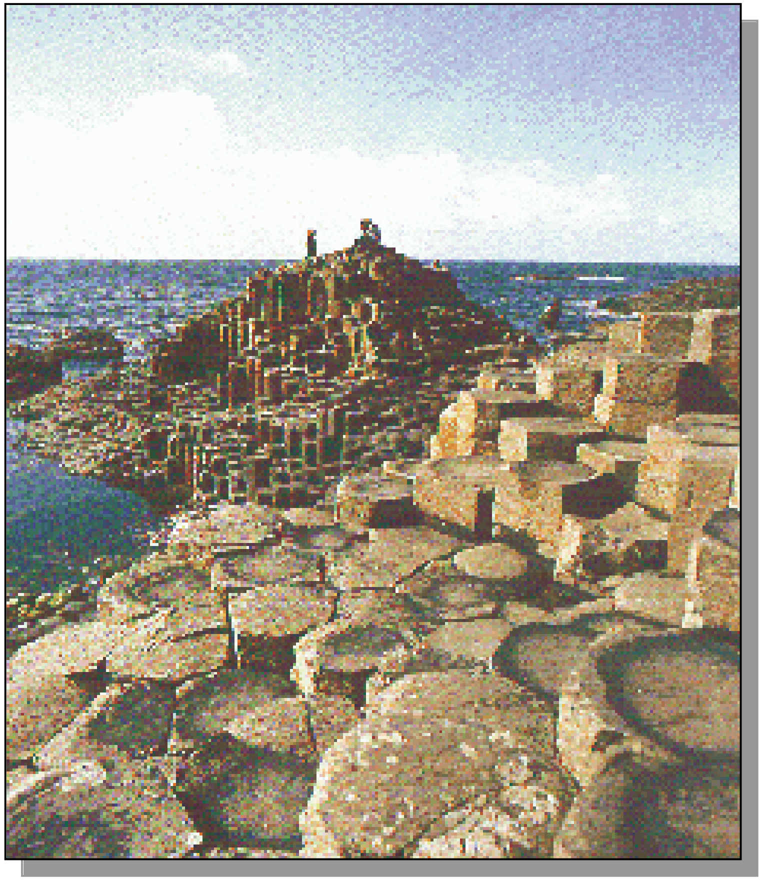


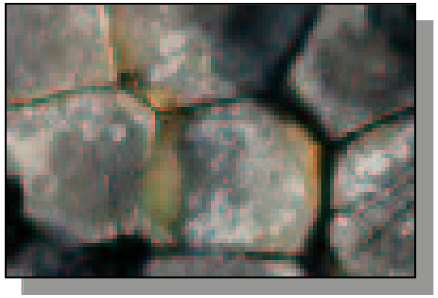


Finite volumes – volumes and surfaces

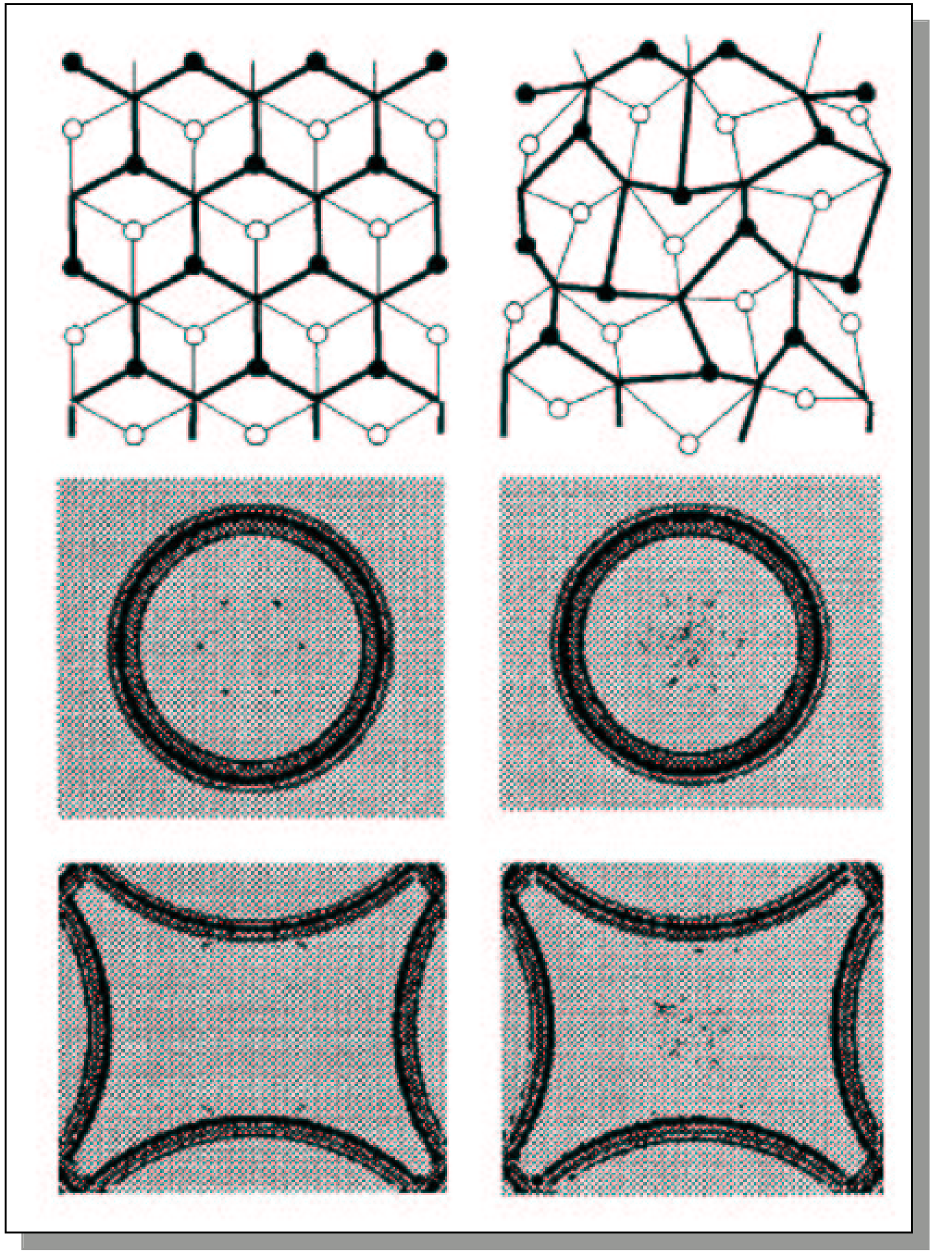


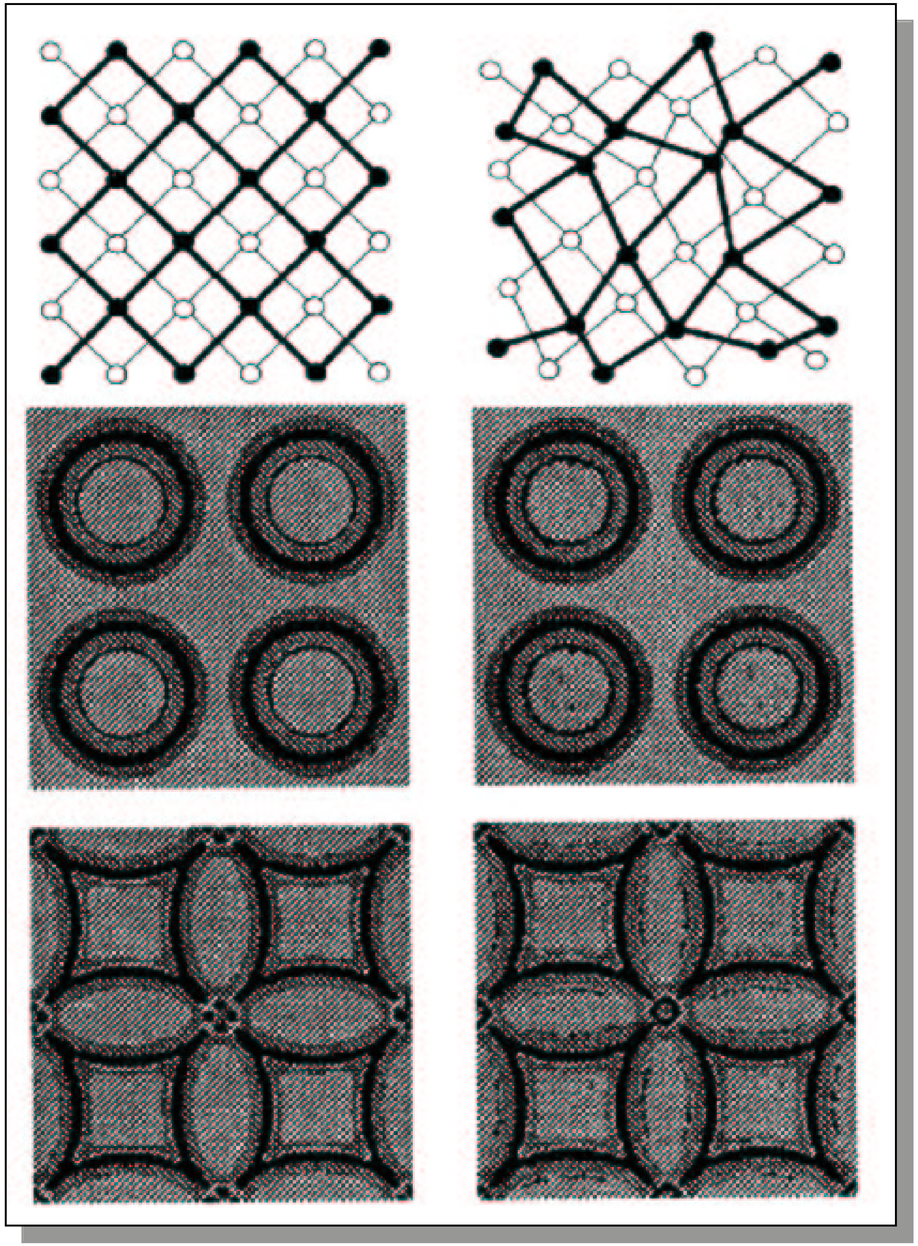
Voronoi Cells in Nature



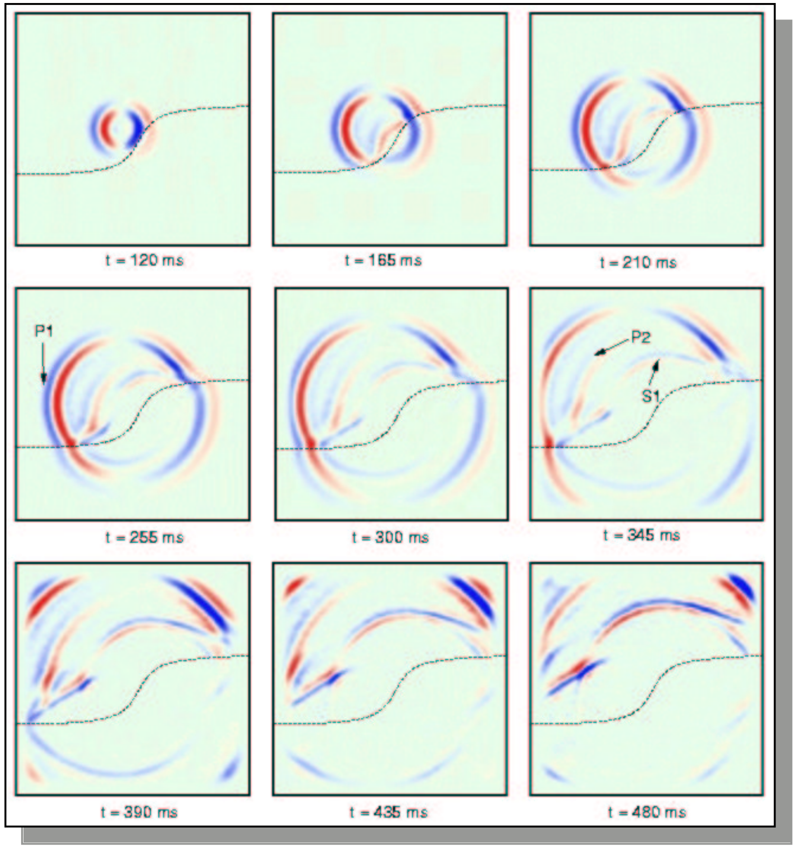


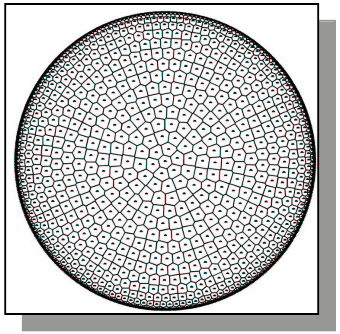
Finite volumes – wave propagation

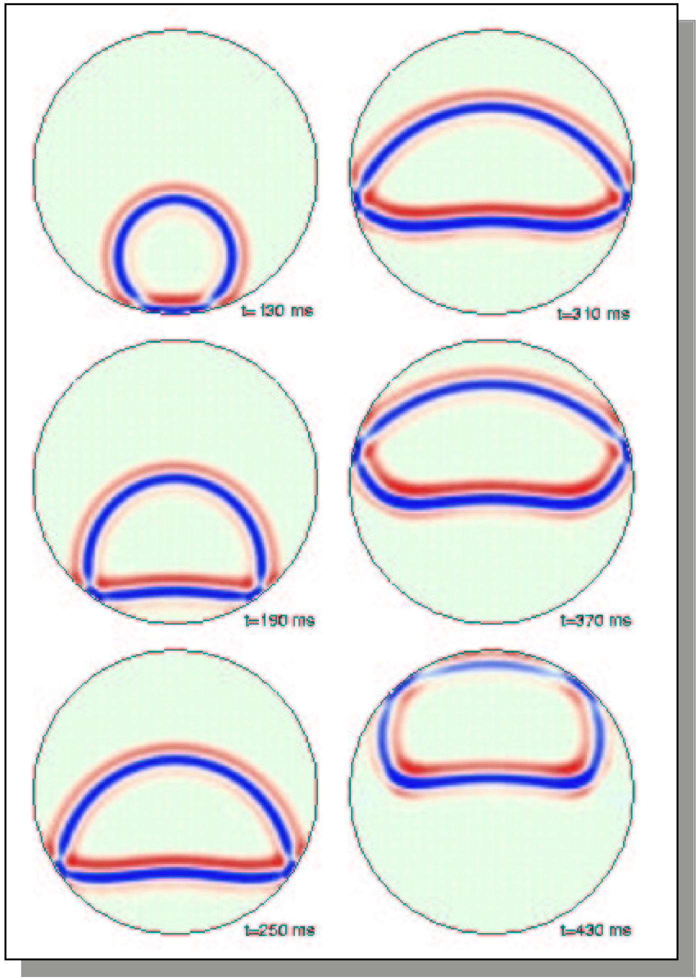




Waves with finite volumes







Käser, Igel, Sambridge, Braun, 2001
Käser and Igel, 2001

Finite volumes: summary

The **finite volume method** is an elegant approach to solving partial differential equations on unstructured grids.

The finite volume method is based on a discretization of **Gauss´ theorem**.

The FV method is frequently applied to **flow problems**. High-order approaches have been recently developed.

The FV method requires the calculation of volumes and surfaces for each cell. This may involve the calculation of **Voronoi cells** and **triangulation**.

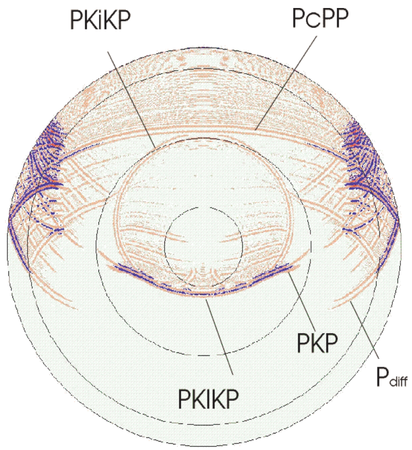
Numerical methods – current challenges

Most numerical methods have been applied to problems in **Earth sciences**, but …

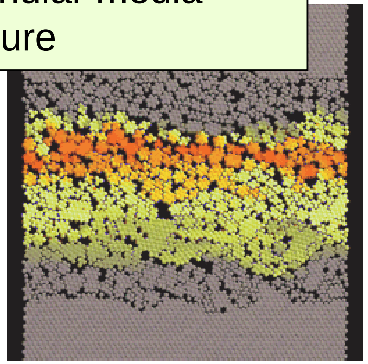
- … often there is not one particular method that solves all problems with the same efficiency …
- … there are still problems when complex shapes are involved (grid generation) …
- … often it is useful to combine the "good" properties of various methods (e.g. FE with FV, pseudospectral methods with FE, etc.) for specific applications …
- … for realistic problems the methods need to work well on parallel computers …

Numerical methods … in all fields of Earth sciences

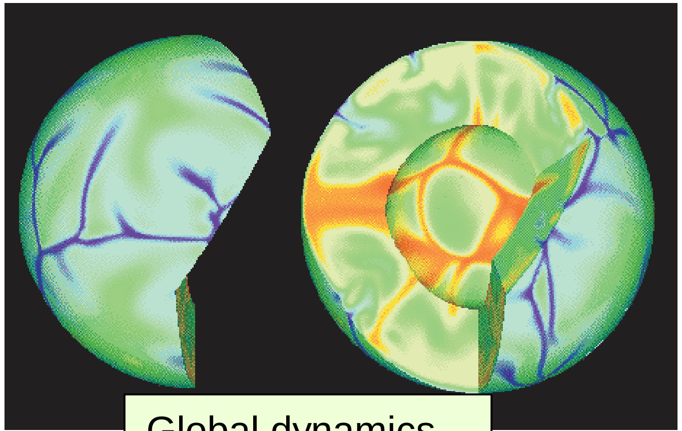
Seismology



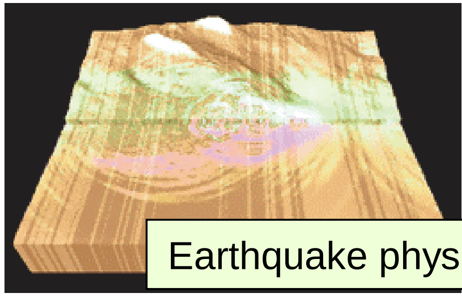
Granular media - rupture



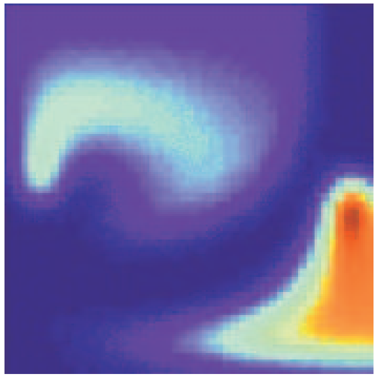
Global dynamics



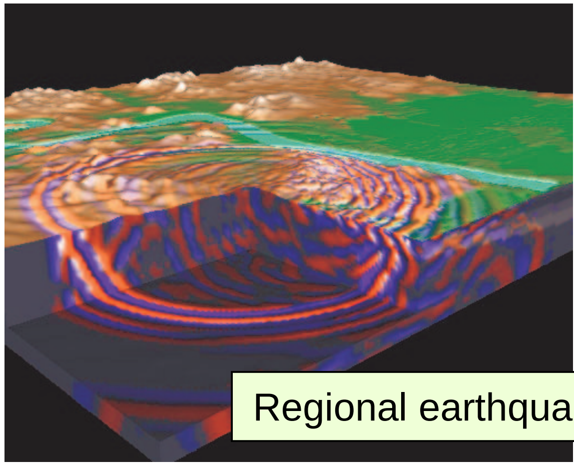
Earthquake physics



Mixing - Geochemistry



Regional earthquakes



Earth’s magnetic field