Finite Difference Approximations

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

- Simple geophysical partial differential equations
- Finite differences - definitions
- Finite-difference approximations to pde‘s
  - Exercises
    - Acoustic wave equation in 2D
    - Seismometer equations
    - Diffusion-reaction equation
- Finite differences and Taylor Expansion
- Stability -> The Courant Criterion
- Numerical dispersion

Partial Differential Equations in Geophysics

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

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

p	pressure
c	acoustic wave speed
s	sources

The acoustic wave equation

- seismology
- acoustics
- oceanography
- meteorology

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

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

Diffusion, advection, Reaction

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

Numerical methods: properties

Finite differences →

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

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

Finite elements →

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

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

Finite volumes →

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

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

Other numerical methods

Particle-based methods

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

-> very heterogeneous problems, nonlinear problems

Boundary element methods

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

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

***Pseudo*spectral methods**

- orthogonal basis functions, special case of FD
- spectral accuracy of space derivatives
- wave propagation, ground penetrating radar

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

What is a finite difference?

Common definitions of the derivative of f(x):

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

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

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

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

But we want dx to remain **FINITE**

What is a finite difference?

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

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

forward difference

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

backward difference

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

centered difference

The **big** question

How good are the FD approximations?

$$\approx \neq =$$

This leads us to Taylor series....

Taylor Series

Taylor series are expansions of a function f(x) for some finite distance dx to f(x+dx)

$$f(x \pm dx) = f(x) \pm \mathrm{dx} f^{'}(x) + \frac{dx^2}{2!} f^{''}(x) \pm \frac{dx^3}{3!} f^{'''}(x) + \frac{dx^4}{4!} f^{''''}(x) \pm \ldots$$

What happens, if we use this expression for

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

?

Taylor Series

... that leads to :

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

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

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:

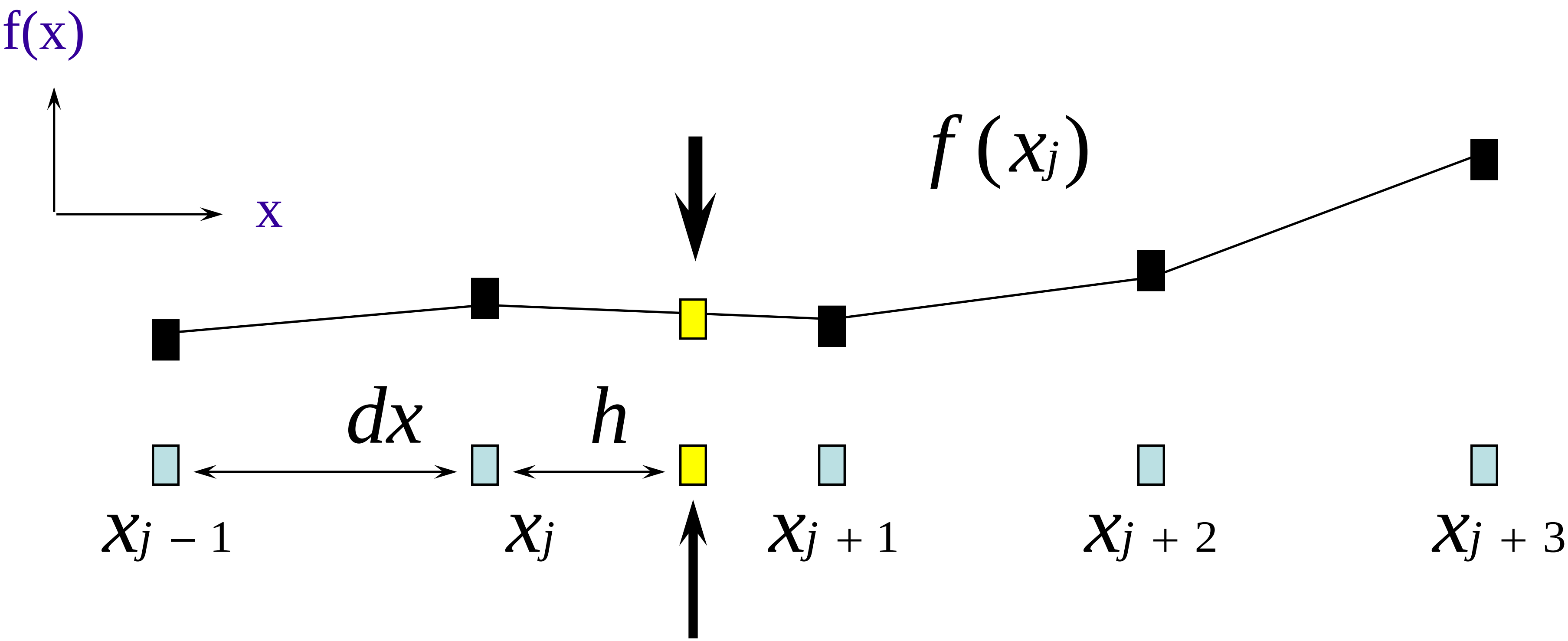
$$\frac{f(x+dx/2)-f(x-dx/2)}{dx}=\frac{1}{dx}\left[dx f^{'}(x)+\frac{dx^3}{3!}f^{'''}(x)+\ldots\right]$$

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

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!

Alternative Derivation



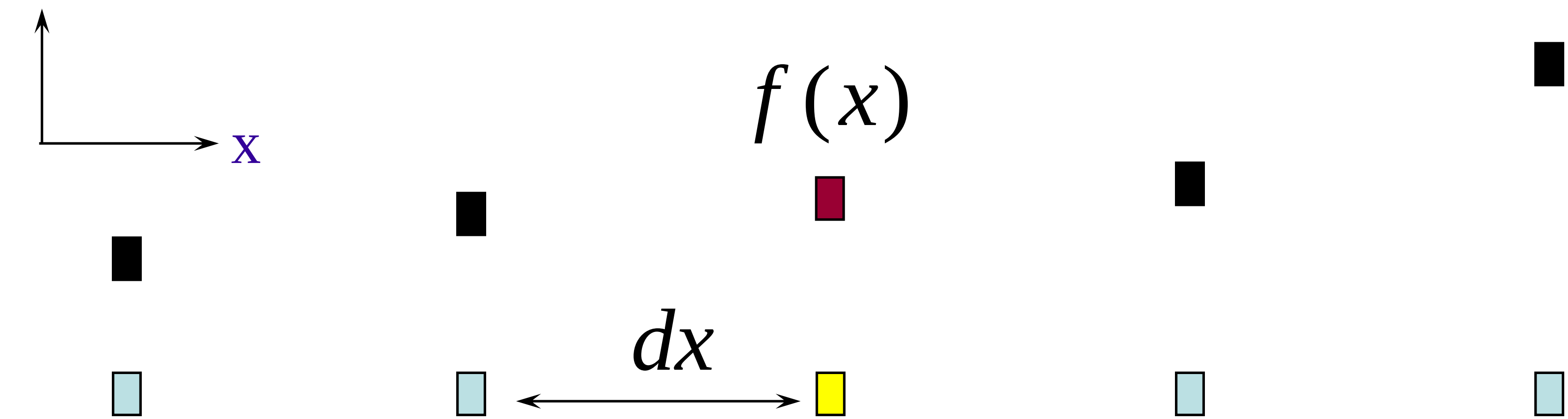
desired x location

What is the (approximate) value of the function or its (first, second ..) derivative at the desired location ?

How can we calculate the weights for the neighboring points?

Lets' try Taylor's Expansion

f(x)



$$f(x+dx) \approx f(x) + f'(x)dx \qquad (1)$$

$$f(x-dx) \approx f(x) - f'(x)dx \qquad (2)$$

we are looking for something like

$$f^{(i)}(x) \approx \sum_{j=1,L} w_j^{(i)} f(x_{index(j)})$$

2nd order weights

deriving the second-order scheme …

$$af^{+} \approx af + af' dx$$

$$bf^{-} \approx bf - bf' dx$$

$$\Rightarrow af^{+} + bf^{-} \approx (a+b) f + (a-b) f' dx$$

the solution to this equation for a and b leads to
a system of equations which can be cast in matrix form

Interpolation

$$a + b = 1$$

$$a - b = 0$$

Derivative

$$a + b = 0$$

$$a - b = 1/dx$$

Taylor Operators

... in matrix form ...

Interpolation	Derivative
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 0 \\ 1/dx \end{pmatrix}$

... so that the solution for the *weights* is ...

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{-1}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{-1}\begin{pmatrix} 0 \\ 1/dx \end{pmatrix}$$

Interpolation and difference weights

... and the result ...

Interpolation

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$$

Derivative

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{2dx}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Can we generalise this idea to longer operators?

Let us start by extending the Taylor expansion beyond f(x±dx):

Higher order operators

*a | $$f(x-2dx) \approx f - (2dx)f' + \frac{(2dx)^2}{2!} f'' - \frac{(2dx)^3}{3!} f'''$$

*b | $$f(x-dx) \approx f - (dx)f' + \frac{(dx)^2}{2!} f'' - \frac{(dx)^3}{3!} f'''$$

*c | $$f(x+dx) \approx f + (dx)f' + \frac{(dx)^2}{2!} f'' + \frac{(dx)^3}{3!} f'''$$

*d | $$f(x+2dx) \approx f + (2dx)f' + \frac{(2dx)^2}{2!} f'' + \frac{(2dx)^3}{3!} f'''$$

... again we are looking for the coefficients a,b,c,d with which the function values at x±(2)dx have to be multiplied in order to obtain the interpolated value or the first (or second) derivative!

... Let us add up all these equations like in the previous case ...

Higher order operators

$$af^{--} + bf^{-} + cf^{+} + df^{++} \approx$$

$$f(a+b+c+d)+$$

$$dxf'(-2a-b+c+2d)+$$

$$dx^2 f''(2a+\frac{b}{2}+\frac{c}{2}+2d)+$$

$$dx^3 f'''(-\frac{8}{6}a-\frac{1}{6}b+\frac{1}{6}c+\frac{8}{6}d)$$

... we can now ask for the coefficients a,b,c,d, so that the left-hand-side yields either f,f',f'',f''' ...

Linear system

... if you want the interpolated value ...

$$a + b + c + d = 1$$

$$-2a - b + c + 2d = 0$$

$$2a + \frac{b}{2} + \frac{c}{2} + 2d = 0$$

$$-\frac{8}{6}a - \frac{1}{6}b + \frac{1}{6}c + \frac{8}{6}d = 0$$

... you need to solve the matrix system ...

High-order interpolation

… Interpolation …

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 1 & 2 \\ 2 & 1/2 & 1/2 & 2 \\ -8/6 & -1/6 & 1/6 & 8/6 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

… with the result after inverting the matrix on the lhs …

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} -1/6 \\ 2/3 \\ 2/3 \\ -1/6 \end{pmatrix}$$

First derivative

… first derivative …

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -2 & -1 & 1 & 2 \\ 2 & 1/2 & 1/2 & 2 \\ -8/6 & -1/6 & 1/6 & 8/6 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 1/dx \\ 0 \\ 0 \end{pmatrix}$$

… with the result …

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \frac{1}{2dx}\begin{pmatrix} 1/6 \\ -4/3 \\ 4/3 \\ -1/6 \end{pmatrix}$$

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:

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

Problems: Stability

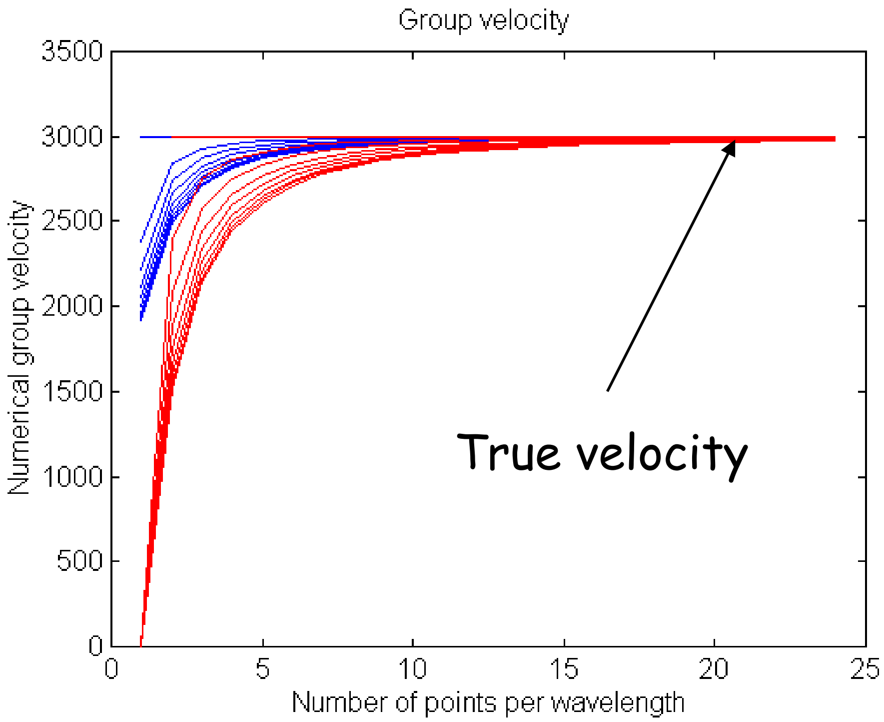
$$p(t+dt) = \frac{c^2 dt^2}{dx^2}\left[p(x+dx) - 2p(x) + p(x-dx)\right] + 2p(t) - p(t-dt) + s\,dt^2$$

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. (Derivation on the board).

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

Problems: Dispersion

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



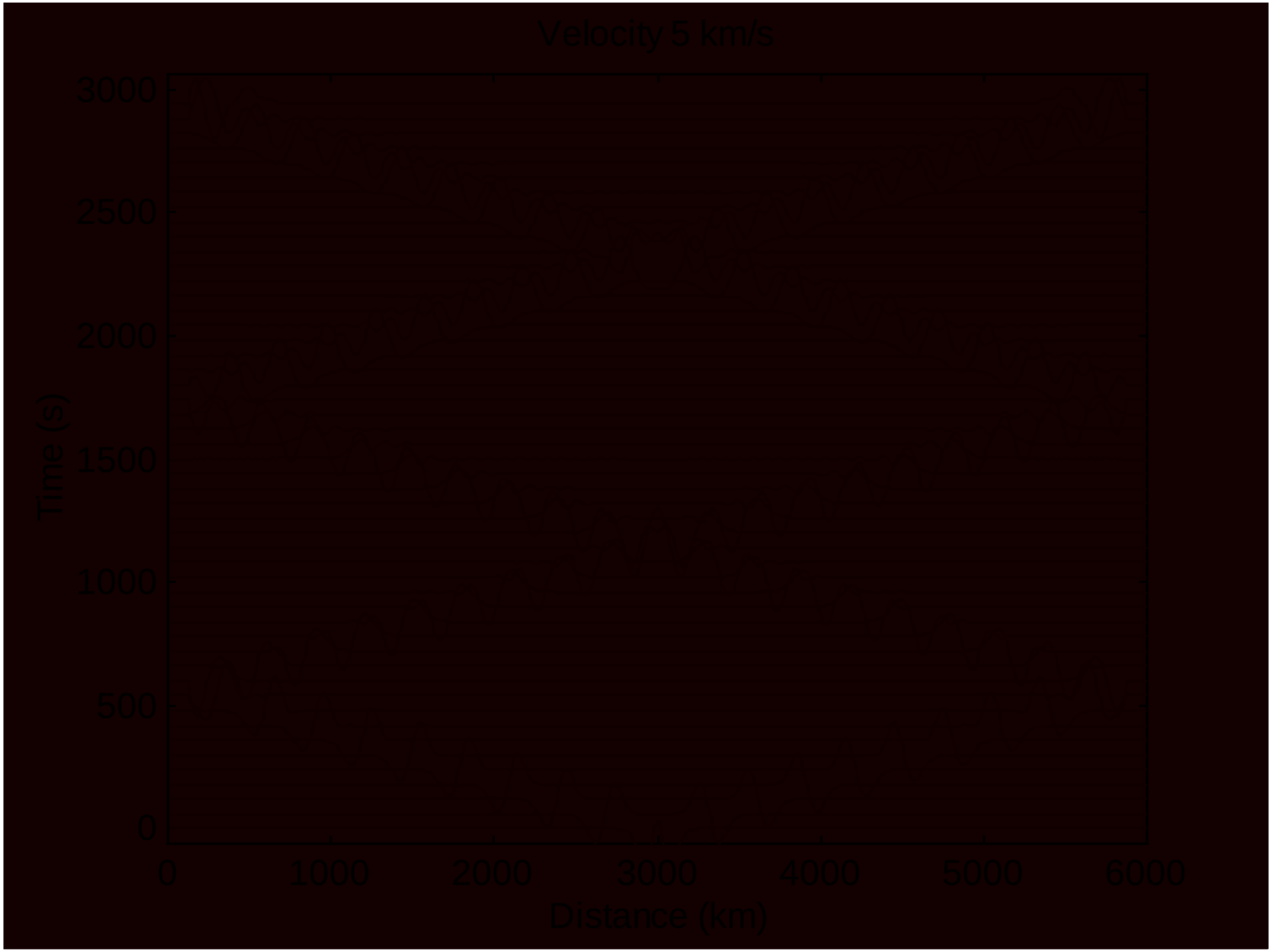
Dispersion: The numerical approximation has artificial dispersion, in other words, the wave speed becomes frequency dependent (Derivation in the board). You have to find a frequency bandwidth where this effect is small. The solution is to use a sufficient number of **grid points per wavelength.**

Our first FD code!

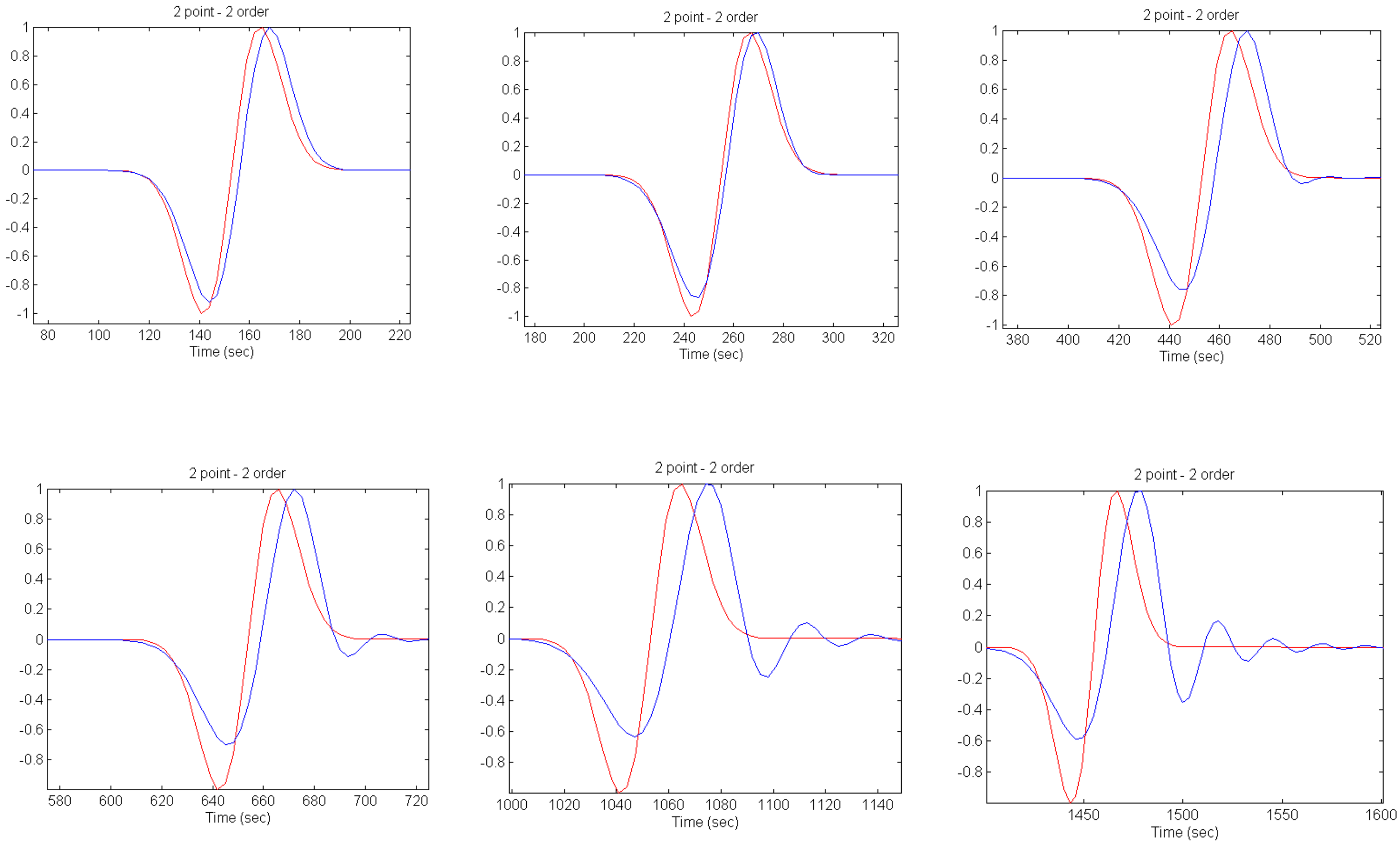
$$p(t+dt)=\frac{c^2dt^2}{dx^2}\left[p(x+dx)-2p(x)+p(x-dx)\right]$$
$$+2p(t)-p(t-dt)+sdt^2$$

```
% 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