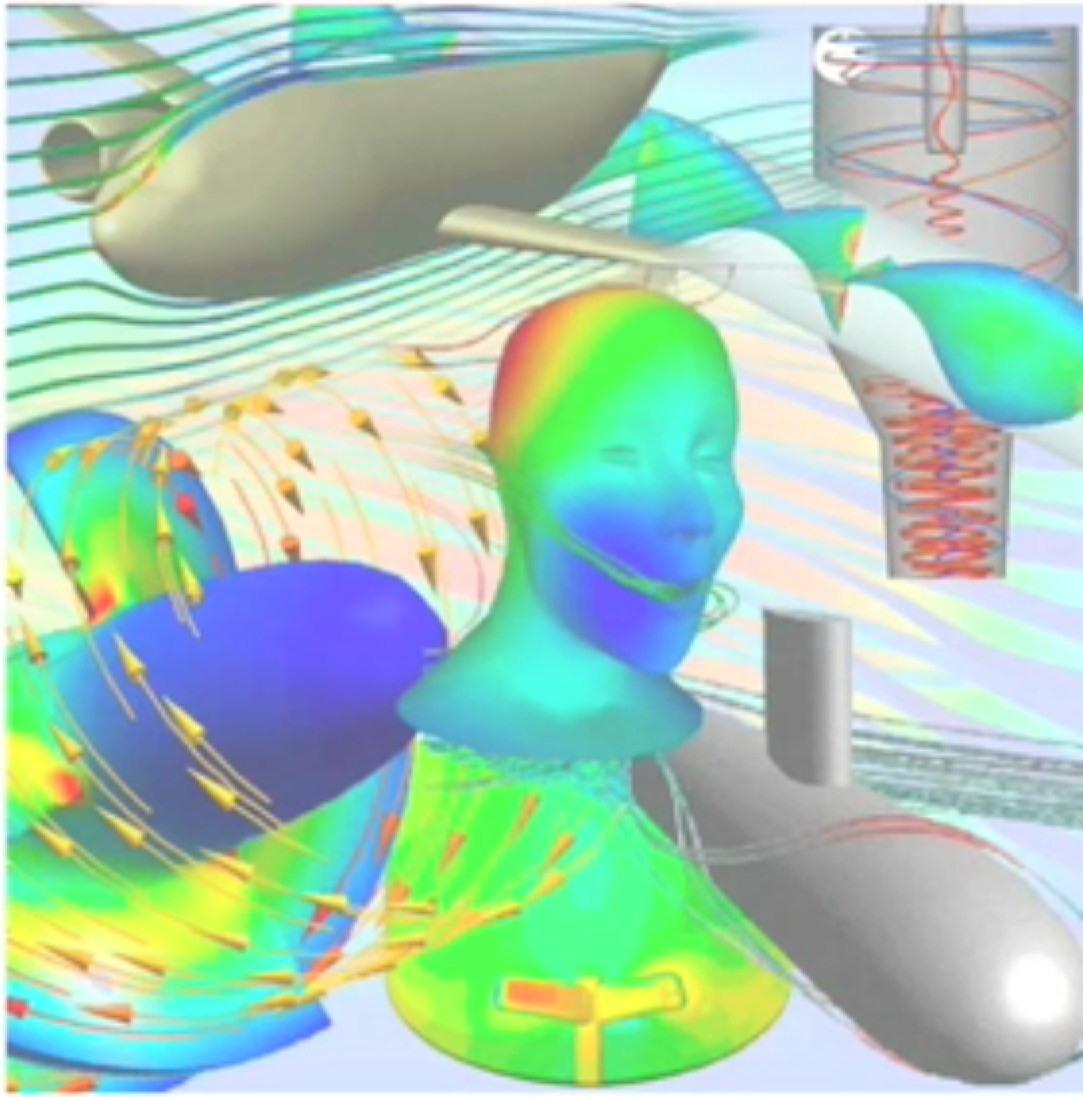
ME 702

Computational Fluid Dynamics, CFD

Lecture 9

Schemes for hyperbolic eqns.



BU College of Engineering

Spectral analysis of numerical errors (cont.)

- … for convection

  - Lax-Friedrichs / Lax-Wendroff / Leapfrog

Lax-Friedrichs: considerable numerical diffusion

Leapfrog: "double solution"

Lax-Wendroff / Leapfrog: little numerical diffusion but oscillations appear near "non-smooth" spots

- Spectral Analysis
  - * Error in amplitude $\epsilon_D = \dfrac{|G|}{|\tilde{G}|} \rightarrow$ diffusion error
  - * Error in the phase $\epsilon_\phi = \Phi / \tilde{\Phi} \rightarrow$ dispersion errors

Hyperbolic problem: $\epsilon_D = |G|$, $\epsilon_\phi = \dfrac{C_{num}}{C} = \dfrac{\tan^{-1}(-\mathrm{Im}(G)/\mathrm{Re}(G))}{\sigma\phi}$

Looked at plots of $\epsilon_D, \epsilon_\phi$ vs $\phi \ (= k\Delta x)$, with σ as parameter

▸ Summary ...

1st order upwind ϵ_D decreases away from 1 quickly with increasing frequencies.

ϵ_ϕ < 1 (lagging) for $\sigma > 0.5$
> 1 (leading) for $\sigma < 0.5$
$= 1$ for $\sigma = 0.5$

Lax-Friedrichs : ϵ_D shows strong damping for smaller σ
no damping $\phi = \pi$ $(\lambda = 2\Delta x)$
$\epsilon_\phi > 1 \rightarrow$ leading phase error

Lax-Wendroff : ϵ_D shows larger "accurate region", where $\epsilon_D \approx 1$
ϵ_ϕ mostly $< 1 \rightarrow$ lagging

Leapfrog (pending)

- Leapfrog scheme

Note that $|G| = 1 \Rightarrow$ No diffusion error

Leapfrog scheme is particularly useful for long-term simulations
e.g weather forecast codes

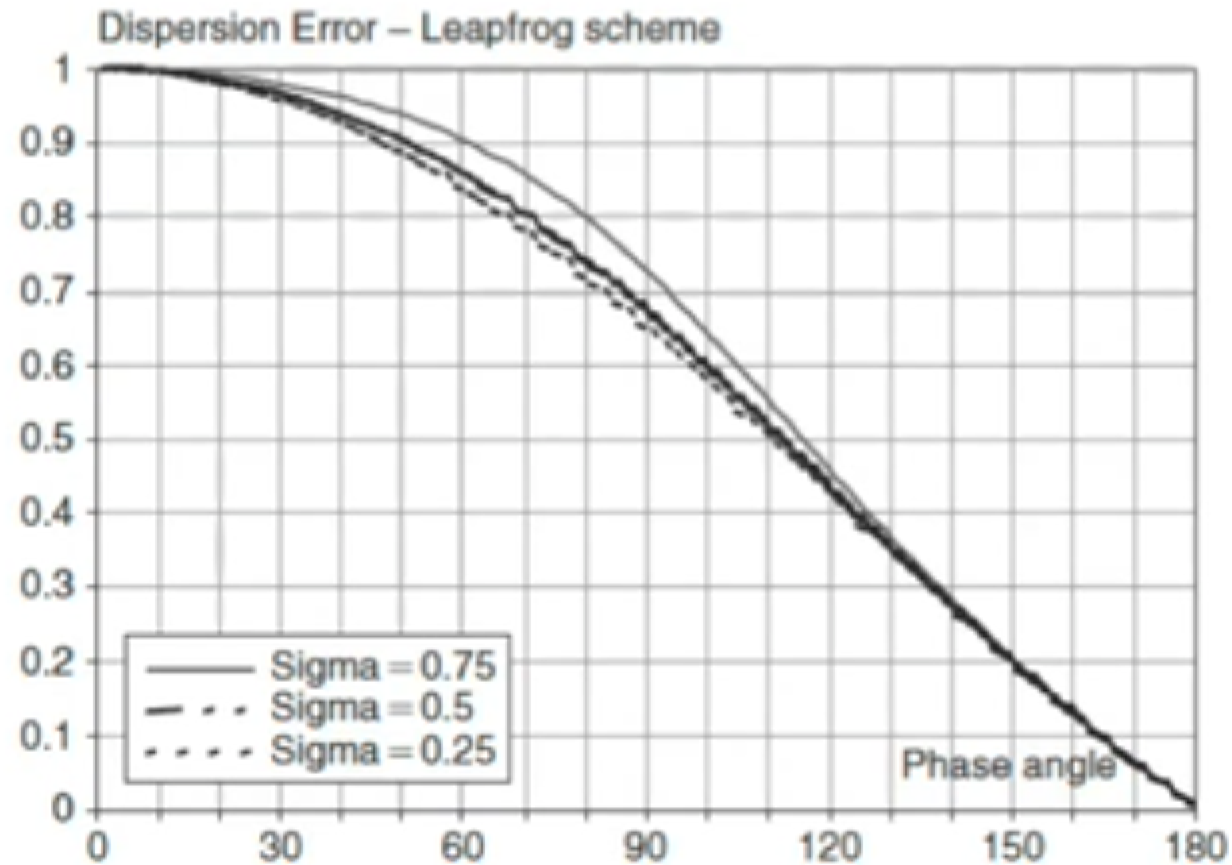
* Dispersion error

$$\epsilon_\phi = \pm \frac{\tan^{-1}\left[(\sigma \sin\phi)/\sqrt{1-\sigma^2\sin^2\phi}\right]}{\sigma\phi} = \pm \frac{\sin^{-1}(\sigma\sin\phi)}{\sigma\phi}$$

- Leapfrog scheme → accurate results for $u(x)$ smooth
  - * amplitudes correctly modeles
  - * low frequencies, the phase error close to 1

ϵ_ϕ mostly < 1 → lagging

Neutral stability
$|G| = 1 \quad \forall \sigma < 1$
some problems
* high frequency errors NOT damping



Leapfrog
Unstable for Burgers eqn !!

NOT good for high speed flows where shocks can occur.

- *Note on the oscillations ...* (LW + Leapfrog)

— Have not explained the origin of oscillations

Why do they occur behind the traveling wave?

- oscillations have frequency
- ϵ_ϕ for LW & Leapfrog predominantly < 1 (especially at higher frequencies)

  → convection speed of errors SLOWER than physical one

Leapfrog $\epsilon_\phi \to 0$ for $\phi \to \pi$ and so oscillations are stronger.

Consider an alternative scheme, due to Beam & Warming:

Recall derivation of LW ....

$$u_i^{n+1} = u_i^n - c\Delta t\,(u_x)_i + c^2\frac{\Delta t^2}{2}(u_{xx})_i + O(\Delta t^3)$$

Then discretize using BD (upwind) ....

④ — Analysis of *Beam-Warming* (use BD instead of CD of LW)

$$u_i^{n+1} = u_i^n - \frac{\sigma}{2}\left(3u_i^n - 4u_{i-1}^n + u_{i-2}^n\right) + \frac{\sigma^2}{2}\left(u_i^n - 2u_{i-1}^n + u_{i-2}^n\right)$$

Stability gives:

$$G = 1 - I\sigma\left[1 + 2(1-\sigma)\sin^2(\phi/2)\right]\sin\phi - 2\sigma\left[1-(1-\sigma)\cos\phi\right]\sin^2(\phi/2)$$

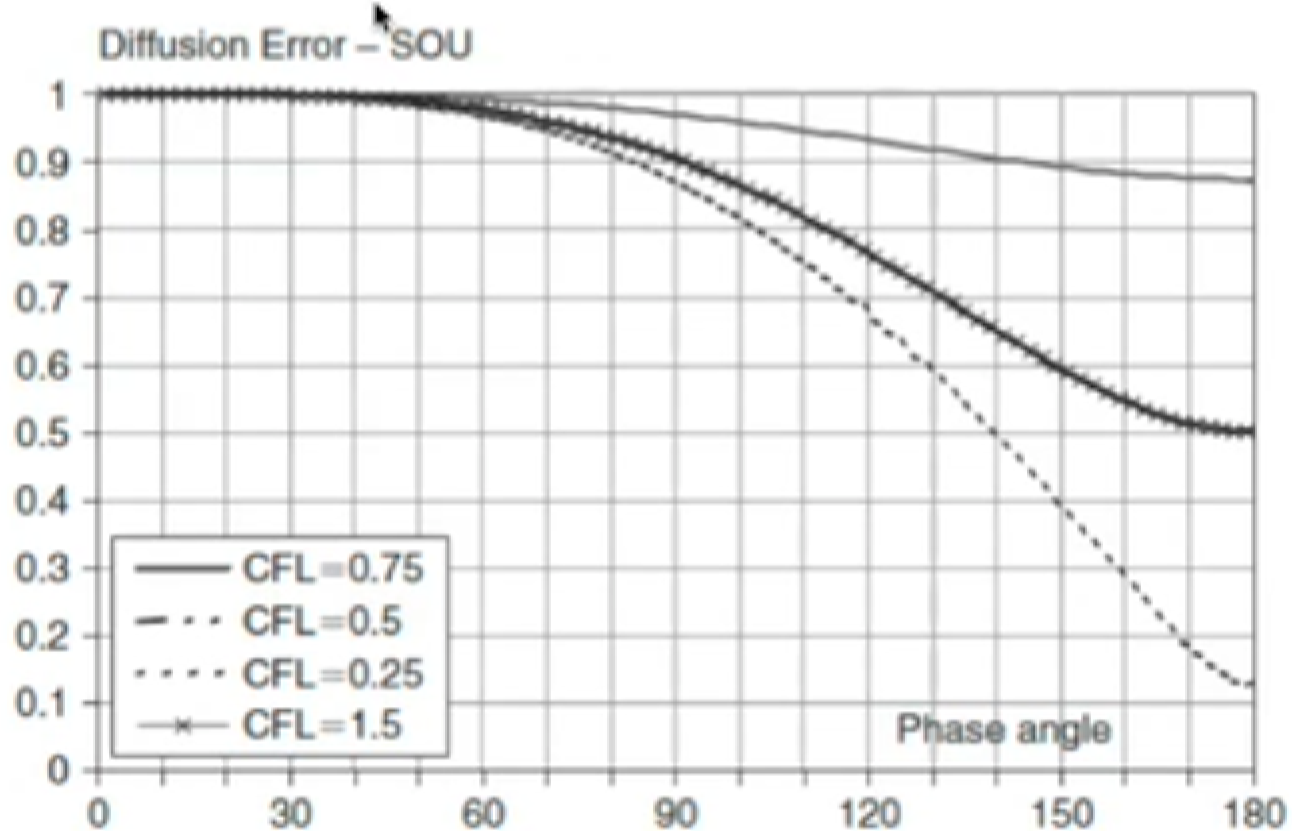
→ conditionally stable for $0 \leq \sigma \leq 2$

* Diffusion error:

$$\varepsilon_D = |G| = \sqrt{1 - \sigma(1-\sigma)^2(2-\sigma)(1-\cos\phi)^2}$$

ε_ϕ ... more messy See plots!

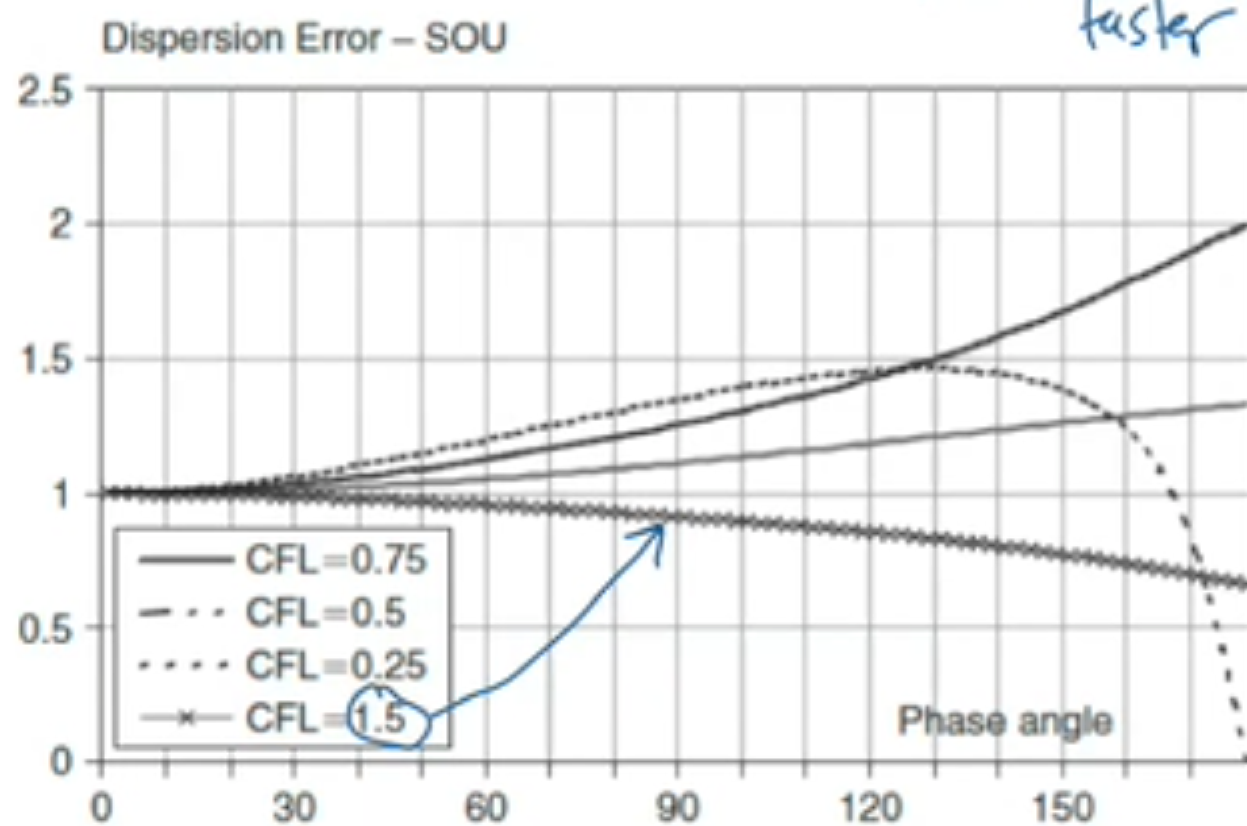
④ — Analysis of *Beam-Warming*



④ — Analysis of *Beam-Warming*

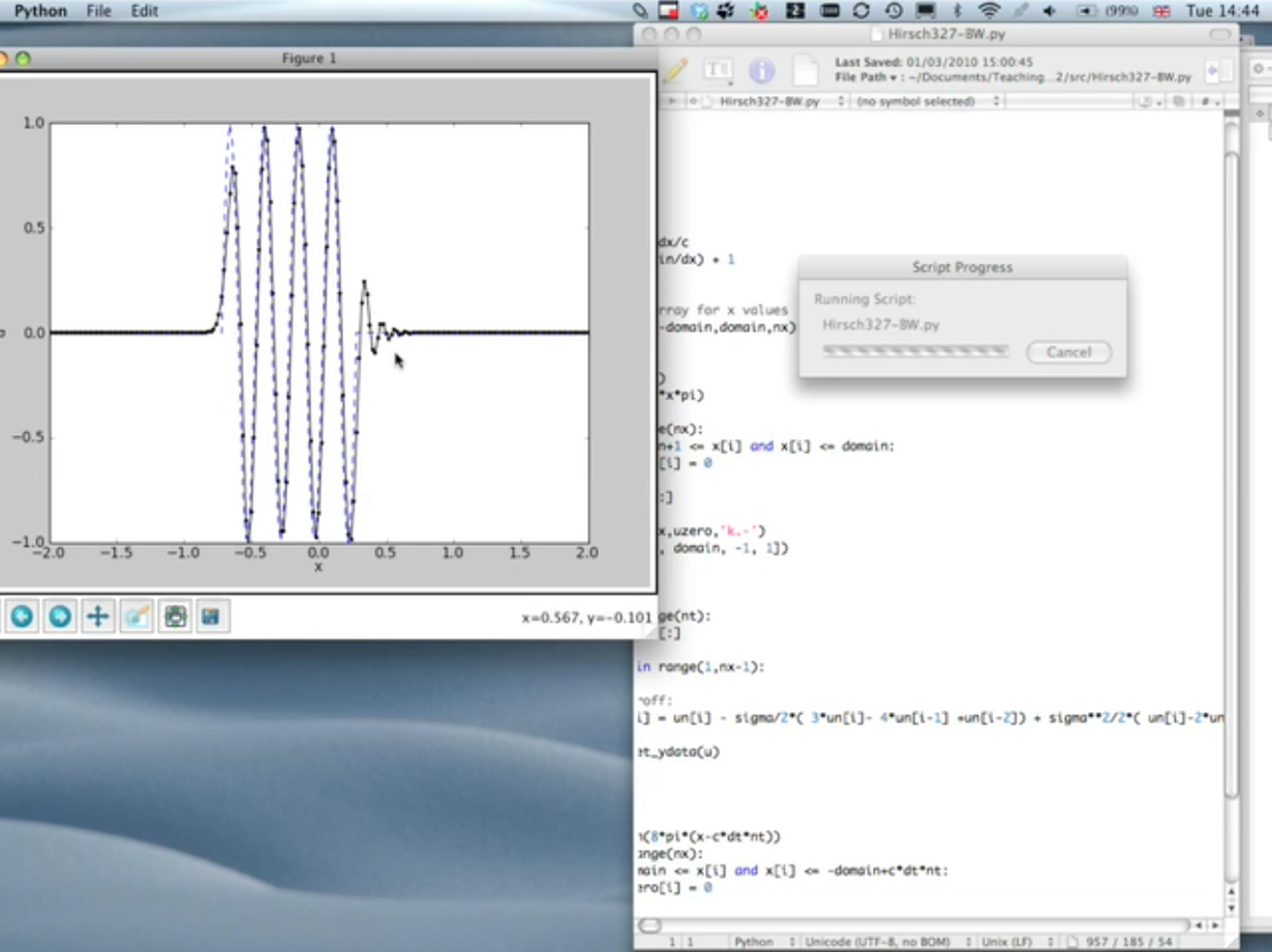
ϵ_ϕ predominantly > 1 for $\sigma < 1$

Numerical solution will move faster $\rightarrow$ oscillations move **ahead** of the soln.



$\uparrow \sigma < 1$

$\leftarrow \sigma > 1$


Python File Edit
Tue 14:44
Figure 1
Hirsch327-BW.py
Last Saved: 01/03/2010 15:00:45
File Path ▾ : ~/Documents/Teaching...2/src/Hirsch327-BW.py
Hirsch327-BW.py
(no symbol selected)
1.0
0.5
0.0
−0.5
−1.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x
x=0.567, y=-0.101
Script Progress
Running Script:
Hirsch327-BW.py
Cancel
dx/c
in/dx) + 1
rray for x values
-domain,domain,nx)
*x*pi)
e(nx):
n+1 <= x[i] and x[i] <= domain:
[i] = 0
:]
x,uzero,'k.-')
, domain, -1, 1])
ge(nt):
[:]
in range(1,nx-1):
off:
i] = un[i] - sigma/2*(3*un[i]- 4*un[i-1] +un[i-2]) + sigma**2/2*(un[i]-2*un
et_ydata(u)
n(8*pi*(x-c*dt*nt))
nge(nx):
main <= x[i] and x[i] <= -domain+c*dt*nt:
ero[i] = 0
Python
Unicode (UTF-8, no BOM)
Unix (LF)
957 / 185 / 54

TextWrangler File Edit Text View Search Window #! Help Tue 14:45

Hirsch327-BW.py

Last Saved: 01/03/2010 15:00:45
File Path ▾ : ~/Documents/Teaching...2/src/Hirsch327-BW.py

Hirsch327-BW.py (no symbol selected)

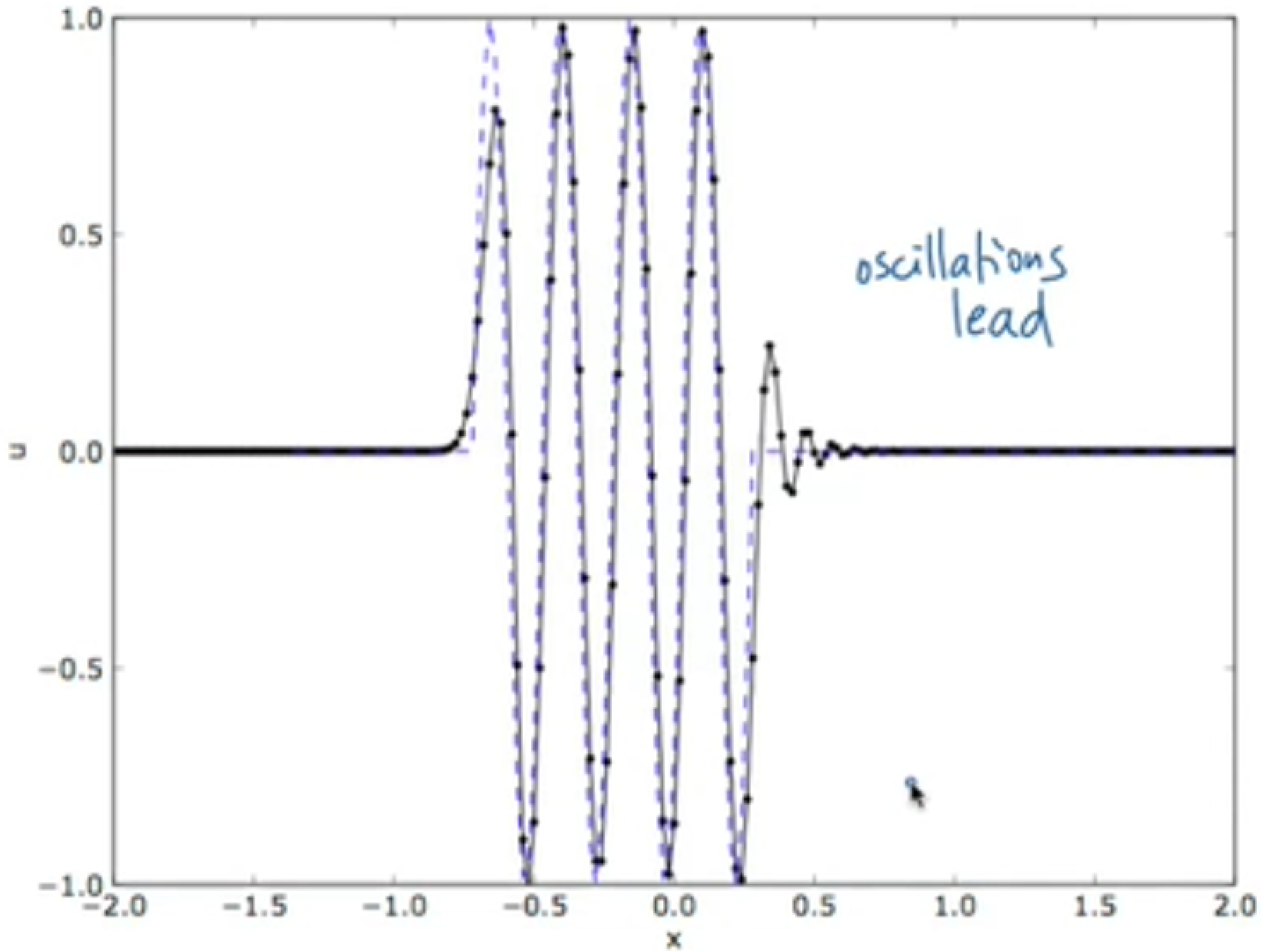
```python
from pylab import *

ion()

domain = 2.0
nt = 80
sigma = 0.8
c  = 1.0
dx = 0.01
dt = sigma * dx/c
nx = int(domain/dx) + 1
print nx

# create an array for x values
x = linspace(-domain,domain,nx)

u = zeros(nx)
un = zeros(nx)
uzero = sin(8*x*pi)

for i in range(nx):
    if -domain+1 <= x[i] and x[i] <= domain:
        uzero[i] = 0

u[:] = uzero[:]

line, = plot(x,uzero,'k.-')
axis([-domain, domain, -1, 1])
xlabel('x')
ylabel('u')

for it in range(nt):
    un[:] = u[:]

    for i in range(1,nx-1):

# Lax-Wendroff:
        u[i] = un[i] - sigma/2*( 3*un[i]- 4*un[i-1] +un[i-2]) + sigma**2/2*(

    line.set_ydata(u)
    draw()


hold(True)
uzero = sin(8*pi*(x-c*dt*nt))
for i in range(nx):
```

1 | 1 Python Unicode (UTF-8, no BOM) Unix (LF) 957 / 185 / 54

④ — Beam-Warming

(Test #3)

$$\sigma = 0.8$$
$$\Delta x = 0.02$$
$$n_{\text{steps}} = 80$$



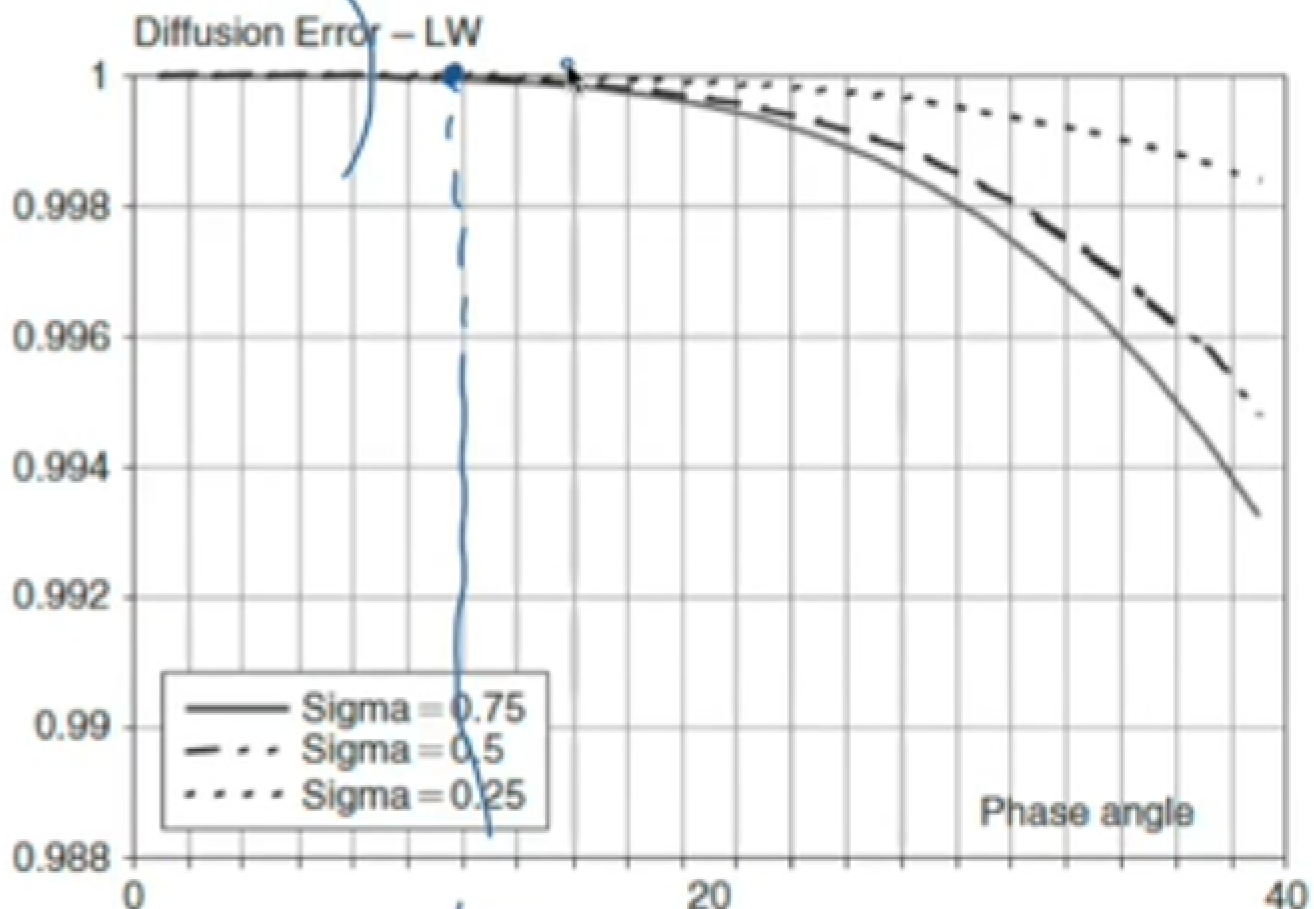
- *Final notes*
  - 1st order schemes ... generate large errors
  - 2nd order schemes ... acceptable errors BUT should be careful especially at higher frequencies

Look at plot ϵ_D for LW, establish a phase angle limit

ϕ_{lim} want $\epsilon_D \approx 1$

- zoom-in

LW scheme



$\phi \approx 10$

- *Final notes*
  - 1st order schemes ...
  - 2nd order schemes ... LW

$$\phi_{lim} \sim 10^\circ \sim \frac{\pi}{18} \rightarrow \text{for } \epsilon \approx 1$$

Key quantity : NUMBER OF MESH POINTS PER WAVELENGTH

$$N_\lambda = \frac{\lambda}{\Delta x}$$

We require $\phi = k\Delta x = \frac{2\pi}{\lambda} \cdot \Delta x \leq \phi_{lim}$ phase angle limit

or $\boxed{N_\lambda = \frac{\lambda}{\Delta x} \geq \frac{2\pi}{\phi_{lim}}}$ severe

$\phi_{lim} = \frac{\pi}{18} \Rightarrow$ at least 36 points / wavelength

$\phi_{lim} = \frac{\pi}{12} \Rightarrow 24$ pts / wavelength (safer)

Test#3 only 12 points / wavelength

- Second-order one-sided differences

Finite formulas for $\partial u/\partial x|_i$ of higher order, requiring increasing number of mesh points

E.g. Use upwind points $(i-2), (i-1), i$

can write: $$(u_x)_i = \frac{a u_i + b u_{i-1} + c u_{i-2}}{\Delta x} + O(\Delta x^2)$$

Taylor expansions

$$(1) \quad u_{i-2} = u_i - 2\Delta x (u_x)_i + \frac{(2\Delta x)^2}{2}(u_{xx})_i - \frac{(2\Delta x)^3}{6}(u_{xxx})_i + \dots$$

$$(2) \quad u_{i-1} = u_i - \Delta x (u_x)_i + \frac{(\Delta x)^2}{2}(u_{xx})_i - \frac{(\Delta x)^3}{6}(u_{xxx})_i + \dots$$

Mult. (1) by c, (2) by b, add $a u_i$:

$$\boxed{a u_i + b u_{i-1} + c u_{i-2}} = \cancel{(a+b+c) u_i} - \Delta x (b+2c)(u_x)_i + \frac{\Delta x^2}{2}\cancel{(b+4c)}(u_{xx})_i + O(\Delta x^3)$$

$$\left.\begin{aligned} &\boxed{a+b+c=0} \quad !! \\ &b+2c=-1 \\ &b+4c=0 \end{aligned}\right\} \quad (u_x)_i = \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x} + O(\Delta x)^2$$

- Multi-step methods — FD schemes at split time levels
work well in NONLINEAR hyperbolic problems

Also called PREDICTOR - CORRECTOR
* 1st step, a "temporary value" for $u(x)$ is "predicted"
* 2nd step, a "corrected value" is computed

① — Richtmyer / Lax-Wendroff
{ variant 1 (Richtmyer) → at point i
{ variant 2 (2-step LW) → at point $i+1/2$

(V1) : Step 1 always LF method at time level $(n+1/2)$

$$\frac{u_i^{n+1/2} - \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right)}{\Delta t/2} = -c\,\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$$

Step 2 — Leapfrog (with $\Delta t/2$) Leapfrog $(u_i^{n+1} - u_i^{n-1})/(2\Delta t$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -c\,\frac{\left(u_{i+1}^{n+1/2} - u_{i-1}^{n+1/2}\right)}{2\Delta x}$$

Rearrange:
$$u_i^{n+1/2} = \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right) - \frac{c\Delta t}{4\Delta x}\left(u_{i+1}^n - u_{i-1}^n\right)$$

and
$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{2\Delta x}\left(u_{i+1}^{n+1/2} - u_{i-1}^{n+1/2}\right)$$

→ Stable for $\frac{c\Delta t}{\Delta x} \le 2$.

(V2) Step 1, LF on $i+1/2$

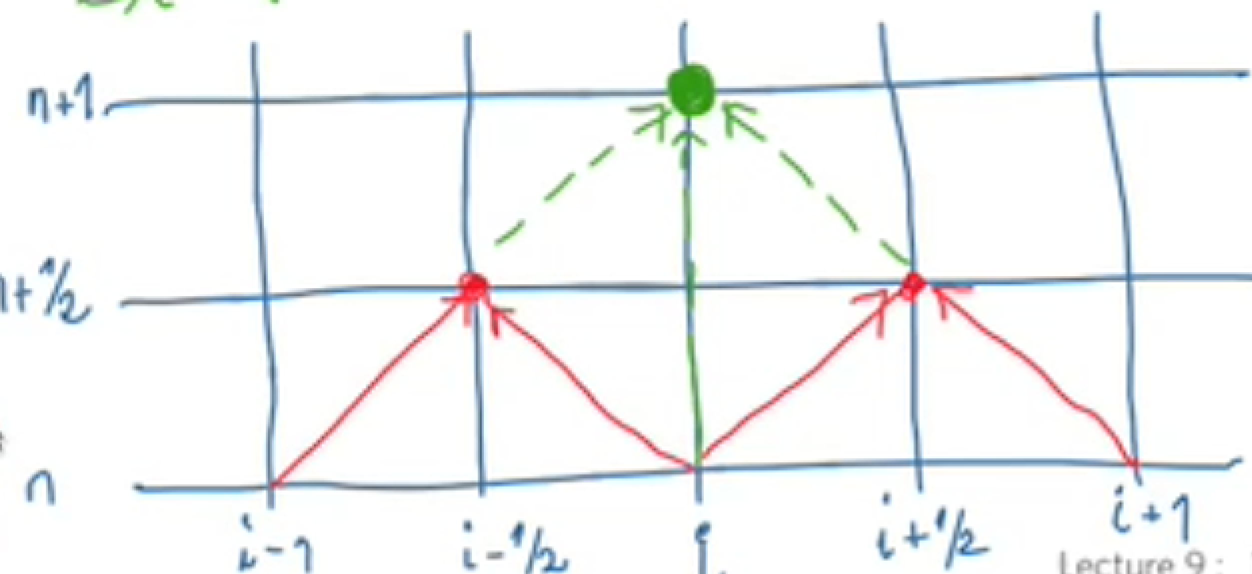
used $\Delta x/2$

$$u_{i+1/2}^{n+1/2} = \frac{1}{2}\left(u_{i+1}^n + u_{i-1}^n\right) - \frac{c\Delta t}{2\Delta x}\left(u_{i+1}^n - u_i^n\right)$$

and
$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_{i+1/2}^{n+1/2} - u_{i-1/2}^{n+1/2}\right)$$

→ Stable for $\frac{c\Delta t}{\Delta x} \le 1$

- 2nd order
- For LINEAR PDEs it is equivalent to single-step LW



② — MacCormack method

Step 1 uses FD scheme — call u^* the temporary solution

$$\frac{u_i^* - u_i^n}{\Delta t} = -c\,\frac{u_{i+1}^n - u_i^n}{\Delta x}$$

Step 2 uses BD scheme, with $\Delta t/2$

$$\frac{u_i^{n+1} - u_i^{n+1/2}}{\Delta t/2} = -c\,\frac{u_i^* - u_{i-1}^*}{\Delta x}$$

and replace the value $u_i^{n+1/2}$ by the average

$$u_i^{n+1/2} = \frac{1}{2}\left(u_i^n + u_i^*\right)$$

- 2nd order
- stability $\sigma < 1$
- For linear PDEs equivalent to LW!
- Can alternate FD/BD — BD/FD works well nonlinear

Predictor $\quad u_i^* = u_i^n - \frac{c\Delta t}{\Delta x}\left(u_{i+1}^n - u_i^n\right)$

Corrector $\quad u_i^{n+1} = \frac{1}{2}\left[\left(u_i^n + u_i^*\right) - \frac{c\Delta t}{\Delta x}\left(u_i^* - u_{i-1}^*\right)\right]$

Non-linear convection

- inviscid Burgers equation $$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x}$$

= Conservative Form =

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{u^2}{2}\right)$$

or $\frac{\partial u}{\partial t} = -\frac{\partial E}{\partial x}$, where $E = \frac{u^2}{2}$

Physical interpretation → a wave propagating with different speeds at different points

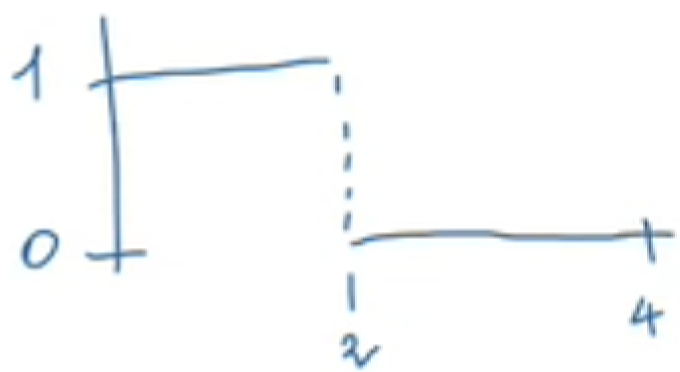
eventually → a shock will be formed

Non-linear convection

- inviscid Burgers equation $\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x}$

Practical Module — Nonlinear problems

test $u(x,0) = \begin{cases} 1 & 0 \le x \le 2 \\ 0 & 2 \le x \le 4 \end{cases}$



Investigate the propagation with

1. LF
2. LW
3. MacCormack

start $\Delta x = \Delta t = 0.1$

then change $\frac{\Delta t}{\Delta x} = 1$ to 0.5 → effect of Courant numer

change to a finer mesh → effect of step size