Plan for Today

ATMS 502
CSE 566

Tuesday,
9 April 2019

Class #23

Program #5 is due Tuesday, April 16

- 1) Review:
  - Staggered grids
  - Diffusion: exact vs. numerical amplitude
  - Advection-diffusion –
    - consistency, numerical Peclet number
    - small wavelengths: advection vs. diffusion
    - time differencing
- 2) Program 5, continued
- 3) Lee & Wilhelmson
  - Intro: Density currents, 3D model

Review from last class

Review: Staggered grid phase

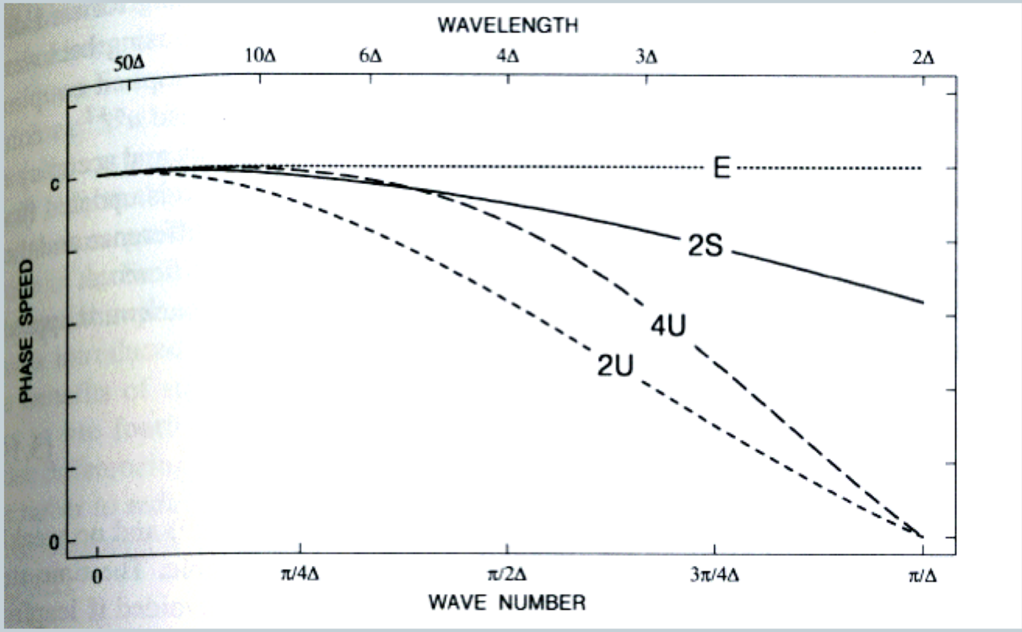
Phase speed analysis

$$c_{unstag} = \frac{\omega}{k} = \frac{c}{k\Delta x}\sin k\Delta x$$

unstaggered

$$c_{stag} = \frac{\omega}{k} = \frac{2c}{k\Delta x}\sin\frac{k\Delta x}{2}$$

staggered



- Staggered 2nd order better than 2nd *or 4th* order unstaggered.
- Group velocity: unstaggered 2x worse

Review: Diffusion: exact vs. FTCS

- Forward time, centered space: $$\frac{\phi_j^{n+1}-\phi_j^n}{\Delta t} = M\left(\frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{\Delta x^2}\right)$$
- Amplitude behavior: $$\lambda = 1 - 4\nu \sin^2\left(\frac{\beta}{2}\right); \quad \nu = M\Delta t / \Delta x^2$$
  - Damps 2Δx most
  - stable for 0< ν ≤ ½
    - But ν>¼ has λ<0 (sign flips) for 2Δx
    - To avoid 'over damping' use 0 < ν ≤ ¼

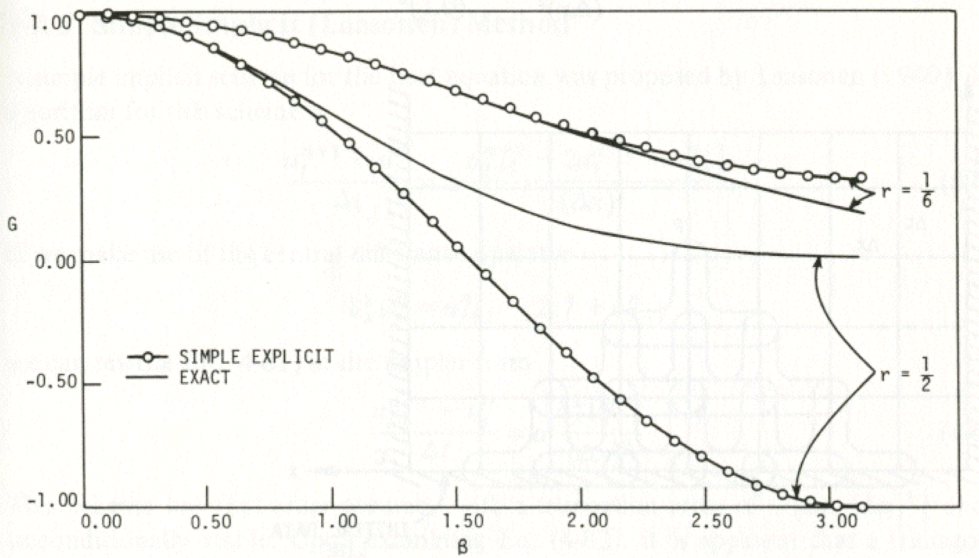


Figure 4-12 Amplification factor for simple explicit method.

ANDERSON ET AL. 1984

Review: Advection + diffusion

We are *trying* to solve:

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = M\frac{\partial^2 \phi}{\partial x^2}$$

- *Diffusion must dominate!*
  - Upstream advection, centered diffusion
    - Modified equation:

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = M\left(1+\frac{P_e}{2}\right)\frac{\partial^2 \phi}{\partial x^2} \quad \text{where } P_e = \frac{c\Delta x}{M}$$

  - Centered advection, centered diffusion
    - Modified equation:

$$\phi_t + c\phi_x = \left(M - \frac{c^2\Delta t}{2}\right)\phi_{xx} + ()\phi_{xxx} + ()\phi_{xxxx} + \ldots$$

- Summary:
  - The *Péclet number* must be small, or the solution is no longer consistent. Small P_e requires
    - small flow speed *c*
    - small grid spacing Δx, or
    - large damping coefficient *M*
      - M may be prognostic, and so vary throughout the flow.

French physicist Jean Claude Eugène Péclet

Review: Small wavelengths: Advection/diffusion

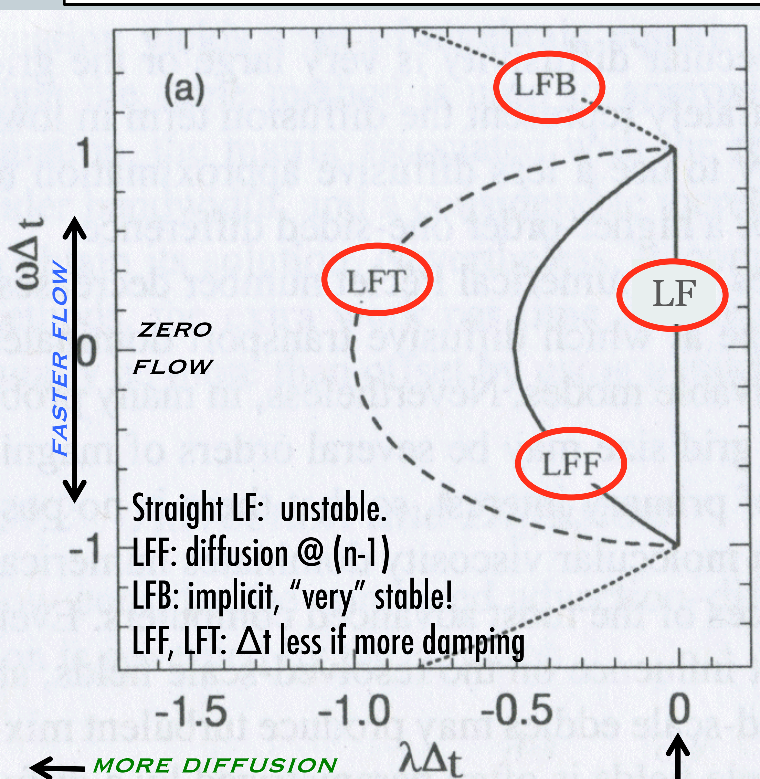
- General form: $$\frac{\phi_j^{n+1}-\phi_j^n}{\Delta t} = M\frac{\theta\delta_{xx}\phi_j^{n+1}+(1-\theta)\delta_{xx}\phi_j^n}{\Delta x^2} \text{ with } 0 \le \theta \le 1$$
  - $\theta=0$: simple explicit; $\theta=1$: simple implicit; $\theta=½$: Crank-Nicolson
  - Accuracy: $\theta=0,1$: $O(\Delta t,\Delta x^2)$; $\theta=½$: $O(\Delta t^2,\Delta x^2)$

- Stability and accuracy: advection vs. diffusion
  - Advection problem:
    - Small wavelengths unstable first; may propagate away and/or amplify, impacting overall solution
  - Diffusion problem:
    - Smallest wavelengths damped the most
    - Durran: accuracy of treatment of smallest waves is "irrelevant" if they are decaying anyway..

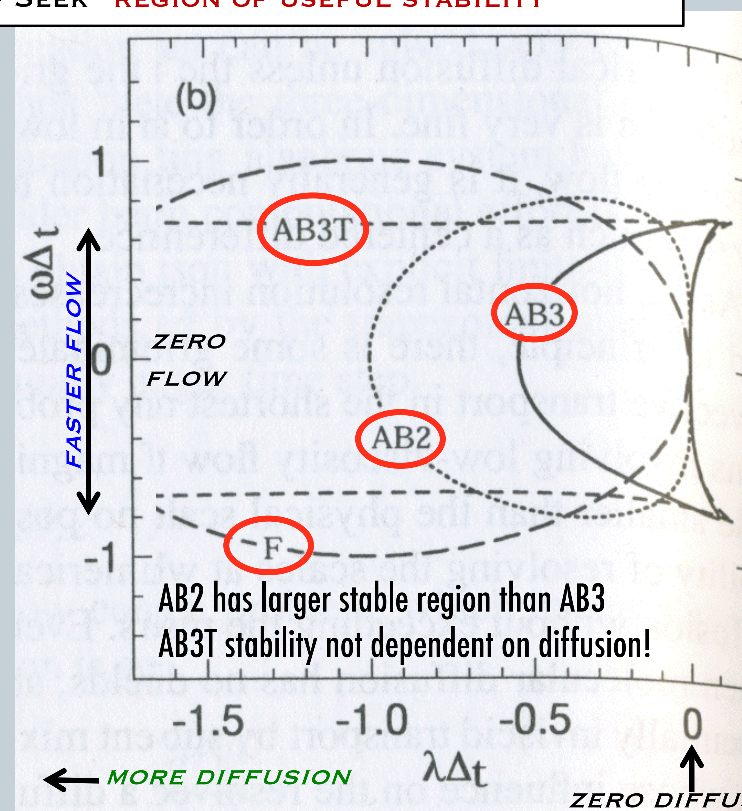
Review: Advection+diffusion: Time differencing

- Leapfrog time differencing
  - Straight LF, LFF (forward) *as in P6*
- Adams-Bashforth stability
  - 2^{nd}, 3^{rd}-order, trapezoidal

AXES ARE SCALED ADVECTION/DIFFUSION COEFFICIENTS. >> SEEK "REGION OF USEFUL STABILITY"



DURRAN (1ST ED). FIG. 3.7A



DURRAN (1ST ED). FIG. 3.7B

$$\tilde{\omega} = \omega\Delta t = -kv\Delta x = -\beta v; \quad \tilde{\lambda} = \lambda\Delta t = \left(-Mk^2\right)\Delta t$$

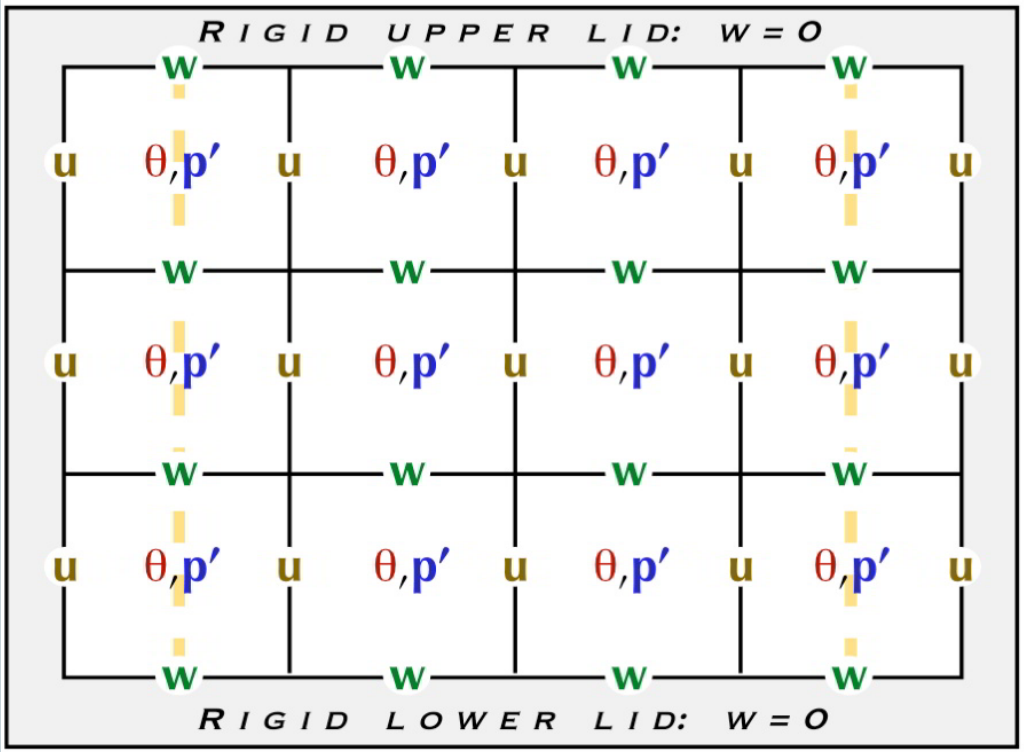
Notes

Program #5

2D NONLINEAR, COMPRESSIBLE FLOW *CONTINUED*

Review: Program 5: boundary conditions

- Variable loop range
  - u is solved 2:nx,1:nz
  - w is solved 1:nx,2:nz
  - p and theta are solved 1:nx,1:nz
- BCs
  - u at i=1, n=nx+1 are -(value inside grid)
  - w at k=1, k=nz *always 0*



Program 5: dt vs. tstep

- Straightforward coding would look like …
  - Forward step:
    - u2 = u1 + Δt • *[forcing terms]*
  - Centered step:
    - u3 = u1 + 2Δt • *[forcing terms]*
  - Writing all that code out twice is annoying.
- Instead, we will do …
  - For the first time step, *tstep=Δt;* otherwise, *tstep=2Δt*
  - And so our equations look like …
    - u3 = u1 + tstep • *[forcing terms] (same for W, P)*
    - works because we *also* initialize our arrays u1=u2=0 *(same for W, P)*
  - *Except* for the temperature: θ is *always* forward in time.

Program 5: Coding practice

- Starting a time step
  - Before doing anything else:
    - t2 = t1
    - u3 = u1
    - w3 = w1
    - All later routines *add to* these "n+1" arrays.

This also lets us turn processes on or off – we have taken the 'first part' of each time step – before starting.

  - So in *advection, diffusion, PGF, you will code …*
    - *t2 = t2 + … Δt • [forcing terms]*
    - *u3 = u3 + … 2Δt • [forcing terms] (same for W)*
  - Exception: pressure
    - Only one step to pressure: *p3 = p1 + 2Δt • [forcing terms]*

Program 5: Variables / equations

- What arrays do I need?
  - Really need only 1 ghost point for u, w, p; 2 for theta
  - Arrays:
    - Leapfrog time differencing: 3 arrays for u,w,p
      - u1,u2,u3   w1,w2,w3   p1,p2,p3
    - Forward time differencing: scalar field theta/s/q
      - t1,t2
- Equations
  - Watch your indexing w/staggered grids!
  - Each equation: indexing is relative *to the variable being solved*
  - Theta advection/diffusion *uses dt*; u/w/p *use tstep.*

Program 5: PGF test

- PGF test
  - PGF = *Pressure Gradient Force* (and buoyancy)
  - No advection / diffusion! Theta perturbation *does not evolve.*
  - Steps to code/debug:
    - 1) Get IC correct
      - *be careful with velocities – make All = 0 in IC*
      - *check your 1-D density array values against mine in handout*
    - 2) Check if *W* at end of first time step is correct. (u still = 0)
    - 3) Check if *P* at end of first time step is correct
    - 4) Only now – check U evolution
  - The first 5 steps are key!!

Notes

Modeling 3d density currents – background for Program 6

NONLINEAR COMPRESSIBLE FLOW