Shallow water equations

- From last time, using eta for depth=h+H:

$$\frac{D\eta}{Dt} = -\eta \nabla \cdot u$$
$$\frac{Du}{Dt} = -g\nabla h$$

- We'll discretize this using "splitting"
  - Handle the material derivative first, then the right-hand side terms next
  - Intermediate depth and velocity from just the advection part

Advection

- Let's discretize just the material derivative (advection equation):

$$q_t + u \cdot \nabla q = 0 \quad \text{or} \quad \frac{Dq}{Dt} = 0$$

- For a Lagrangian scheme this is trivial: just move the particle that stores q, don't change the value of q at all

$$q\big(x(t),t\big) = q\big(x_0,0\big)$$

- For Eulerian schemes it's not so simple

Scalar advection in 1D

- Let's simplify even more, to just one dimension: $q_t+uq_x=0$
- Further assume u=constant
- And let's ignore boundary conditions for now
  - E.g. use a periodic boundary
- True solution just translates q around at speed u - shouldn't change shape

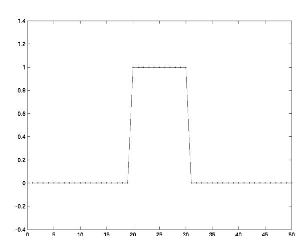
First try: central differences

- Centred-differences give better accuracy
  - More terms cancel in Taylor series
- Example: $\frac{\partial q_i}{\partial t} = -u\left(\frac{q_{i+1} - q_{i-1}}{2\Delta x}\right)$
  - 2nd order accurate in space
- Eigenvalues are pure imaginary - rules out Forward Euler and RK2 in time
- But what does the solution look like?

Testing a pulse

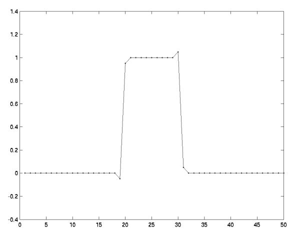
- We know things have to work out nicely in the limit (second order accurate)
  - I.e. when the grid is fine enough
  - What does that mean? -- when the sampled function looks smooth on the grid
- In graphics, it's just redundant to use a grid that fine
  - we can fill in smooth variations with interpolation later
- So we're always concerned about coarse grids == not very smooth data
- Discontinuous pulse is a nice test case

A pulse (initial conditions)



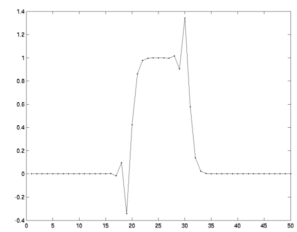
Centered finite differences

- A few time steps (RK4, small Δt) later
  - u=1, so pulse should just move right without changing shape



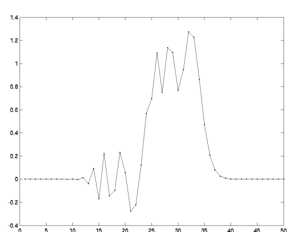
Centred finite differences

- A little bit later…



Centred finite differences

- A fair bit later



What went wrong?

- Lots of ways to interpret this error
- Example - phase analysis
  - Take a look at what happens to a sinusoid wave numerically
  - The amplitude stays constant (good), but the wave speed depends on wave number (bad) - we get dispersion
  - So the sinusoids that initially sum up to be a square pulse move at different speeds and separate out
    - We see the low frequency ones moving faster…
  - But this analysis won't help so much in multi-dimensions, variable u…

Modified PDE's

- Another way to interpret error - try to account for it in the physics
- Look at truncation error more carefully:

$$q_{i+1} = q_i + \Delta x \frac{\partial q}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 q}{\partial x^2} + \frac{\Delta x^3}{6}\frac{\partial^3 q}{\partial x^3} + O\left(\Delta x^4\right)$$

$$q_{i-1} = q_i - \Delta x \frac{\partial q}{\partial x} + \frac{\Delta x^2}{2}\frac{\partial^2 q}{\partial x^2} - \frac{\Delta x^3}{6}\frac{\partial^3 q}{\partial x^3} + O\left(\Delta x^4\right)$$

$$\frac{q_{i+1} - q_{i-1}}{2\Delta x} = \frac{\partial q}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 q}{\partial x^3} + O\left(\Delta x^3\right)$$

- Up to high order error, we numerically solve

$$q_t + uq_x = -\frac{u\Delta x^2}{6} q_{xxx}$$

Interpretation

$$q_t + uq_x = -\frac{u\Delta x^6}{6} q_{xxx}$$

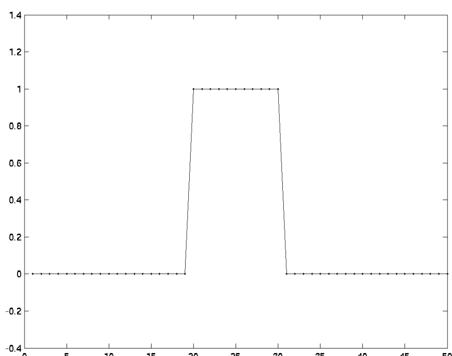
- Extra term is "dispersion"
  - Does exactly what phase analysis tells us
  - Behaves a bit like surface tension…
- We want a numerical method with a different sort of truncation error
  - Any centred scheme ends up giving an odd truncation error --- dispersion
- Let's look at one-sided schemes

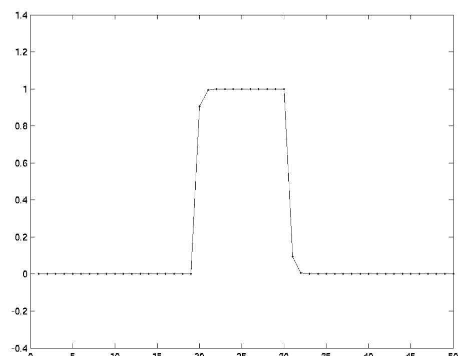
Upwind differencing

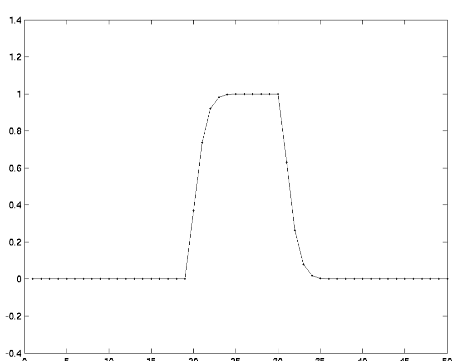
- Think physically:
  - True solution is that q just translates at velocity u
- Information flows with u
- So to determine future values of q at a grid point, need to look "upwind" -- where the information will blow from
  - Values of q "downwind" only have any relevance if we know q is smooth -- and we're assuming it isn't

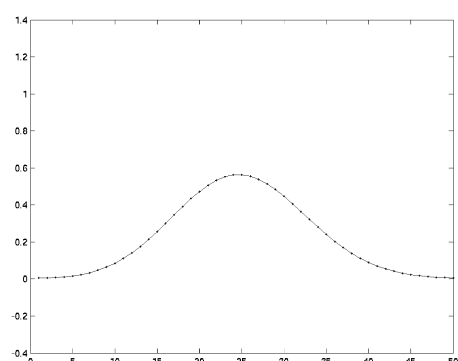
1st order upwind

- Basic idea: look at sign of u to figure out which direction we should get information
- If u<0 then $\partial q/\partial x \approx (q_{i+1}-q_i)/\Delta x$
- If u>0 then $\partial q/\partial x \approx (q_i-q_{i-1})/\Delta x$
- Only 1st order accurate though
  - Let's see how it does on the pulse…









Modified PDE again

- Let's see what the modified PDE is this time

$$q_{i+1} = q_i + \Delta x \frac{\partial q}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 q}{\partial x^2} + O(\Delta x^3)$$

$$\frac{q_{i+1} - q_i}{\Delta x} = \frac{\partial q}{\partial x} + \frac{\Delta x}{2} \frac{\partial^2 q}{\partial x^2} + O(\Delta x^2)$$

- For u<0 then we have $q_t + uq_x = -\frac{u\Delta x}{2} q_{xx}$
- And for u>0 we have (basically flip sign of Δx)

$$q_t + uq_x = \frac{u\Delta x}{2} q_{xx}$$

- Putting them together, 1st order upwind numerical solves (to 2nd order accuracy)

$$q_t + uq_x = \left| \frac{u\Delta x}{2} \right| q_{xx}$$

Interpretation

- The extra term (that disappears as we refine the grid) is **diffusion** or **viscosity**
- So sharp pulse gets blurred out into a hump, and eventually melts to nothing
- It looks a lot better, but still not great
  - Again, we want to pack as much detail as possible onto our coarse grid
  - With this scheme, the detail melts away to nothing pretty fast
- Note: unless grid is really fine, the numerical viscosity is much larger than physical viscosity - so might as well not use the latter

Fixing upwind method

- Natural answer - reduce the error by going to higher order - but life isn't so simple
- High order difference formulas can overshoot in extrapolating
  - If we difference over a discontinuity
  - Stability becomes a real problem
- Go nonlinear (even though problem is linear)
  - "limiters" - use high order unless you detect a (near-)overshoot, then go back to 1st order upwind
  - "ENO" - try a few different high order formulas, pick smoothest

Hamilton-Jacobi Equations

- In fact, the advection step is a simple example of a Hamilton-Jacobi equation (HJ)
  - $q_t+H(q,q_x)=0$
- Come up in lots of places
  - Level sets…
- Lots of research on them, and numerical methods to solve them
  - E.g. 5th order HJ-WENO
- We don't want to get into that complication

Other problems

- Even if we use top-notch numerical methods for HJ, we have problems
  - Time step limit: CFL condition
    - Have to pick time step small enough that information physically moves less than a grid cell: $\Delta t<\Delta x/u$
  - Schemes can get messy at boundaries
  - Discontinuous data still gets smoothed out to some extent (high resolution schemes drop to first order upwinding)

Exploiting Lagrangian view

- But wait! This was trivial for Lagrangian (particle) methods!
- We still want to stick an Eulerian grid for now, but somehow exploit the fact that
  - If we know q at some point x at time t, we just follow a particle through the flow starting at x to see where that value of q ends up

$$q\big(x(t+\Delta t), t+\Delta t\big) = q\big(x(t), t\big)$$

$$\frac{dx}{dt} = u(x), \quad x(t) = x_0$$

Looking backwards

- Problem with tracing particles - we want values at **grid nodes** at the end of the step
  - Particles could end up anywhere
- But… we can look backwards in time

$$q_{ijk} = q\big(x(t-\Delta t), t-\Delta t\big)$$

$$\frac{dx}{dt} = u\big(x\big), \quad x(t) = x_{ijk}$$

- Same formulas as before - but new interpretation
  - To get value of q at a grid point, follow a particle backwards through flow to wherever it started

Semi-Lagrangian method

- Developed in weather prediction, going back to the 50's
- Also dubbed "stable fluids" in graphics (reinvention by Stam '99)
- To find new value of q at a grid point, trace particle backwards from grid point (with velocity u) for -Δt and interpolate from old values of q
- Two questions
  - How do we trace?
  - How do we interpolate?

Tracing

- The errors we make in tracing backwards aren't too big a deal
  - We don't compound them - stability isn't an issue
  - How accurate we are in tracing doesn't effect shape of q much, just location
    - Whether we get too much blurring, oscillations, or a nice result is really up to interpolation
- Thus look at "Forward" Euler and RK2

Tracing: 1st order

- We're at grid node (i,j,k) at position x_{ijk}
- Trace backwards through flow for -Δt

$$x_{old} = x_{ijk} - \Delta t\, u_{ijk}$$

  - Note - using u value at grid point (what we know already) like Forward Euler.
- Then can get new q value (with interpolation)

$$\begin{aligned} q_{ijk}^{n+1} &= q^n\left(x_{old}\right) \\ &= q^n\left(x_{ijk} - \Delta t\, u_{ijk}\right) \end{aligned}$$

Interpolation

- "First" order accurate: nearest neighbour
  - Just pick q value at grid node closest to x_{old}
  - Doesn't work for slow fluid (small time steps) -- nothing changes!
  - When we divide by grid spacing to put in terms of advection equation, drops to zero'th order accuracy
- "Second" order accurate: linear or bilinear (2D)
  - Ends up first order in advection equation
  - Still fast, easy to handle boundary conditions…
  - How well does it work?

Linear interpolation

- Error in linear interpolation is proportional to

$$\Delta x^2 \frac{\partial^2 q}{\partial x^2}$$

- Modified PDE ends up something like…

$$\frac{Dq}{Dt} = k\big(\Delta t\big)\Delta x^2 \frac{\partial^2 q}{\partial x^2}$$

  - We have numerical viscosity, things will smear out
  - For reasonable time steps, k looks like 1/Δt ~ 1/Δx
- [Equivalent to 1st order upwind for CFL Δt]
- In practice, too much smearing for inviscid fluids

Nice properties of lerping

- Linear interpolation is completely stable
  - Interpolated value of q must lie between the old values of q on the grid
  - Thus with each time step, max(q) cannot increase, and min(q) cannot decrease
- Thus we end up with a fully stable algorithm - take Δt as big as you want
  - Great for interactive applications
  - Also simplifies whole issue of picking time steps

Cubic interpolation

- To fix the problem of excessive smearing, go to higher order
- E.g. use cubic splines
  - Finding interpolating C^2 cubic spline is a little painful, an alternative is the
  - C^1 Catmull-Rom (cubic Hermite) spline
    - [derive]
- Introduces overshoot problems
  - Stability isn't so easy to guarantee anymore

Min-mod limited Catmull-Rom

- See Fedkiw, Stam, Jensen '01
- Trick is to check if either slope at the endpoints of the interval has the wrong sign
  - If so, clamp the slope to zero
  - Still use cubic Hermite formulas with more reliable slopes
- This has same stability guarantee as linear interpolation
  - But in smoother parts of flow, higher order accurate
  - Called "high resolution"
- Still has issues with boundary conditions…

Back to Shallow Water

- So we can now handle advection of both water depth and each component of water velocity
- Left with the divergence and gradient terms

$$\frac{\partial \eta}{\partial t} = -\eta\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}\right)$$

$$\frac{\partial u}{\partial t} = -g\frac{\partial h}{\partial x}$$

$$\frac{\partial w}{\partial t} = -g\frac{\partial h}{\partial z}$$

MAC grid

- We like central differences - more accurate, unbiased
- So natural to use a staggered grid for velocity and height variables
  - Called MAC grid after the Marker-and-Cell method (Harlow and Welch '65) that introduced it
- Heights at cell centres
- u-velocities at x-faces of cells
- w-velocities at z-faces of cells

Spatial Discretization

- So on the MAC grid:

$$\frac{\partial \eta_{ij}}{\partial t} = -\eta_{ij}\left(\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{w_{i,j+1/2} - w_{i,j-1/2}}{\Delta z}\right)$$

$$\frac{\partial u_{i+1/2,j}}{\partial t} = -g\frac{h_{i+1,j} - h_{i,j}}{\Delta x}$$

$$\frac{\partial w_{i,j+1/2}}{\partial t} = -g\frac{h_{i,j+1} - h_{i,j}}{\Delta z}$$