Notes

- Please read
  - Kass and Miller, “Rapid, Stable Fluid Dynamics for Computer Graphics”, SIGGRAPH’90

Shallow water

- Simplified linear analysis before had dispersion relation

$$c = \sqrt{\frac{g}{k} \tanh kH}$$

  - For shallow water, kH is small (that is, wave lengths are comparable to depth)
  - Approximate tanh(x)=x for small x:

$$c = \sqrt{gH}$$

- Now wave speed is independent of wave number, but **dependent** on depth
  - Waves slow down as they approach the beach

What does this mean?

- We see the effect of the bottom
  - Submerged objects (H decreased) show up as places where surface waves pile up on each other
  - Waves pile up on each other (eventually should break) at the beach
  - Waves refract to be parallel to the beach
- We can’t use Fourier analysis

PDE’s

- Saving grace: wave speed independent of k means we can solve as a 2D PDE
- We’ll derive these “shallow water equations”
  - When we linearize, we’ll get same wave speed
- Going to PDE’s also let’s us handle non-square domains, changing boundaries
  - The beach, puddles, …
  - Objects sticking out of the water (piers, walls, …) with the right reflections, diffraction, …
  - Dropping objects in the water

Kinematic assumptions

- We'll assume as before water surface is a height field y=h(x,z,t)
- Water bottom is y=-H(x,z,t)
- On top of this, assume velocity field doesn't vary much in the y direction
  - u=u(x,z,t), w=w(x,z,t)
  - Can't assume v is independent of y, but assume least variation possible: linear in y
- For "shallow" water (or any other nearly 2D flow) this is a good approximation
  - "shallow" meaning whenever this is a good approximation - little variation in y ☺

Conservation of mass

- Integrate over a column of water with cross-section A and height h+H
  - Total mass is ρ(h+H)A
  - Mass flux around cross-section is ρ(h+H)(u,w)
- Write down the conservation law
- In differential form (assuming constant density): $\frac{\partial}{\partial t}\left(h+H\right)+\nabla\cdot\left(\left(h+H\right)u\right)=0$
  - Note: switched to 2D so u=(u,w) and ∇=(∂/∂x, ∂/∂z)

Pressure

- Look at y-component of momentum equation:

$$v_t + u\cdot\nabla v + \frac{1}{\rho}\frac{\partial p}{\partial y} = -g + \nu\nabla^2 v$$

- Assume small velocity variation - so dominant terms are pressure gradient and gravity:

$$\frac{1}{\rho}\frac{\partial p}{\partial y} = -g$$

- Boundary condition at water surface is p=0 again, so can solve for p:

$$p = \rho g\left(h-y\right)$$

Momentum Equation

- So now look at x and z components of momentum equation, plugging in our formula for pressure (in 2D again)

$$u_t + u\cdot\nabla u + g\nabla h = \nu\nabla^2 u$$

- In conservation law form (just rewriting):

$$u_t + \frac{\partial}{\partial x}\left(\tfrac{1}{2}u^2 + gh - \nu\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial z}\left(uw - \nu\frac{\partial u}{\partial z}\right) = 0$$

$$w_t + \frac{\partial}{\partial x}\left(uw - \nu\frac{\partial w}{\partial x}\right) + \frac{\partial}{\partial z}\left(\tfrac{1}{2}w^2 + gh - \nu\frac{\partial w}{\partial z}\right) = 0$$

The Shallow Water Equations

- Typically assume viscosity=0, write as a system of conservation laws

$$\begin{pmatrix} h \\ u \end{pmatrix}_t + \nabla \cdot F(h,u) = G$$

- Lots of work done on conservation laws such as this
- Without viscosity, “shocks” may develop
  - Discontinuities in solution (need to go to weak integral form of equations)
  - Corresponds to breaking waves - getting steeper and steeper until heightfield assumption breaks down

Breaking things down

- Rewrite equations to highlight advection

$$(h+H)_t + u \cdot \nabla (h+H) = -(h+H)\nabla \cdot u$$

$$u_t + u \cdot \nabla u = -g\nabla h$$

- Recognize the material derivative
  - So water height gets carried around by flow, and increases if velocity is converging
  - Water velocity gets carried around by flow, water accelerates down slope of water according to gravity
- Suggests a numerical approach (“splitting” or “fractional steps”)
  - move stuff around on the grid first, then change it

Linearization

- Again assume not too much velocity variation (i.e. waves move, but water basically doesn’t)
  - No currents, just small waves
  - Alternatively: inertia not important compared to gravity
  - Or: numerical method treats the advection separately (see next week!)
- Then drop the nonlinear advection terms
- Also assume H doesn’t vary in time

$$h_t = -(h+H)\nabla \cdot u$$

$$u_t = -g\nabla h$$

Wave equation

- Only really care about heightfield for rendering
- Differentiate height equation in time

$$h_{tt} = -h_t \nabla \cdot u - (h+H)\nabla \cdot u_t$$

- Plug in u equation

$$h_{tt} = -h_t \nabla \cdot u + g(h+H)\nabla^2 h$$

- Finally, neglect nonlinear (quadratically small) terms on right to get

$$h_{tt} = gH\nabla^2 h$$

Deja vu

- This is the linear wave equation, with constant wave speed $c^2=gH$
- Which is exactly what we derived from the dispersion relation before (after linearizing the equations in a different way)
- But now we have it in a PDE that we have some confidence in
  - Can handle varying H, irregular domains…
- Caveat: to handle H going to 0 or negative, we'll in fact use $h_{tt} = g(h+H)\nabla^2 h$

Anti-water

- In linearizing, lost conservation properties
- Now possible for a large wave out in deeper water to come into where its shallower and…
- Stay large, with $h<-H$
  - I.e. the water level beneath the sea floor
- Obviously a bad thing
- Simple solution: ignore it
- Slightly better: clamp h to be at least -H
  - And directly enforce global conservation of mass in each connected region of water - if mass changes, scale h+H by just the right amount

Initial + boundary conditions

- We can specify initial h and h_t
  - Since it's a second order equation
- We can specify h at "open" boundaries
  - Water is free to flow in and out
- Specify $\partial h/\partial n=0$ at "closed" boundaries
  - Water does not pass through boundary
  - Equivalent to reflection symmetry
  - Waves reflect off these boundaries
- Note: dry beaches etc. don't have to be treated as boundaries -- instead just have h=-H initially

Example conditions

- Start with quiet water h=0, beach on one side of domain
- On far side, specify h by 1D Fourier synthesis (e.g. see last lecture)
- On lateral sides, specify $\partial h/\partial n=0$ (reflect solution)
- Keep beach side dry h=-H
- Start integrating

Space Discretization

- In space, let's use finite-differences on a regular grid
- Need to discretize $\nabla^2 h = h_{xx} + h_{zz}$
- Standard 5-point approximation good:

$$\left(\nabla^2 h\right)_{ij} \approx \frac{h_{i+1j} - 2h_{ij} + h_{i-1j}}{\Delta x^2} + \frac{h_{ij+1} - 2h_{ij} + h_{ij-1}}{\Delta z^2}$$

- At boundaries where h is specified, plug in those values instead of grid unknowns
- At boundaries where normal derivative is specifed, use finite difference too
  - Example $h_{i+1j} - h_{ij} = 0$ which gives $h_{i+1j} = h_{ij}$

Time discretization

- We're doing the "method of lines" - discretize PDE in space to get a bunch of ODE's
- It's (roughly) of the form $h_{tt} = -Ah$ where A is a symmetric positive-semidefinite matrix
  - Actually A depends on h as well as H
- Can go back to original set of methods
- A few other choices available

Central explicit

- Use same second-order accurate discretization of second derivative in space as in time:

$$\frac{h^{n+1} - 2h^n + h^{n-1}}{\Delta t^2} = -Ah^n$$

$$h^{n+1} = h^n + \left(h^n - h^{n-1}\right) - \Delta t^2 A h^n$$

- Instead of keeping velocity around, keep an older value of h
  - Equivalent to central method of before
- Can derive stability limit
  - Same as CFL limit: numerical speed > wave speed

Implicit methods

- Need to solve system involving A
  - Will need to use an iterative method
  - Or call an efficient sparse direct solver
  - Not so efficient
- Alternative: ADI (alternating direction implicit)
  - Split time discretiztion into an implicit step with d/dx part, then another implicit step with d/dz
  - Each implicit solve only uses one part of A, logically a tridiagonal matrix: can solve really easily in linear time
  - But, can cause weird grid artifacts if dt too big
- Note: "lagging" - use old value of h to get A

Surface tension

- Let's go back to nonlinear shallow water equations for a moment
- If we include surface tension, then there's an extra normal traction (i.e. pressure) on surface
  - Proportional to the mean curvature
  - The more curved the surface, the more it wants to get flat again
  - Actually arises out of different molecular attractions between water-water, water-air, air-air
- We can model this by changing pressure BC to $p=\sigma\kappa$ from $p=0$ at surface $y=h$

Mean curvature

- If surface is fairly flat, can approximate

$$\kappa \approx -\nabla^2 h$$

  - [show in 1D]
  - Laplacian is rotationally invariant, so rotate to line up directions of maximum curvature
- Plugging this pressure into momentum gives

$$u_t + u \cdot \nabla u + g\nabla h - \frac{\sigma}{\rho}\nabla\nabla^2 h = \nu\nabla^2 u$$

Simplifying

- Doing same linearization as before, but now in 1D (forget z) get

$$h_{tt} = gHh_{xx} - \frac{\sigma H}{\rho} h_{xxxx}$$

- Should look familiar - it's the bending equation from long ago
- Capillary (surface tension) waves important at small length scales

Other shallow water eq's

- General idea of ignoring variation (except linear pressure) in one dimension applicable elsewhere
- Especially geophysical flows: the weather
- Need to account for the fact that Earth is rotating, not an inertial frame
  - Add Coriolis pseudo-forces