Modelling Water

• Real water has lots of interesting properties
  • smooth (laminar) and turbulent flow
  • waves
  • splashes
  • droplets
  • surface tension
  • wetting (surfaces)
  • mixing (with other liquids)
  • cavitation (when you drop a solid in)

• DAMTP has a research group studying how to model water
  • graphics people shouldn't expect to find an easy and accurate model
  • but might find an “OK” model with “reasonable” computational requirements

  • “OK” means that it looks right (or close enough to right...)
Three ways you could model water

1. Height field
   - $h_i$:
   - Water height
   - $b_i$:
   - Solid base level

2. Implicit surface

3. Voxel space
   - A voxel can be:
     - Full
     - Empty
     - Surface
**HEIGHT FIELD MODEL**

- cannot model splashes, breaking waves, droplets, ...
- based on Kass & Miller “Rapid, stable fluid dynamics for CG” SIGGRAPH 90
- simple implementation in Java at:
  
  http://www.cl.cam.ac.uk/~nad/applts/AGwater.html

---

Each column of water has a velocity

\[ d_i = h_i - b_i \]

(depth)

**Basic Equations**

1. \[ \frac{dv}{dt} = -v \frac{dv}{dx} - g \frac{dh}{dx} \]  
   
   \( (F=ma) \)

   \( \text{acceleration} \) \quad \text{velocity difference} \quad \text{pressure difference} \quad \text{force} \)

2. \[ \frac{dd}{dt} = -\frac{d}{dx} (rd) \]

   \( \text{(volume conservation)} \)

   \( \text{change in depth} \) \quad \text{amount of water flowing} \)
Kass & Miller assume:

- adjacent columns have (roughly) the same velocity

\[ \therefore \text{ approximate } 1 \text{ by } \]
\[ 1' \quad \frac{dv}{dt} = -g \frac{dh}{dx} \]

- depths do not change rapidly (?)

\[ \therefore \text{ approximate } 2 \text{ by } \]
\[ 2' \quad \frac{dh}{dt} = -d \frac{dv}{dx} \]

Combine 1' & 2' to get

\[ 3 \quad \frac{d^2h}{dt^2} = gd \frac{d^2h}{dx^2} \]

which (it transpires) is the “shallow water Navier-Stokes simplification”

Kass & Miller go on to produce a discrete approximation and solve a tridiagonal system of linear equations to advance from one timestep to the next because "... the wave equation 3 is a notoriously bad example for explicit differential equation methods such as Euler"
Nevertheless...

we can implement it using Euler evaluation

\[ v_i' = v_i + \Delta t \left( v_i \frac{v_{i+1} - v_i}{\Delta x} - g \frac{h_{i+1} - h_i}{\Delta x} \right) \]

\[ h_i' = h_i + \Delta t \left( [v_{i-1} - v_i] \frac{h_i - b_i}{\Delta x} \right) \]

*my implementation includes several tweaks to*

these equations

What goes wrong?

\[ \text{if } v_i > \frac{\Delta x}{\Delta t} \]

then the algorithm takes more water
from column \( i \) than is actually

present

* as \( v_i \) approaches \( \frac{\Delta x}{\Delta t} \) the Euler
solution diverges from the accurate solution

* one solution: decrease \( \Delta t \)

* another: use a better solver (e.g. Runge-Kutta)

* & another: solve a system of equations
  (e.g. Koss & Miller)
Solving the full *Navier-Stokes* equation

Variables

- position \((x, y, z)\)
- velocity \((u, v, w)\)
- time \(t\)
- pressure \(p\)
- body force \((F_x, F_y, F_z)\)

\[
\frac{du}{dx} + \frac{dv}{dy} + \frac{dw}{dz} = 0
\]

\[
\rho \left( \frac{du}{dt} + u \frac{du}{dx} + v \frac{du}{dy} + w \frac{du}{dz} \right) = - \frac{dp}{dx} + \eta \left( \frac{d^2 u}{dx^2} + \frac{d^2 u}{dy^2} + \frac{d^2 u}{dz^2} \right) + F_x
\]

\[
\rho \left( \frac{dv}{dt} + u \frac{dv}{dx} + v \frac{dv}{dy} + w \frac{dv}{dz} \right) = - \frac{dp}{dy} + \eta \left( \frac{d^2 v}{dx^2} + \frac{d^2 v}{dy^2} + \frac{d^2 v}{dz^2} \right) + F_y
\]

\[
\rho \left( \frac{dw}{dt} + u \frac{dw}{dx} + v \frac{dw}{dy} + w \frac{dw}{dz} \right) = - \frac{dp}{dz} + \eta \left( \frac{d^2 w}{dx^2} + \frac{d^2 w}{dy^2} + \frac{d^2 w}{dz^2} \right) + F_z
\]

- Simplify as much as possible (e.g. assume no body force) & discretise
- Solve on a voxel grid for each time step

* this is actually *Navier-Stokes* for an irrotational, incompressible liquid — it gets worse if you remove these 2 constraints