FLUID SIMULATION

Robert Bridson, UBC

Matthias Müller-Fischer, AGEIA Inc.

Adapted from SIGGRAPH 2007 course notes. The original presentation is here: https://www.cs.ubc.ca/~rbridson/fluidsimulation/

The Basic Equations

Symbols

- $\vec{u}$: velocity with components (u,v,w)
- ρ: fluid density
- p: pressure
- $\vec{g}$: acceleration due to gravity or animator
- μ: dynamic viscosity

The Equations

- Incompressible Navier-Stokes:
  - "Momentum Equation"

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \bullet \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + \frac{\mu}{\rho} \nabla \bullet \nabla \vec{u}$$

  - "Incompressibility condition"

$$\nabla \bullet \vec{u} = 0$$

The Momentum Equation

The Momentum Equation

- Just a specialized version of F=ma
- Let's build it up intuitively
- Imagine modeling a fluid with a bunch of particles (e.g. blobs of water)
  - A blob has a mass m, a volume V, velocity u
  - We'll write the acceleration as Du/Dt ("material derivative")

$$m \frac{D\vec{u}}{Dt} = \vec{F}$$

  - What forces act on the blob?

Forces on Fluids

- Gravity: mg
  - or other “body forces” designed by animator

$$m\frac{D\vec{u}}{Dt} = m\vec{g} + \ldots$$

- And a blob of fluid also exerts contact forces on its neighbouring blobs…

Pressure

- The “normal” contact force is pressure (force/area)
  - How blobs push against each other,
    and how they stick together

- If pressure is equal in every direction, net force is zero…
  Important quantity is **pressure gradient**:

$$m\frac{D\vec{u}}{Dt} = m\vec{g} - V\nabla p + \ldots$$

- What is the pressure? Coming soon…

Viscosity

- Think of it as frictional part of contact force: a sticky (viscous) fluid blob resists other blobs moving past it
- For now, simple model is that we want velocity to be blurred/diffused/…
- Blurring in PDE form comes from the Laplacian:

$$m \frac{D\vec{u}}{Dt} = m\vec{g} - V\nabla p + V\mu\nabla \bullet \nabla \vec{u}$$

The Continuum Limit (1)

- Model the world as a continuum:
  - # particles → ∞
    Mass and volume → 0
- We want F=ma to be more than 0=0:
  - Divide by mass

$$\frac{D\vec{u}}{Dt} = \vec{g} - \frac{V}{m}\nabla p + \frac{V}{m}\mu\nabla\bullet\nabla\vec{u}$$

The Continuum Limit (2)

- The fluid density is ρ=m/V:

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla p = \vec{g} + \frac{\mu}{\rho}\nabla \bullet \nabla \vec{u}$$

- This is almost the same as the stanford eq'n (in fact, the form we mostly use!)
- The only weird thing is Du/Dt…

Lagrangian vs. Eulerian

- Lagrangian viewpoint:
  - Treat the world like a particle system
  - Label each speck of matter, track where it goes (how fast, acceleration, etc.)
- Eulerian viewpoint:
  - Fix your point in space
  - Measure stuff as it flows past
- Think of measuring the temperature:
  - Lagrangian: in a balloon, floating with the wind
  - Eulerian: on the ground, wind blows past

The Material Derivative (1)

- We have fluid moving in a velocity field u
- It possesses some quality q
- At an instant in time t and a position in space x, the fluid at x has q(x,t)
  - q(x,t) is an **Eulerian** field
- How fast is that blob of fluid's q changing?
  - A **Lagrangian** question
- Answer: the material derivative Dq/Dt

The Material Derivative (2)

- It all boils down to the chain rule:

$$\frac{D}{Dt}q(x,t) = \frac{\partial q}{\partial t} + \frac{\partial q}{\partial x} \bullet \frac{dx}{dt}$$

$$= \frac{\partial q}{\partial t} + \nabla q \bullet \vec{u}$$

- We usually rearrange it:

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \vec{u} \bullet \nabla q$$

Turning Dq/Dt Around

- For a thought experiment, turn it around:

$$\frac{\partial q}{\partial t} = \frac{Dq}{Dt} - \vec{u} \bullet \nabla q$$

- That is, how fast q is changing at a fixed point in space (∂q/∂t) comes from two things:
  - How fast q is changing for the blob of fluid at x
  - How fast fluid with different values of q is flowing past

Writing D/Dt Out

- We can explicitly write it out from components:

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \vec{u} \bullet \nabla q$$

$$= \frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} + v\frac{\partial q}{\partial y} + w\frac{\partial q}{\partial z}$$

D/Dt For Vector Fields

- Say our fluid has a colour variable (RGB vector) C
- We still write

$$\frac{D\vec{C}}{Dt} = \frac{\partial \vec{C}}{\partial t} + \vec{u} \bullet \nabla \vec{C}$$

  - The dot-product and gradient confusing?
  - Just do it component-by-component:

$$\frac{D\vec{C}}{Dt} = \begin{bmatrix} DR/Dt \\ DG/Dt \\ DB/Dt \end{bmatrix} = \begin{bmatrix} \partial R/\partial t + \vec{u} \bullet \nabla R \\ \partial G/\partial t + \vec{u} \bullet \nabla G \\ \partial B/\partial t + \vec{u} \bullet \nabla B \end{bmatrix}$$

Du/Dt

- This holds even if the vector field is velocity itself:

$$\frac{D\vec{u}}{Dt} = \frac{\partial \vec{u}}{\partial t} + \vec{u} \bullet \nabla \vec{u}$$

$$\begin{bmatrix} Du/Dt \\ Dv/Dt \\ Dw/Dt \end{bmatrix} = \begin{bmatrix} \partial u/\partial t + \vec{u} \bullet \nabla u \\ \partial v/\partial t + \vec{u} \bullet \nabla v \\ \partial w/\partial t + \vec{u} \bullet \nabla w \end{bmatrix}$$

- Nothing different about this, just that the fluid blobs are moving at the velocity they're carrying.

The Incompressibility Condition

Compressibility

- Real fluids are compressible
- Shock waves, acoustic waves, pistons…
  - Note: liquids change their volume as well as gases, otherwise there would be no sound underwater
- But this is nearly irrelevant for animation
  - Shocks move too fast to normally be seen (easier/better to hack in their effects)
  - Acoustic waves usually have little effect on visible fluid motion
  - Pistons are boring

Incompressibility

- Rather than having to simulate acoustic and shock waves, eliminate them from our model: assume fluid is **incompressible**
  - Turn stiff system into a constraint, just like rigid bodies!
- If you fix your eyes on any volume of space, volume of fluid in = volume of fluid out:

$$\iint_{\partial\Omega} \vec{u} \cdot \hat{n} = 0$$

Divergence

- Let's use the divergence theorem:

$$\iint_{\partial\Omega} \vec{u} \bullet \hat{n} = \iiint_{\Omega} \nabla \bullet \vec{u}$$

- So for any region, the integral of $\nabla \bullet \vec{u}$ is zero
- Thus it's zero everywhere:

$$\nabla \bullet \vec{u} = 0$$

Pressure

- Pressure p:
  whatever it takes to make the velocity field divergence free
- If you know constrained dynamics, $\nabla \bullet \vec{u} = 0$ is a constraint, and pressure is the matching Lagrange multiplier
- Our simulator will follow this approach:
  - solve for a pressure that makes our fluid incompressible at each time step.

Aside: A Few Figures

- Dynamic viscosity of air: $\mu_{air} \approx 1.8 \times 10^{-5}\ Ns/m^2$
- Density of air: $\rho_{air} \approx 1.3\, kg/m^3$

- Dynamic viscosity of water: $\mu_{water} \approx 1.1 \times 10^{-3}\ Ns/m^2$
- Density of water: $\rho_{water} \approx 1000\ kg/m^3$

- The ratio, μ_{air}/μ_{water} ("kinematic viscosity") is what's important for the motion of the fluid…
  … air is 10 times more viscous than water!

Boundary Conditions

Boundary Conditions

- We know what's going on inside the fluid: what about at the surface?
- Three types of surface
  - Solid wall: fluid is adjacent to a solid body
  - Free surface: fluid is adjacent to nothing (e.g. water is so much denser than air, might as well forget about the air)
  - Other fluid: possibly discontinuous jump in quantities (density, …)

Solid Wall Boundaries

- No fluid can enter or come out of a solid wall:

$$\vec{u} \bullet \hat{n} = \vec{u}_{solid} \bullet \hat{n}$$

- For common case of $u_{solid}=0$:

$$\vec{u} \bullet \hat{n} = 0$$

- Sometimes called the "no-stick" condition, since we let fluid slip past tangentially
  - For viscous fluids, can additionally impose "no-slip" condition:

$$\vec{u} = \vec{u}_{solid}$$

Free Surface

- Neglecting the other fluid, we model it simply as pressure=constant
  - Since only pressure gradient is important, we can choose the constant to be zero:

$$p = 0$$

- If surface tension is important (not covered today), pressure is instead related to mean curvature of surface

Multiple Fluids

- At fluid-fluid boundaries, the trick is to determine "jump conditions"
  - For a fluid quantity q, $[q]=q_1-q_2$
- Density jump $[\rho]$ is known
- Normal velocity jump must be zero: $[\vec{u} \bullet \hat{n}] = 0$
- For inviscid flow, tangential velocities may be unrelated (jump is unknown)
- With no surface tension, pressure jump $[p]=0$

Numerical Simulation Overview

Splitting

- We have lots of terms in the momentum equation: a pain to handle them all simultaneously
- Instead we split up the equation into its terms, and integrate them one after the other
  - Makes for easier software design too: a separate solution module for each term
- First order accurate in time
  - Can be made more accurate, not covered today.

A Splitting Example

- Say we have a differential equation

$$\frac{dq}{dt} = f(q) + g(q)$$

- And we can solve the component parts:
  - SolveF(q,Δt) solves dq/dt=f(q) for time Δt
  - SolveG(q,Δt) solves dq/dt=g(q) for time Δt
- Put them together to solve the full thing:
  - $q^* = \text{SolveF}(q^n, \Delta t)$
  - $q^{n+1} = \text{SolveG}(q^*, \Delta t)$

Does it Work?

- Up to O(Δt):

$$\frac{dq}{dt} \approx \frac{q^{n+1} - q^n}{\Delta t}$$

$$= \frac{q^{n+1} - q^*}{\Delta t} + \frac{q^* - q^n}{\Delta t}$$

$$\approx g(q) + f(q)$$

Splitting Momentum

- We have three terms: $$\frac{\partial \vec{u}}{\partial t} = -\vec{u} \bullet \nabla \vec{u} + \vec{g} - \frac{1}{\rho} \nabla p$$
- First term: **advection** $$\frac{\partial \vec{u}}{\partial t} = -\vec{u} \bullet \nabla \vec{u}$$
  - Move the fluid through its velocity field (Du/Dt=0)
- Second term: **gravity** $$\frac{\partial \vec{u}}{\partial t} = \vec{g}$$
- Final term: **pressure update** $$\frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \nabla p$$
  - How we'll make the fluid incompressible: $\nabla \bullet \vec{u} = 0$

Space

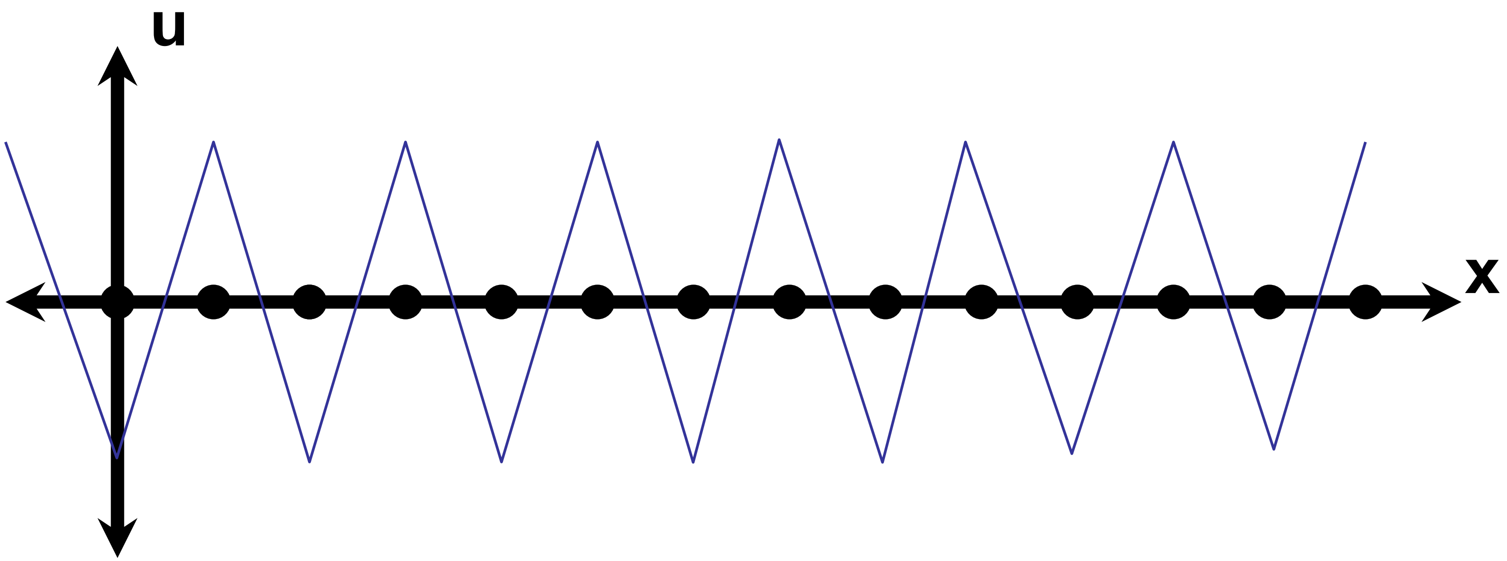
- That's our general strategy in time; what about space?
- We'll begin with a fixed Eulerian grid
  - Trivial to set up
  - Easy to approximate spatial derivatives
  - Particularly good for the effect of pressure
- Disadvantage: advection doesn't work so well
  - Later: particle methods that fix this

A Simple Grid

- We could put all our fluid variables at the nodes of a regular grid
- But this causes some major problems
- In 1D: incompressibility means $\frac{\partial u}{\partial x} = 0$
- Approximate at a grid point: $\frac{u_{i+1} - u_{i-1}}{2\Delta x} = 0$
- Note the velocity at the grid point isn't involved!

A Simple Grid Disaster

- The only solutions to $\frac{\partial u}{\partial x} = 0$ are u=constant
- But our numerical version has other solutions:



Staggered Grids

- Problem is solved if we don't skip over grid points
- To make it unbiased, we **stagger** the grid: put velocities halfway between grid points
- In 1D, we estimate divergence at a grid point as:

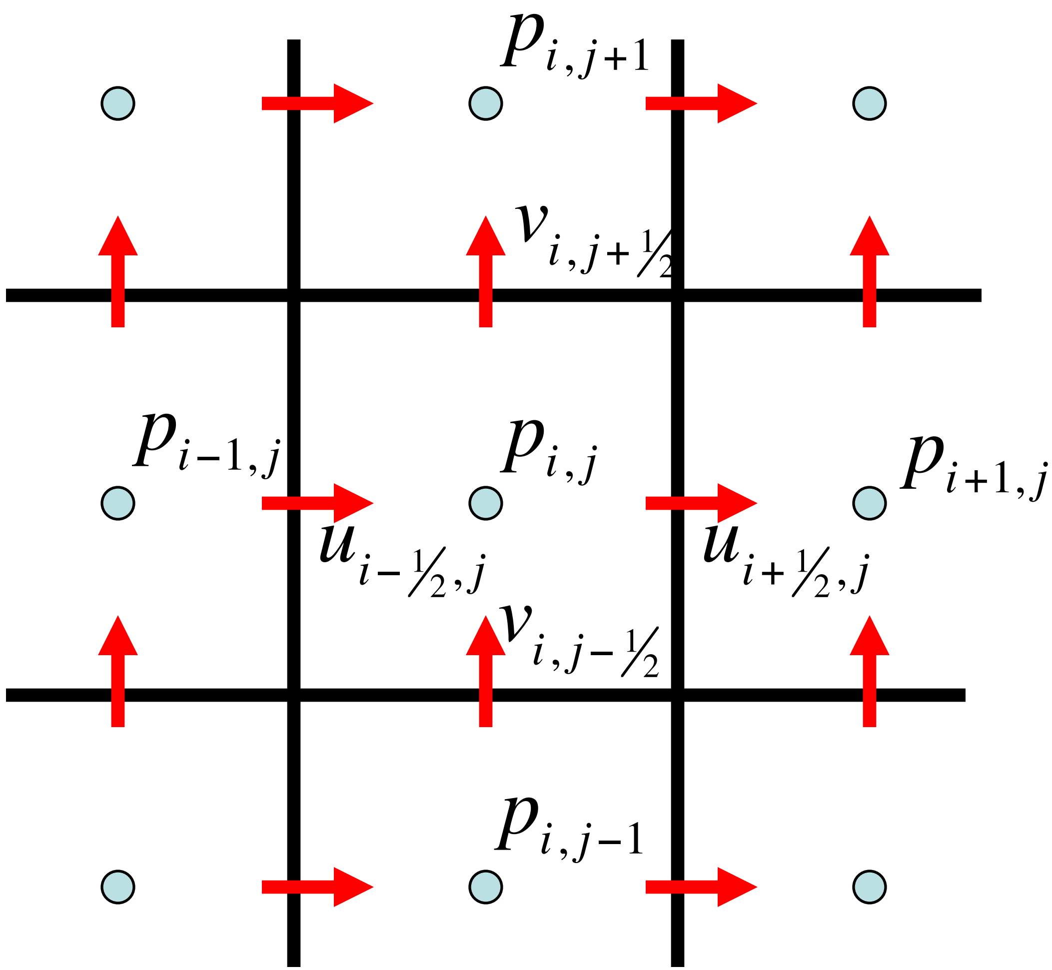
$$\frac{\partial u}{\partial x}(x_i) \approx \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x}$$

- Problem solved!

The MAC Grid

- From the Marker-and-Cell (MAC) method [Harlow&Welch'65]
- A particular staggering of variables in 2D/3D that works well for incompressible fluids:
  - Grid cell (i,j,k) has pressure $\mathbf{p}_{i,j,k}$ at its center
  - x-part of velocity $\mathbf{u}_{i+1/2,jk}$ in middle of x-face between grid cells (i,j,k) and (i+1,j,k)
  - y-part of velocity $\mathbf{v}_{i,j+1/2,k}$ in middle of y-face
  - z-part of velocity $\mathbf{w}_{i,j,k+1/2}$ in middle of z-face

MAC Grid in 2D



Array storage

- Then for a nx X ny X nz grid, we store them as 3D arrays:
  - p[nx, ny, nz]
  - u[nx+1, ny, nz]
  - v[nx, ny+1, nz]
  - w[nx, ny, nz+1]
- And translate indices in code, e.g.

$$u_{i+\frac{1}{2},j,k} \equiv u[i+1,j,k]$$

The downside

- Not having all quantities at the same spot makes some algorithms a pain
  - Even interpolation of velocities for pushing particles is annoying
- One strategy: switch back and forth (colocated/staggered) by averaging
- My philosophy: avoid averaging as much as possible!

Advection Algorithms

Advecting Quantities

- The goal is to solve $$\frac{Dq}{Dt} = 0$$ "the advection equation" for any grid quantity q
  - in particular, the components of velocity
- Rather than directly approximate spatial term, we'll use the Lagrangian notion of advection directly
- We're on an Eulerian grid, though, so the result will be called "semi-Lagrangian"
  - Introduced to graphics by [Stam'99]

Semi-Lagrangian Advection

- $Dq/Dt=0$ says q doesn't change as you follow the fluid.
- So $$q(x^{new}, t^{new}) = q(x^{old}, t^{old})$$
- We want to know q at each grid point at t^{new} (that is, we're interested in $x^{new}=x_{ijk}$)
- So we just need to figure out
  x^{old} (where fluid at x_{ijk} came from)
  and
  $q(x^{old})$ (what value of q was there before)

Finding x^{old}

- We need to trace backwards through the velocity field.
- Up to $O(\Delta t)$ we can assume velocity field constant over the time step
- The simplest estimate is then

$$x^{old} \approx x_{ijk} - \Delta t\, \vec{u}(x_{ijk})$$

  - I.e. tracing through the time-reversed flow with one step of Forward Euler
  - Other ODE methods can (and **should**) be used

Which u to Use?

- Note that we only want to advect quantities in an incompressible velocity field
  - Otherwise the quantity gets compressed (often an obvious unphysical artifact)
- For example, when we advect u, v, and w themselves, we use the old incompressible values stored in the grid
  - Do not update as you go!

Finding $q(x^{old})$

- Odds are when we trace back from a grid point to x^{old} we won't land on a grid point
  - So we don't have an old value of q there
- Solution: interpolate from nearby grid points
  - Simplest method: bi/trilinear interpolation
  - Know your grid: be careful to get it right for staggered quantities!

Boundary Conditions

- What if x^{old} isn't in the fluid? (or a nearest grid point we're interpolating from is not in the fluid?)
- Solution: **extrapolate** from boundary of fluid
  - Extrapolate before advection, to all grid points in the domain that aren't fluid
- ALSO: if fluid can move to a grid point that isn't fluid now, make sure to do semi-Lagrangian advection there
  - Use the extrapolated velocity

Body Forces

Integrating Body Forces

- Gravity vector or volumetric animator forces:

$$\frac{\partial \vec{u}}{\partial t} = \vec{g}$$

- Simplest scheme: at every grid point just add

$$\vec{u}^{*} = \vec{u}^{advected} + \Delta t\, \vec{g}$$

Making Fluids Incompressible

The Continuous Version

- We want to find a pressure p so that the updated velocity:

$$\vec{u}^{n+1} = \vec{u}^{*} - \frac{\Delta t}{\rho} \nabla p$$

is divergence free:

$$\nabla \bullet \vec{u}^{n+1} = 0$$

while respecting the boundary conditions:

$$\vec{u} \bullet \hat{n} = \vec{u}_{solid} \bullet \hat{n} \quad \text{at solid walls}$$

$$p = 0 \quad \text{at the free surface}$$

The Poisson Problem

- Plug in the pressure update formula into incompressibility:

$$\nabla \bullet \left(\vec{u}^{\,*} - \frac{\Delta t}{\rho} \nabla p \right) = 0$$

- Turns into a “Poisson equation” for pressure:

$$\begin{cases} \nabla \bullet \left(\dfrac{\Delta t}{\rho} \nabla p \right) = \nabla \bullet \vec{u}^{\,*} & \\ \dfrac{\Delta t}{\rho} \nabla p \bullet \hat{n} = (\vec{u} - \vec{u}_{solid}) \bullet \hat{n} & \text{at solid walls} \\ p = 0 & \text{at the free surface} \end{cases}$$

Discrete Pressure Update

- The discrete pressure update on the MAC grid:

$$u^{n+1}_{i+1/2\,jk} = u^{*}_{i+1/2\,jk} - \frac{\Delta t}{\rho}\left(\frac{p_{i+1\,jk} - p_{ijk}}{\Delta x}\right)$$

$$v^{n+1}_{ij+1/2\,k} = v^{*}_{ij+1/2\,k} - \frac{\Delta t}{\rho}\left(\frac{p_{ij+1\,k} - p_{ijk}}{\Delta x}\right)$$

$$w^{n+1}_{ijk+1/2} = w^{*}_{ijk+1/2} - \frac{\Delta t}{\rho}\left(\frac{p_{ijk+1} - p_{ijk}}{\Delta x}\right)$$

Discrete Divergence

- The discretized incompressibility condition on the new velocity (estimated at i,j,k):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

$$\left(\frac{u^{n+1}_{i+1/2\,jk} - u^{n+1}_{i-1/2\,jk}}{\Delta x} + \frac{v^{n+1}_{ij+1/2\,k} - v^{n+1}_{ij-1/2\,k}}{\Delta x} + \cdots \right) = 0$$

Discrete Pressure Equations

- Substitute in pressure update formula to discrete divergence
- In each fluid cell (i,j,k) get an equation:

$$\frac{1}{\Delta x}\left[\begin{array}{l}\left(u_{i+\frac{1}{2}jk}-\frac{\Delta t}{\rho}\frac{p_{i+1jk}-p_{ijk}}{\Delta x}\right)-\left(u_{i-\frac{1}{2}jk}-\frac{\Delta t}{\rho}\frac{p_{ijk}-p_{i-1jk}}{\Delta x}\right)\\+\left(v_{ij+\frac{1}{2}k}-\frac{\Delta t}{\rho}\frac{p_{ij+1k}-p_{ijk}}{\Delta x}\right)-\left(v_{ij-\frac{1}{2}k}-\frac{\Delta t}{\rho}\frac{p_{ijk}-p_{ij-1k}}{\Delta x}\right)\\+\left(w_{ijk+\frac{1}{2}}-\frac{\Delta t}{\rho}\frac{p_{ijk+1}-p_{ijk}}{\Delta x}\right)-\left(w_{ijk-\frac{1}{2}}-\frac{\Delta t}{\rho}\frac{p_{ijk}-p_{ijk-1}}{\Delta x}\right)\end{array}\right]=0$$

Linear Equations

- End up with a sparse set of linear equations to solve for pressure
  - Matrix is symmetric positive (semi-)definite
- In 3D on large grids, direct methods unsatisfactory
- Instead use Preconditioned Conjugate Gradient, with Incomplete Cholesky preconditioner
- See course notes for full details (pseudo-code)
- Residual is how much divergence there is in u^{n+1}
  - Iterate until satisfied it's small enough

Voxelization is Suboptimal

- Free surface artifacts:
  - Waves less than a grid cell high aren't "seen" by the fluid solver – thus they don't behave right
  - Left with strangely textured surface
- Solid wall artifacts:
  - If boundary not grid-aligned, O(1) error – it doesn't even go away as you refine!
  - Slopes are turned into stairs, water will pool on artificial steps.
- More on this later…