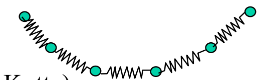
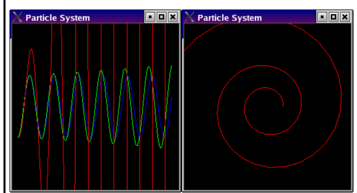
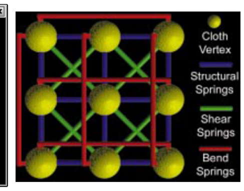
Navier-Stokes
&
Flow Simulation

Last Time?

- Spring-Mass Systems
- Numerical Integration (Euler, Midpoint, Runge-Kutta)
- Modeling string, hair, & cloth

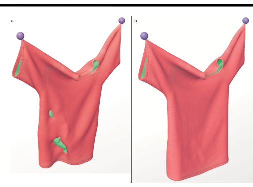


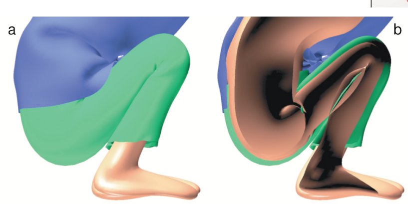




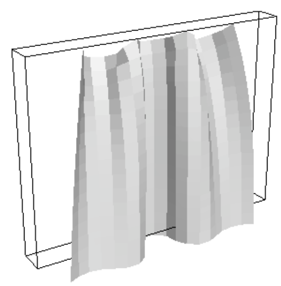
Optional Reading for Last Time:

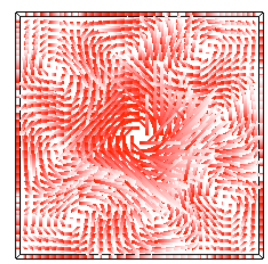
- Baraff, Witkin & Kass
  Untangling Cloth
  SIGGRAPH 2003





HW2: Cloth & Fluid Simulation





Today

- Flow Simulations in Computer Graphics
  – water, smoke, viscous fluids
- Navier-Stokes Equations
  – incompressibility, conservation of mass
  – conservation of momentum & energy
- Fluid Representations
- Basic Algorithm
- Data Representation

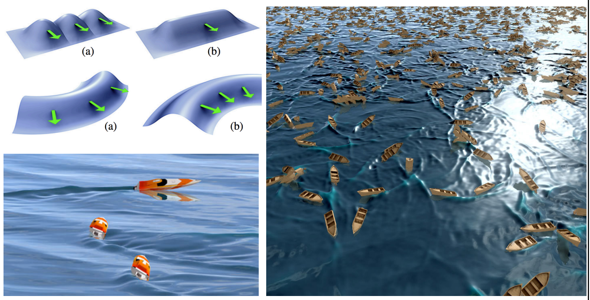
Flow Simulations in Graphics

- Random velocity fields
  – with averaging to get simple background motion
- Shallow water equations
  – height field only, can't represent crashing waves, etc.
- Full Navier-Stokes

- *note: typically we ignore surface tension and focus on macroscopic behavior*

Heightfield Wave Simulation

- Cem Yuksel, Donald H. House, and John Keyser, "Wave Particles", SIGGRAPH 2007



Flow in a Voxel Grid

- conservation of mass:

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

For a single phase simulation (e.g., water only, air only)

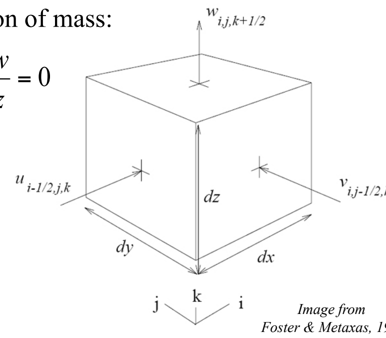


Image from Foster & Metaxas, 1996

Navier-Stokes Equations

- conservation of momentum:

gravity (& other external forces)
pressure
viscosity

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} = -\frac{\partial p}{\partial x} + g_x + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

$$\frac{\partial v}{\partial t} + \frac{\partial vu}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial vw}{\partial z} = -\frac{\partial p}{\partial y} + g_y + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right)$$

$$\frac{\partial w}{\partial t} + \frac{\partial wu}{\partial x} + \frac{\partial wv}{\partial y} + \frac{\partial w^2}{\partial z} = -\frac{\partial p}{\partial z} + g_z + \nu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right)$$

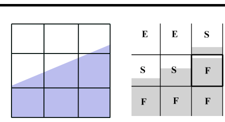
acceleration
Convection: internal movement in a fluid (e.g., caused by variation in density due to a transfer of heat)
drag

Today

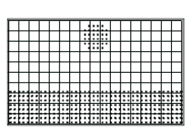
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Basic Algorithm
- Data Representation

Modeling the Air/Water Surface

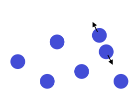
- Volume-of-fluid tracking



- Marker and Cell (MAC)



- Smoothed Particle Hydrodynamics (SPH)

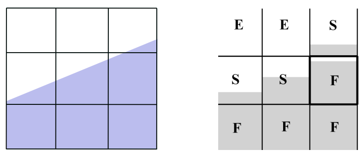


Comparing Representations

- How do we render the resulting surface?
- Are we guaranteed not to lose mass/volume? (is the simulation incompressible?)
- How is each affected by the grid resolution and timestep?
- Can we guarantee stability?

Volume-of-fluid-tracking

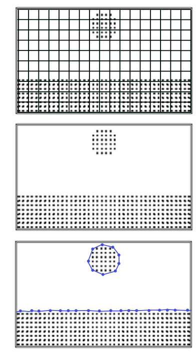
- Each cell stores a scalar value indicating that cell's "full"-ness



\+ preserves volume
\- difficult to render
\- very dependent on grid resolution

Marker and Cell (MAC)

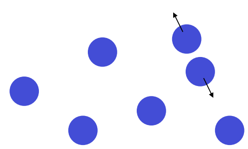
- Harlow & Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface", *The Physics of Fluids*, 1965.
- *Volume marker particles* identify location of fluid within the volume
- (Optional) *surface marker particles* track the detailed shape of the fluid/air boundary
- But… marker particles don't have or represent a mass/volume of fluid



\+ rendering
\- does not preserve volume
\- dependent on grid resolution

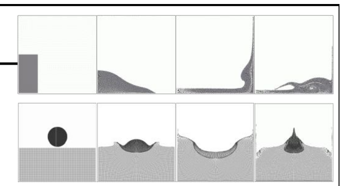
Smoothed Particle Hydrodynamics (SPH)

- Each particle represents a specific mass of fluid
- "Meshless" (no voxel grid)
- Repulsive forces between neighboring particles maintain constant volume



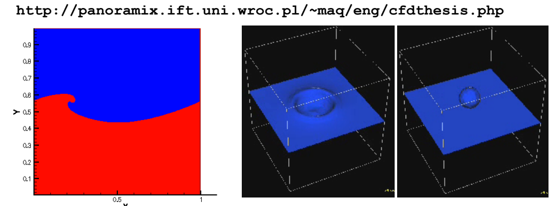
\+ no grid resolution concerns (now accuracy depends on number/size of particles)
\+ volume is preserved*
\+ render similar to MAC
\- much more expensive (particle-particle interactions)

Demos



- Nice Marker and Cell (MAC) videos at:

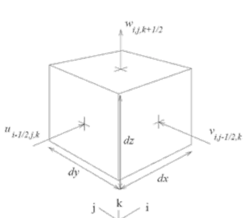
http://panoramix.ift.uni.wroc.pl/~maq/eng/cfdthesis.php

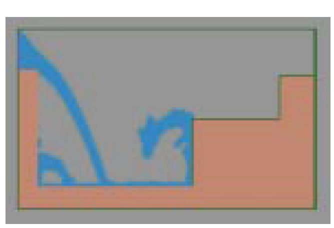


http://mme.uwaterloo.ca/~fslien/free_surface/free_surface.htm

Reading for Today

- "Realistic Animation of Liquids", Foster & Metaxas, 1996





Today

- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Basic Algorithm
- Data Representation

Each Grid Cell Stores:

- Velocity at the cell faces (offset grid)
- Pressure
- List of particles

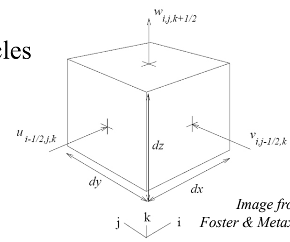


Image from Foster & Metaxas, 1996

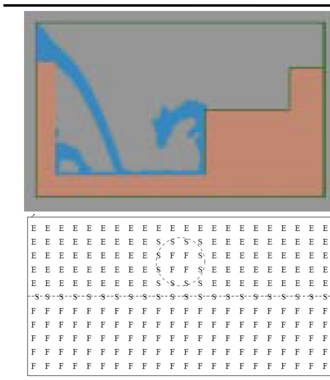
Initialization

- Choose a voxel resolution
- Choose a particle density
- Create grid & place the particles
- Initialize pressure & velocity of each cell
- Set the viscosity & gravity
- Choose a timestep & go!

At each Timestep:

- Identify which cells are Empty, Full, or on the Surface
- Compute new velocities
- Adjust the velocities to maintain an incompressible flow
- Move the particles
  - Interpolate the velocities at the faces
- Render the geometry and repeat!

Empty, Surface & Full Cells



Images from Foster & Metaxas, 1996

At each Timestep:

- Identify which cells are Empty, Full, or on the Surface
- Compute new velocities
- Adjust the velocities to maintain an incompressible flow
- Move the particles
  - Interpolate the velocities at the faces
- Render the geometry and repeat!

Compute New Velocities

$$\begin{aligned}
\tilde{u}_{i+1/2,j,k} = u_{i+1/2,j,k} + \delta t\{&(1/\delta x)[(u_{i,j,k})^2 - (u_{i+1,j,k})^2] \\
&+(1/\delta y)[(uv)_{i+1/2,j-1/2,k} - (uv)_{i+1/2,j+1/2,k}] \\
&+(1/\delta z)[(uw)_{i+1/2,j,k-1/2} - (uw)_{i+1/2,j,k+1/2}] + g_x \\
&+(1/\delta x)(p_{i,j,k} - p_{i+1,j,k}) + (\nu/\delta x^2)(u_{i+3/2,j,k} \\
&-2u_{i+1/2,j,k} + u_{i-1/2,j,k}) + (\nu/\delta y^2)(u_{i+1/2,j+1,k} \\
&-2u_{i+1/2,j,k} + u_{i+1/2,j-1,k}) + (\nu/\delta z^2)(u_{i+1/2,j,k+1} \\
&-2u_{i+1/2,j,k} + u_{i+1/2,j,k-1})\},
\end{aligned}$$

Note: some of these values are the *average velocity* within the cell rather than the velocity at a cell face

At each Timestep:

- Identify which cells are Empty, Full, or on the Surface
- Compute new velocities
- Adjust the velocities to maintain an incompressible flow
- Move the particles
  - Interpolate the velocities at the faces
- Render the geometry and repeat!

Adjusting the Velocities

- Calculate the *divergence* of the cell (the extra in/out flow)
- The divergence is used to update the *pressure* within the cell
- Adjust each face velocity uniformly to bring the divergence to zero
- Iterate across the entire grid until divergence is < ε

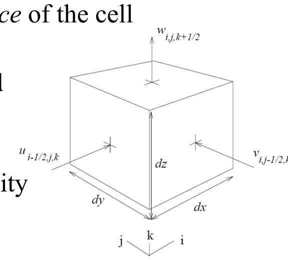
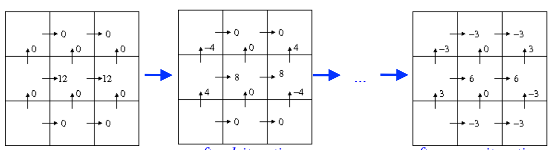


Image from Foster & Metaxas, 1996

Calculating/Eliminating Divergence



initial flow field — *after 1 iteration* — *after many iterations*
(results will vary with different calculation order)

0	0	0
–12	0	+12
0	0	0

corresponding divergence

–4	0	4
0	0	0
–4	0	4

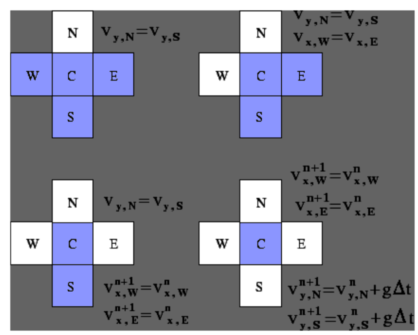
corresponding divergence

0	0	0
0	0	0
0	0	0

corresponding divergence

Handing Free Surface with MAC

- Divergence in surface cells:
  - Is divided equally amongst neighboring empty cells
  - Or other similar strategies?
- Zero out the divergence & pressure in empty cells

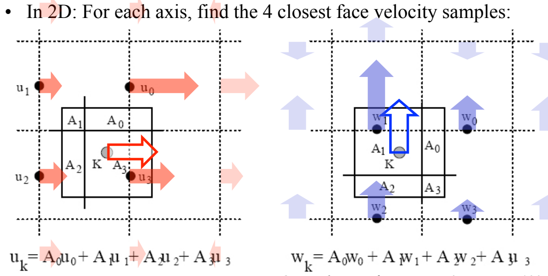


At each Timestep:

- Identify which cells are Empty, Full, or on the Surface
- Compute new velocities
- Adjust the velocities to maintain an incompressible flow
- Move the particles
  - Interpolate the velocities at the faces
- Render the geometry and repeat!

Velocity Interpolation

- In 2D: For each axis, find the 4 closest face velocity samples:

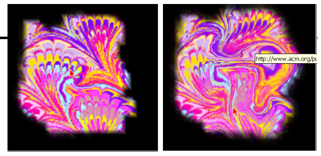


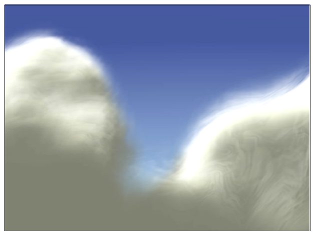
Original image from Foster & Metaxas, 1996

- In 3D… find 8 closest face velocities in each dimension
- NOTE: The complete implementation isn't particularly elegant

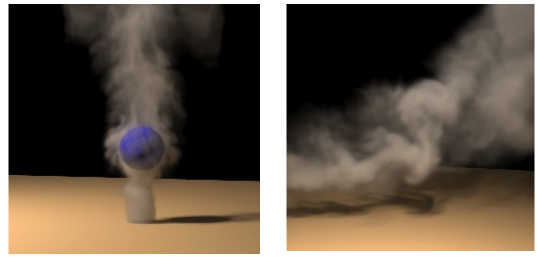
Stable Fluids

- "Stable Fluids", Jos Stam, SIGGRAPH 1999.





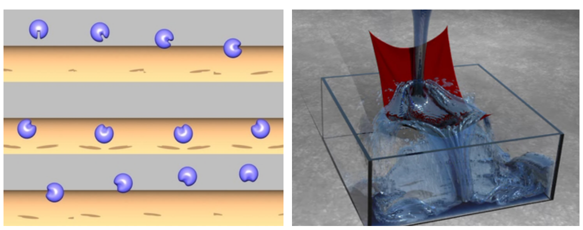
Smoke Simulation & Rendering



"Visual Simulation of Smoke"
Fedkiw, Stam & Jensen
SIGGRAPH 2001

Ron Fedkiw http://physbam.stanford.edu/~fedkiw/

- Enright, Marschner, & Fedkiw, "Animation and Rendering of Complex Water Surfaces", SIGGRAPH 2002.
- Guendelman, Selle, Losasso, & Fedkiw, "Coupling Water and Smoke to Thin Deformable and Rigid Shells", SIGGRAPH 2005.



Reading for Friday:

"Melting and Flowing"
Carlson, Mucha, Van Horn III & Turk
Symposium on Computer Animation 2002

Reading for Friday:

"Real-time Large-deformation Substructuring",
Barbic & Zhao, SIGGRAPH 2011