Navier-Stokes & Flow Simulation

Last Time?
- Implicit Surfaces
- Marching Cubes/Tetras
- Collision Detection & Response
- Conservative Bounding Regions

Flow Simulations in Computer Graphics
- 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

Flow in a Voxel Grid
- conservation of mass:
  \[ \frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{v}}{\partial y} + \frac{\partial \hat{w}}{\partial z} = 0 \]

Navier-Stokes Equations
- conservation of momentum:
  \[
  \begin{align*}
  \frac{\partial \hat{u}}{\partial t} + \frac{\partial \hat{u}^2}{\partial x} + \frac{\partial \hat{u} \hat{v}}{\partial y} + \frac{\partial \hat{u} \hat{w}}{\partial z} &= -\frac{\partial p}{\partial x} + \hat{u}_g + \nu(\frac{\partial^2 \hat{u}}{\partial x^2} + \frac{\partial^2 \hat{u}}{\partial y^2} + \frac{\partial^2 \hat{u}}{\partial z^2}) \\
  \frac{\partial \hat{v}}{\partial t} + \frac{\partial \hat{v}^2}{\partial y} + \frac{\partial \hat{u} \hat{v}}{\partial x} + \frac{\partial \hat{v} \hat{w}}{\partial z} &= -\frac{\partial p}{\partial y} + \hat{v}_g + \nu(\frac{\partial^2 \hat{v}}{\partial x^2} + \frac{\partial^2 \hat{v}}{\partial y^2} + \frac{\partial^2 \hat{v}}{\partial z^2}) \\
  \frac{\partial \hat{w}}{\partial t} + \frac{\partial \hat{w}^2}{\partial z} + \frac{\partial \hat{u} \hat{w}}{\partial x} + \frac{\partial \hat{v} \hat{w}}{\partial y} &= -\frac{\partial p}{\partial z} + \hat{w}_g + \nu(\frac{\partial^2 \hat{w}}{\partial x^2} + \frac{\partial^2 \hat{w}}{\partial y^2} + \frac{\partial^2 \hat{w}}{\partial z^2})
  \end{align*}
\]
Today

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

Modeling the Water Surface

• Volume-of-fluid tracking
  – a scalar saying how “full” each cell is
• Particle In Cell (PIC)
  – the particles have mass
• Marker and Cell (MAC)
  – the particles don’t affect computation, just identify
    which cells the surface passes through
  – Harlow & Welch, “Numerical calculation of time-dependent
    viscous incompressible flow of fluid with free surface”,

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?

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

Each Grid Cell Stores:

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

Today

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

Modeling the Water Surface

• Volume-of-fluid tracking
  – a scalar saying how “full” each cell is
• Particle In Cell (PIC)
  – the particles have mass
• Marker and Cell (MAC)
  – the particles don’t affect computation, just identify
    which cells the surface passes through
  – Harlow & Welch, “Numerical calculation of time-dependent
    viscous incompressible flow of fluid with free surface”,
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

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

\[ \hat{u}_{i+1/2,j,k} = u_{i+1/2,j,k} + \frac{\Delta t}{\Delta x} \left[ \frac{1}{2} \left( \frac{u_{i+1,j,k+1/2} - u_{i+1,j,k-1/2}}{\Delta y} \right) + \frac{1}{2} \left( \frac{u_{i+1,j+1/2} - u_{i+1,j-1/2}}{\Delta z} \right) \right] \]

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

Handing Boundaries with MAC

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!

Efficient Smoke Simulation

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

Solid/Liquid: Time-varying viscosity

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