Navier-Stokes & Flow Simulation

<table>
<thead>
<tr>
<th>Last Time?</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Spring-Mass Systems</td>
</tr>
<tr>
<td>• Numerical Integration (Euler, Midpoint, Runge-Kutta)</td>
</tr>
<tr>
<td>• Modeling string, hair, &amp; cloth</td>
</tr>
</tbody>
</table>

Optional Reading for Last Time:

- Baraff, Witkin & Kass  
  *Untangling Cloth*  
  SIGGRAPH 2003

<table>
<thead>
<tr>
<th>HW2: Cloth &amp; Fluid Simulation</th>
</tr>
</thead>
</table>

Today

<table>
<thead>
<tr>
<th>Flow Simulations in Computer Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• water, smoke, viscous fluids</td>
</tr>
<tr>
<td>• Navier-Stokes Equations</td>
</tr>
<tr>
<td>- incompressibility, conservation of mass</td>
</tr>
<tr>
<td>- conservation of momentum &amp; energy</td>
</tr>
<tr>
<td>• Fluid Representations</td>
</tr>
<tr>
<td>• Basic Algorithm</td>
</tr>
<tr>
<td>• Data Representation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flow Simulations in Graphics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Random velocity fields</td>
</tr>
<tr>
<td>- with averaging to get simple background motion</td>
</tr>
<tr>
<td>• Shallow water equations</td>
</tr>
<tr>
<td>- height field only, can’t represent crashing waves, etc.</td>
</tr>
<tr>
<td>• Full Navier-Stokes</td>
</tr>
<tr>
<td>• note: typically we ignore surface tension and focus on macroscopic behavior</td>
</tr>
</tbody>
</table>
**Heightfield Wave Simulation**


**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)

**Navier-Stokes Equations**

- conservation of momentum:
  \[
  \begin{align*}
  \frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uw}{\partial y} + \frac{\partial uw}{\partial z} &= -\frac{\partial p}{\partial x} + \rho g_x + \nu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \\
  \frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial vw}{\partial z} &= -\frac{\partial p}{\partial y} + \rho g_y + \nu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) \\
  \frac{\partial w}{\partial t} + \frac{\partial w}{\partial x} + \frac{\partial w}{\partial y} + \frac{\partial w^2}{\partial z} &= -\frac{\partial p}{\partial z} + \rho g_z + \nu \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right)
  \end{align*}
  \]

- pressure
- gravity (and other external forces)
- viscosity
- 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"

<table>
<thead>
<tr>
<th>E</th>
<th>E</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

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

Marker and Cell (MAC)

- *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://www.waterloo.ca/~fslien/free_surface/free_surface.htm](http://www.waterloo.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

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

\[
\begin{align*}
\delta u_{i+1/2,j,k} &= u_{i+1,j,k} + \delta t \left( \frac{1}{\partial x} \left( \left( u_{i+1,j,k} - u_{i,j,k} \right)^{2} \right) \right) \\
&- \delta t \left( \frac{1}{\partial y} \left( \left( u_{i+1,j+1/2,k} - u_{i+1,j-1/2,k} \right)^{2} \right) \right) \\
&+ \delta t \left( \frac{1}{\partial z} \left( \left( u_{i+1,j,k+1/2} - u_{i+1,j,k-1/2} \right)^{2} \right) \right) + g_x \\
&+ \delta t \left( \frac{1}{\partial x} \left( p_{i,j,k} - p_{i+1,j,k} \right) + \nu \delta^2 \left( u_{i+1,j,k} \right) \right) \\
&- 2\delta u_{i+1/2,j,k} + u_{i,j,k} + \nu \left( u_{i+1,j+1/2,k} + u_{i+1,j-1/2,k} \right) \\
&- 2\delta u_{i+1/2,j,k+1} + u_{i+1,j,k+1} + \nu \left( u_{i+1,j+1/2,k+1} + u_{i+1,j+1/2,k-1} \right) \\
&- 2\delta u_{i+1/2,j,k-1} + u_{i+1,j,k-1} + \nu \left( u_{i+1,j+1/2,k-1} + u_{i+1,j-1/2,k-1} \right)
\end{align*}
\]

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

Calculating/Eliminating Divergence

<table>
<thead>
<tr>
<th>Initial flow field</th>
<th>after 1 iteration</th>
<th>after many iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 1</td>
<td>1 0 0 1</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>-12 1 -12</td>
<td>0 1 0 1</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>-1 0 1 0</td>
<td>0 0 0 0</td>
</tr>
</tbody>
</table>

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

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


Smoke Simulation & Rendering

“Visual Simulation of Smoke”
Fedkiw, Stam & Jensen
SIGGRAPH 2001

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


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