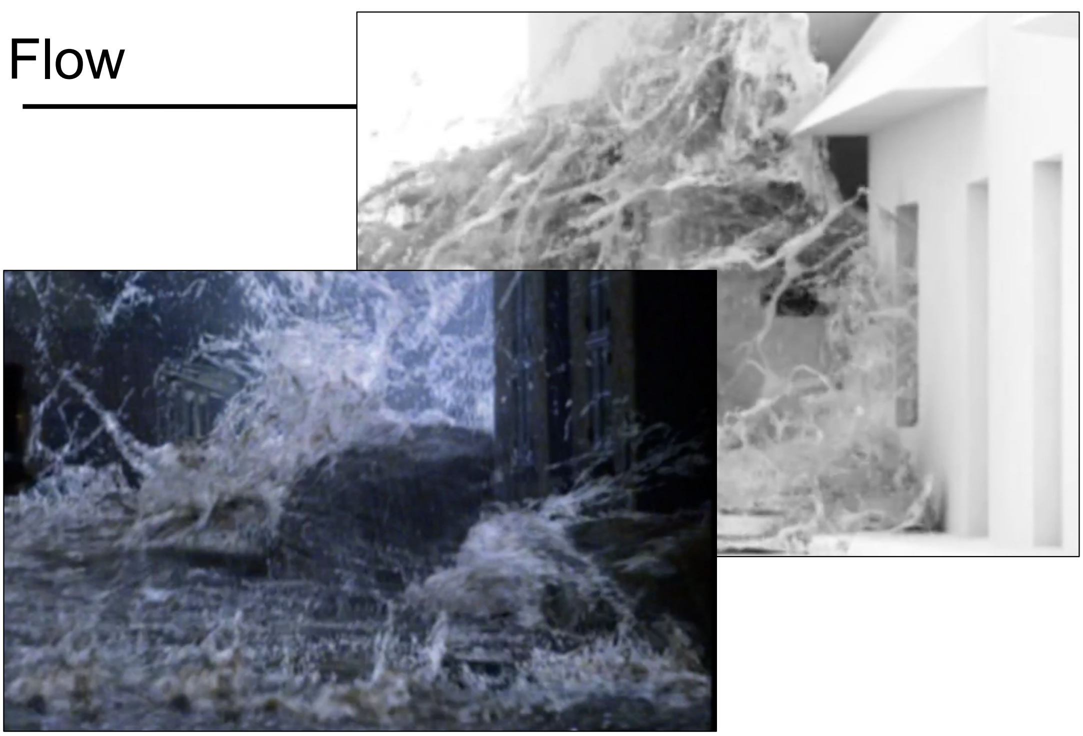
Navier-Stokes & Flow Simulation

Flow

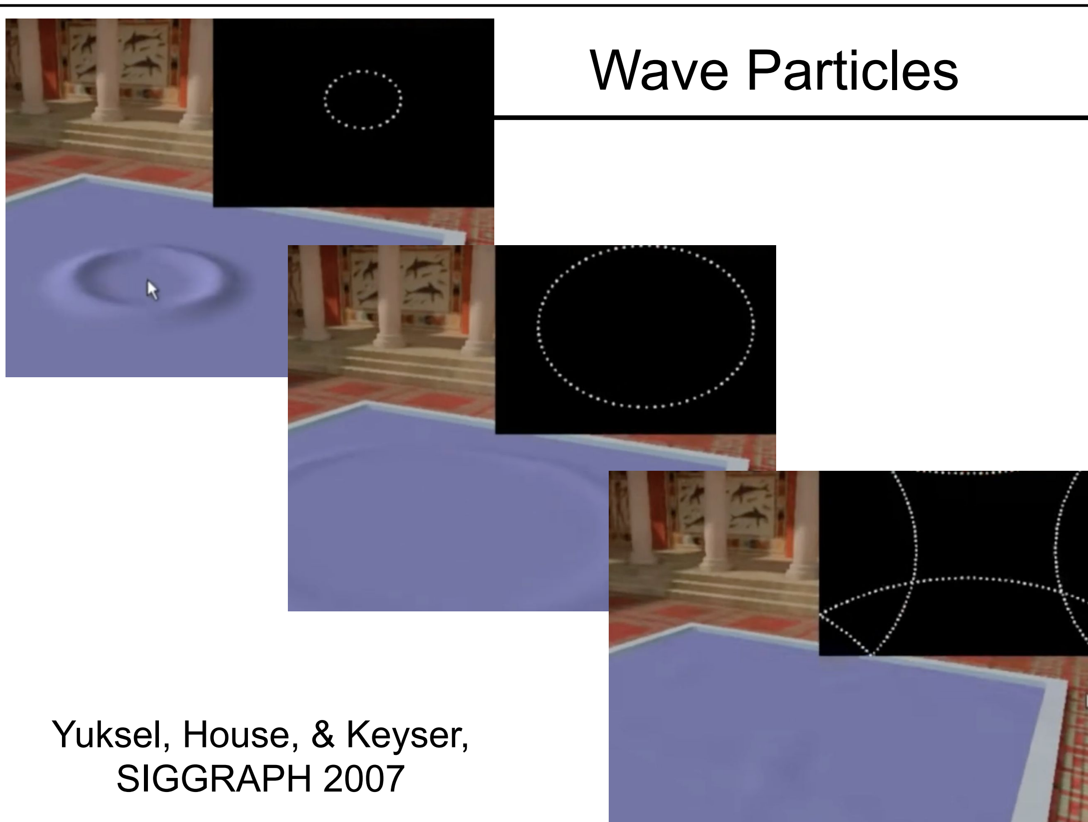


Scanline Flowline VFX, SIGGRAPH 2006

Flow

Scanline Flowline VFX, SIGGRAPH 2006

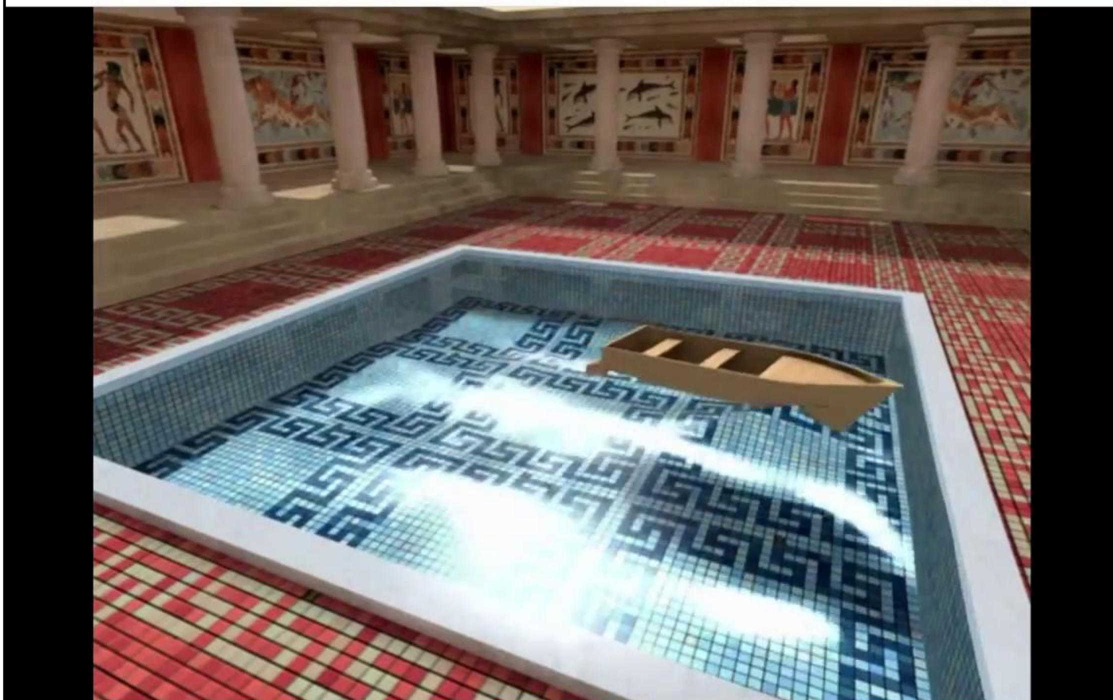
Wave Particles



Yuksel, House, & Keyser, SIGGRAPH 2007

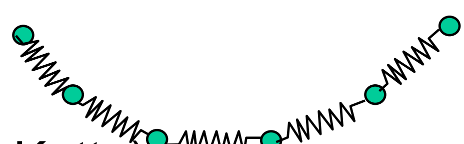
Wave Particles

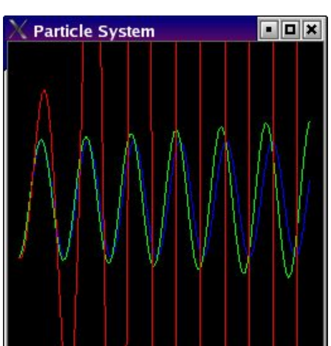
Yuksel, House, & Keyser,
SIGGRAPH 2007

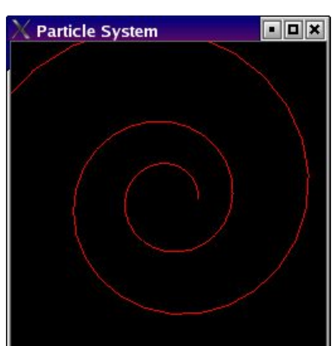


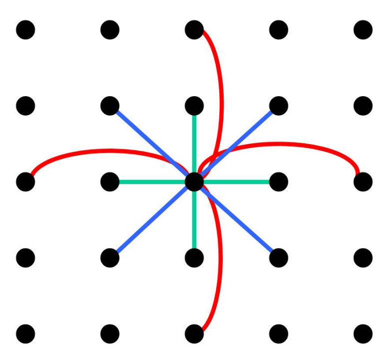
Last Time?

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







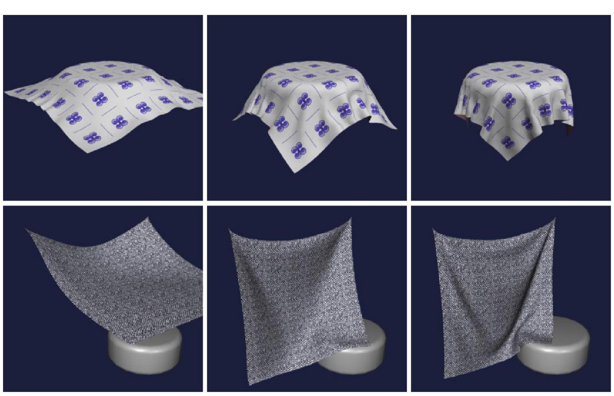


Today

- Papers for Today
  - How to read a research paper
  - Components of a well written paper
- Worksheet: 2D Mass Spring System
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm
- Papers for Next Time...

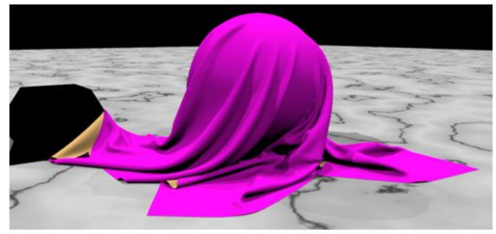
Papers for Today *(pick one)*

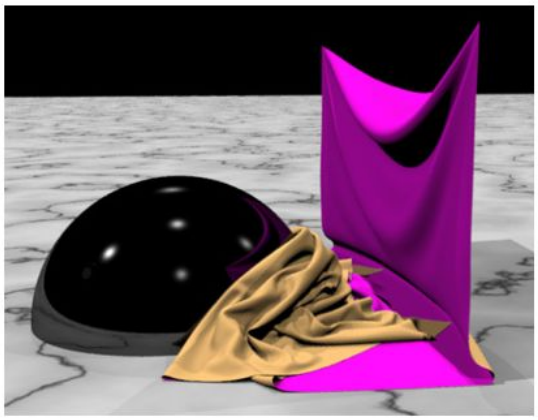
"Large Steps in Cloth Simulation",
Baraff & Witkin,
SIGGRAPH 1998

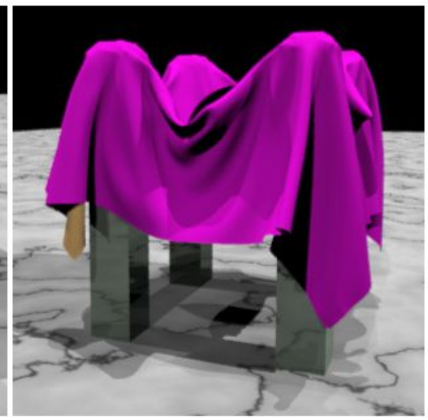


Papers for Today *(pick one)*

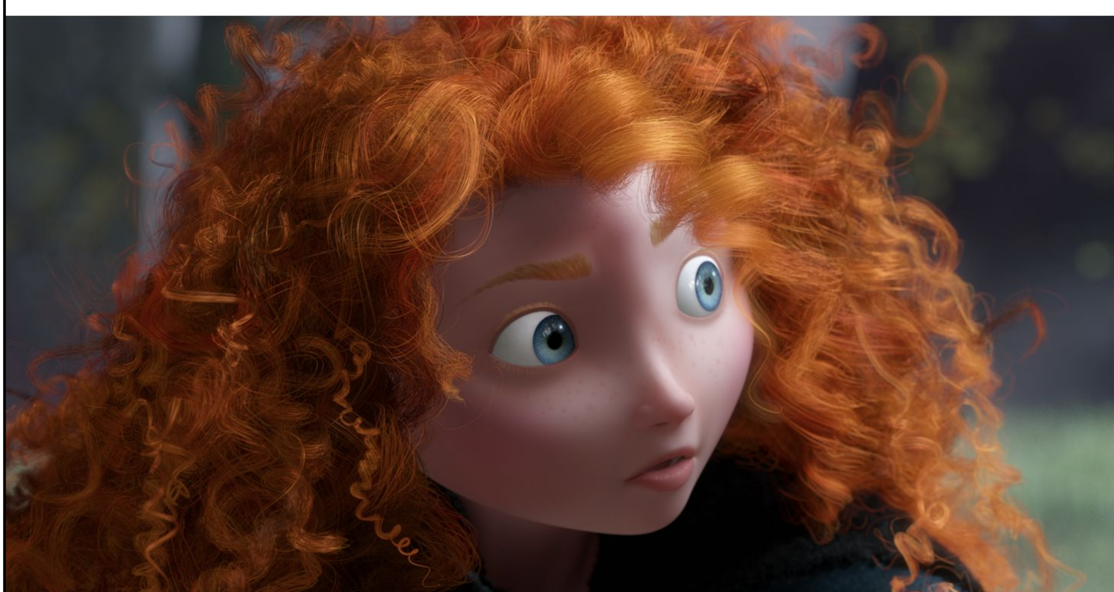
"Robust Treatment of Collisions, Contact and Friction for Cloth Animation", Bridson, Fedkiw & Anderson, SIGGRAPH 2002







Papers for Today *(pick one)*

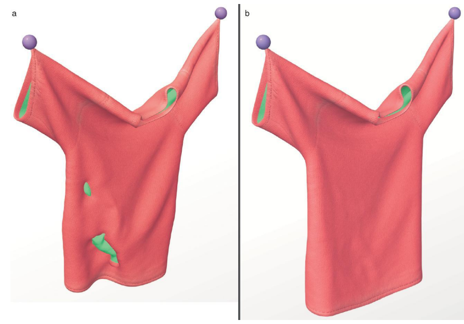


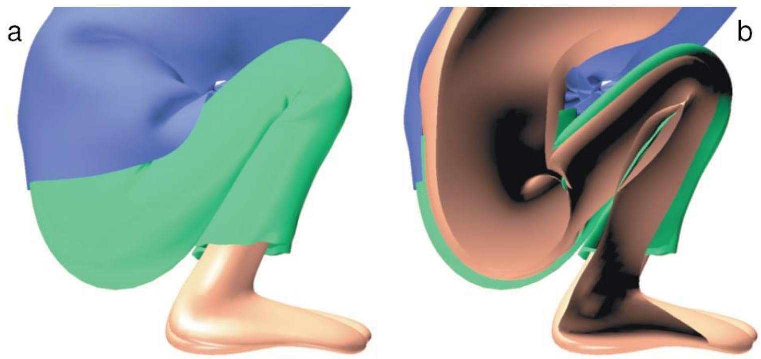
"Artistic Simulation of Curly Hair", Iben, Meyer, Petrovic, Soares, Anderson, and Witkin, Symposium on Computer Animation 2013

Cloth in Practice (w/ Animation)

OPTIONAL READING

- Baraff, Witkin & Kass
  Untangling Cloth
  SIGGRAPH 2003





How to read a research paper?

-

How to read a research paper?

(especially an advanced paper in a new area)

- Multiple readings are often necessary
- Don't necessarily read from front to back
- Lookup important terms
- Target application & claimed contributions
- Experimental procedure
- How well results & examples support the claims
- Scalability of the technique (Big O Notation)
- Limitations of technique, places for future research
- Possibilities for hybrid systems with other work

Components of a well-written research paper?

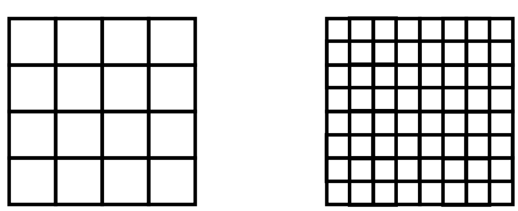
-

Components of a well-written research paper?

- Motivation/context/related work
- Contributions of this work
- Clear description of algorithm
  - Sufficiently-detailed to allow work to be reproduced
  - Work is theoretically sound (hacks/arbitrary constants discouraged)
- Results
  - well chosen examples
  - clear tables/illustrations/visualizations
- Conclusions
  - limitations of the method are clearly stated

The Discretization Problem

- What happens if we discretize our cloth more finely, or with a different mesh structure?



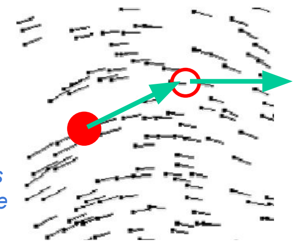
- Do we get the same behavior?
  - Usually not! It takes a lot of effort to design a scheme that does not depend on the discretization.
- Using (explicit) Euler, how many timesteps before a force propagates across the mesh?

Explicit vs. Implicit Integration

- With an explicit/forward integration scheme:

  $\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\mathbf{g}(y_k)$ *The future state (position & velocity) of this particle is a function of the current state of the particle.*

  we must use a very small timestep to simulate *stable, stiff* cloth.



- Alternatively we can use an implicit/backwards scheme:

  $\mathbf{y}_{k+1} = \mathbf{y}_k + h\,\mathbf{g}(y_{k+1})$

  $\mathbf{y}_k = \mathbf{y}_{k+1} - h\,\mathbf{g}(y_{k+1})$

  The future state of this particle depends on the current state AND the future state.

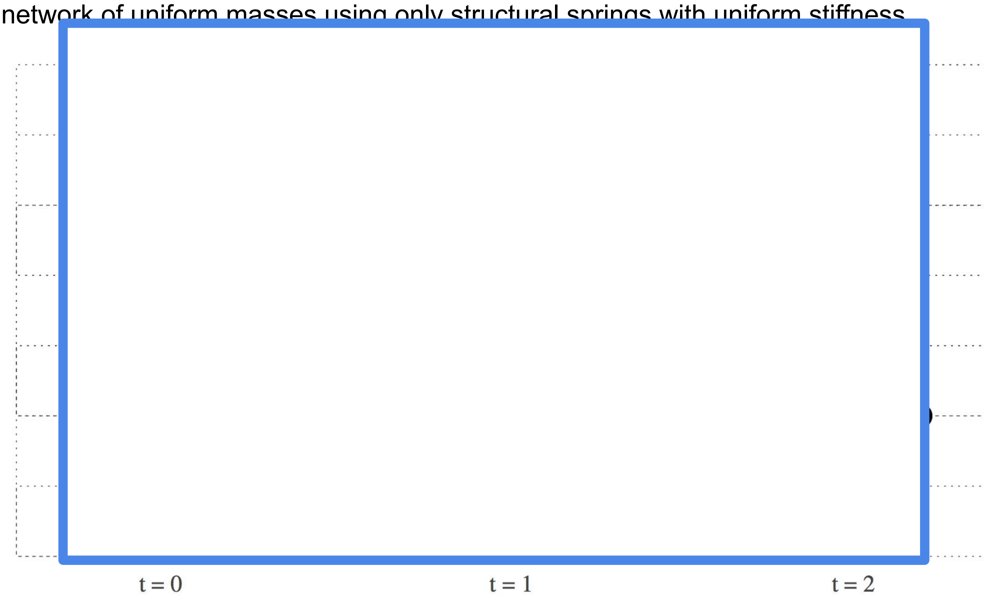
  Solving one step is much more expensive (Newton's Method, Conjugate Gradients, …) but overall faster than the thousands of explicit timesteps required for very stiff springs.
- Larger timesteps are possible with implicit methods.

Today

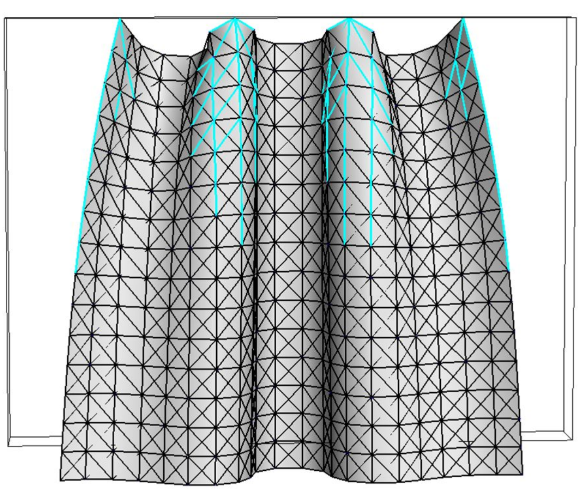
- Papers for Today
- Worksheet: 2D Mass Spring System
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm
- Papers for Next Time...

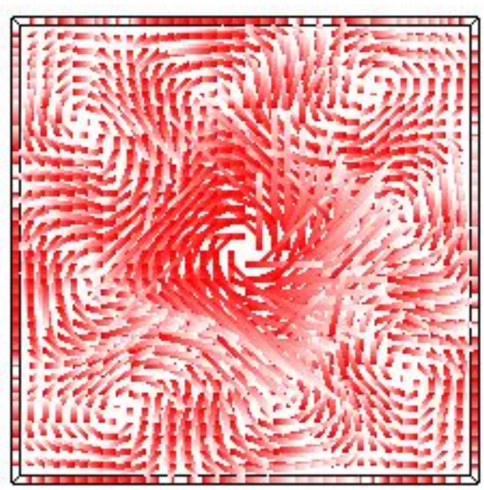
Pop Worksheet!

Sketch the first few frames of a 2D explicit Euler mass-spring simulation for a 2x3 cloth network of uniform masses using only structural springs with uniform stiffness.



HW2: Cloth & Fluid Simulation



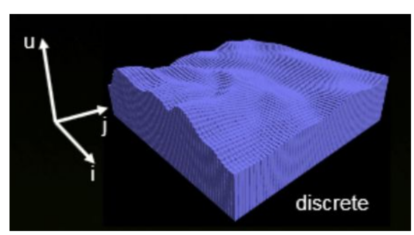


Today

- Papers for Today
- Worksheet: 2D Mass Spring System
- Flow Simulations in Computer Graphics
  – water, smoke, viscous fluids
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm
- Papers for Next Time...

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.

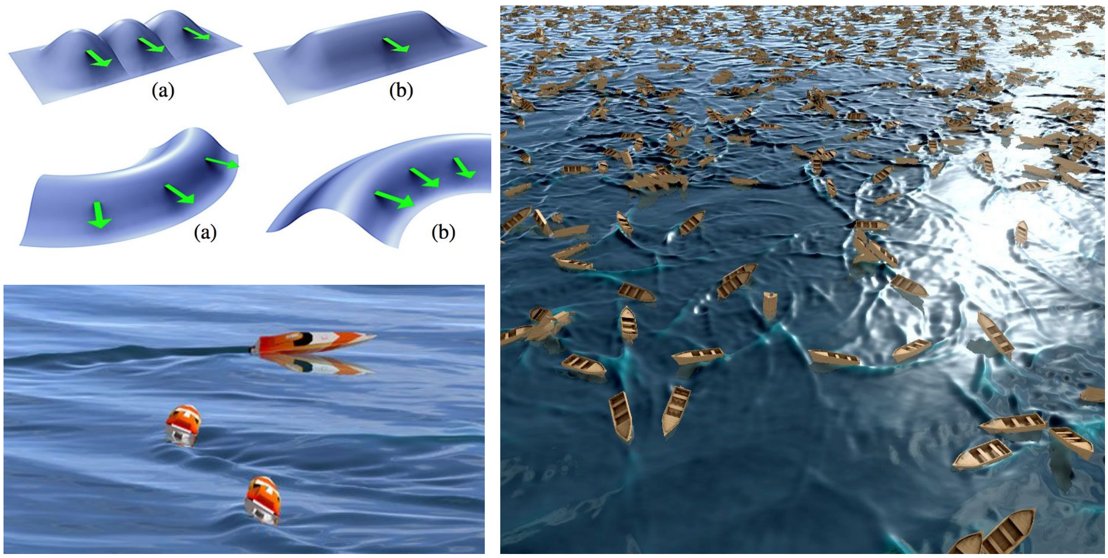


- Full Navier-Stokes

- *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



Today

- Papers for Today
- Worksheet: 2D Mass Spring System
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
  - incompressibility, conservation of mass
  - conservation of momentum & energy
- Fluid Representations
- Data Structure & Algorithm
- Papers for Next Time...

Flow in a Volume (continuous or 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)

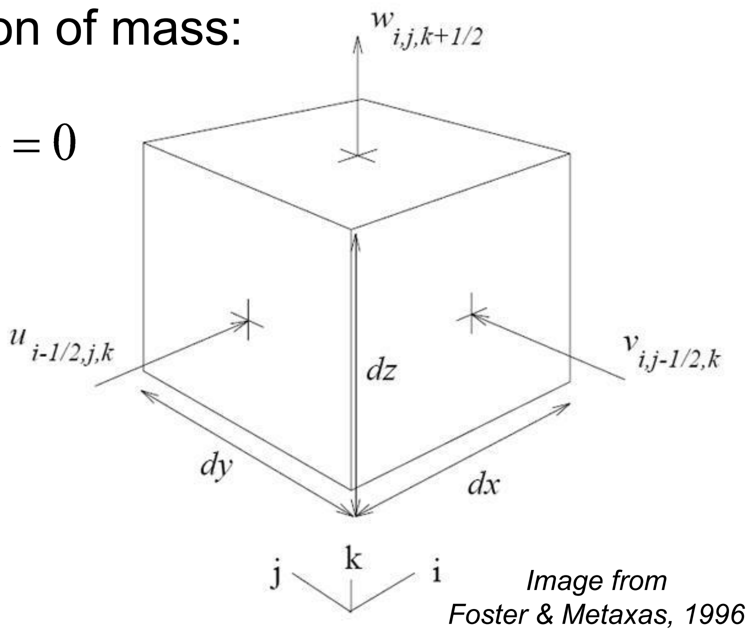


Image from Foster & Metaxas, 1996

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)

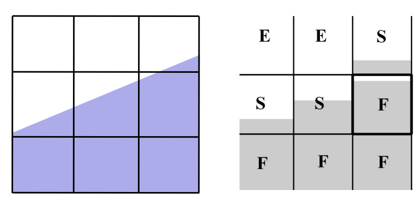
drag

Today

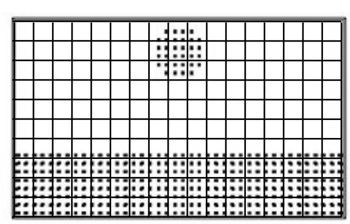
- Papers for Today
- Worksheet: 2D Mass Spring System
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm
- Papers for Next Time...

Modeling the Air/Water Surface

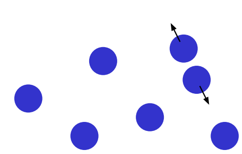
- Volume-of-fluid tracking



- Marker and Cell (MAC)



- Smoothed Particle Hydrodynamics (SPH)

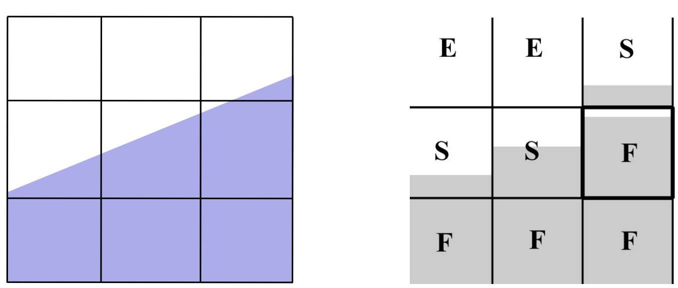


Comparing Representations

- 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 floating point value indicating that cell's "full"-ness



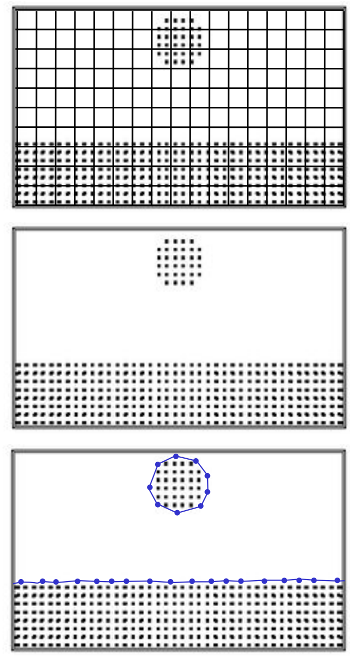
\+ preserves volume

– 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



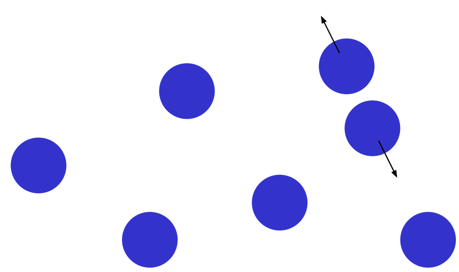
\+ rendering

– 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*

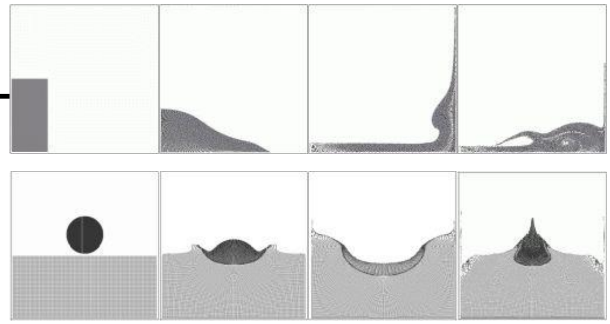
\+ render similar to Marker and Cell (MAC)

– much more expensive (particle-particle interactions)

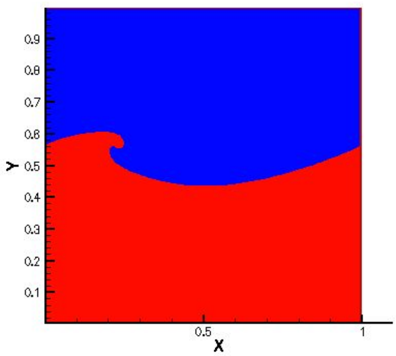
Note: Usually a spatial data structure (grid!) is added to reduce the number of particle-particle comparisons!

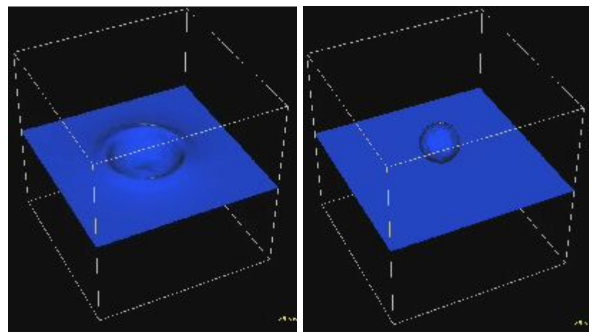
Demos

- 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

Today

- Papers for Today
- Worksheet: 2D Mass Spring System
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm
- Papers for Next Time...

Each Grid Cell Stores:

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

This is a critically important detail! (and makes correct implementation rather annoying)

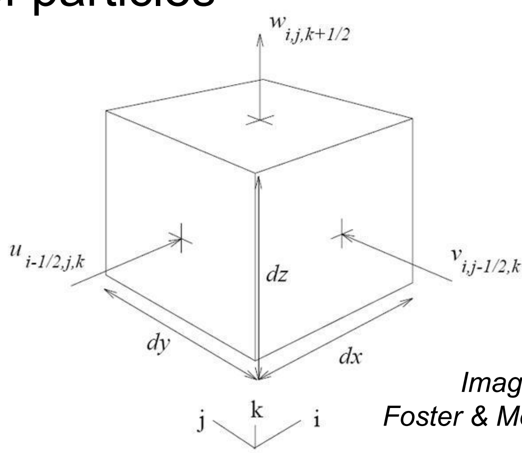


Image from Foster & Metaxas, 1996

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!

This piece needs explanation!

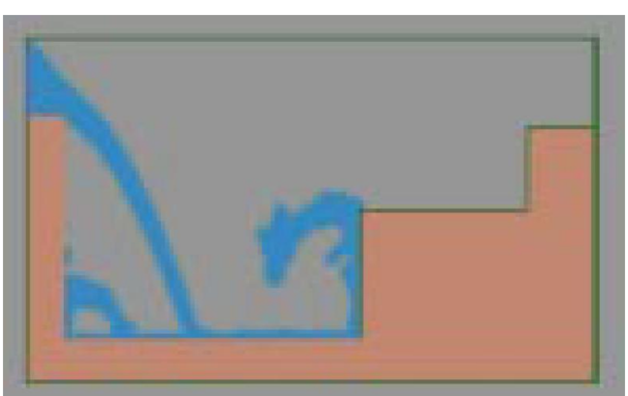
At each Timestep:

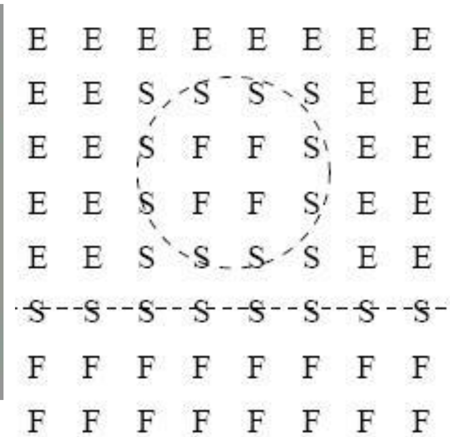
- Identify which cells are Empty, Full, or on the Surface
- Compute new velocities
- Adjust the velocities to maintain an incompressible flow
- Move the particles
  - Interpolate the velocities at the faces
- Render the geometry and repeat!

Empty, Surface & Full Cells

For 2-phase simulations, where we enforce incompressibility for only one phase!

- Empty: no marker particles
- Surface: has an neighbor that is “Empty”
- Full: not “Empty” or “Surface”





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

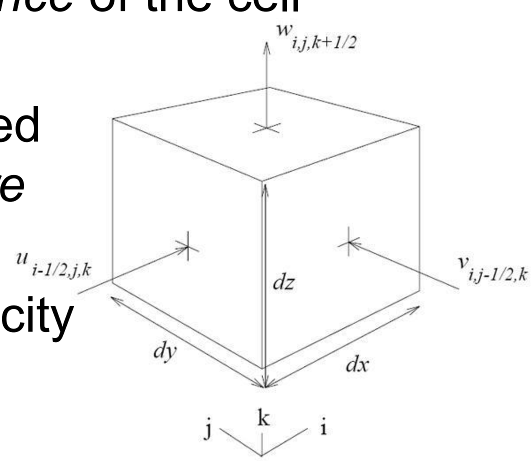
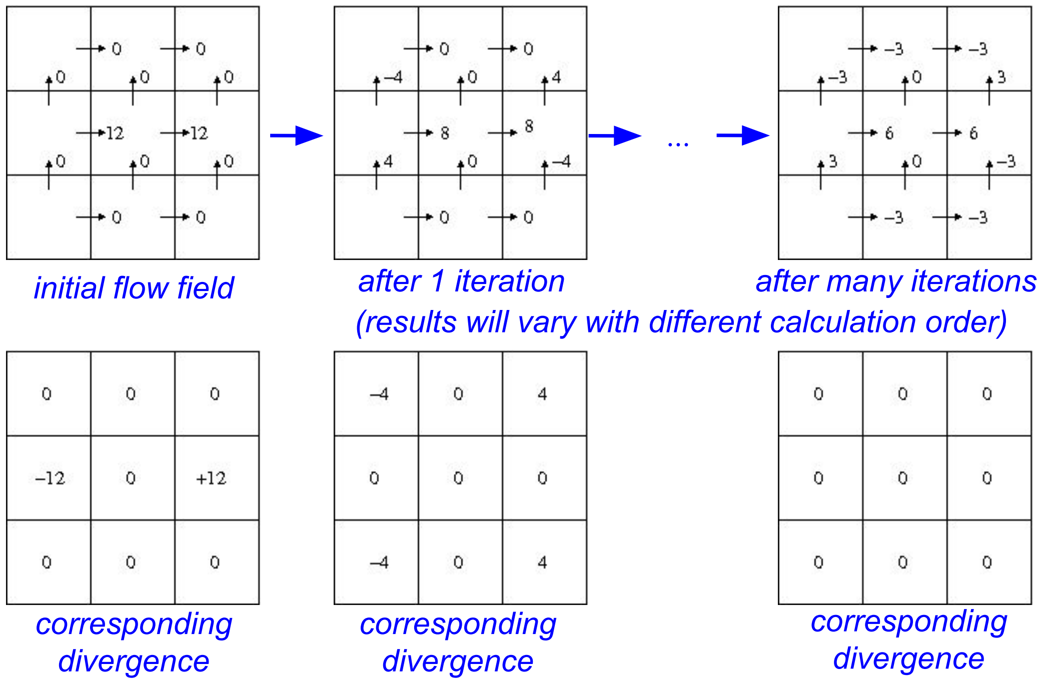


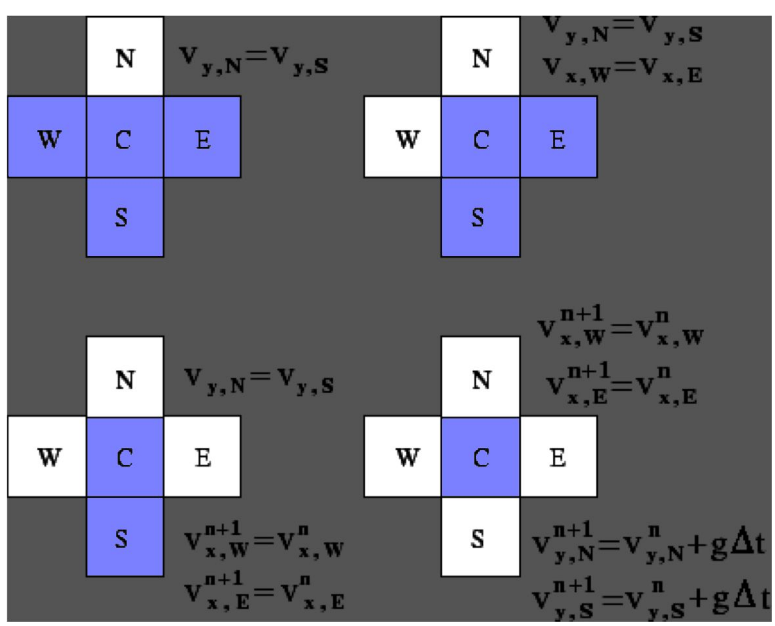
Image from Foster & Metaxas, 1996

Calculating/Eliminating Divergence



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



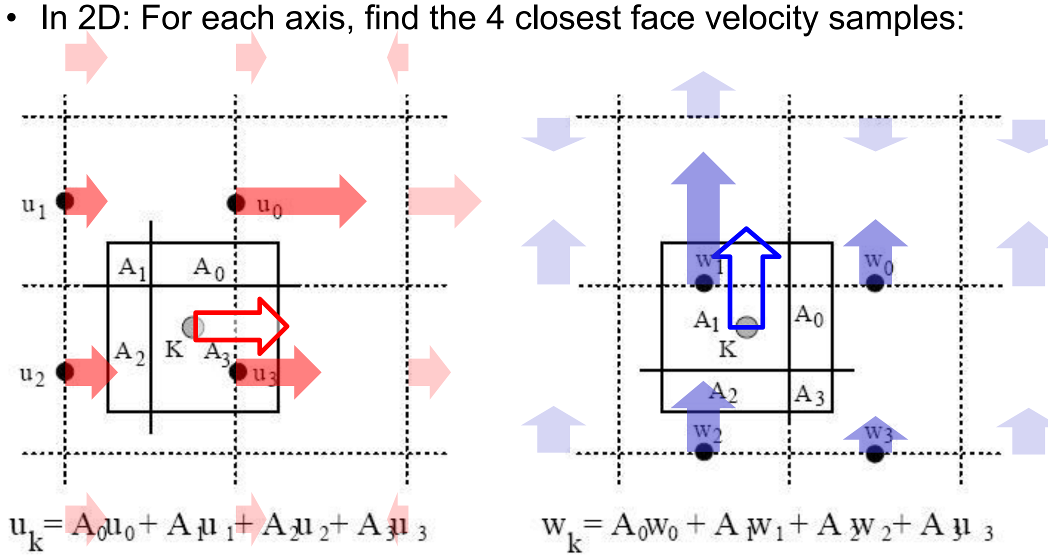
At each Timestep:

- 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

Original image from Foster & Metaxas, 1996

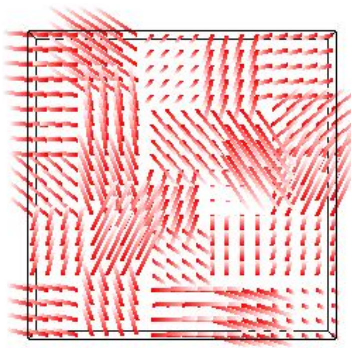
- In 2D: For each axis, find the 4 closest face velocity samples:



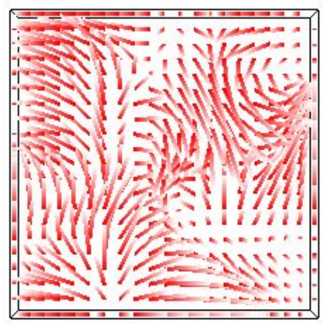
- (In 3D… Find 8 closest face velocities in each dimension)

Correct Velocity Interpolation

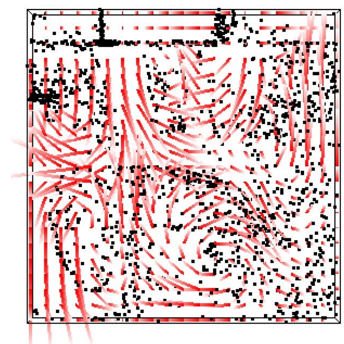
- NOTE: The complete implementation isn't particularly elegant… Storing velocities at face midpoints (req'd for conservation of mass) makes the index math messy!



No interpolation (just use the left/bottom face velocity)
Note the discontinuities in velocity at cell boundaries



Correct Interpolation
Note that the velocity perpendicular to the outer box is zero



Buggy Interpolation
Note the clumping particles, and the discontinuities at some of the cell borders (& particles might escape the box!)

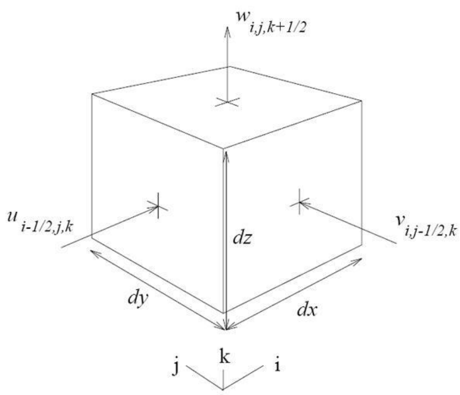
Today

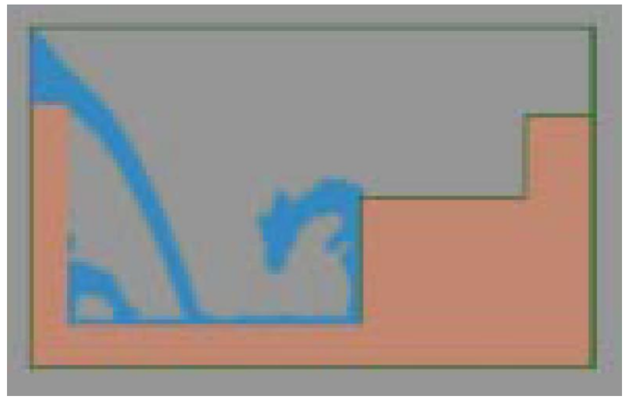
- Papers for Today
- Worksheet: 2D Mass Spring System
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm
- Papers for Next Time...

Reading for Next Time

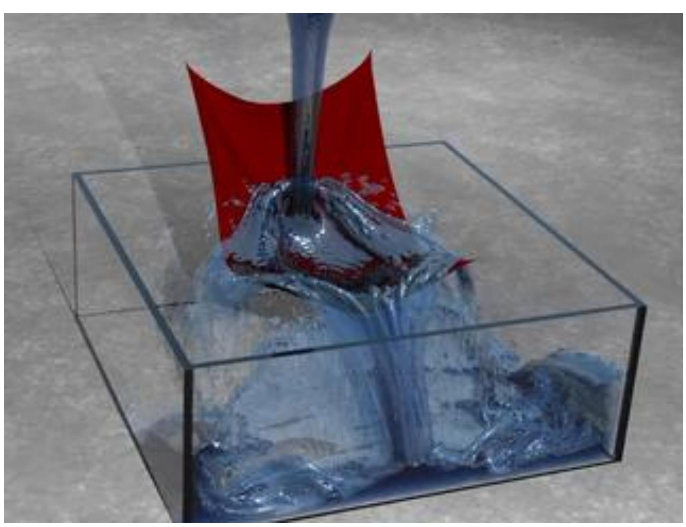
Everyone should read this
(simple fluid model used in HW2)

- “Realistic Animation of Liquids”, Foster & Metaxas, 1996



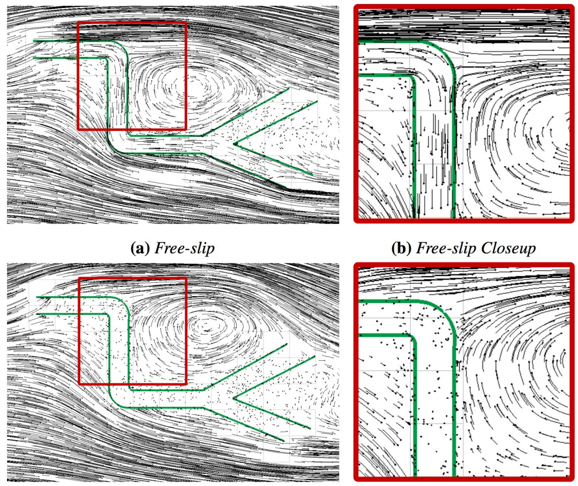


Optional Reading



“Coupling Water and Smoke to Thin Deformable and Rigid Shells”, Guendelman, Selle, Losasso, & Fedkiw, SIGGRAPH 2005.

Optional Reading



(a) *Free-slip* **(b)** *Free-slip Closeup*

(c) *No-slip* **(d)** *No-slip Closeup*

"Preserving Geometry and Topology for Fluid Flows with Thin Obstacles and Narrow Gaps"
Azevedo, Batty, & Oliveira, SIGGRAPH 2016