CSCI 4530/6530 Advanced Computer Graphics

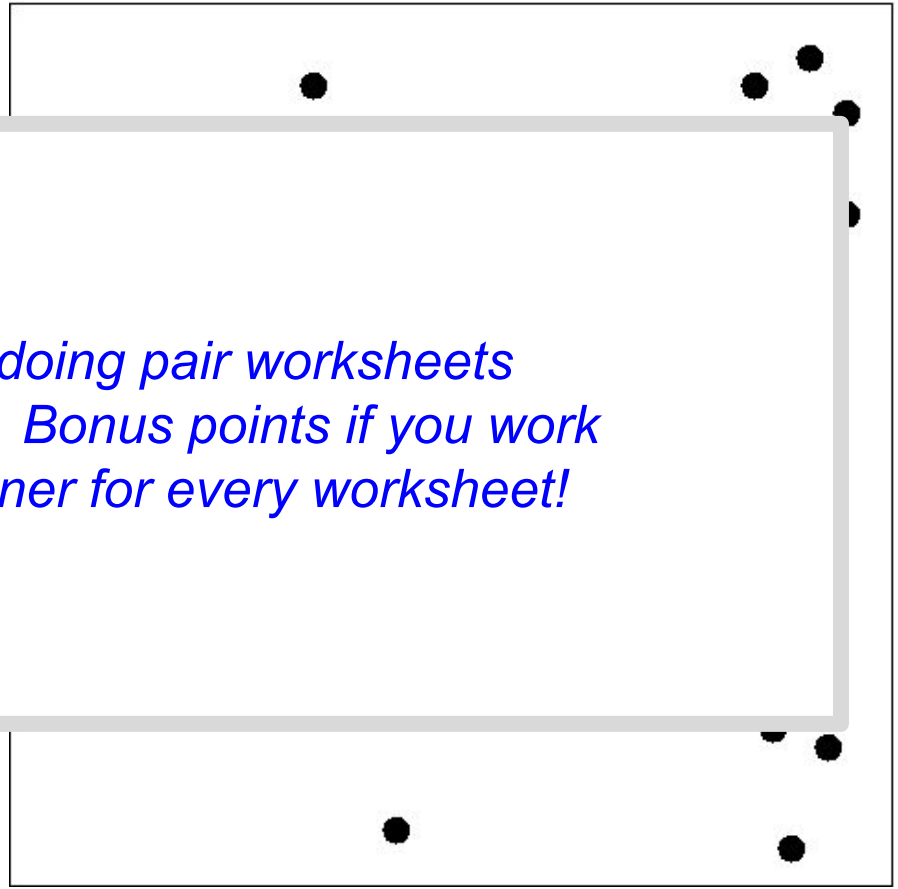
https://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S25/

Lecture 6: Navier-Stokes & Fluid Simulation

Worksheet: Spatial Data Structures

For each adaptive grid method
(quad
partitio
if we s
and all
of 5 (m

Wher
minim
and n
from each point to the split



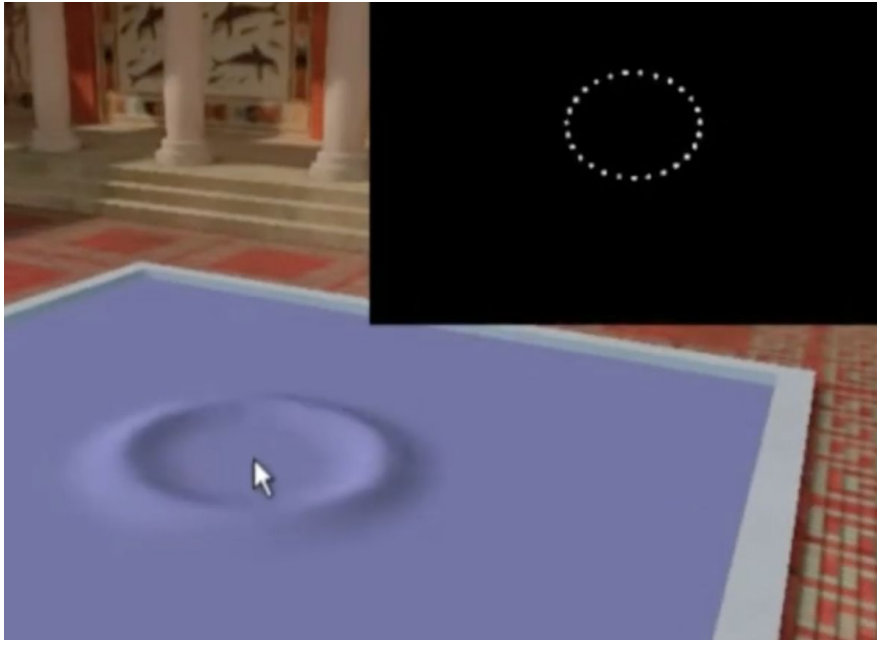
NOTE: We'll be doing pair worksheets throughout the term. Bonus points if you work with a different partner for every worksheet!

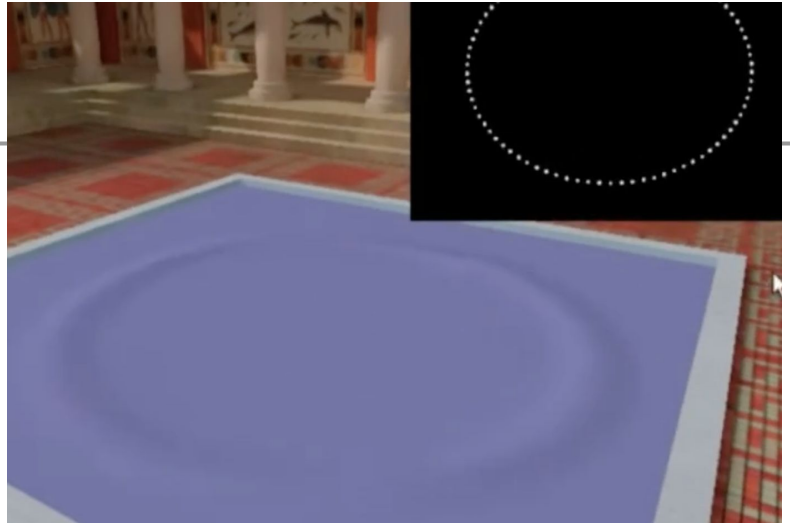
Flow

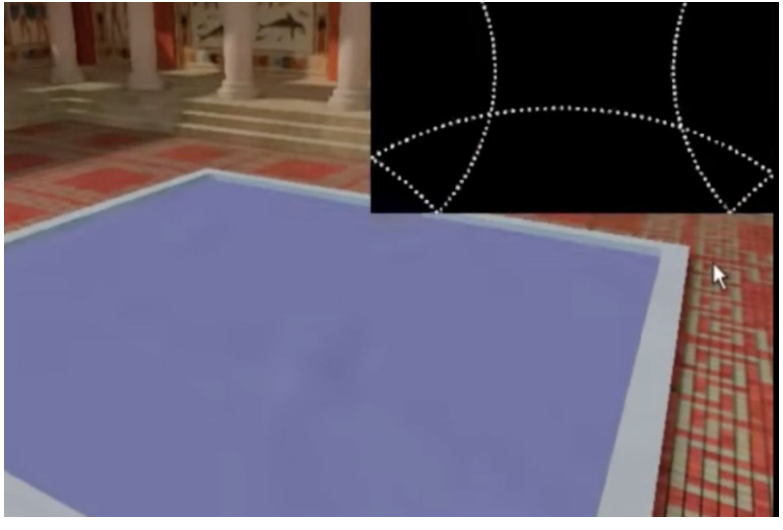
Scanline Flowline VFX, SIGGRAPH 2006


100% FULL CG WATER

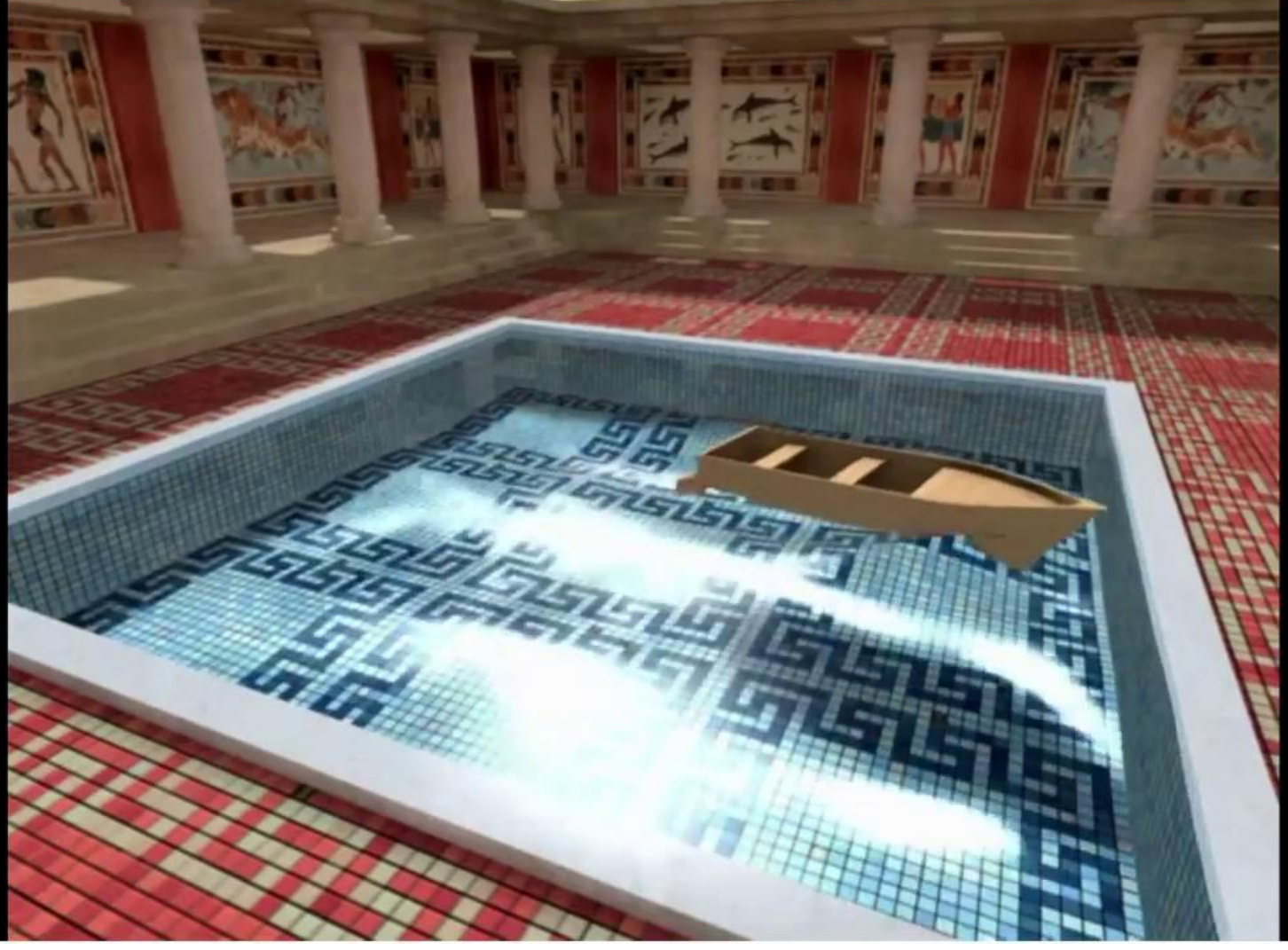
Wave Particles





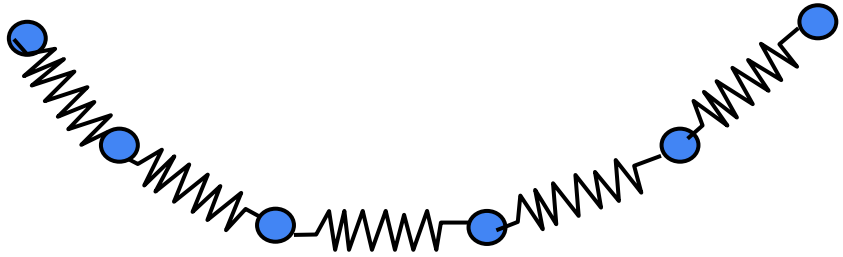


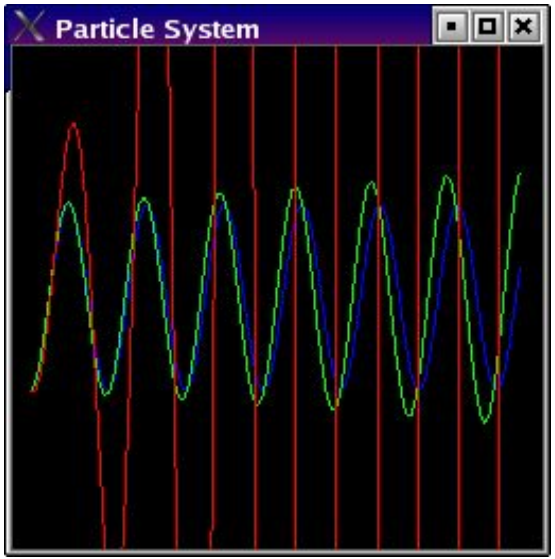
Yuksel, House, & Keyser, SIGGRAPH 2007

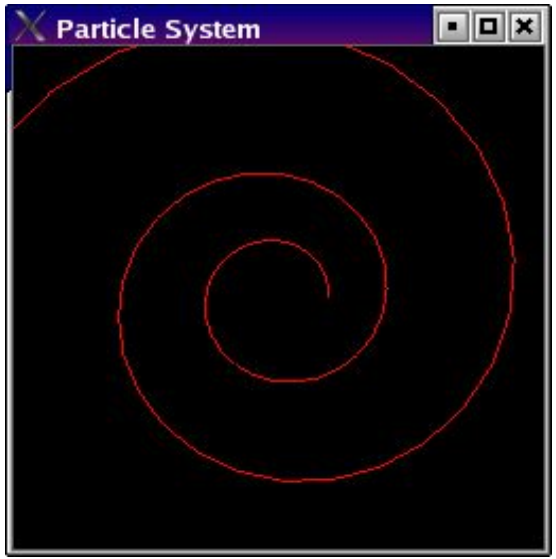


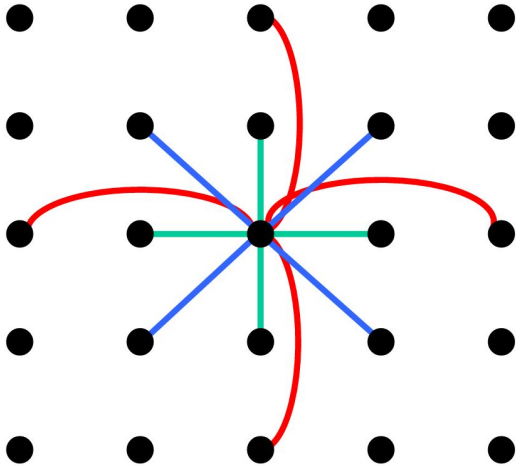
Last Time?

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







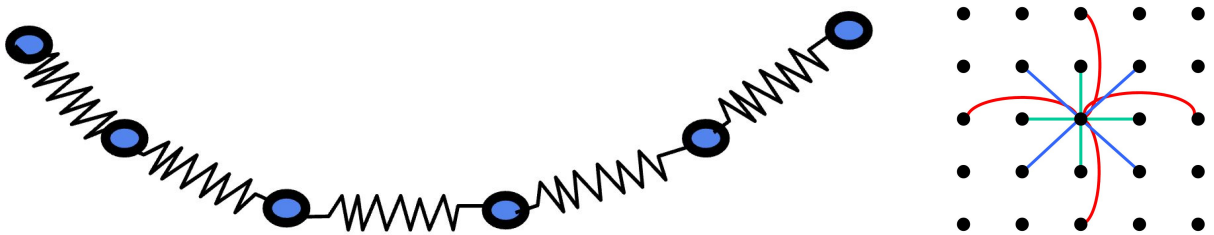


Today

- Worksheet: Spatial Data Structures
- From Last Time: Stiffness & Discretization
- Papers for Today
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm for Fluid Simulation
- Papers for Next Time...

The Stiffness Issue

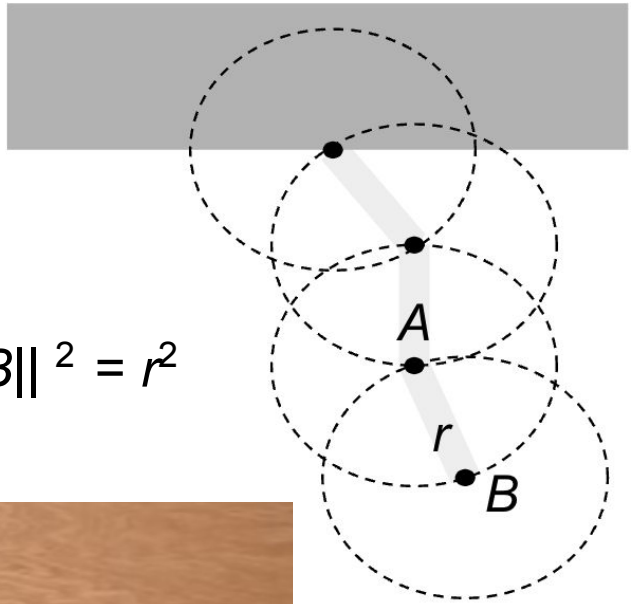
- What relative stiffness ***K*** do we want for the different springs in the network?



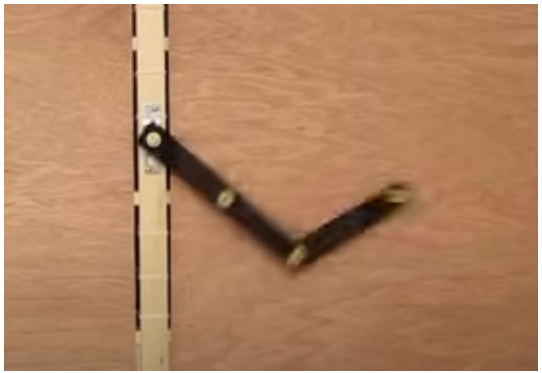
- Simple (non-spandex) cloth is barely elastic - *it shouldn't stretch much!*
- The actual spring length will always be greater than rest length
- Challenge/Unpleasant Compromise: Inverse relationship between stiffness & Δt necessary for stable simulation
  - Numerical oscillation and instability if Δt is too big
  - Simulation is costly & slow if Δt is small

What about Rigid Constraints?

- What we really want is *no stretch or maximum stretch constraints* (not springs!)
- *E.g., rigid, fixed-length bars that link the particles*
  - Dynamics + constraints must be solved simultaneously
  - non-trivial, even for tiny systems



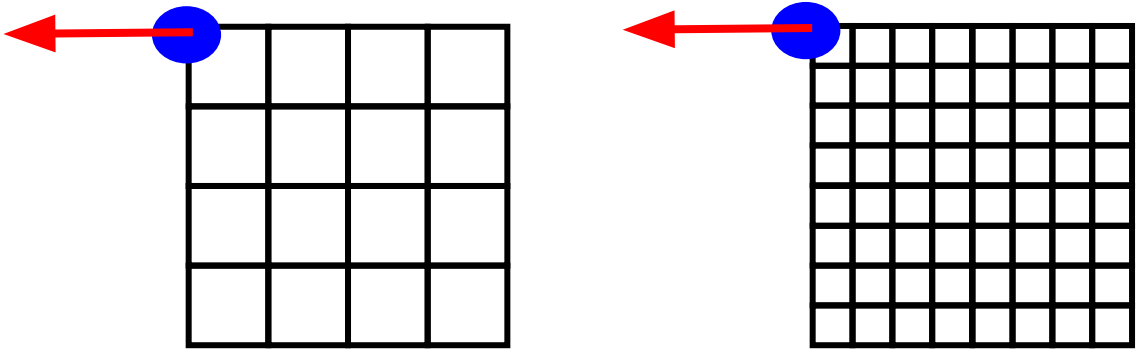
$$||A-B||^2 = r^2$$



*even 2 rigid links = **Double Pendulum** is chaotic!*
https://www.youtube.com/watch?v=AwT0k09w-jw

The Discretization Problem

- What is the impact of the grid resolution?
- What happens if we double the resolution of our mesh?



- Do we get the same simulation behavior for the two meshes?
  - *Usually not! It takes a lot of effort to design a scheme that does not depend on the discretization.*
- Using (explicit) Euler simulation, how many timesteps before a force propagates across the mesh in the two meshes?
  - *It will take twice as many timesteps in the higher resolution mesh!*

A Better Solution: Explicit vs. Implicit Integration

- Explicit/forward integration :

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

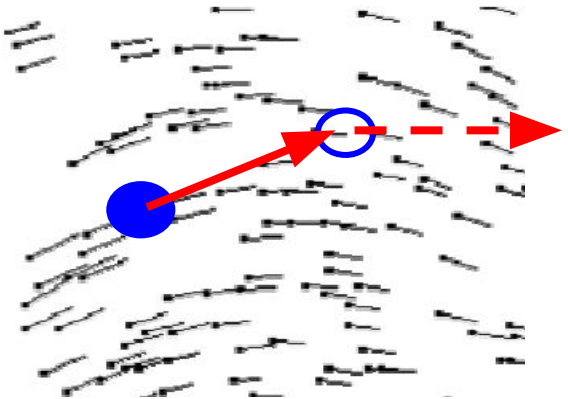
The future state (position & velocity) of this particle is a function of the current state of the particle.

- Implicit/backwards integration :

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

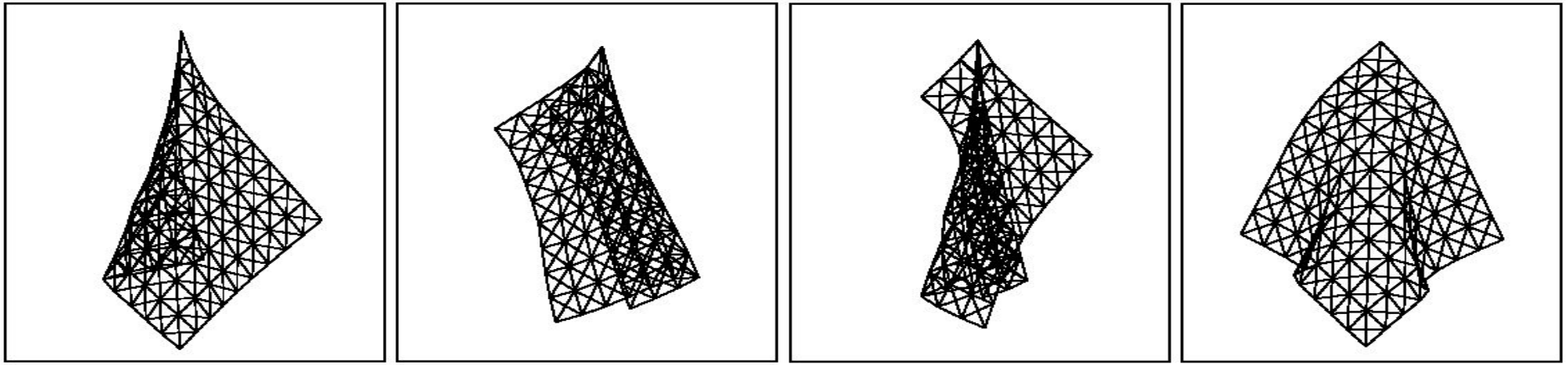
$$\mathbf{y}_{k+1} - h\,\mathbf{g}(\mathbf{y}_{k+1}) = \mathbf{y}_k$$

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



  - Because particles are interconnected, must solve global system (not just local)
  - Solving each each step is more expensive (Newton's Method, Conjugate Gradients, …)
  - *Larger timesteps are possible with implicit methods!*
  - Thus it can be overall faster than the equivalent explicit simulation

Questions?



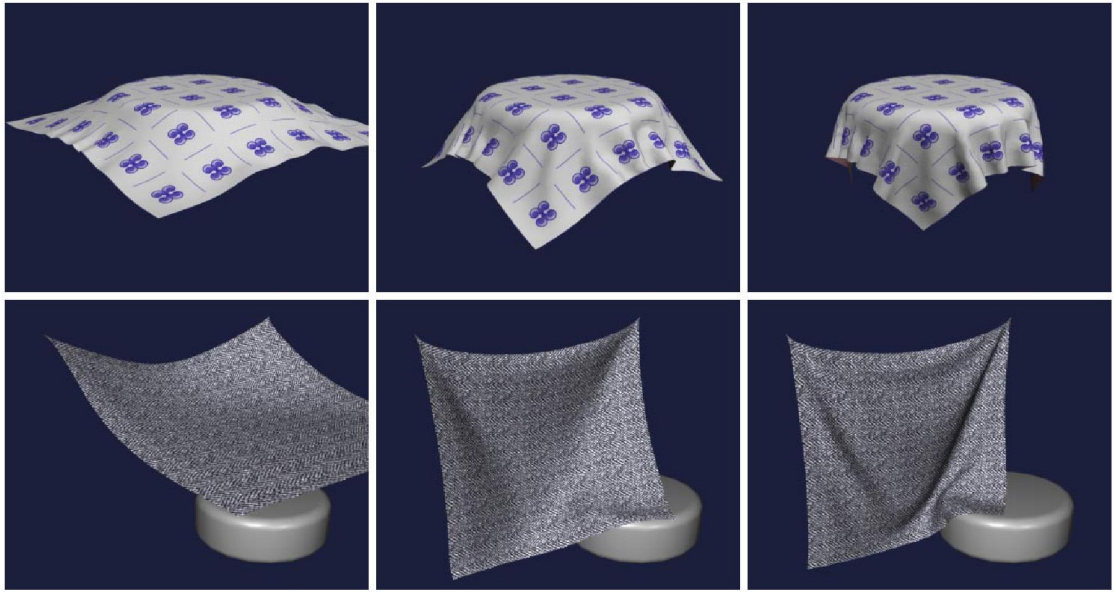
Interactive Animation of
Structured Deformable Objects
Desbrun, Schröder, & Barr 1999

Today

- Worksheet: Spatial Data Structures
- From Last Time: Stiffness & Discretization
- Papers for Today
  - How to read a research paper
  - Components of a well written paper
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm for Fluid Simulation
- Papers for Next Time...

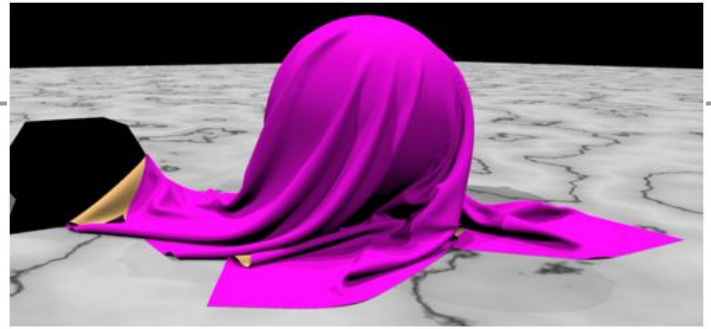
Papers for Tuesday *(pick one)*

- “Large Steps in Cloth Simulation”,
  Baraff & Witkin,
  SIGGRAPH 1998

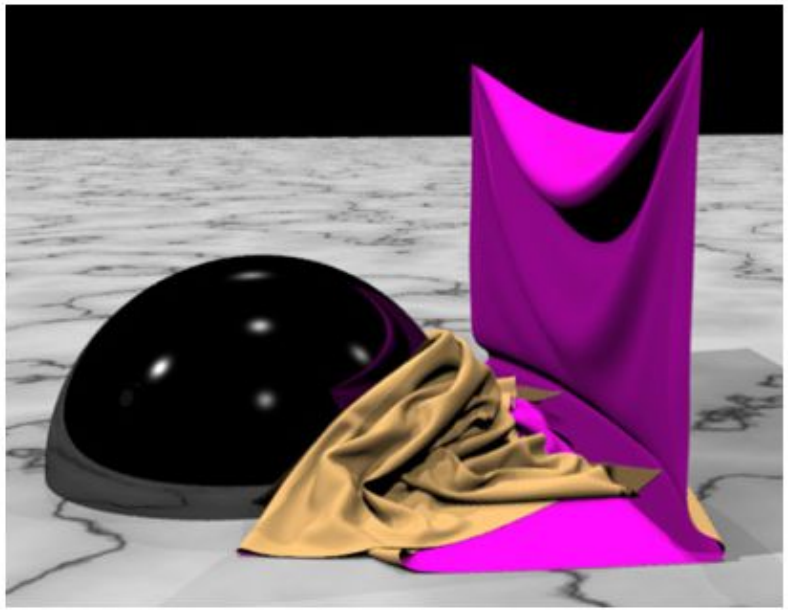


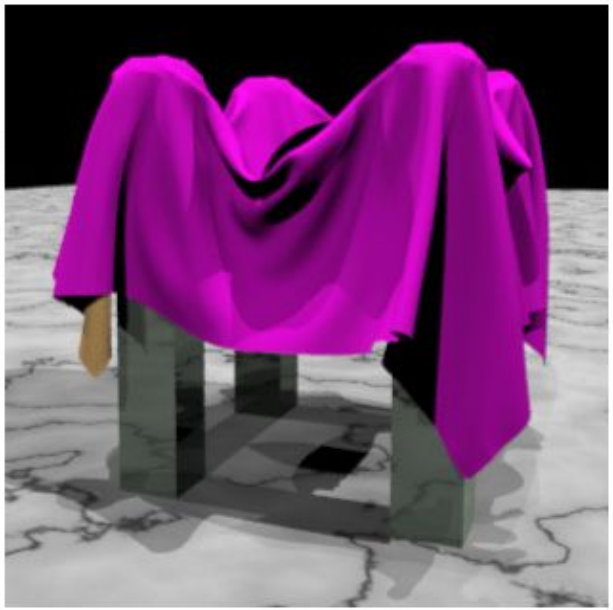
- Implicit integration is important for stability, realism, efficiency
- Nice to see progression/improvement from earlier paper
- Good to see proper use of numerical computing technology
- Related work section thorough in discussion / comparisons
- Runtime varies alot based on properties of cloth
- Inverting mass

Papers for Tuesday *(pick one)*



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





- Predict/anticipate collisions and adjust velocity in advance
- No “tricks”
- How does this perform (runtime) in very dense/bunched areas of cloth?
- Cloth simulation + subdivision surfaces
- High resolution meshes
- Friction
- Adaptive timestep to avoid collisions
- Realistic cloth has near collisions & friction
- Algorithms described well - helpful to readers who want to apply techniques to solve other problems

Papers for Tuesday *(pick one)*

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

- Hair in Brave film was seamless – didn't realize how challenging the problem was!
- Can't simulate *every hair*, use "guide hairs"
- Goal: believable realism, not perfectly accurate realism
- Animators should still be able to control hair & tell a story
- Could this (or any hair simulation) ever be "real time"?
- Hair pruning and parallelization
- People who have curly hair know how it actually moves and behaves (and how challenging it is to maintain)
- Is this model appropriate for all hair types? Acknowledge limitations and press forward with more diverse research & development.

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?

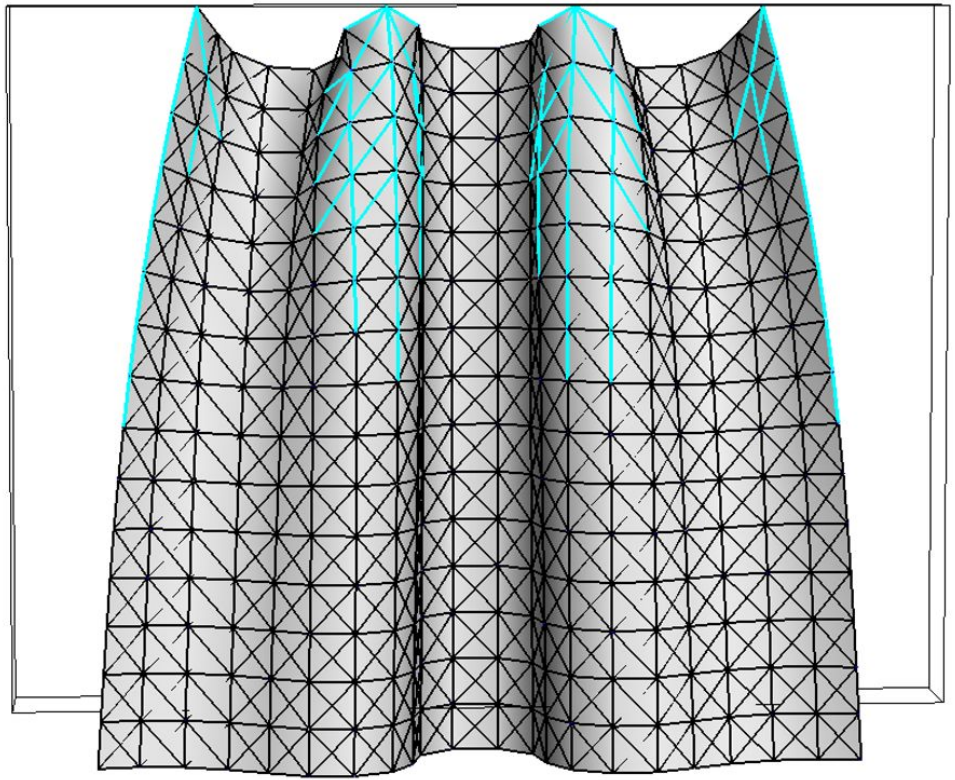
- Motivation/context/related work
- Contributions of this work
- Clear description of algorithm
  - Sufficiently-detailed to allow work to be reproduced
  - Work is theoretically sound
    (hacks/arbitrary constants discouraged)
- Results
  - well chosen examples
  - clear tables/illustrations/visualizations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

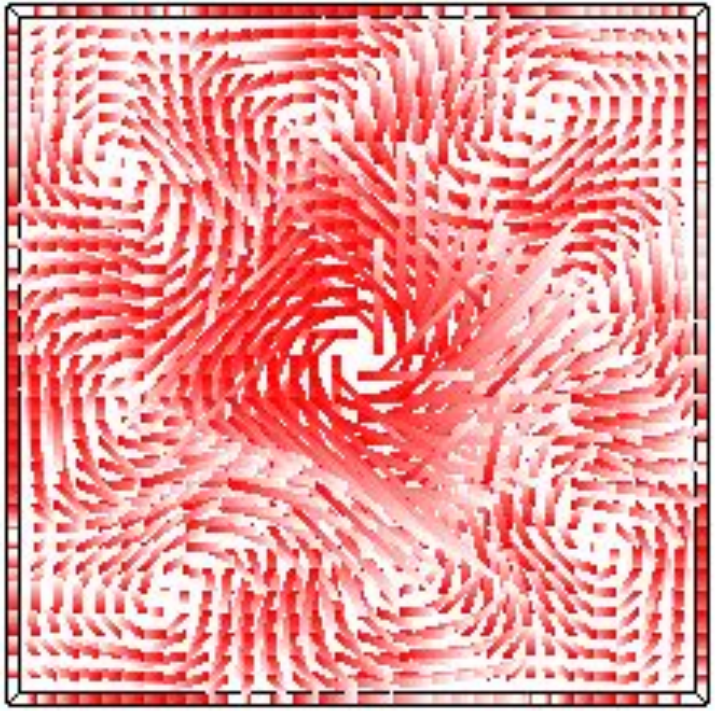
Today

- Worksheet: Spatial Data Structures
- From Last Time: Stiffness & Discretization
- Papers for Today
- Flow Simulations in Computer Graphics
  - water, smoke, viscous fluids
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm for Fluid Simulation
- Papers for Next Time...

HW2: Cloth & Fluid Simulation

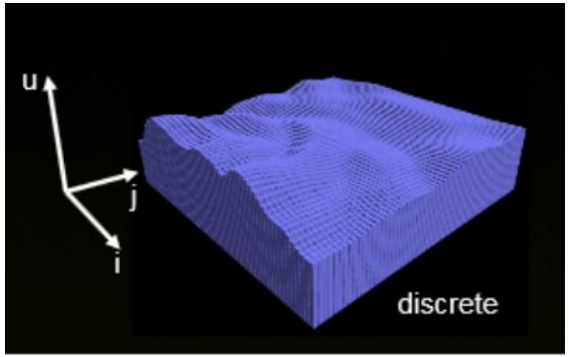
progress post due Thursday evening





Flow Simulations in Graphics

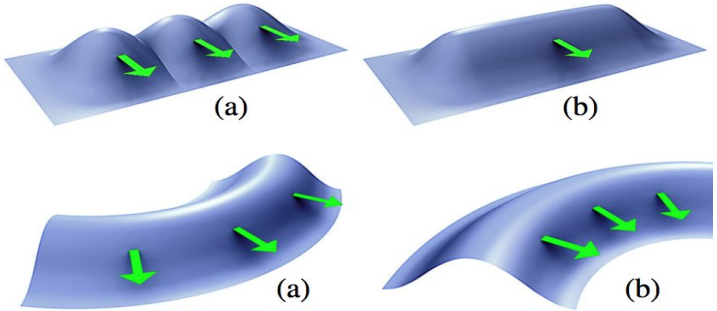
- Random velocity fields – good for simple background motion
  - *E.g. dust/leaves blowing randomly in the wind*
- Shallow water equations
  - height field only - “2.5 D”
  - *cannot represent crashing/overturning waves*
- Navier-Stokes
  - *Full 3D!*

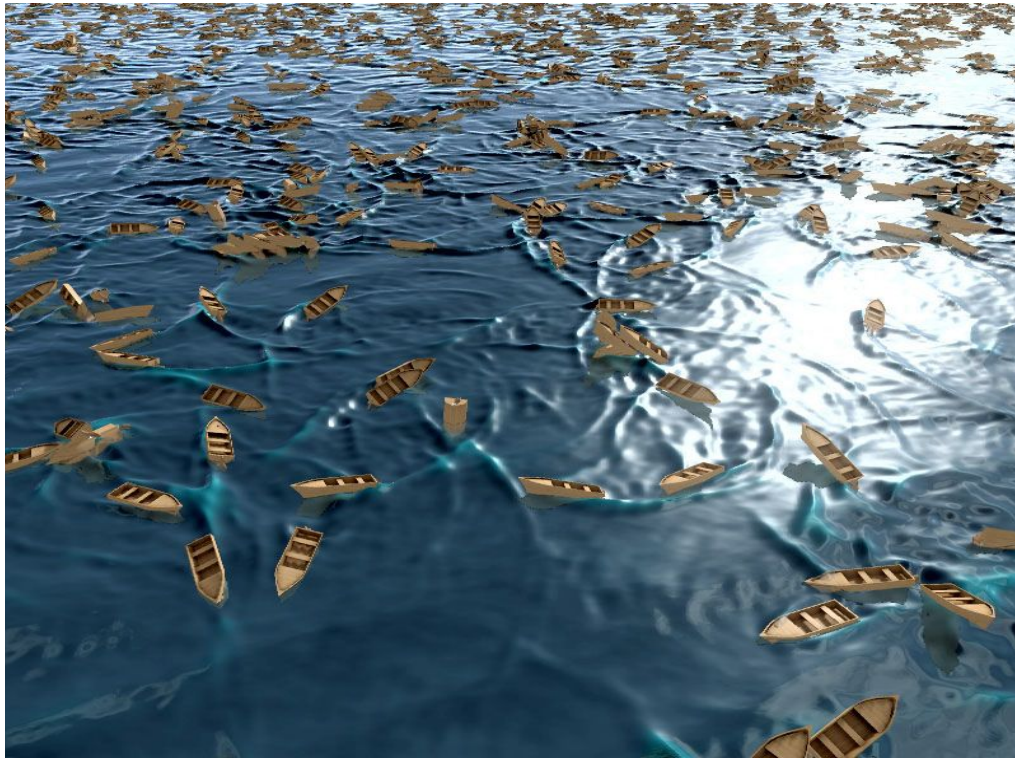


- *Note: typically we ignore surface tension and focus on macroscopic behavior*

Heightfield Wave Simulation

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





Today

- Worksheet: Spatial Data Structures
- From Last Time: Stiffness & Discretization
- Papers for Today
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
  - incompressibility, conservation of mass
  - conservation of momentum & energy
- Fluid Representations
- Data Structure & Algorithm for Fluid Simulation
- 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, or air only)

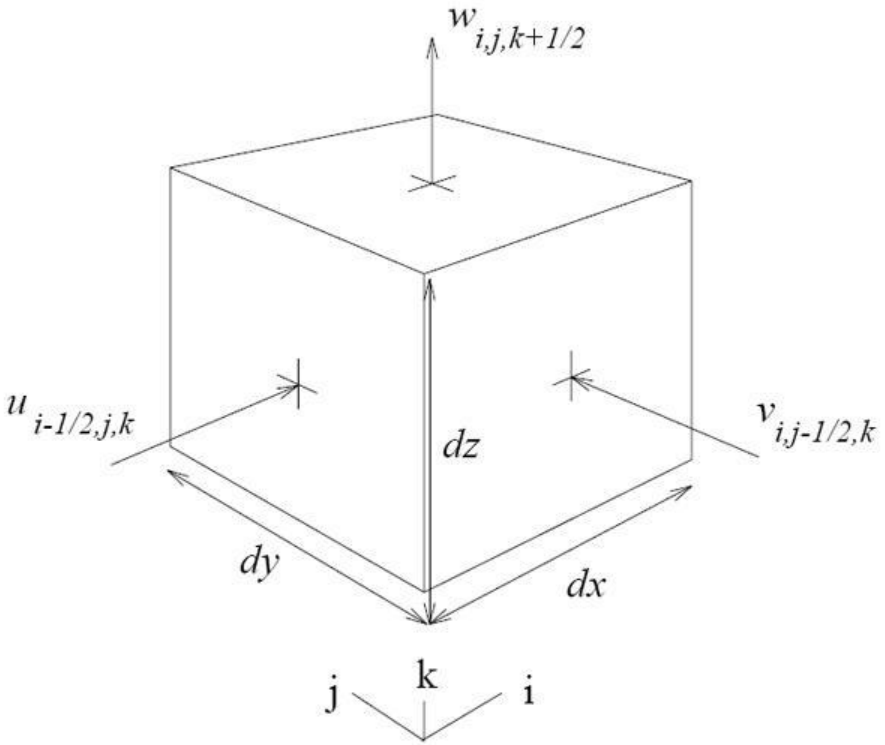


Image from Foster & Metaxas, 1996

Navier-Stokes Equations

- conservation of momentum:

gravity (& other external forces)

pressure *viscosity*

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

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

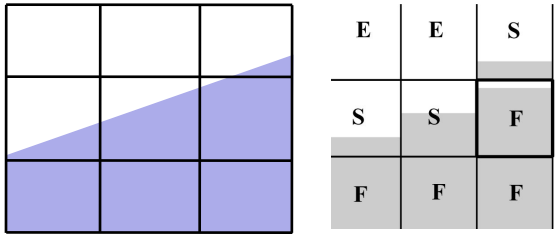
$$\underbrace{\frac{\partial w}{\partial t}}_{\text{acceleration}} + \underbrace{\frac{\partial wu}{\partial x} + \frac{\partial wv}{\partial y} + \frac{\partial w^2}{\partial z}}_{\text{convection: internal movement in a fluid (e.g., caused by variation in density due to a transfer of heat)}} = -\frac{\partial p}{\partial z} + g_z + \nu\underbrace{\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right)}_{\text{drag}}$$

Today

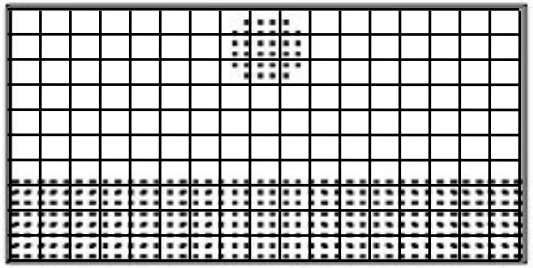
- Worksheet: Spatial Data Structures
- From Last Time: Stiffness & Discretization
- Papers for Today
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm for Fluid Simulation
- Papers for Next Time...

Modeling the Air/Water Surface

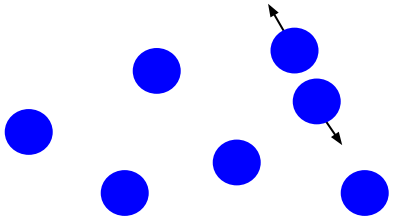
- Volume-of-fluid tracking



- Marker and Cell (MAC)



- Smoothed Particle Hydrodynamics (SPH)

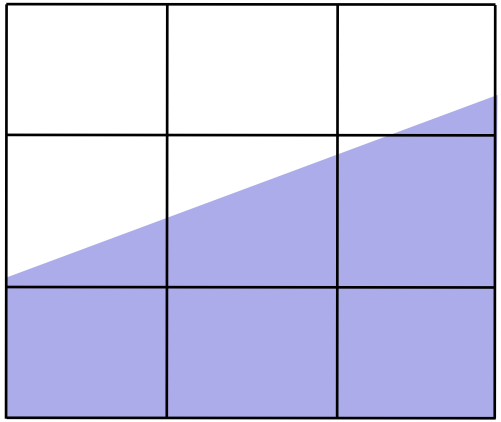


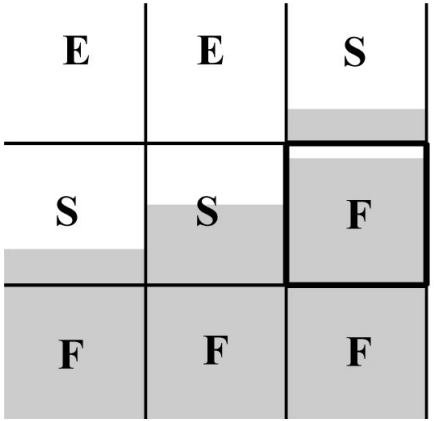
Comparing Representations

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





+ *preserves volume*
– *difficult to render*
– *very dependent on grid resolution*

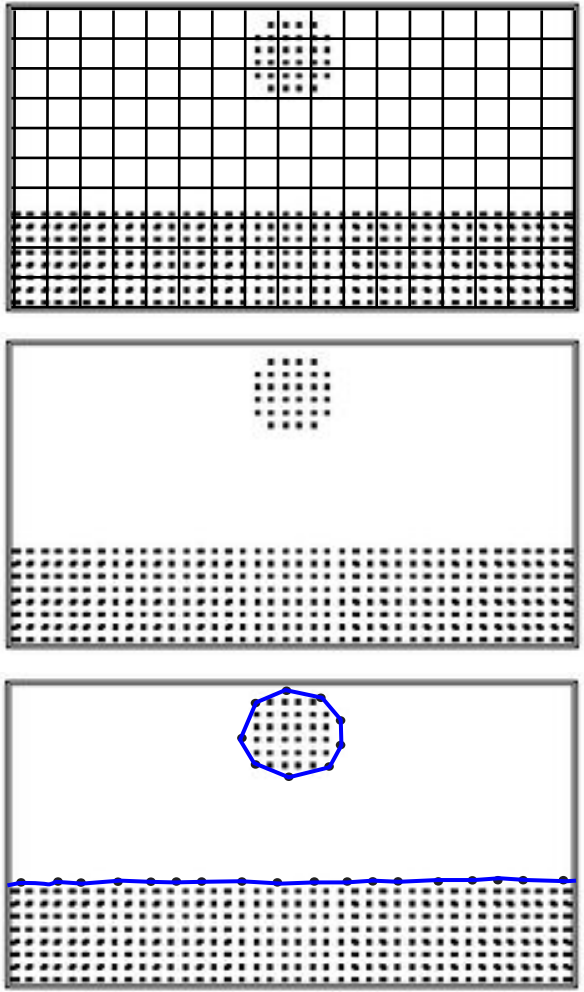
Marker and Cell (MAC)

- ***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*

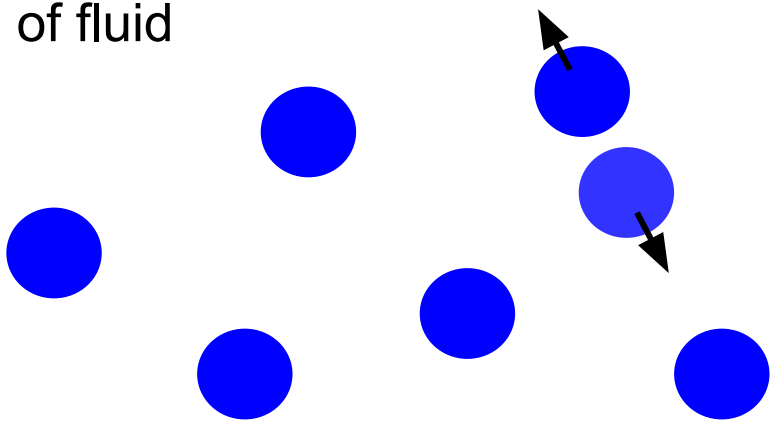
– *dependent on grid resolution*



Harlow & Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface", The Physics of Fluids, 1965.

Smoothed Particle Hydrodynamics (SPH)

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



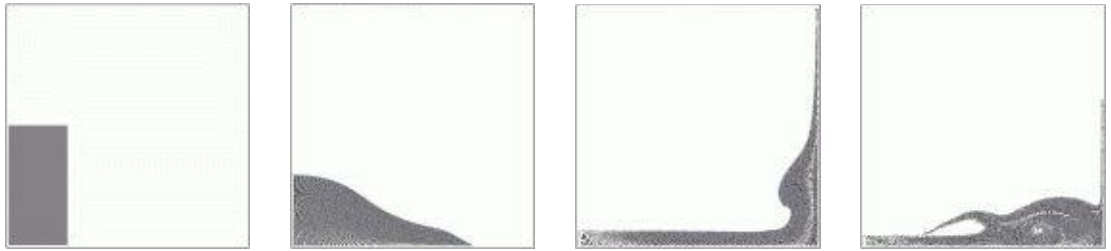
+ *no grid resolution concerns (accuracy depends on number/size of particles)*
+ *volume is preserved**
+ *render similar to Marker and Cell (MAC)*
– *much more expensive (particle-particle interactions)*

Note: A spatial data structure is used to reduce the number of particle-particle comparisons!

Marker and Cell (MAC) Examples

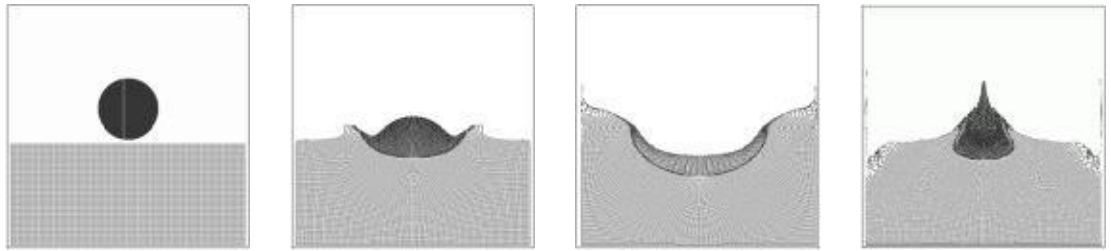
Maciej Matyka

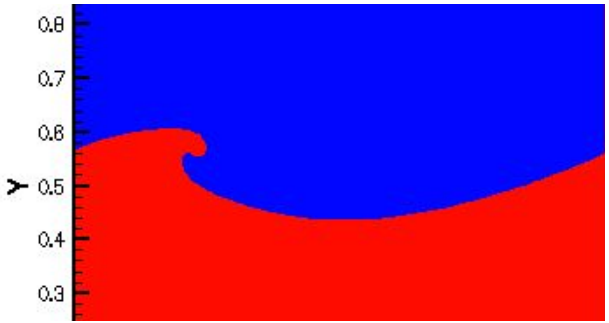
https://panoramx.ift.uni.wroc.pl/~MAQ/eng/cfdthesis.php

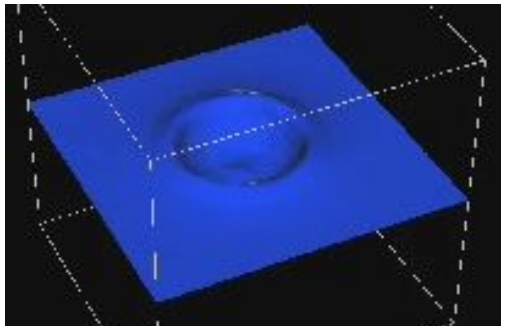


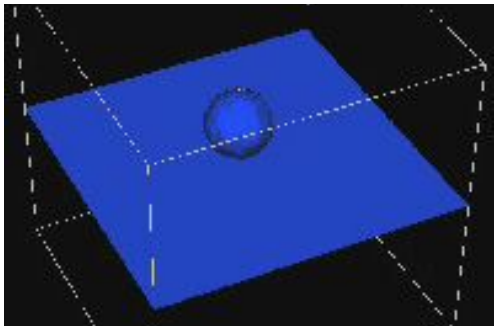
Fue-Sang Lien

https://uwaterloo.ca/computational-fluid-dynamics-turbulence-modeling-laboratory/







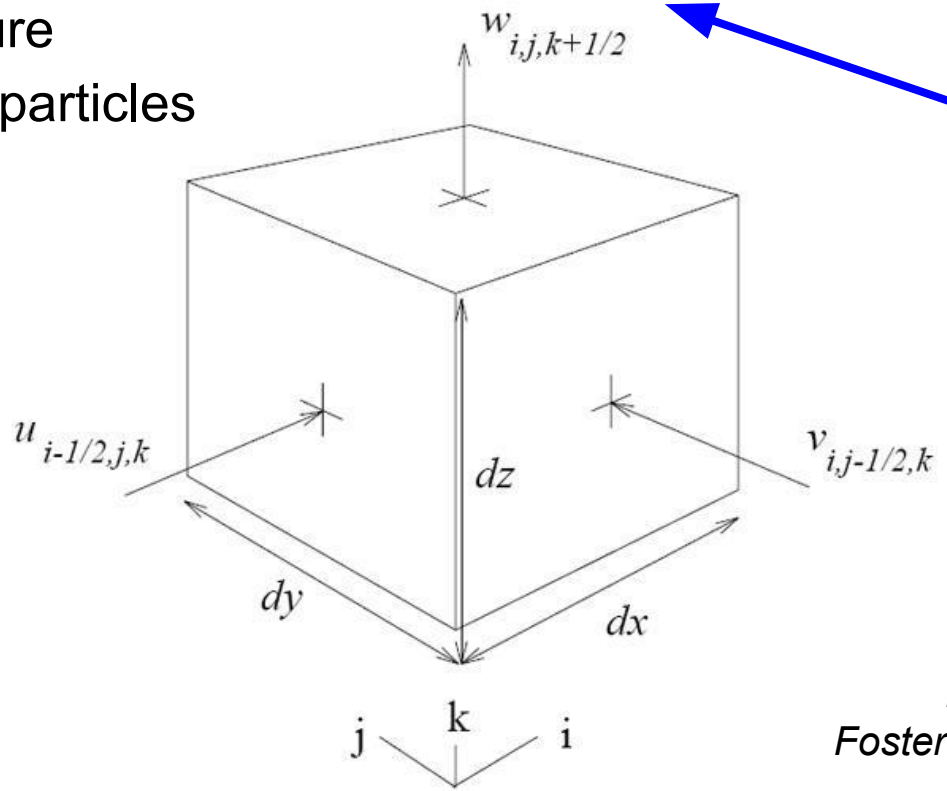


Today

- Worksheet: Spatial Data Structures
- From Last Time: Stiffness & Discretization
- Papers for Today
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm for Fluid Simulation
- 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

Fluid Simulation Implementation

- 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!

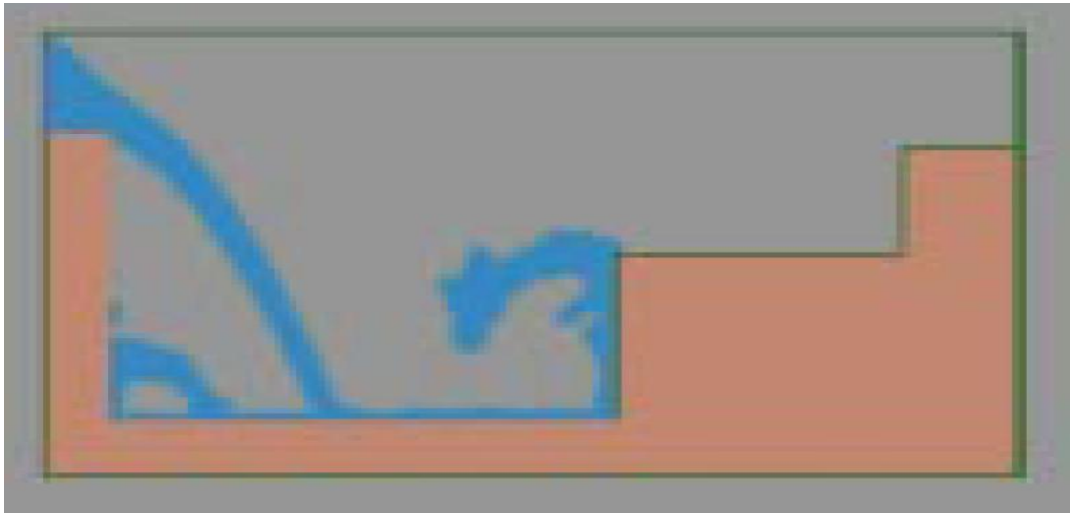
This piece needs
more explanation!

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!

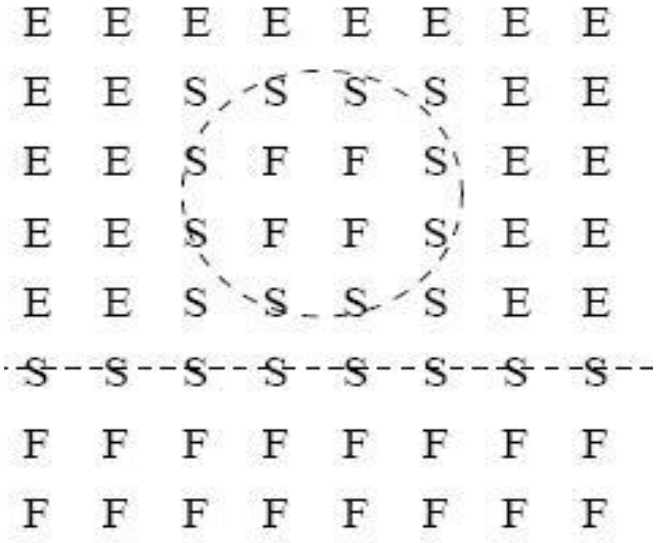
Empty, Surface, & Full Cells

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



Images from Foster & Metaxas, 1996

This step is necessary for 2-phase simulations (e.g. air+water), where we enforce incompressibility for only one phase!



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

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

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

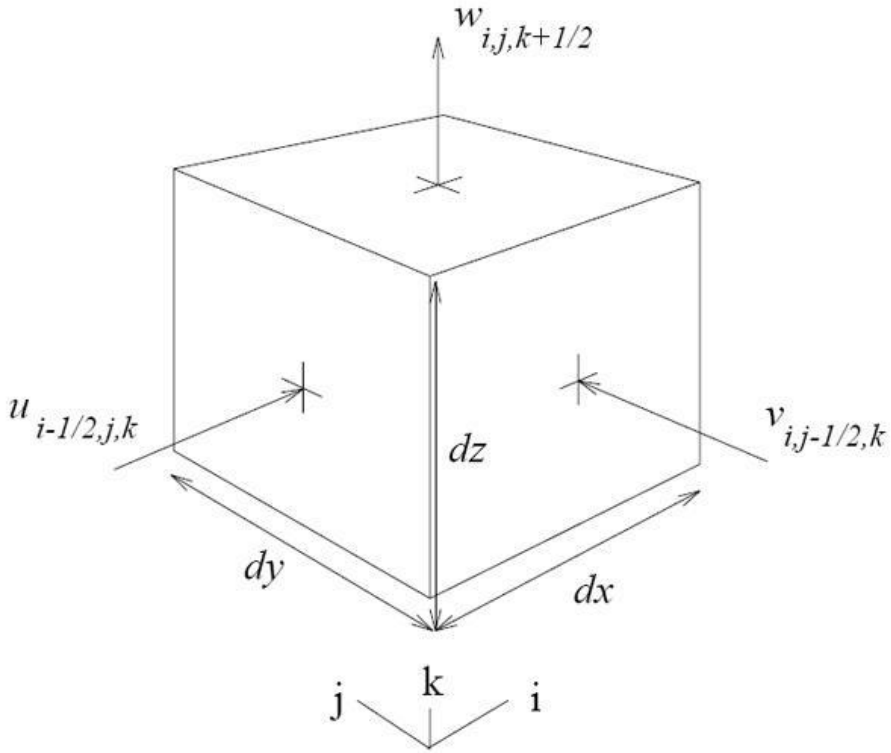
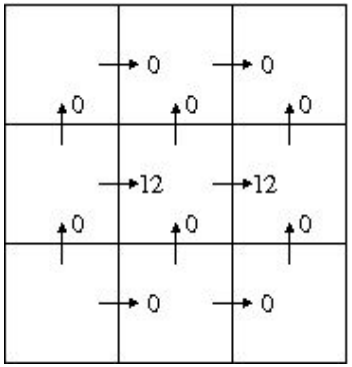
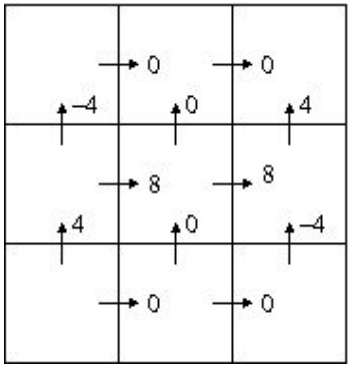


Image from Foster & Metaxas, 1996

Calculating/Eliminating Divergence

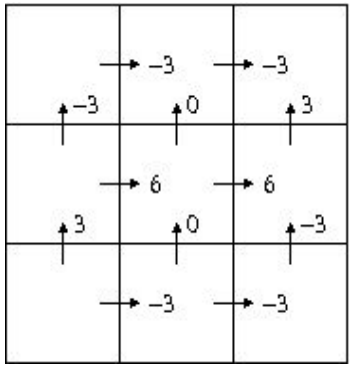


initial flow field



after 1 iteration

…



after many iterations

NOTE: results will vary with a different calculation order

0	0	0
−12	0	+12
0	0	0

corresponding divergence

−4	0	4
0	0	0
−4	0	4

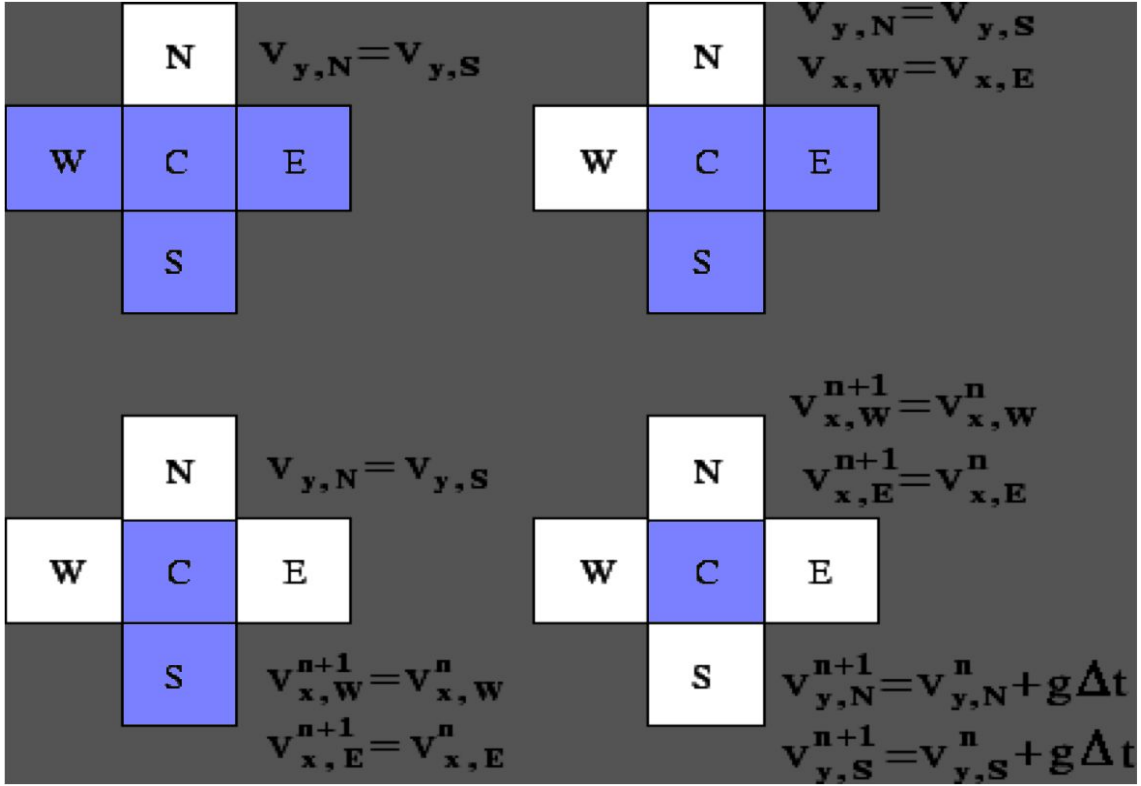
corresponding divergence

0	0	0
0	0	0
0	0	0

corresponding divergence

Handing Divergence with a Free Surface with MAC

- Divergence in **surface cells**:
  - Divide excess equally amongst neighboring **empty cells**
  - Handle the special cases (include gravity)
- Zero out the divergence & pressure in empty cells



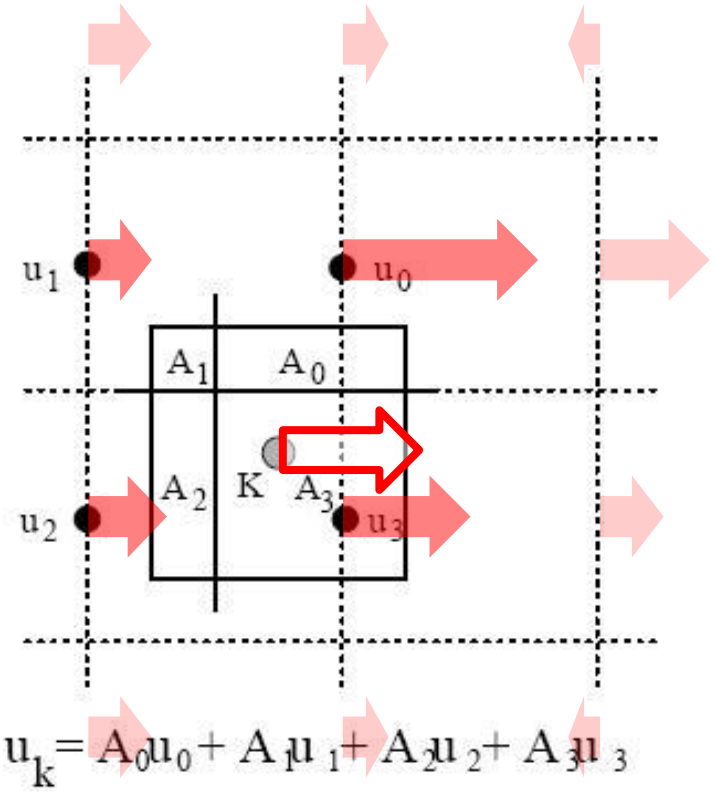
At each Timestep:

- 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!

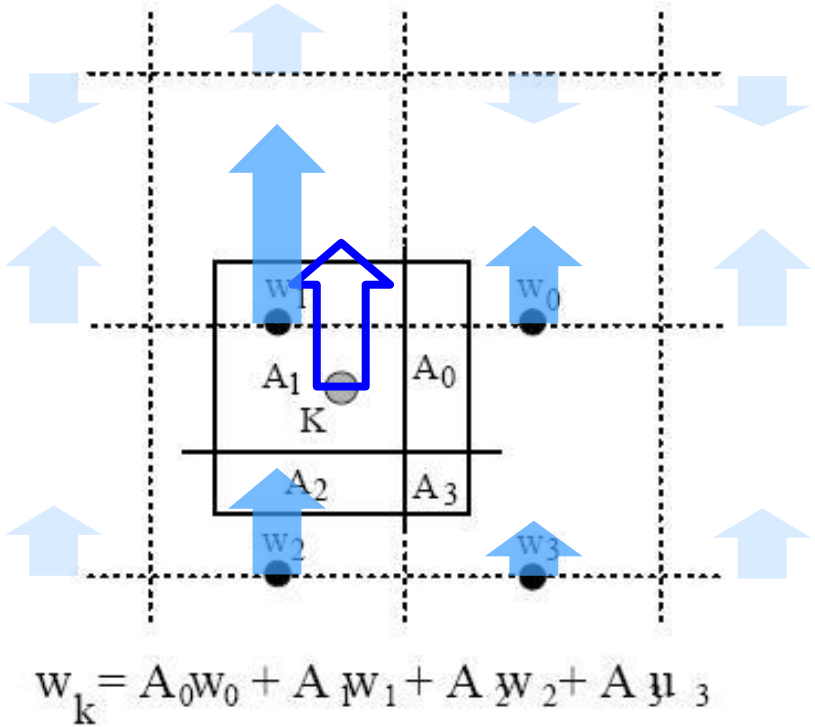
Velocity Interpolation

Original image from Foster & Metaxas, 1996

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



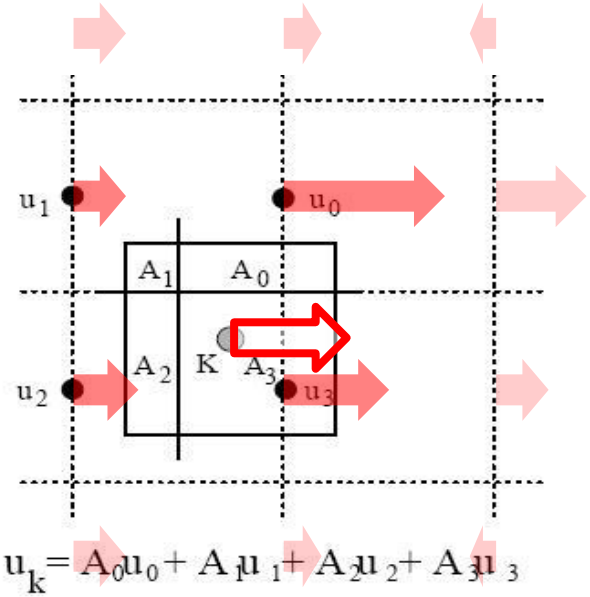
$$u_k = A_0 u_0 + A_1 u_1 + A_2 u_2 + A_3 u_3$$

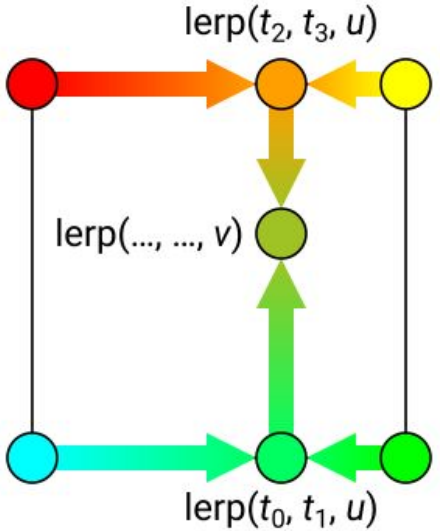


$$w_k = A_0 w_0 + A_1 w_1 + A_2 w_2 + A_3 w_3$$

Velocity Interpolation = *Bilinear Interpolation*

- In 2D: For each dimension, find the 4 closest ***face velocity*** samples
- *In 3D: Find 8 closest face velocities in each dimension*
- Separately interpolate velocity for each axis, then combine

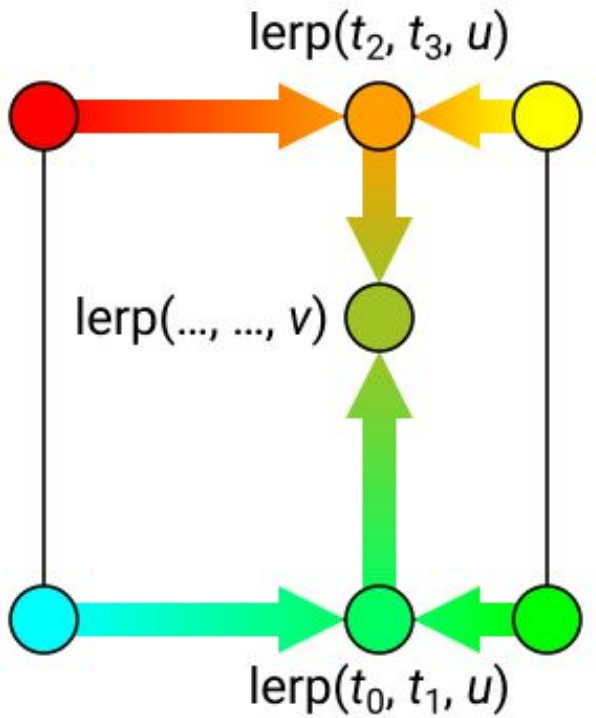




http://reedbeta.com/blog/quadrilateral-interpolation-part-2/

Bilinear Interpolation

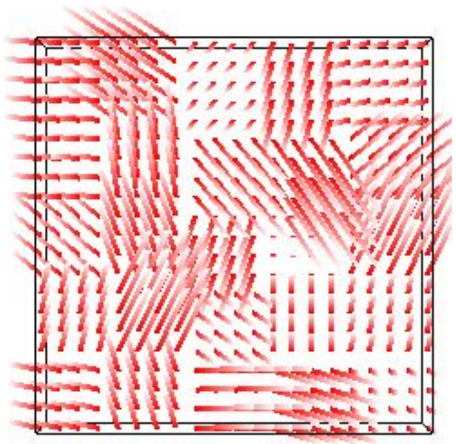
- It might be simplest to think about interpolating one axis at a time
  It doesn't matter which axis you start with!
- Calculate *u*, the fraction of the distance along the horizontal axis, ***e.g., u=0.65***
- Then calculate the top & bottom averages:
  *orange = (1-u)*red + u*yellow*
  *bluegreen = (1-u)*cyan + u*green*
- Calculate *v*, the fraction of the distance along the vertical axis, ***e.g., v=0.6***
- Then calculate the final average:
  *pukegreen = (1-v)*bluegreen + v*orange*



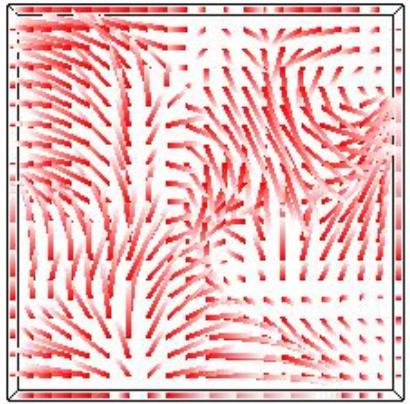
http://reedbeta.com/blog/quadrilateral-interpolation-part-2/

Correct Velocity Interpolation

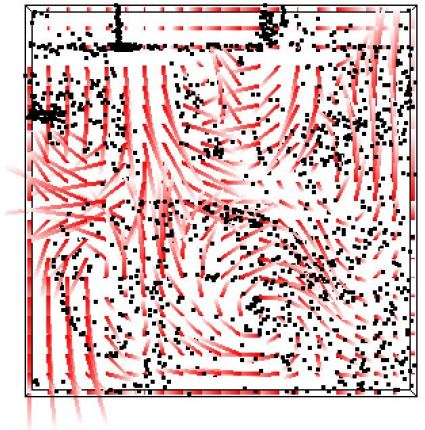
- WARNING: The finished code for 3D face velocity interpolation isn't particularly elegant… Storing velocities at face midpoints (req'd for conservation of mass) makes the index math messy!



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



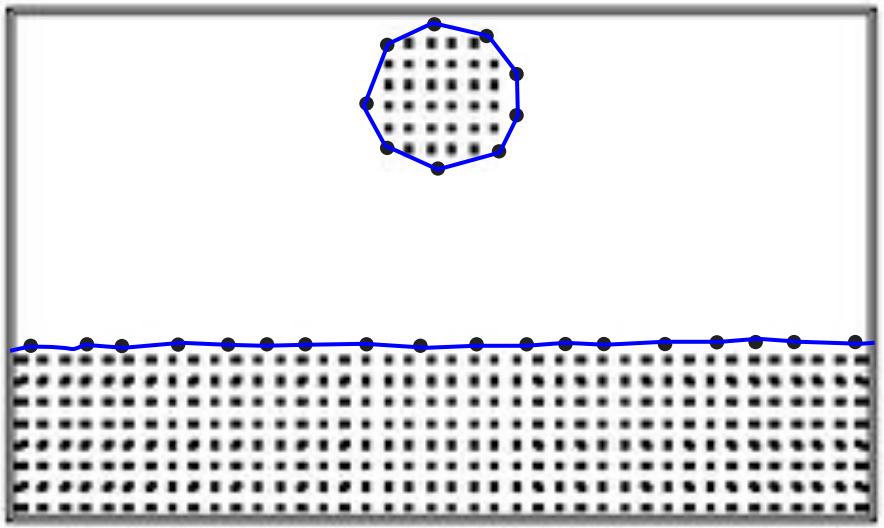
Buggy Interpolation
Note the clumping particles, and the discontinuities at some of the cell borders (& particles might escape the box!)

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!

Rendering via Surface Marker Particles

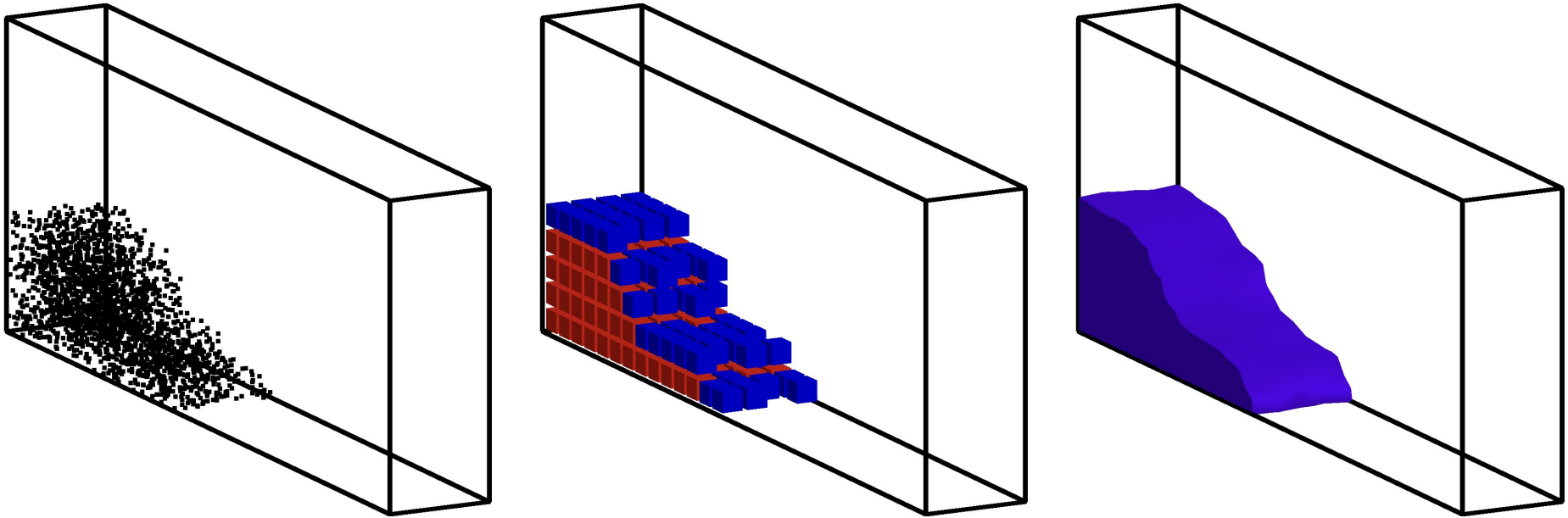
- Volumetric marker particles (black) track the volume the fluid/water (which cells contain fluid)
- Additionally, surface marker particles (blue) track the boundary between water & air
- Nearby surface particles are connected with edges & triangles



- *Where the surface geometry changes significantly, surface marker particles must be added and/or removed. The surface is re-triangulated. The surface topology may change as regions of fluid merge or separate.*

Fluid Surface Rendering using Marching Cubes

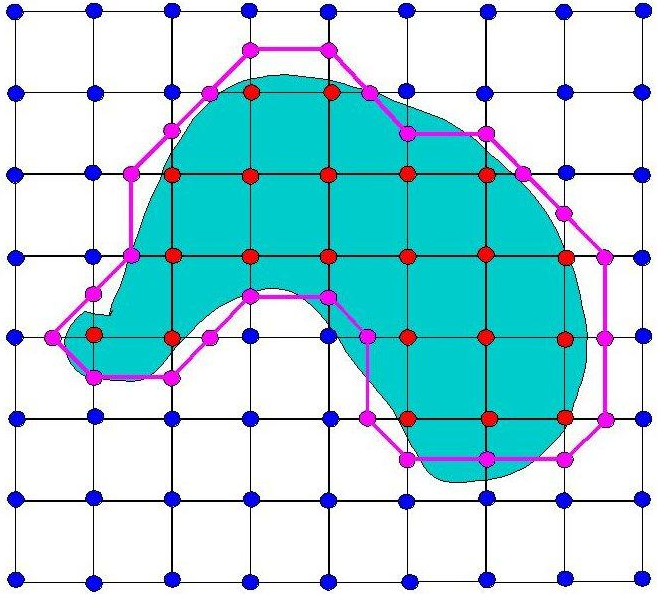
- Provided HW2 code approximates the surface using Marching Cubes (this is not the best/typical/recommended method)



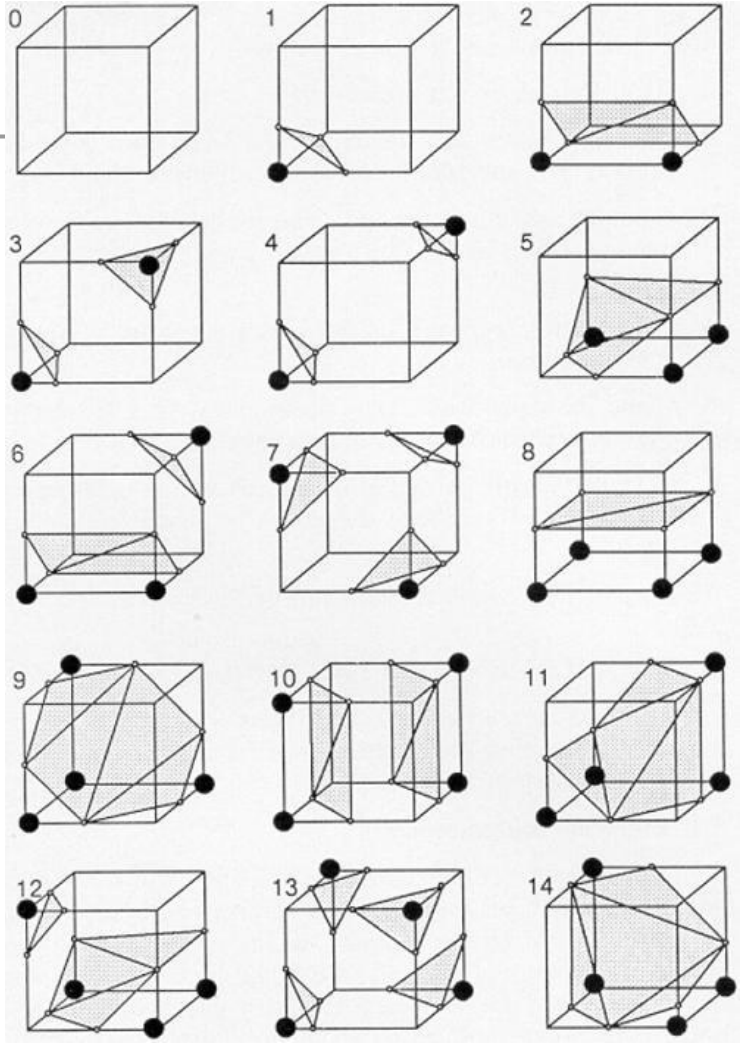
Marching Cubes

"Marching Cubes: A High Resolution 3D Surface Construction Algorithm", Lorensen and Cline, SIGGRAPH '87.

- Classic technique for extracting surface from scalar voxel data



http://www.cs.carleton.edu/cs_comps/0405/shape/marching_cubes.html



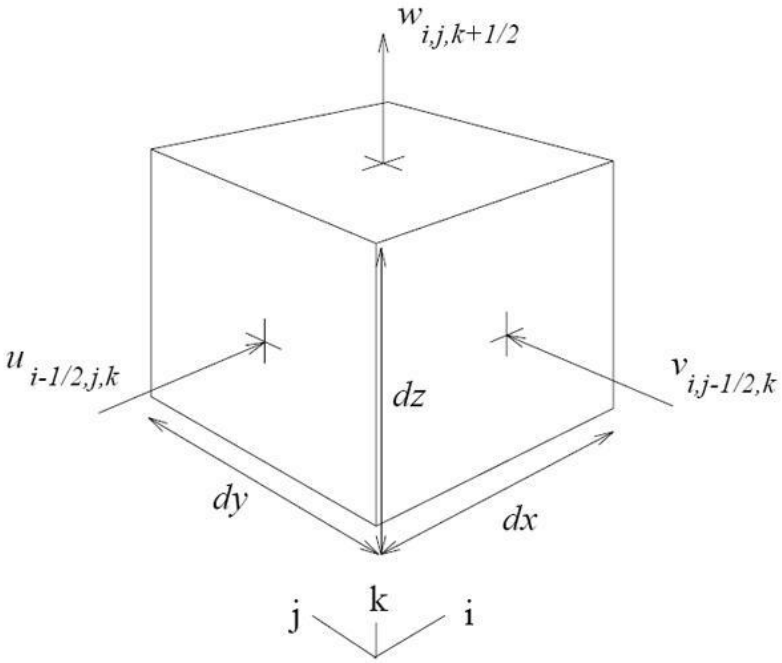
Today

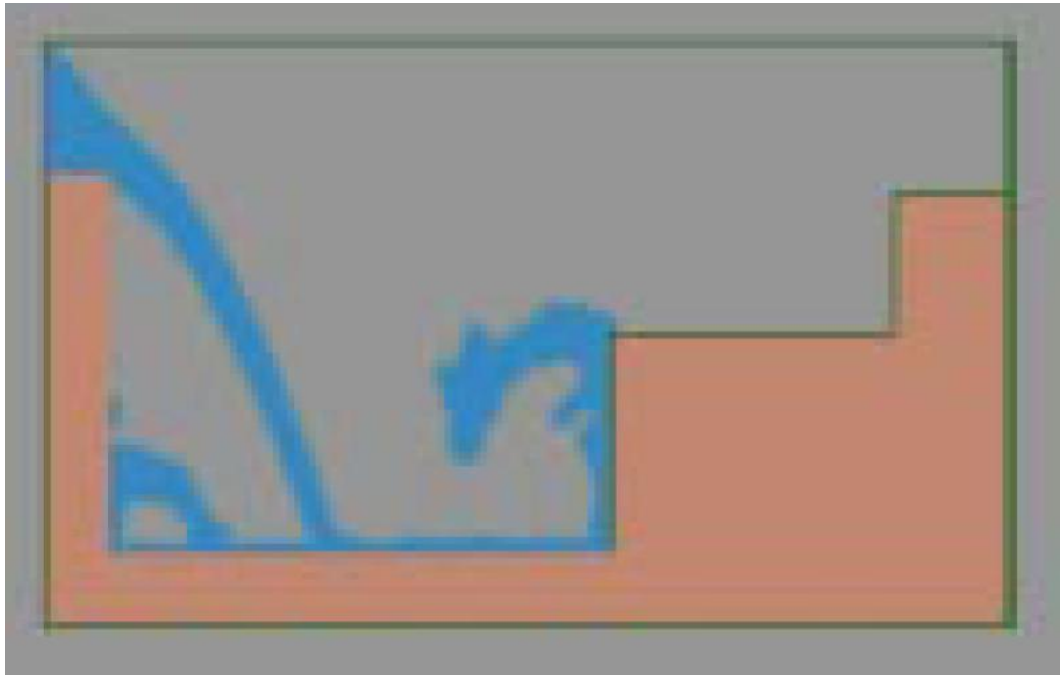
- Worksheet: Spatial Data Structures
- From Last Time: Stiffness & Discretization
- Papers for Today
- Flow Simulations in Computer Graphics
- Navier-Stokes Equations
- Fluid Representations
- Data Structure & Algorithm for Fluid Simulation
- Papers for Next Time...

Reading for Next Time

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

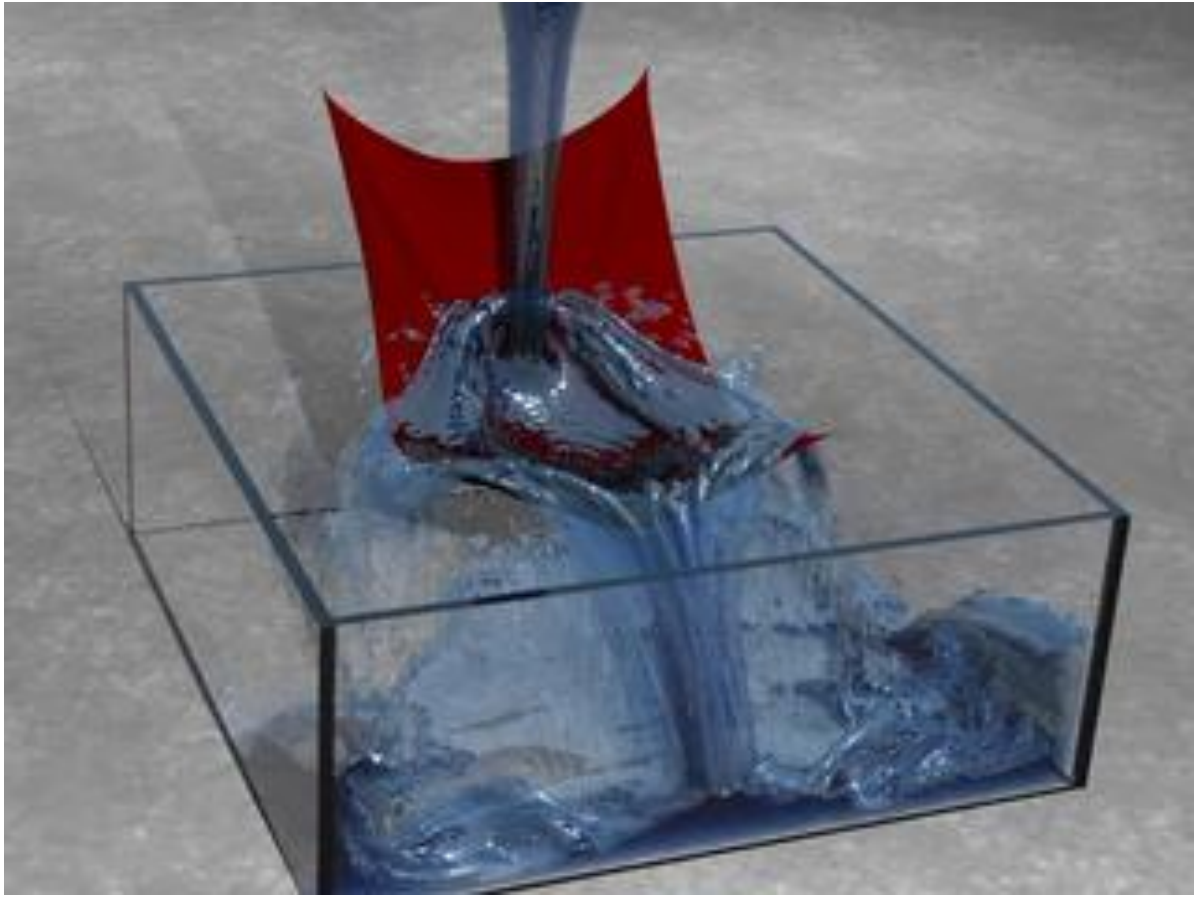
- "Realistic Animation of Liquids", Foster & Metaxas, 1996





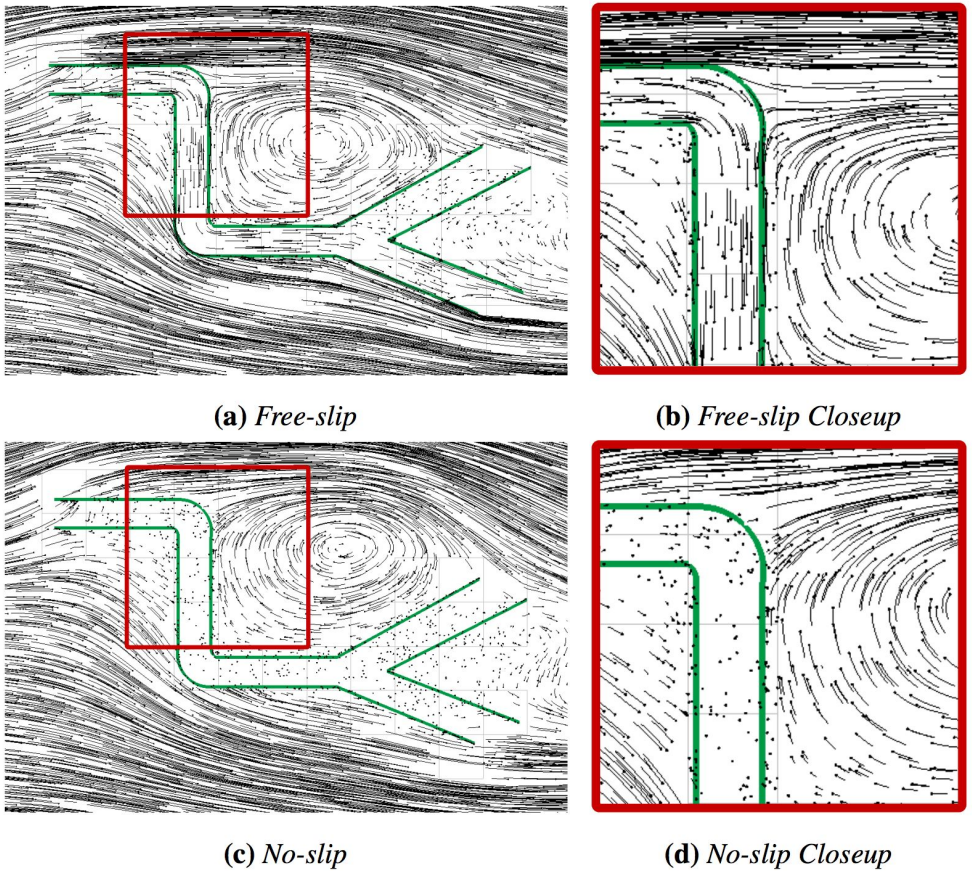
Optional Reading for Next Time

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



Optional Reading for Next Time

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



(a) *Free-slip*

(b) *Free-slip Closeup*

(c) *No-slip*

(d) *No-slip Closeup*