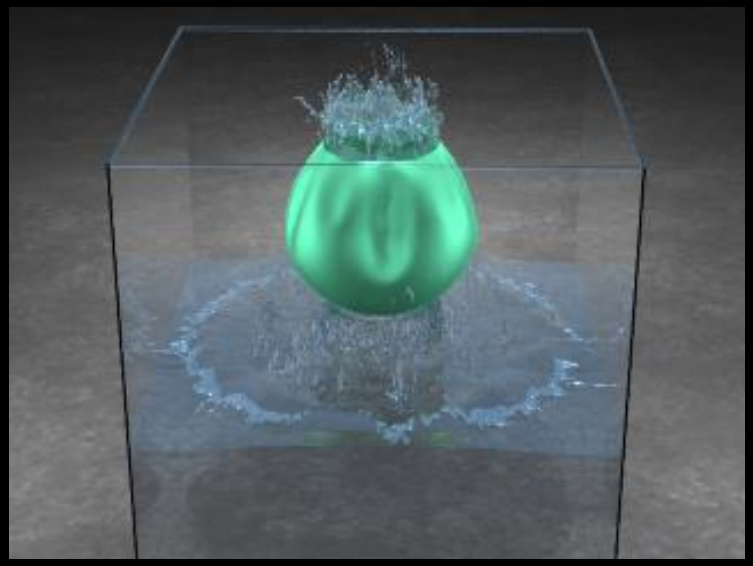
Fluid Simulation

Jos Stam, “Stable Fluid”



solid fluid coupling, Ron Fedkiw’s group

Robert Bridson and Matthias Müller-Fischer, “Fluid Simulation,” SIGGRAPH 2007 Course Notes

Fluid Behavior

- Lots of molecules smack into each other over and over and over
- Modeled as Continuum
  - Molecules way too tiny to simulate, even with the fastest computers
  - Break space into infinitesimal tidbits and do calculus
- Constrained Physics
  - F=ma
  - Conservation of mass
  - Conservation of momentum

What forces act on a fluid?

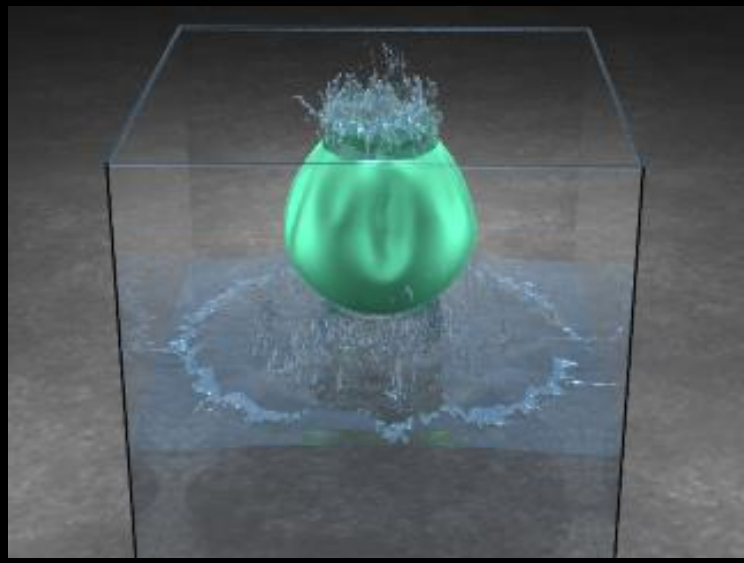
- Gravity
- Pressure

What forces act on a fluid?

- Gravity
- Pressure
- Viscosity

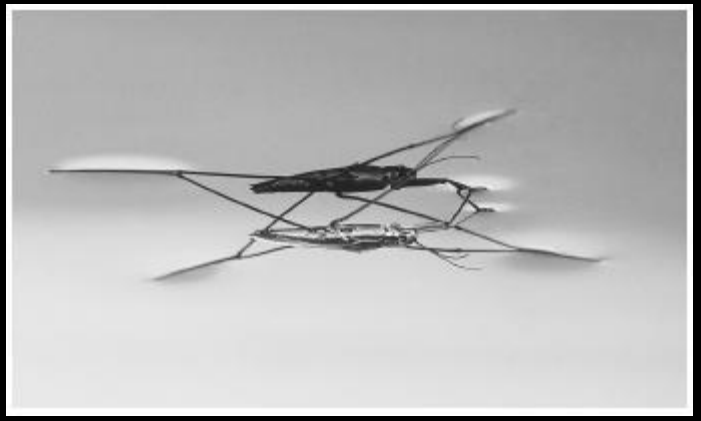
What forces act on a fluid?

- Gravity
- Pressure
- Viscosity
- Collisions with Boundaries
- Collisions with Other Objects



What forces act on a fluid?

- Gravity
- Pressure
- Viscosity
- Collisions with Boundaries
- Collisions with Other Objects
- Surface Tension



What forces act on a fluid?

- Gravity
- Pressure
- Viscosity
- Collisions with Boundaries
- Collisions with Other Objects
- Surface Tension
- Pressure from Other Fluids
- Chemical Reactions
- Elasticity

Vector Calculus Review

- Gradient

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$

  - maps scalar fields to vector fields

- Divergence

$$\nabla \cdot f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}$$

  - maps vector fields to scalar fields

- Laplacian

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$$

(f_x,f_y,f_z): x-,y-,z- component of vector function f

  - maps scalar fields to scalar fields

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$

Symbols

- $\vec{u}$: velocity with components (u,v,w)
- ρ: fluid density
- p: pressure
- $\vec{g}$: acceleration due to gravity or animator
- μ : dynamic viscosity

Force Equations

- Gravity
  - Constant acceleration

$$\vec{F}_{\text{gravity}} = m\vec{g}$$

- Pressure
  - From high to low pressure

$$\vec{F}_{\text{pressure}} = -V\nabla p$$

- Viscosity
  - Friction averages out nearby velocities

$$\vec{F}_{\text{viscosity}} = V\mu\nabla^2\vec{u}$$

- Collision with Boundary
  - Flip velocity in normal direction, for example

Two Viewpoints for Modeling Fluids

- Math depends on our coordinate system
- Eulerian Framework
  - Earth's point of view
  - Fluid moves, Earth stays fixed
  - Fixed grid
- Lagrangian Framework
  - Fluid's point of view
  - Fluid locally stays fixed; Everything else moves
  - Particle system

Eulerian vs. Lagrangian

- Take weather report for an example
- Eulerian
  - You stay on the ground, measuring pressure, temperature, humidity etc. of the air flowing past
- Lagrangian
  - You travel with a balloon in the wind, measuring pressure, temperature, humidity etc. of the air that's flowing alongside you

Eulerian Fluid

Forces derived for a continuum

- We don’t care about forces on a single point
- We want forces *per unit volume*
- Divide all equations by volume, to get *density* in place of *mass*

$$\vec{F} = m\vec{a} \quad \Rightarrow \quad \vec{f} = \rho\vec{a}$$

Navier-Stokes Equations!!!!!

- Derivation of momentum equation

$$\vec{f} = \rho \vec{a}$$

$$= \rho \frac{D\vec{u}}{Dt} = \rho \begin{bmatrix} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial u}{\partial z}\frac{\partial z}{\partial t} \\ \frac{\partial v}{\partial t} + \frac{\partial v}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial v}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial v}{\partial z}\frac{\partial z}{\partial t} \\ \frac{\partial w}{\partial t} + \frac{\partial w}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial w}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial w}{\partial z}\frac{\partial z}{\partial t} \end{bmatrix}$$

$$\vec{u}(t, \vec{x}) = (u, v, w)$$

$$\vec{u} \in R^4 \rightarrow R^3$$

Chain Rule

- Consider a scalar function first

$$\frac{d}{dt}q(t,\vec{x}) = \frac{\partial q}{\partial t}\frac{dt}{dt} + \frac{\partial q}{\partial x}\frac{dx}{dt} + \frac{\partial q}{\partial y}\frac{dy}{dt} + \frac{\partial q}{\partial z}\frac{dz}{dt}$$

$$= \frac{\partial q}{\partial t} + \nabla q \cdot \frac{d\vec{x}}{dt}$$

http://mathworld.wolfram.com/ChainRule.html

- For a vector function, simply apply the chain rule to each component function

Navier-Stokes Equations!!!!!

- Derivation of momentum equation

$$\vec{f} = \rho\vec{a}$$

$$= \rho\frac{D\vec{u}}{Dt} = \rho\begin{bmatrix} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}u + \frac{\partial u}{\partial y}v + \frac{\partial u}{\partial z}w \\ \frac{\partial v}{\partial t} + \frac{\partial v}{\partial x}u + \frac{\partial v}{\partial y}v + \frac{\partial v}{\partial z}w \\ \frac{\partial w}{\partial t} + \frac{\partial w}{\partial x}u + \frac{\partial w}{\partial y}v + \frac{\partial w}{\partial z}w \end{bmatrix}$$

Navier-Stokes Equations!!!!!

- Derivation of momentum equation

$$\vec{f} = \rho\vec{a}$$

$$= \rho\frac{D\vec{u}}{Dt} = \rho\begin{bmatrix} \frac{\partial u}{\partial t} + \vec{u}\cdot\nabla u \\ \frac{\partial v}{\partial t} + \vec{u}\cdot\nabla v \\ \frac{\partial w}{\partial t} + \vec{u}\cdot\nabla w \end{bmatrix} = \rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u}\right)$$

Navier-Stokes Equations!!!!!

- Adding all forces

$$\vec{f} = \rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u}\right)$$

$$= \rho\vec{g} - \nabla p + \mu\nabla^2\vec{u}$$

$$\begin{cases} \vec{f}_{\text{gravity}} = \rho\vec{g} \\ \vec{f}_{\text{pressure}} = -\nabla p \\ \vec{f}_{\text{viscosity}} = \mu\nabla^2\vec{u} \end{cases}$$

- The momentum equation

$$\frac{\partial \vec{u}}{\partial t} + \vec{u}\cdot\nabla\vec{u} + \frac{1}{\rho}\nabla p = \vec{g} + \frac{\mu}{\rho}\nabla^2\vec{u}$$

Incompressibility

- Real fluids are compressible
- Shock waves, acoustic waves, pistons…
  - Note: liquids change their volume as well as gases, otherwise there would be no sound underwater
- But this is nearly irrelevant for animation
  - Shocks move too fast to normally be seen (easier/better to hack in their effects)
  - Acoustic waves usually have little effect on visible fluid motion
  - Pistons are boring

Incompressibility

- Rather than having to simulate acoustic and shock waves, eliminate them from our model: assume fluid is incompressible
  - Turn stiff system into a constraint, just like rigid bodies!

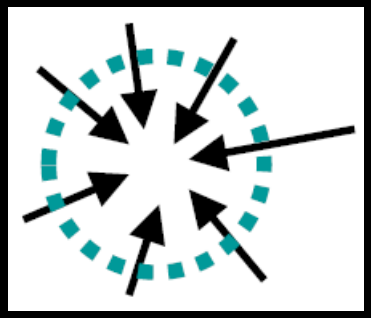
$$\nabla \cdot \vec{u} = 0$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + \frac{\mu}{\rho} \nabla^2 \vec{u}$$

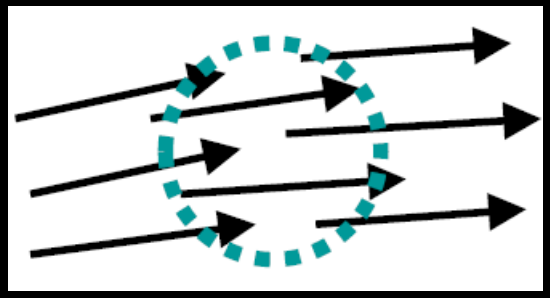
Divergence

$$\nabla \cdot f = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}$$

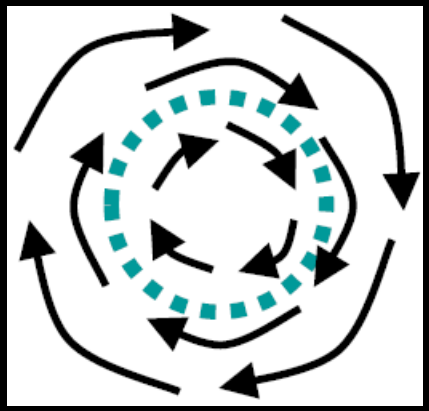
- Total flux through a surface
  - Measure of compressibility



compressible



incompressible



incompressible

$$\nabla \cdot \vec{u} = 0$$

Pressure

- Pressure p:
  whatever it takes to make the velocity field divergence free
- If you know constrained dynamics, $\nabla \cdot \vec{u} = 0$ is a constraint, and pressure is the matching Lagrange multiplier
- Our simulator will follow this approach:
  - solve for a pressure that makes our fluid incompressible at each time step.

Dropping Viscosity

- In most scenarios, viscosity term is much smaller
- Convenient to drop it from the equations:
  - Zero viscosity = “inviscid”
  - Inviscid Navier-Stokes = “Euler equations”
- Numerical simulation typically makes errors that resemble physical viscosity, so we have the visual effect of it anyway
  - Called “numerical dissipation”
  - For animation: often numerical dissipation is larger than the true physical viscosity!

The inviscid equations

- A.k.a. the incompressible Euler equations

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g}$$

$$\nabla \cdot \vec{u} = 0$$

Boundary Conditions

- We know what's going on inside the fluid: what about at the surface?
- Three types of surface
  - Solid wall: fluid is adjacent to a solid body
  - Free surface: fluid is adjacent to nothing (e.g. water is so much denser than air, might as well forget about the air)
  - Other fluid: possibly discontinuous jump in quantities (density, …)

Solid Wall Boundaries

- No fluid can enter or come out of a solid wall:

$$\vec{u} \cdot \hat{n} = \vec{u}_{solid} \cdot \hat{n}$$

- For common case of $\vec{u}_{solid} = 0$

$$\vec{u} \cdot \hat{n} = 0$$

- Sometimes called the “no-stick” condition, since we let fluid slip past tangentially
  - For viscous fluids, can additionally impose “no-slip” condition: $\vec{u} = \vec{u}_{solid}$

Free Surface

- Neglecting the other fluid, we model it simply as pressure=constant
  - Since only pressure gradient is important, we can choose the constant to be zero:

$$p = 0$$

- If surface tension is important (not covered here), pressure is instead related to mean curvature of surface
  - there is a jump in pressure between two fluids

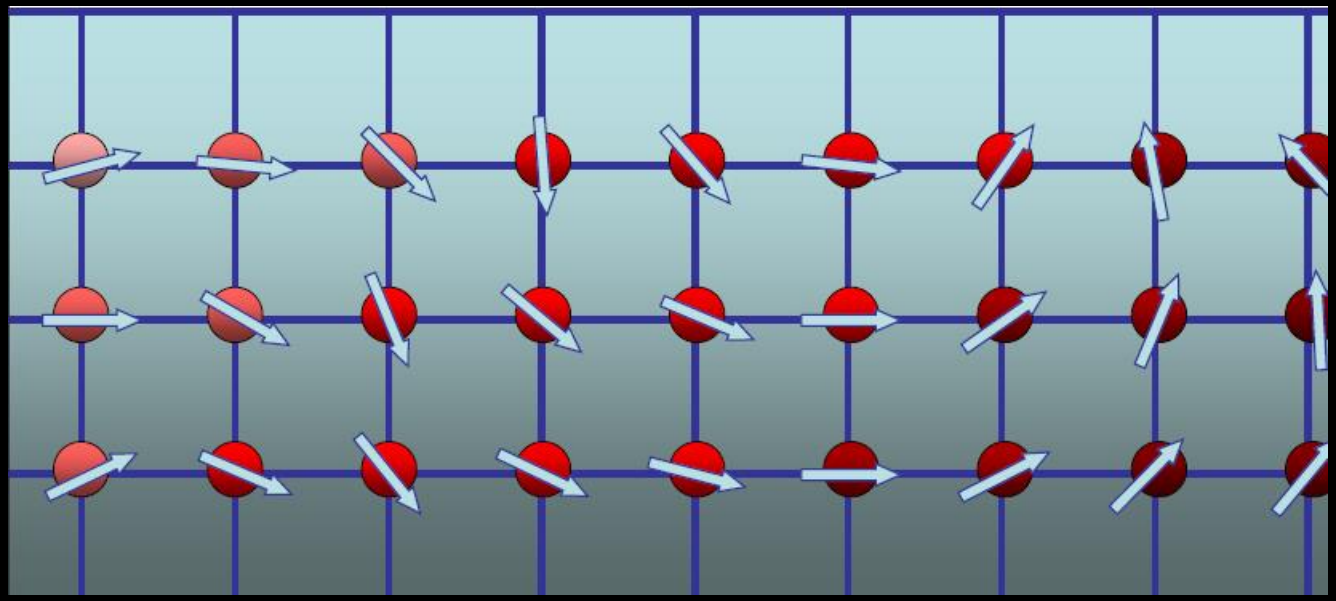
Multiple Fluids

- At fluid-fluid boundaries, the trick is to determine “jump conditions”
  - For a fluid quantity q, $[q]=q_1-q_2$
- Density jump [ρ] is known
- Normal velocity jump must be zero $\vec{u} \cdot \hat{n} = 0$
- For inviscid flow, tangential velocities may be unrelated (jump is unknown)
- With no surface tension, pressure jump [p]=0

Numerical Simulation Overview

Eulerian Grid

- Break space up into tiny cubic cells
  - Store data at cell corners, for example



Advection/Transport

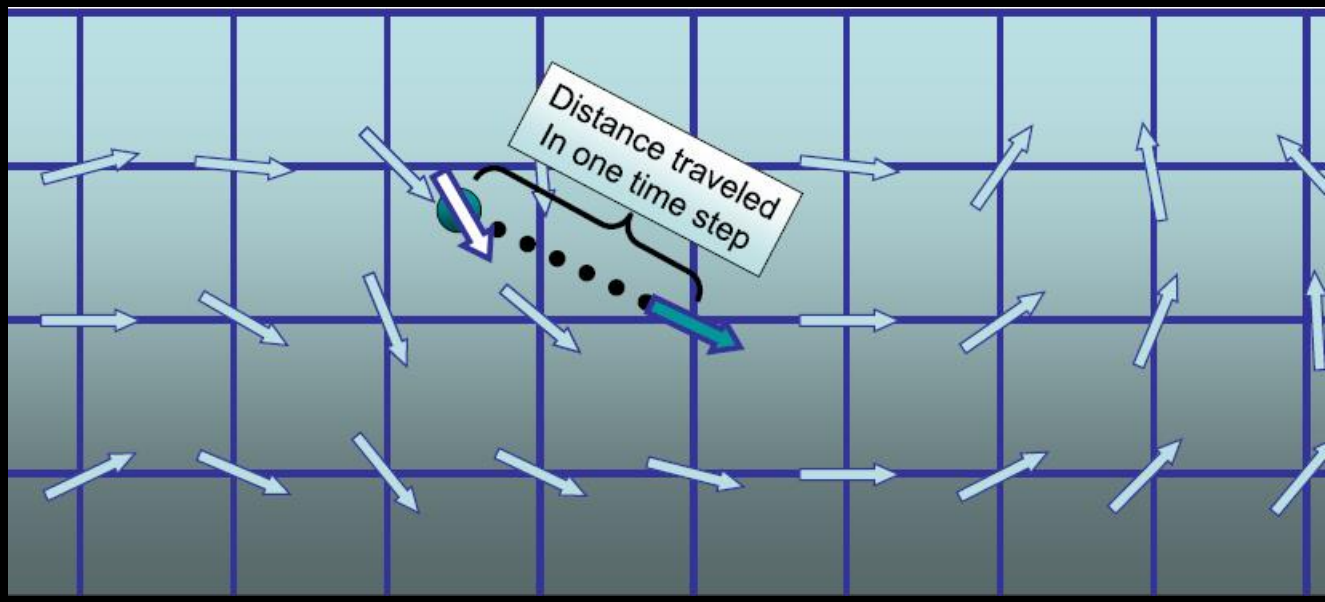
$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \vec{u} \cdot \nabla q$$

- A quantity moves with the velocity field $\vec{u}$
  - e.g. smoke density, or velocity itself

$$\frac{Dq}{Dt} = \frac{\partial q}{\partial t} + \vec{u} \cdot \nabla q$$

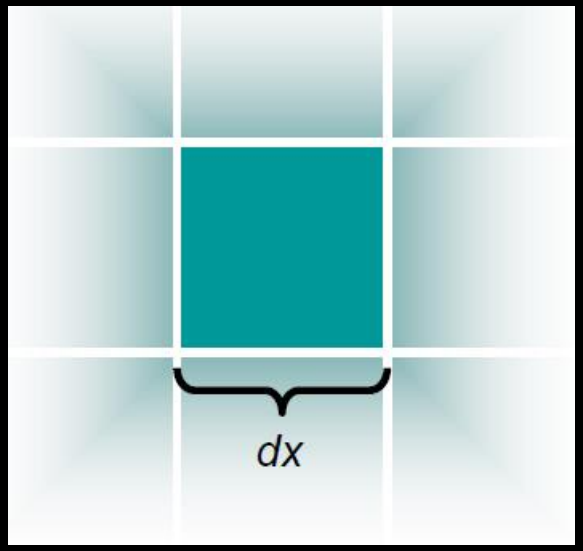
change due to spatial variation ▽q propagated by velocity field u

How fast q is changing at a fixed point



Finite Differences

- Approximate all derivatives
  - Use grid spacing to approximate infinitely small dx



$$\frac{df}{dx} = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}$$

Basic Idea for Solving the Equation

- After discretizing all derivatives with finite differences, we have two linear equations
- Two equations with two unknowns
  - Pressure p
  - Future velocity u
- Solve linear system
  - Done!
    - Repeat with new velocity in the next time step
  - Pressure acts as a Lagrange Multiplier!

Splitting

- We have lots of terms in the momentum equation: a pain to handle them all simultaneously
- Instead we split up the equation into its terms, and integrate them one after the other
  - Makes for easier software design too: a separate solution module for each term
- First order accurate in time
  - Can be made more accurate, not covered today.

A Splitting Example

- Say we have a differential equation

$$\frac{dq}{dt} = f(q) + g(q)$$

- And we can solve the component parts:
  - SolveF(q,Δt) solves dq/dt=f(q) for time Δt
  - SolveG(q,Δt) solves dq/dt=g(q) for time Δt
- Put them together to solve the full thing:
  - $q^* = \text{SolveF}(q^n, \Delta t)$
  - $q^{n+1} = \text{SolveG}(q^*, \Delta t)$

Does it work?

$$\dot{q} = f(q) + g(q) \Rightarrow \begin{cases} \dot{q} = f(q) \\ \dot{q} = g(q) \end{cases}$$

- Suppose Euler method is used
  - q* = SolveF(q^n, Δt) $\longrightarrow$ $q^* \approx q^n + \Delta t f(q^n)$
  - q^{n+1} = SolveG(q*, Δt) $\longrightarrow$ $q^{n+1} \approx q^* + \Delta t g(q^*)$

$$\frac{dq}{dt} \approx \frac{q^{n+1} - q^n}{\Delta t}$$

$$= \frac{q^{n+1} - q^*}{\Delta t} + \frac{q^* - q^n}{\Delta t}$$

$$\approx g(q) + f(q)$$

See course notes for accuracy analysis!

Splitting Momentum into three terms

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g}$$

- First term: advection $\frac{\partial \vec{u}}{\partial t} = -\vec{u} \cdot \nabla \vec{u}$
  - Move the fluid through its velocity field ($\frac{D\vec{u}}{Dt} = 0$)
- Second term: gravity $\frac{\partial \vec{u}}{\partial t} = \vec{g}$
- Final term: pressure update $\frac{\partial \vec{u}}{\partial t} = -\frac{1}{\rho} \nabla p$
  - How we'll make the fluid incompressible $\nabla \cdot \vec{u} = 0$

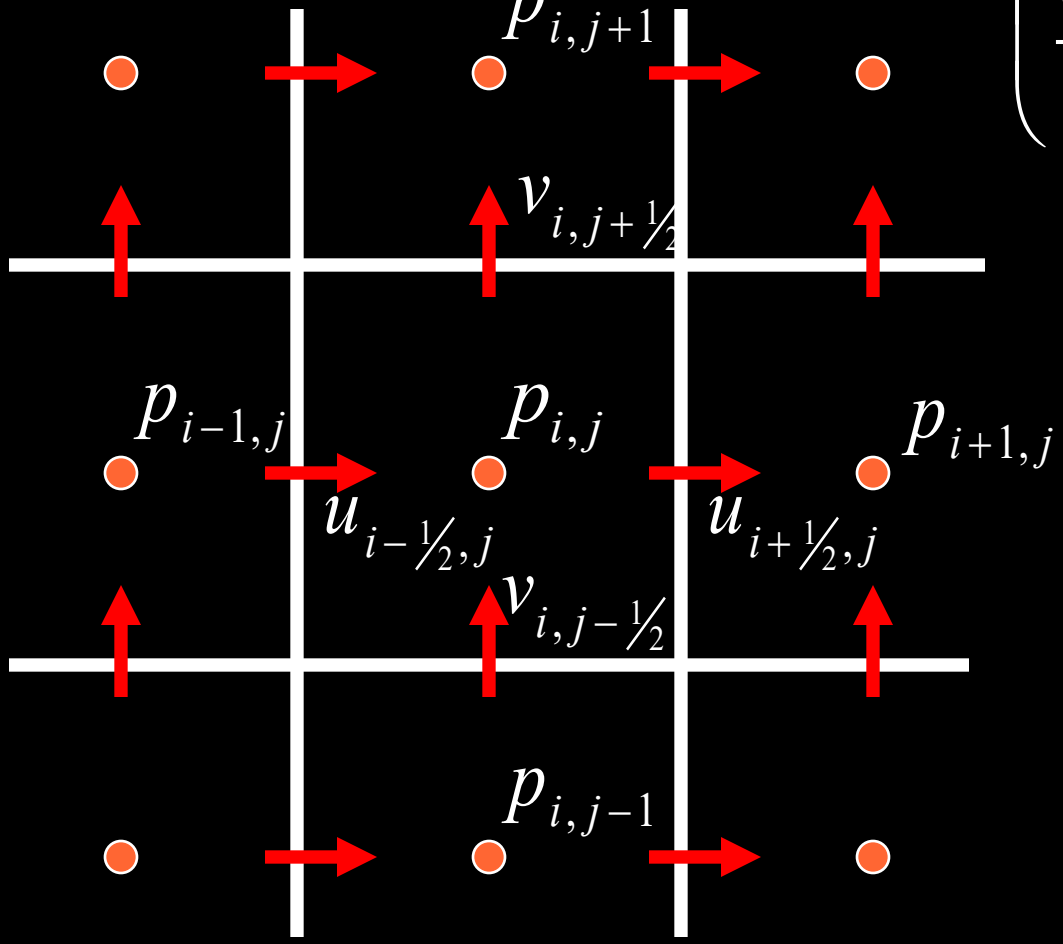
Space

- That's our general strategy in time; what about space?
- We'll begin with a fixed Eulerian grid
  - Trivial to set up
  - Easy to approximate spatial derivatives
  - Particularly good for the effect of pressure
- Disadvantage: advection doesn't work so well
  - Later: particle methods that fix this

The MAC Grid

- From the Marker-and-Cell (MAC) method [Harlow&Welch'65]
- A particular staggering of variables in 2D/3D that works well for incompressible fluids:
  - Grid cell (i,j,k) has pressure $\mathbf{p}_{i,j,k}$ at its center
  - x-part of velocity $\mathbf{u}_{i+1/2,jk}$ in middle of x-face between grid cells (i,j,k) and (i+1,j,k)
  - y-part of velocity $\mathbf{v}_{i,j+1/2,k}$ in middle of y-face
  - z-part of velocity $\mathbf{w}_{i,j,k+1/2}$ in middle of z-face

MAC Grid in 2D



$$\left(\frac{\partial \vec{u}}{\partial x}\right)_i = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x}$$

Put everything together

- Start with a divergence-free velocity field
- At each time step
  - Set $\vec{u}^A = \texttt{advect}(\vec{u}^n, \Delta t, \vec{u}^n)$
  - Add $\vec{u}^B = \vec{u}^A + \Delta t \vec{g}$
  - Set $\vec{u}^{n+1} = \texttt{project}(\Delta t, \vec{u}^B)$

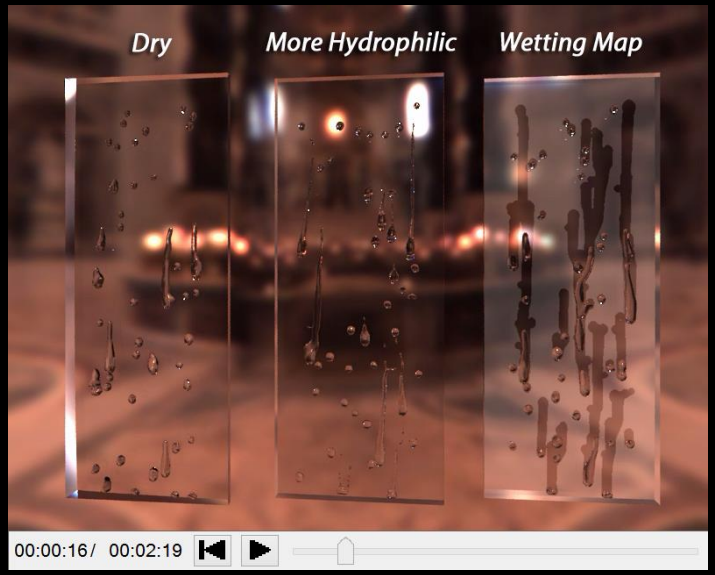
See Chapter 4 in the SIGGRAPH'07 course note

Finite Difference Fluid Pros & Cons

- Good
  - Unconditionally stable!
  - Looks more realistic
  - Smooth surfaces
- Bad
  - Slow!
  - Memory intensive
  - Real time fluid is too viscous
  - Can lose mass over time

Eulerian Fluid Demos

- Jos Stam -Stable Fluids
- Georgia Tech fluids
  - "Rigid Fluid"
  - "Water Drops on Surfaces," SIGGRAPH'05



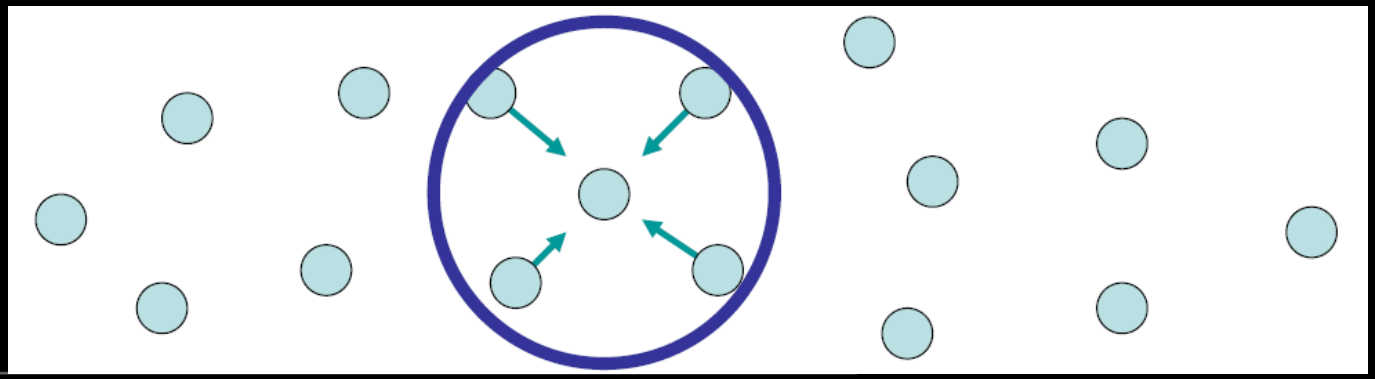
Eulerian Fluid Demos

- Stanford fluids
  - Wrinkled flames and cellular patterns
  - Vortex particle method for smoke, water, and explosions
  - Two-way coupled SPH and particle level set fluid simulation
- Other notable fluid animations
  - Wavelet Turbulence for Fluid Simulation
  - Liquid Simulation on Lattice-Based Tetrahedral Meshes

Lagrangian Fluid

Smoothed Particle Hydrodynamics (SPH)

- Lagrangian Framework
- Model fluid as a particle system
  - Remember particle systems!?!?
  - Forces act on each particle
- A Particle represents a sample of fluid
  - NOT individual molecules!

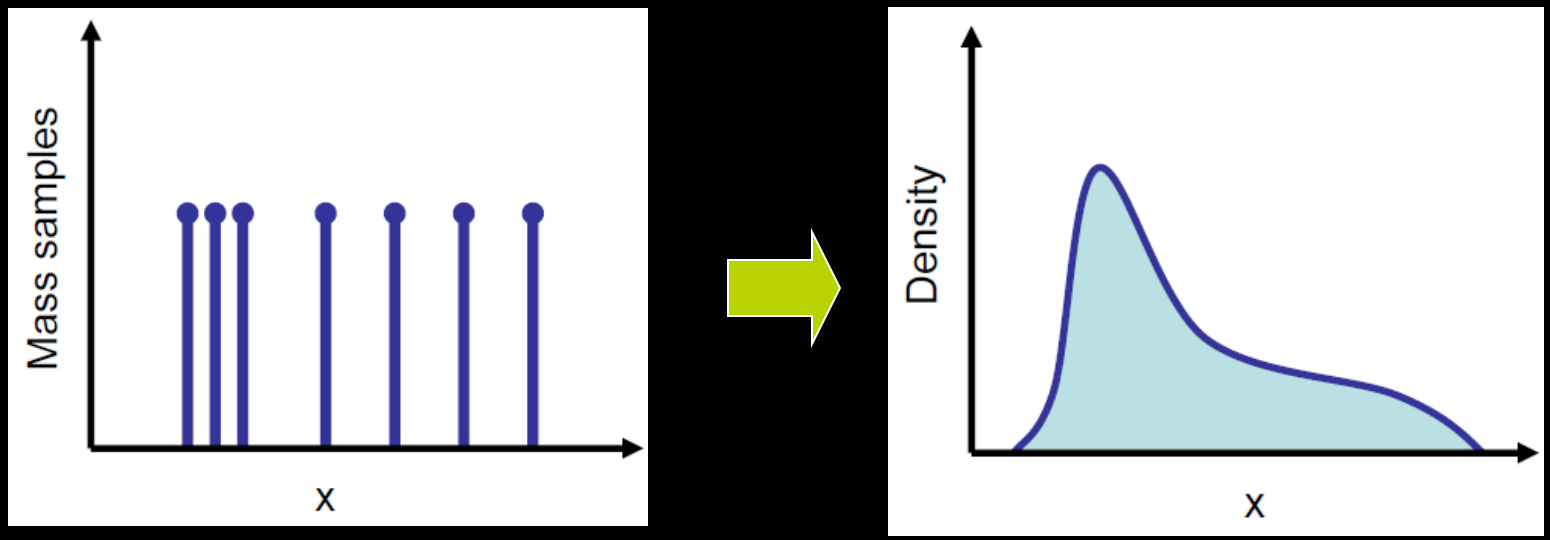


SPH Particles

- Particles only record
  - Position x
  - Velocity v
  - Mass m
- How do we compute
  - density?
  - pressure?
  - gradients?
  - ???

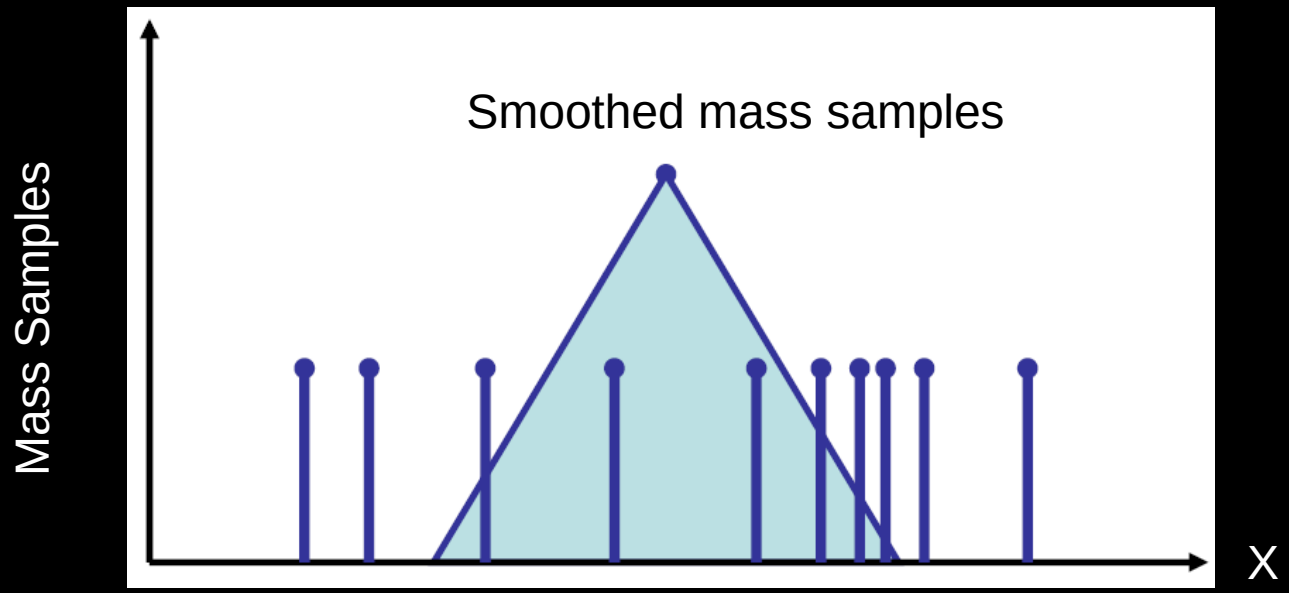
Function Approximation

- We want spatial derivatives (density, pressure, etc)
- We can only take spatial derivatives of smooth functions
- How do we turn infinitesimal point samples into a smooth function?



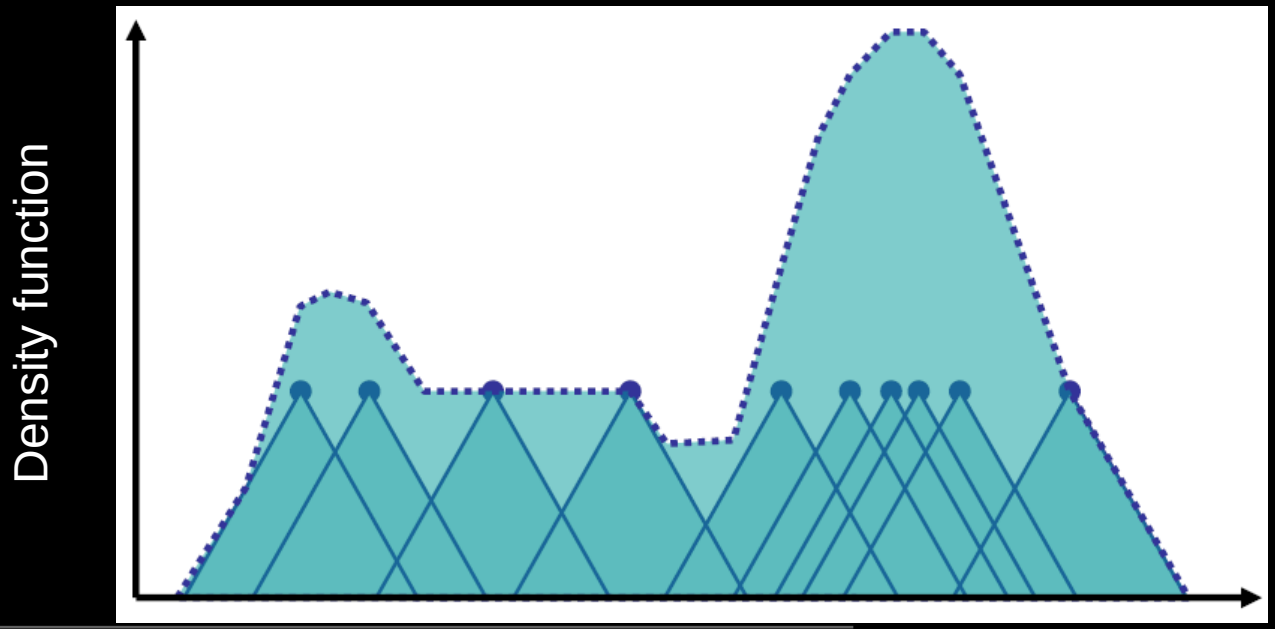
Smoothing Kernel

- Smooth out particle so that it occupies some space



Smoothing Kernel

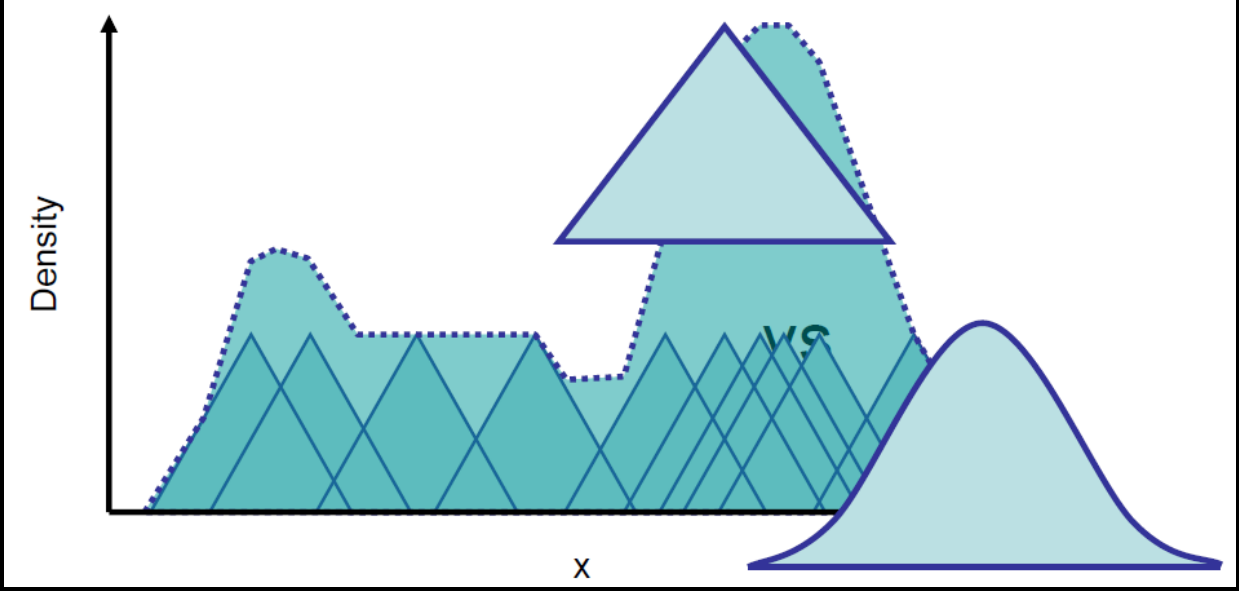
- Smooth out particle so that it occupies some space
- Add all of these smoothed particles together to make a continuous function



Smoothing Kernel

- Smoother kernel means smoother derivatives
- Kernel needs to be normalized, i.e.

$$\int W(|\mathbf{x} - \mathbf{x}_j|)d\mathbf{x} = 1$$

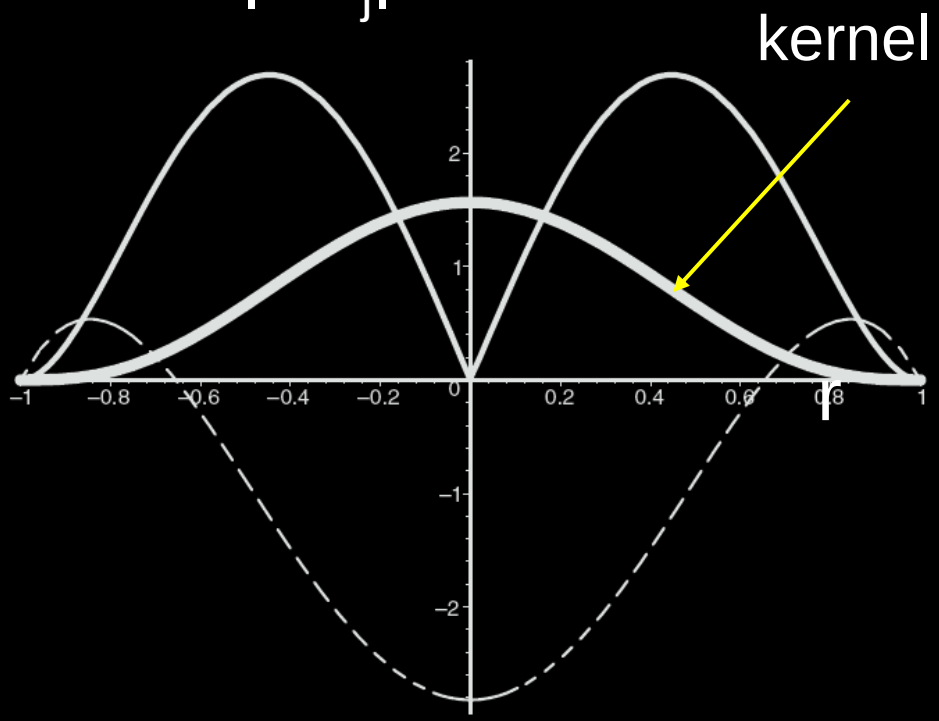


Example Smoothing Kernel

$$W_{\text{poly6}}(\mathbf{r}, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & 0 \le r \le h \\ 0 & \text{otherwise} \end{cases}$$

r = distance from particle = $|x-x_j|$

h = kernel width



Constructing Smoothed Field

- Density field $\rho(\mathbf{x}) = \sum_j m_j W(|\mathbf{x} - \mathbf{x}_j|)$
- Density of a particle $\rho_i = \rho(\mathbf{x}_i)$
- Smoothed field of arbitrary attributes A_i of the particles are

$$A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(|\mathbf{x} - \mathbf{x}_j|)$$

- Gradient of smoothed field

$$\nabla A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(|\mathbf{x} - \mathbf{x}_j|)$$

SPH Forces

$$\nabla A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(|\mathbf{x} - \mathbf{x}_j|)$$

- Pressure

$$\mathbf{f}_i^{pressure} = -\nabla p(\mathbf{x}_i) = \sum_j m_j \frac{p_j}{\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|)$$

Not symmetric! Consider only two particles..
Since the gradient is zero at center, particle i only uses the pressure of particle j to computes its pressure forces and vice versa

$$\mathbf{f}_i^{pressure} = \sum_j m_j \frac{p_i + p_j}{\rho_j} \nabla W(|\mathbf{x}_i - \mathbf{x}_j|)$$

$$p = k(\rho - \rho_0)$$

Note pressure is not carried, so need to be computed

SPH Forces

- Viscosity

$$\mathbf{f}_i^{pressure} = \mu \nabla^2 \mathbf{v}(\mathbf{x}_i) = \mu \sum_j m_j \frac{\mathbf{v}_j}{\rho_j} \nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|)$$

Since viscosity forces are only dependent on velocity differences and not on absolute velocities, there is a natural way to symmetrize the viscosity forces by using velocity differences

$$\mathbf{f}_i^{pressure} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(|\mathbf{x}_i - \mathbf{x}_j|)$$

Interpretation: particle i is accelerated in the direction of the relative speed of its environment

SPH Pros & Cons

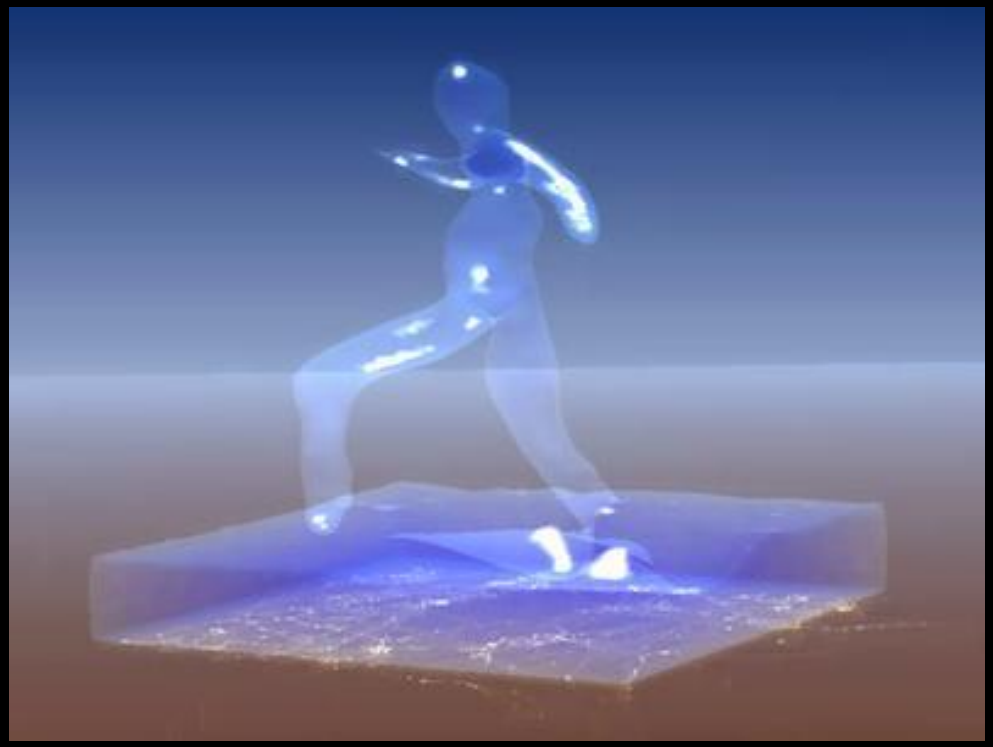
- Good
  - Long term conservation of mass
  - Conservation of momentum
    - Symmetric particle forces
  - Fast!!!
  - Unrestricted simulation domain
- Bad
  - Can become unstable
    - Requires extra damping
  - Compressibility
  - Surface rendering needs improvement

SPH Demos

- Muller et al. “Particle-Based Fluid Simulation for Interactive Applications,” SCA’03, video
- Adams et al. “Adaptively Sampled Particle Fluids,” SIGGRAPH’07, video
- Becker et al. “Weakly compressible SPH for free surface flows,” SCA’07, video

High-level Control of Fluid Simulation

- Smoke
- Cloud
- Liguid



KEYFRAME CONTROL OF SMOKE SIMULATIONS, SIGGRAPH'03

- Adrien Treuille, Antoine McNamara, Zoran Popović, Jos Stam
- Control via optimized "wind" force to the velocity field
- Multiple-shooting approach for animations with several keyframes
  - splitting large problems into smaller overlapping subproblems
- video

Guiding of Smoke Animations

- Nielsen et al., “Improved Variational Guiding of Smoke Animations, EG’10
- Existing methods either lack high frequency detail or may result in undesired features developing over time
- Guide Eulerian-based smoke animations through coupling of simulations at different grid resolutions
- video

Feedback Control of Cloud Formation

- Dobashi, Kusumoto, Nishita, Yamamoto, "Feedback control of cumuliform cloud formation based on computational fluid dynamics," SIGGRAPH'08

Figure 1: *Clouds generated by our method. The pink curve indicates the desired shape specified by the user. Our method controls the cloud formation process in order to make clouds form the desired shape.*

FLUID CONTROL USING THE ADJOINT METHOD, SIGGRAPH'04

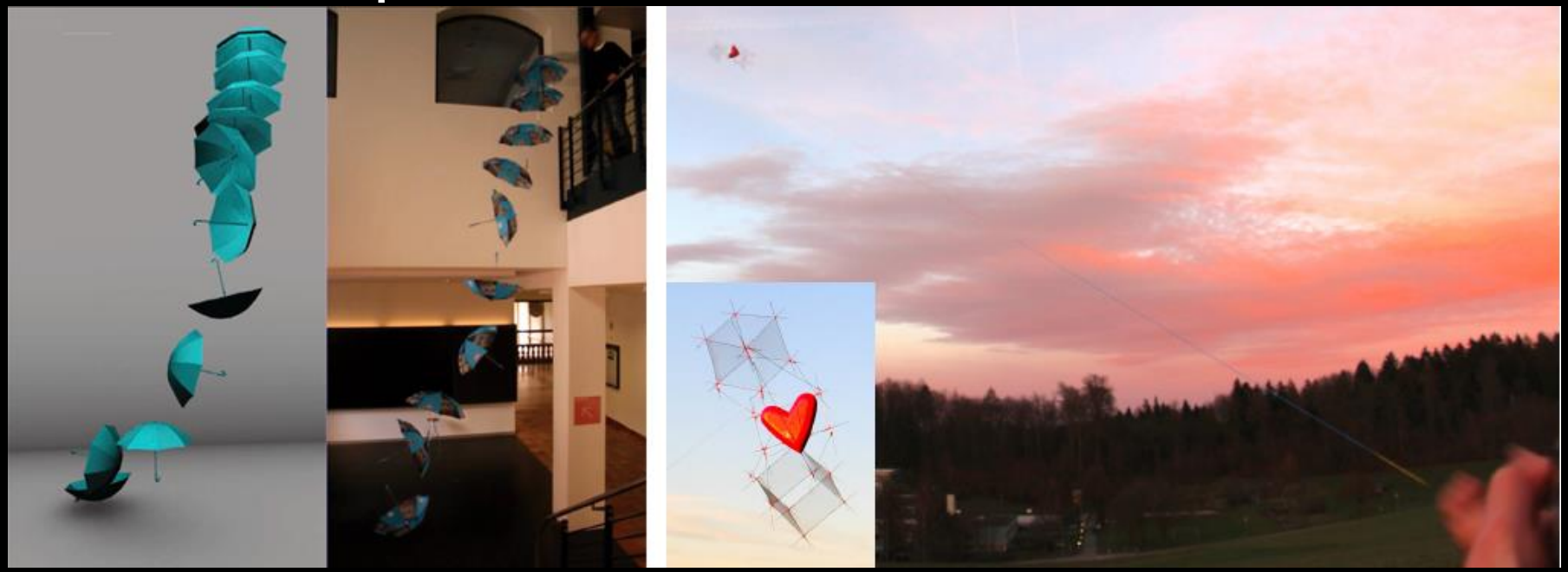
- Antoine McNamara, Adrien Treuille, Zoran Popović, Jos Stam
- Clay man
- Water man
- Smoke man

Physically Guided Liquid Surface Modeling from Videos, SIGGRAPH'09

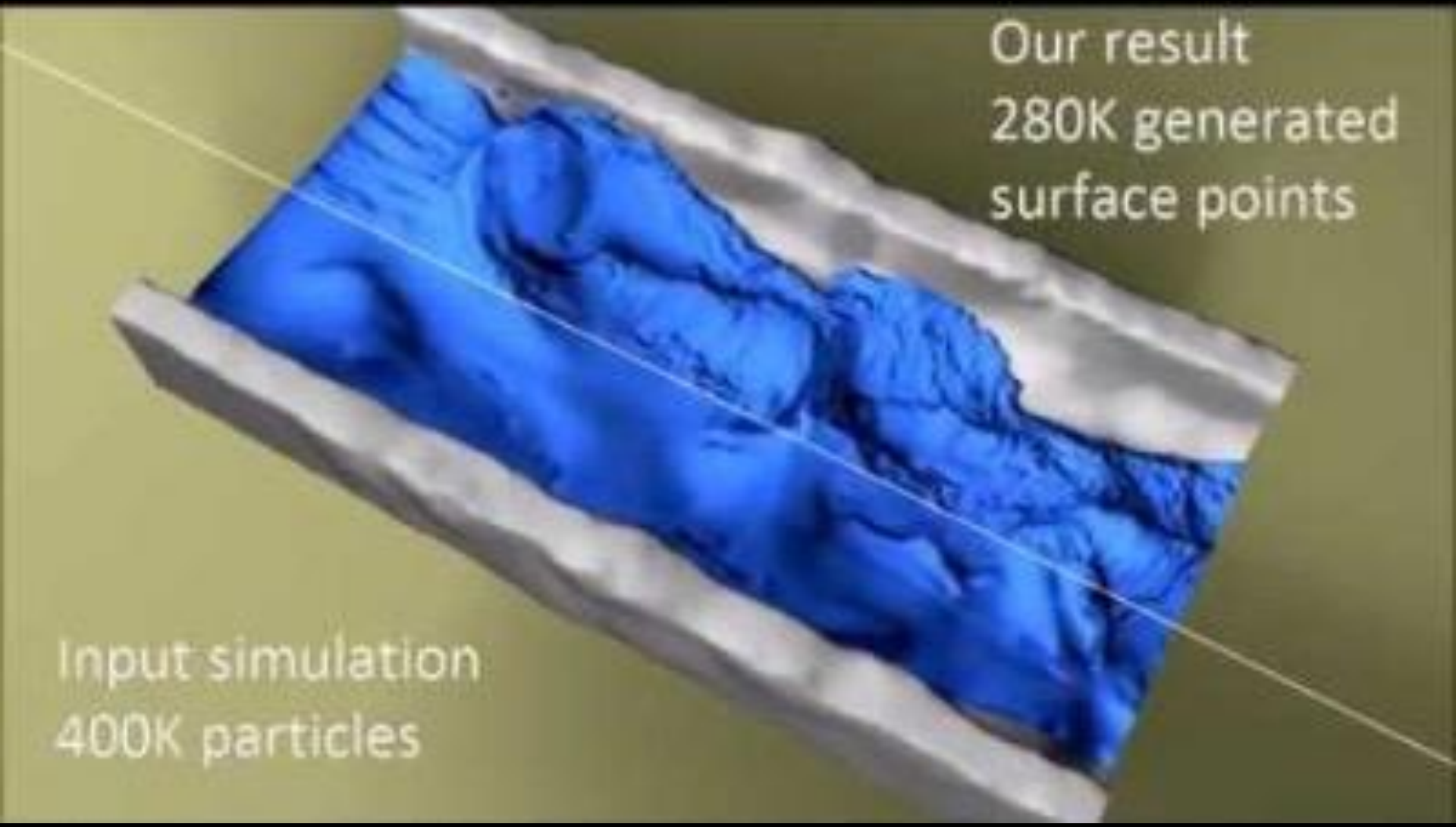
- Huamin Wang, Miao Liao, Qing Zhang, Ruigang Yang and Greg Turk
- Video

OmniAD, SIGGRAPH 2015

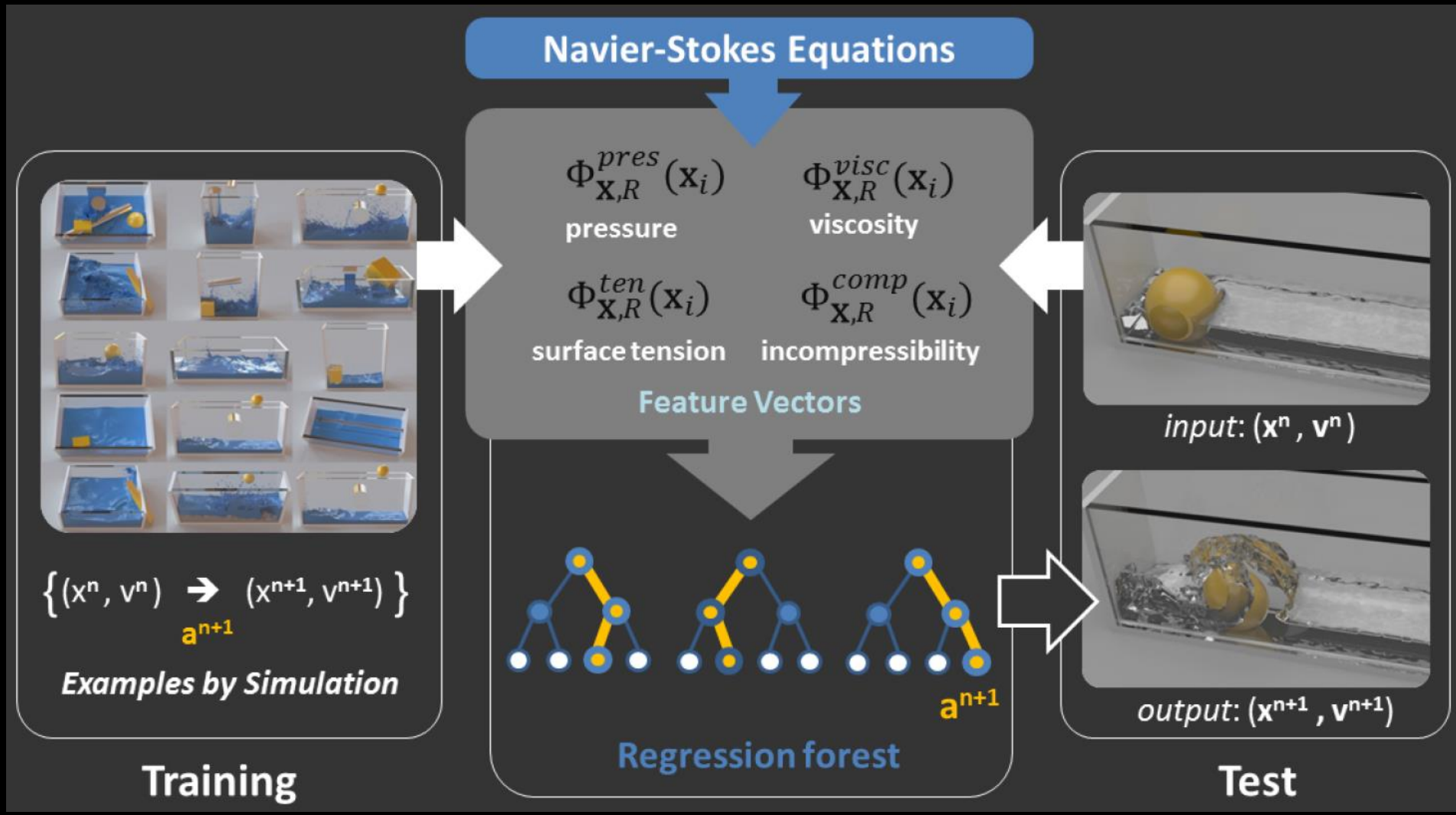
- Data-driven approach to model and acquire the aerodynamics of 3D rigid objects
- Evaluate aerodynamics forces using SH-based representation



Mercier et al., “Surface Turbulence for Particle-Based Liquid Simulations” SIGGRAPH Asia’15



Data-Driven Fluid Simulations using Regression Forests



Further references

- **"SPH Fluids in Computer Graphics,"** *Eurographics 2014 - State of the Art Reports*, pp. 21-42, 2014.