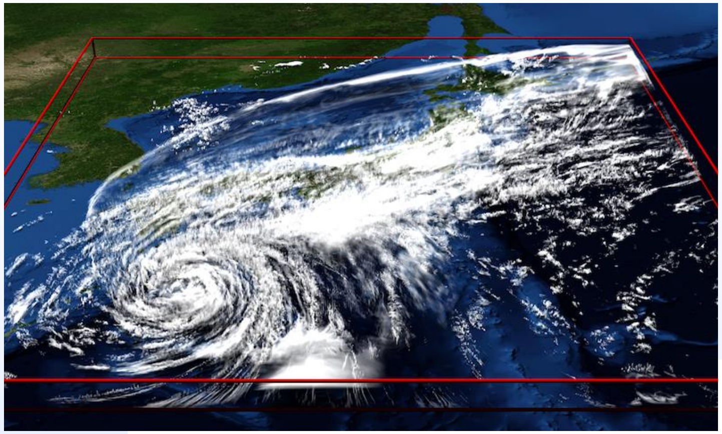
Fluid Simulation

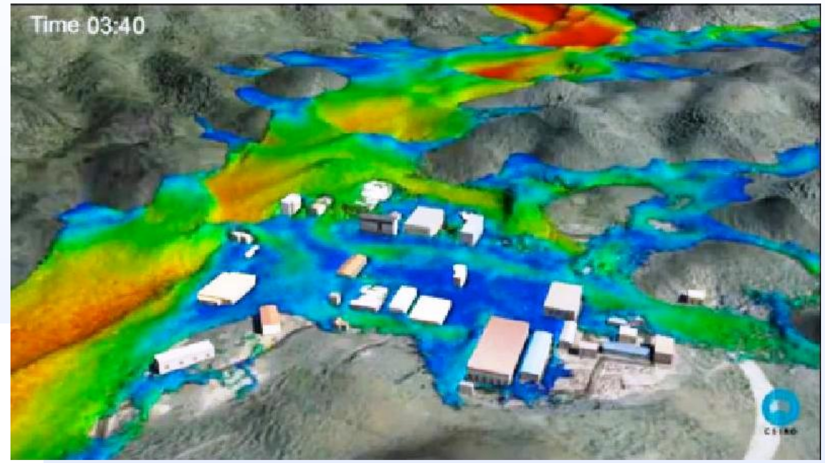

[Thürey 10]

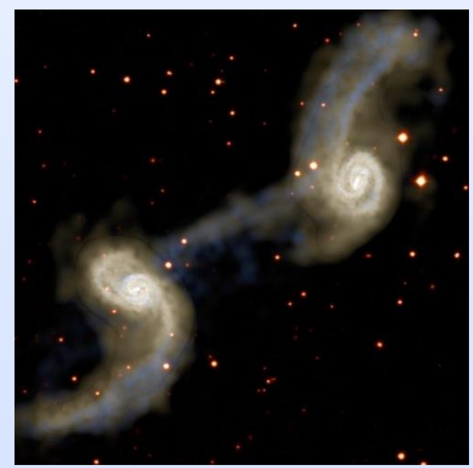

[Pfaff 10]


[Chentanez 11]

Computational Fluid Dynamics







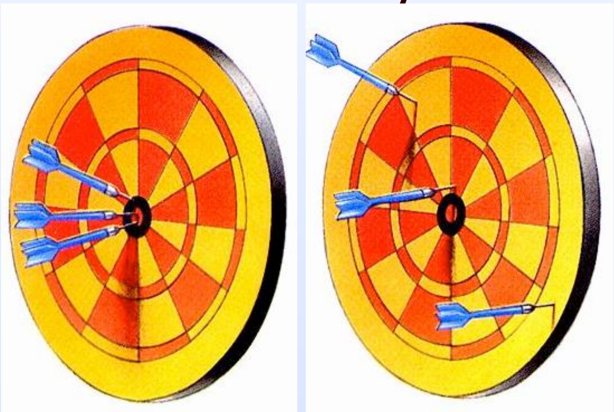
Graphics

Why don’t we just take existing models from CFD for Computer Graphics applications?

Graphics

Why don’t we just take existing models from CFD for Computer Graphics applications?

Accuracy



Performance

Graphics

Why don’t we just take existing models from CFD for Computer Graphics applications?

Visual result

Control

Graphics

Why don’t we just take existing models from CFD for Computer Graphics applications?

“It’s a common misconception that visual effects are about simulating reality. They’re not. Reality is boring. Visual effects are about simulating something dramatic,”

- Jonathan Cohen, Rhythm&Hues

Films

The Abyss, 89

Fluid effects were animated manually, e.g. digitally painting

Films

Antz, 98

First film with fluid simulation

-> very time consuming

Films

Pirates of the Caribbean 3, 07

Control was *key*: manipulate physics to meet creative goals

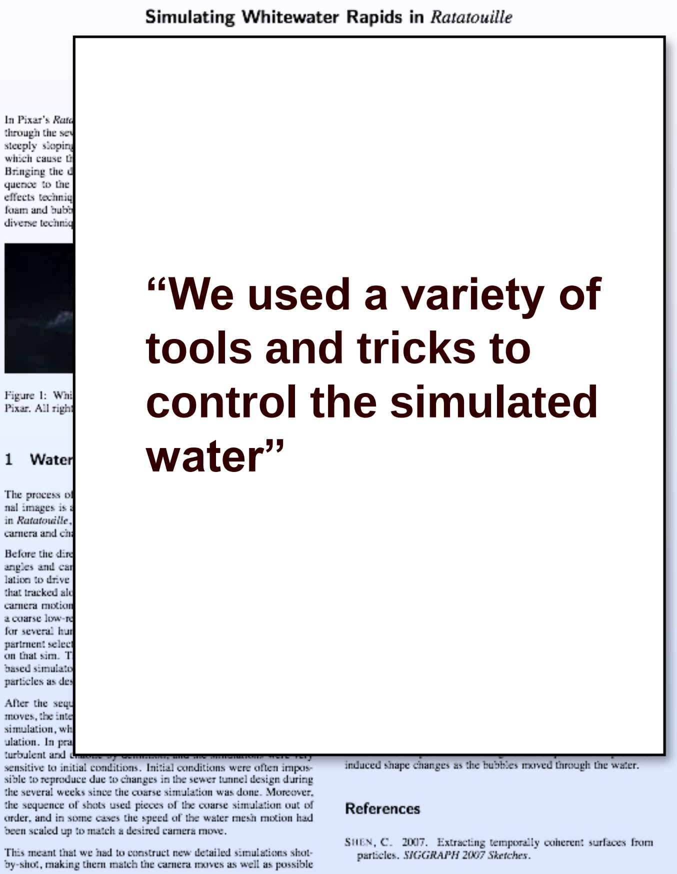
Films

The Day After Tomorrow, 04

Foam, mist, more animator control

Controlling fluid sims is an active area of research

Fluids in production



Simulating Whitewater Rapids in *Ratatouille*

sensitive to initial conditions. Initial conditions were often impossible to reproduce due to changes in the sewer tunnel design during the several weeks since the coarse simulation was done. Moreover, the sequence of shots used pieces of the coarse simulation out of order, and in some cases the speed of the water mesh motion had been scaled up to match a desired camera move.

This meant that we had to construct new detailed simulations shot-by-shot, making them match the camera moves as well as possible

induced shape changes as the bubbles moved through the water.

References

SHEN, C. 2007. Extracting temporally coherent surfaces from particles. *SIGGRAPH 2007 Sketches*.

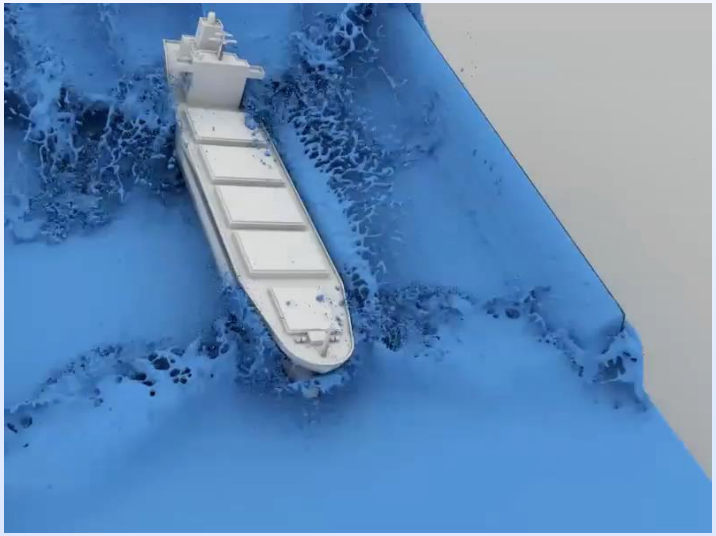
“We used a variety of tools and tricks to control the simulated water”

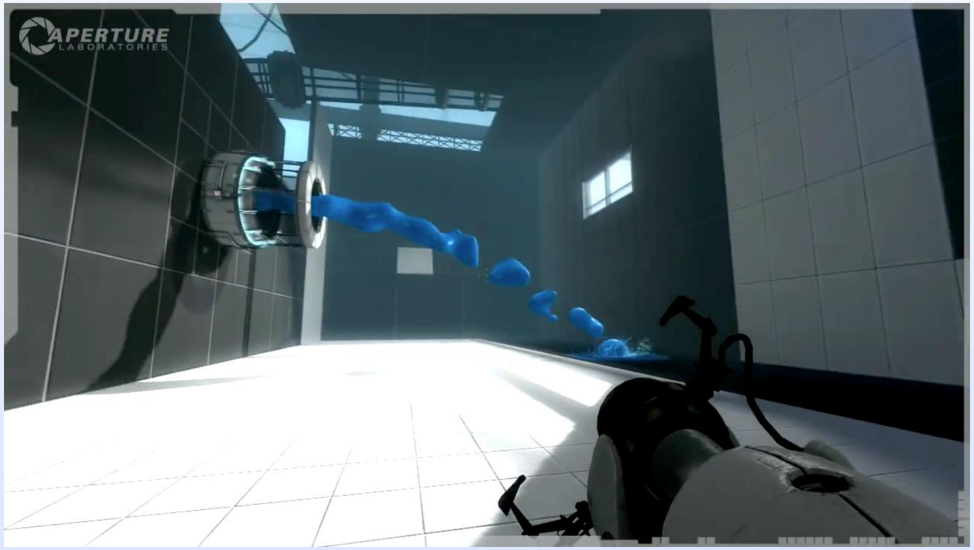
Fluids in production

Scanline VFX: 2012

Fluids in games

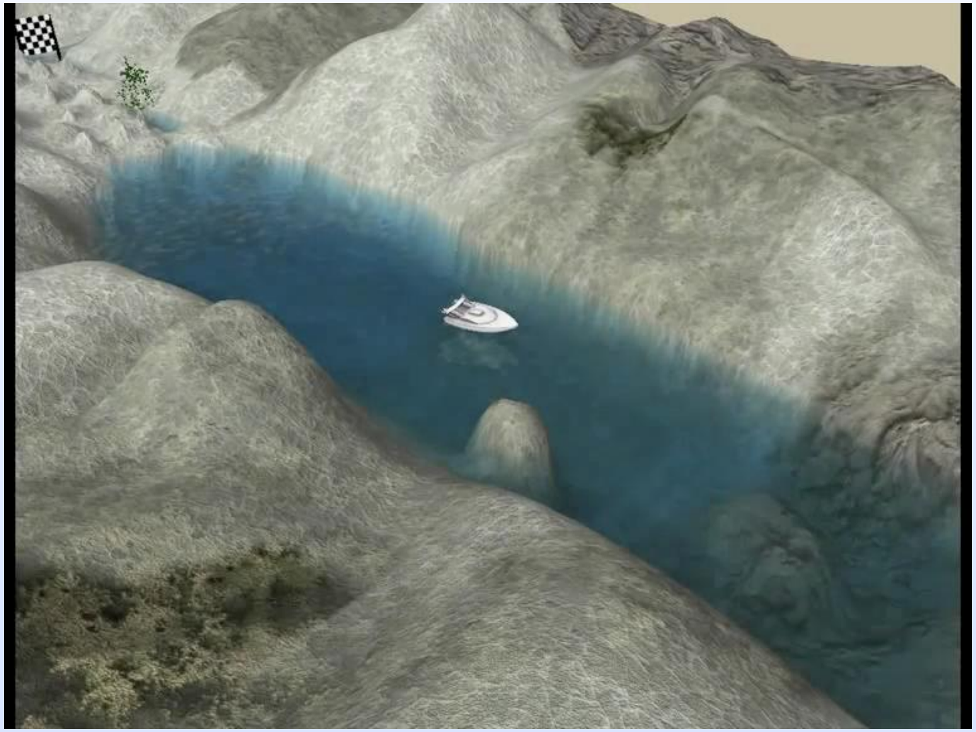
- Real-time, stability
  - Dimension reduction





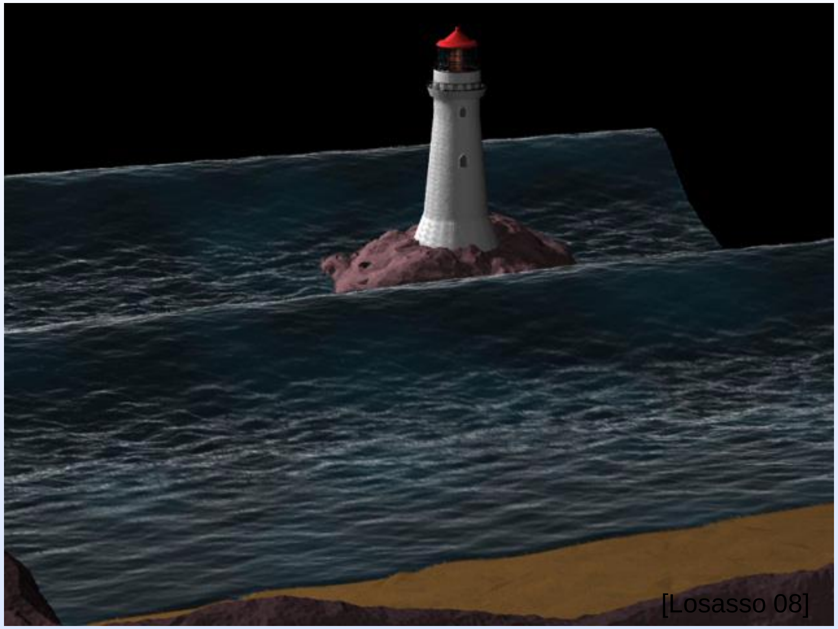
Fluids in games

- Real-time, stability
  - Dimension reduction



Films vs Video Games

Offline Stanford Lighthouse



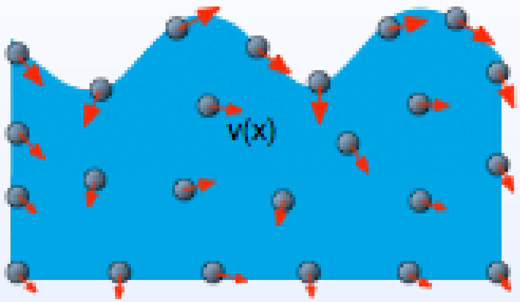
[Losasso 08]

Real-time NVIDIA Lighthouse

[Chentanez 11]

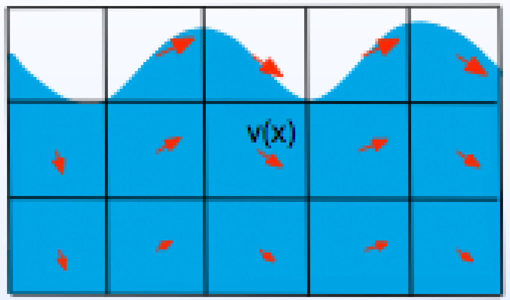
Spatial Discretization

Langrangian viewpoint



- Particles represent the fluid, carry quantities
- Fluid motion by moving particles

Eulerian viewpoint



- Fixed spatial locations
- Measure quantities as it flows past

You are in the balloon floating along with the wind, measuring the pressure, temperature, humidity etc of the air that's flowing alongside you

doing a weather report

You are stuck on the ground, measuring the pressure, temperature, humidity etc of the air that's flowing past

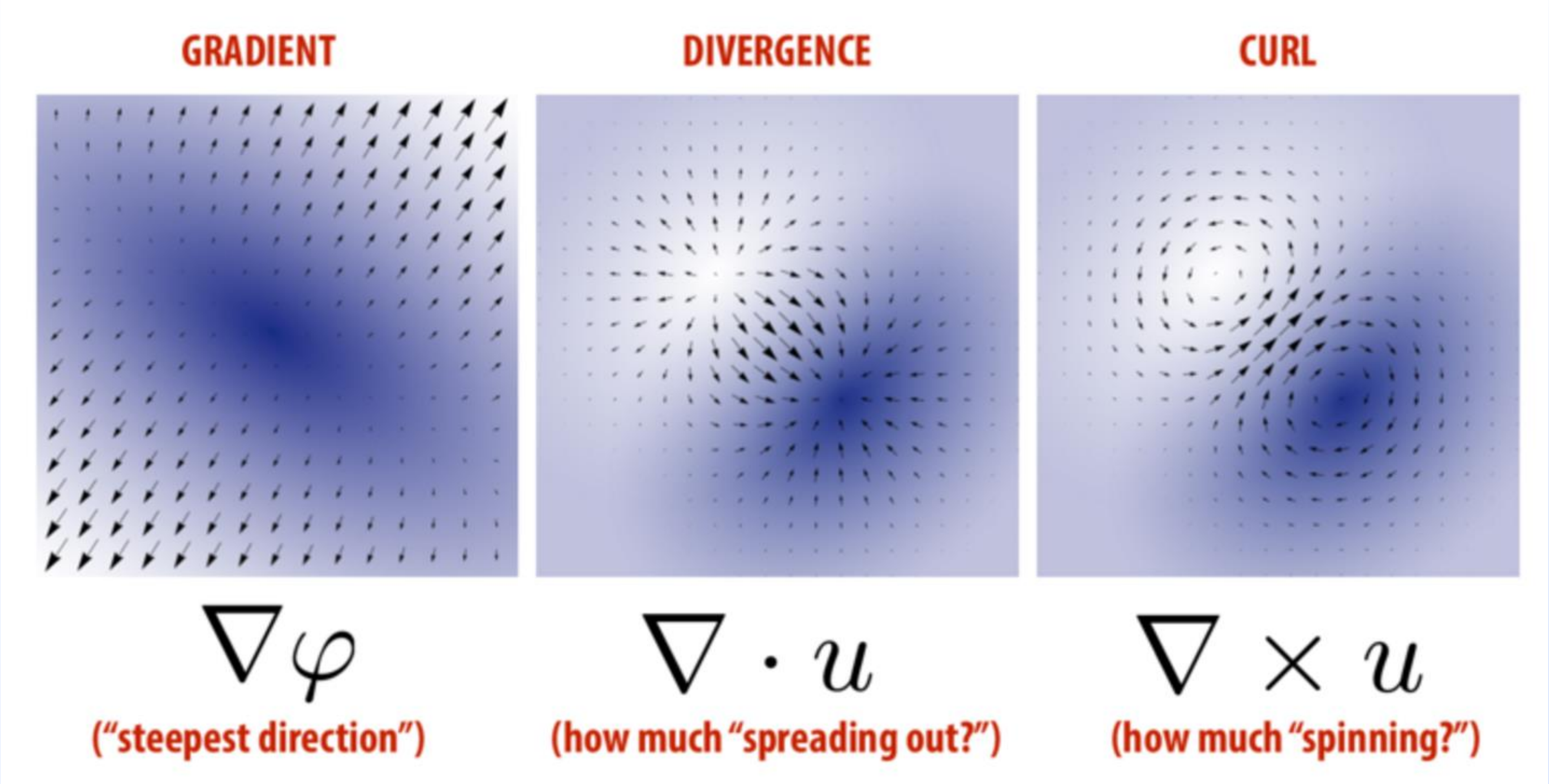
Notation Reminder

- Nabla operator:
  - Gradient (scalar -> vector): $\nabla u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$
  - Divergence (vector -> scalar): $\nabla \cdot \mathbf{u} = u_x + u_y + u_z$

- Laplace operator (scalar->scalar): $\Delta u = \nabla^2 u = \nabla \cdot \nabla u$

$$\Delta u = u_{xx} + u_{yy} + u_{zz}$$

Visual Vector Calculus



Navier-Stokes Equations

- 2 differential equations describing velocity field over time
- Conservation of momentum:

$$\underbrace{\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}}_{\text{advection}} = \underbrace{-\frac{1}{\rho}\nabla p}_{\text{pressure}} + \underbrace{\nu \triangle \mathbf{u}}_{\text{viscosity}} + \mathbf{g}$$

F=ma

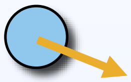
- Conservation of mass:

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

u = velocity
p = pressure
ρ = density
g = gravity
ν = viscosity

Derivation of Momentum Equation

◆ Imagine single blob of fluid *p*:



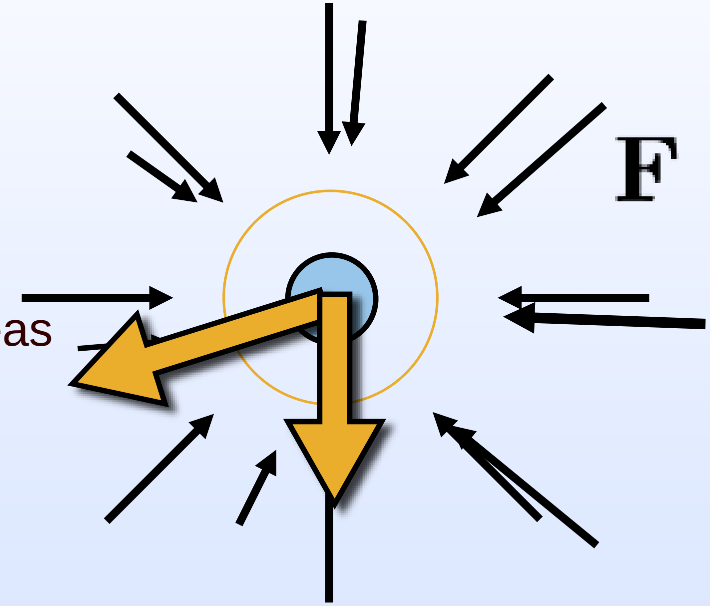
m = mass
V = volume
u = velocity

- Start with Newton's second law: $\mathbf{F} = m\mathbf{a}$

- Rewrite as: $\mathbf{F} = m\frac{D\mathbf{u}}{Dt}$

What Forces Act on p?

- Gravity: $m\mathbf{g}$
- Pressure:
  - Consider nearby fluid
  - Force: negative gradient -> towards largest pressure decrease / low pressure areas
- Compute with:

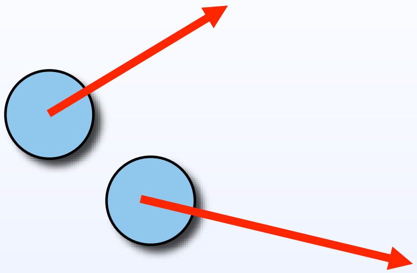


$$\mathbf{F} = 0$$

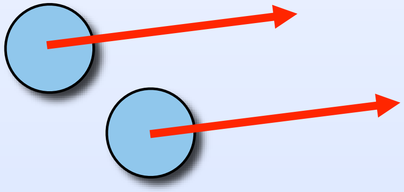
$$-\nabla p \; V$$

What Forces Act on p? (2)

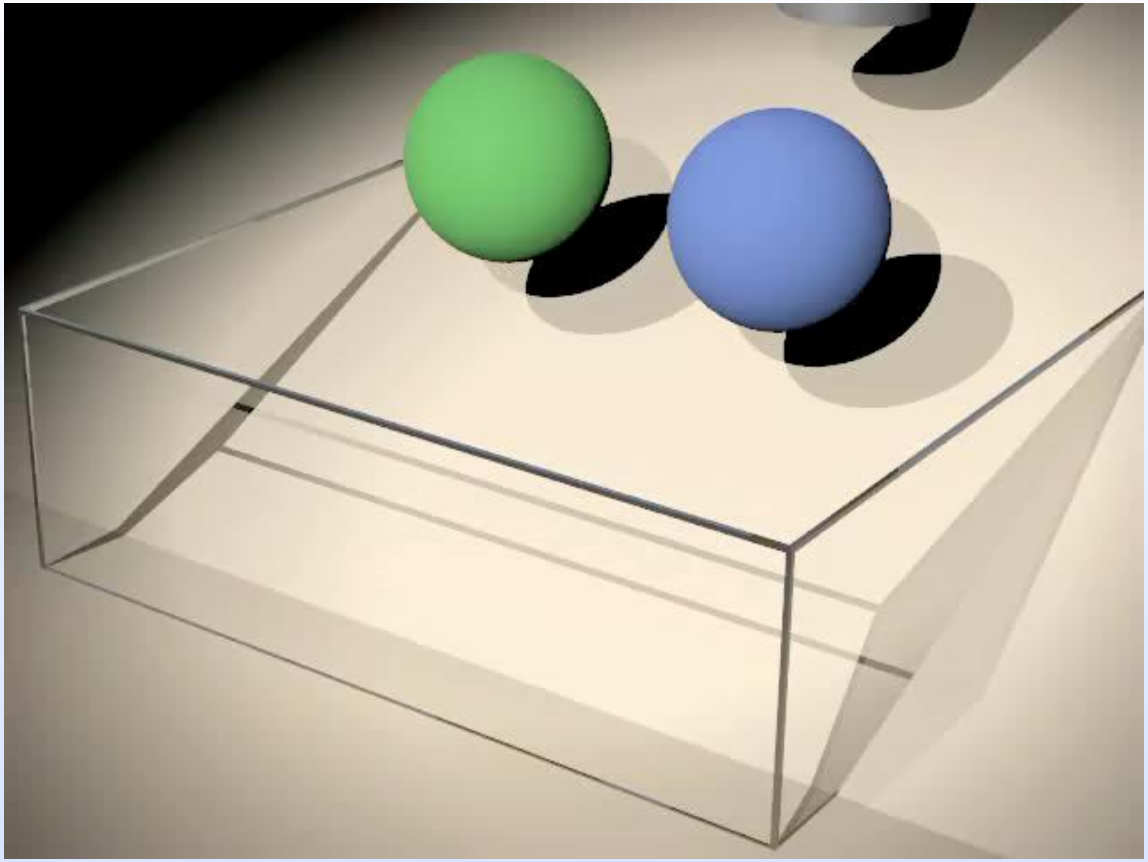
- Viscosity
  - "Internal friction"
  - More accurately: diffusion of (relative) velocities



- Diffusion
  - Laplacian
  - Strength: dynamic viscosity coefficient
  - Compute with: $V\mu\nabla\cdot\nabla\,\mathbf{u} = V\mu\Delta\mathbf{u}$



Viscous Material

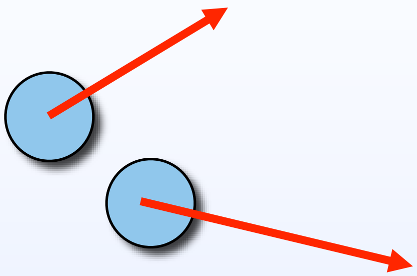


Losasso et al., Multiple Interacting Liquids, Siggraph 06

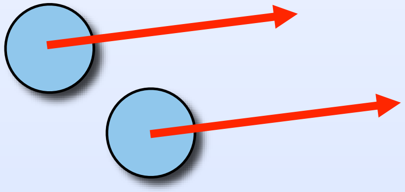
What Forces Act on p? (2)

◆ Viscosity

- "Internal friction"
- More accurately: diffusion of velocities



• Diffusion



- Laplacian
- Strength: dynamic viscosity coefficient
- Compute with: $V\mu\nabla\cdot\nabla\,\mathbf{u} = V\mu\Delta\mathbf{u}$

Note: Particle systems always need viscosity to stabilize system

Total Forces on *p*

(1) $$\mathbf{F} = m\mathbf{a}$$

Newton's 2nd law

(2) $$\mathbf{F} = m\frac{D\mathbf{u}}{Dt}$$

(3) $$m\mathbf{g} - V\nabla p + V\mu\Delta\mathbf{u} = m\frac{D\mathbf{u}}{Dt}$$

Forces so far.

(4) $$\mathbf{g} - \frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\Delta\mathbf{u} = \frac{D\mathbf{u}}{Dt}$$

Divide by mass

(5) $$\boxed{\frac{D\mathbf{u}}{Dt}} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u} + \mathbf{g}$$

Rearrange, using kinematic viscosity

Material Derivative

- We have fluid moving in a velocity field u, possessing some scalar quantity q(t,x)

◆ Change of a quantity q(t,x) during motion:

total derivative of q with respect to time

$$\frac{d}{dt}q(t,\mathbf{x}) = \frac{\partial q}{\partial t} + \frac{\partial q}{\partial \mathbf{x}} \cdot \frac{d\mathbf{x}}{dt}$$

expanded through multivariate chain rule

$$= \frac{\partial q}{\partial t} + \nabla q \cdot \mathbf{u}$$

(1) $\frac{\partial q}{\partial t}$, (2) $\nabla q \cdot \mathbf{u}$

use grad, velocity

$$:= \frac{Dq}{Dt}$$

material derivative

how fast q is changing at fixed point in space x
(temperature is decreasing because of cooling-down)

correcting for how much of that change is due just to differences in the fluid flowing past (inflow / outflow at x)
(temperature is changing because hot air is being replaced by cold air)

Navier-Stokes Equations

how fast q is changing at fixed point in space x
(temperature is decreasing because of cooling-down)

inflow / outflow at x
(temperature is changing because hot air is being replaced by cold air)

$$\underbrace{\frac{\partial q}{\partial t}}_{1} + \underbrace{\nabla q \cdot \mathbf{u}}_{2}$$

Momentum equation:

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + \mathbf{g}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + \mathbf{g}$$

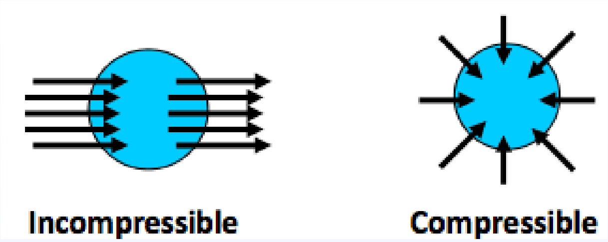
Note: This term not needed for particle-based fluids

Mass conservation equation:

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

Mass Conservation

- Incompressibility



◆ Very small volume change in water:
10m: ≈ 200kPa → 0.0045%
4000m: ≈ 40‘000kPa → 1.8%

- Small effect on how fluids move at macroscopic level
  → Water is *treated as incompressible*

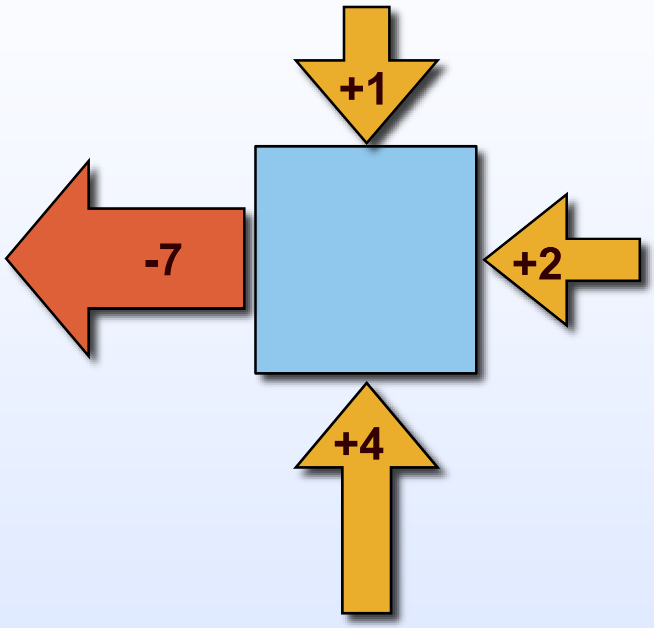
Mass Conservation

- Divergence-free:

"what goes in somewhere,
must go out somewhere else"

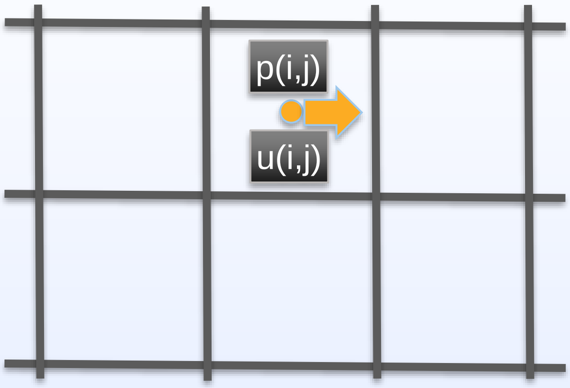
$$\nabla \cdot \mathbf{u} = 0$$

Per unit volume:



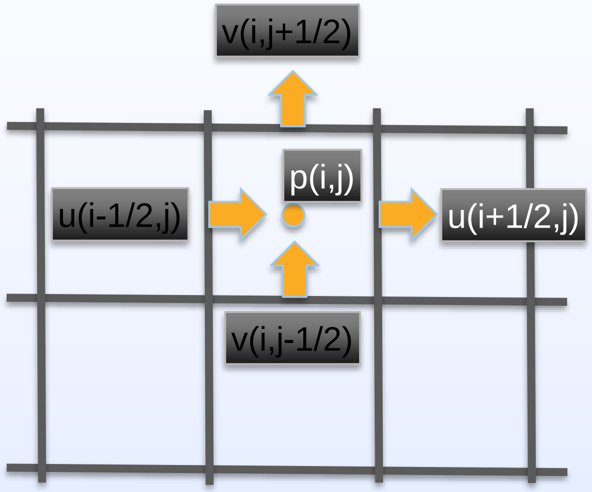
Spatial Discretization: grid-based methods

- Grid discretization
  - Cubical cells
  - Pressure and velocity defined at center



Spatial Discretization: grid-based methods

- Staggered grid (MAC grid)
  - Cubical cells
  - Pressure defined at center
  - Velocity components defined on faces of cells



\+ Staggering more stable, second order accurate

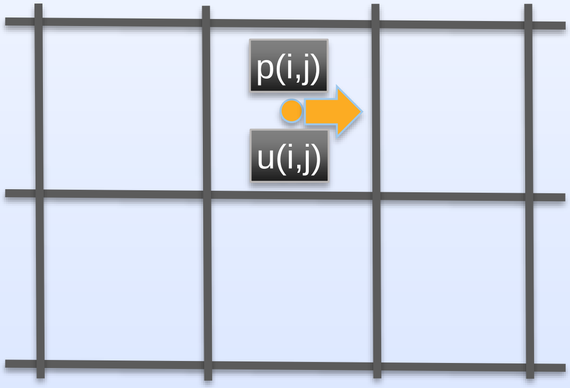
\- Evaluate velocity at any point through interpolation

Spatial Discretization: grid-based methods

- How do we evaluate gradient, divergence, laplacians, etc on a grid?

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + \mathbf{g}$$

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



- Finite difference method (FDM)

Example: Grid Laplacian

$$\Delta u(x,y) = \frac{\partial^2}{\partial x^2} u(x,y) + \frac{\partial^2}{\partial y^2} u(x,y)$$

Discretizing on a grid with cell size *h*:

$$\Delta u_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2}$$

Computation Order

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + \mathbf{g}$$

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

Advection → Body Force → Diffusion → Pressure Solve

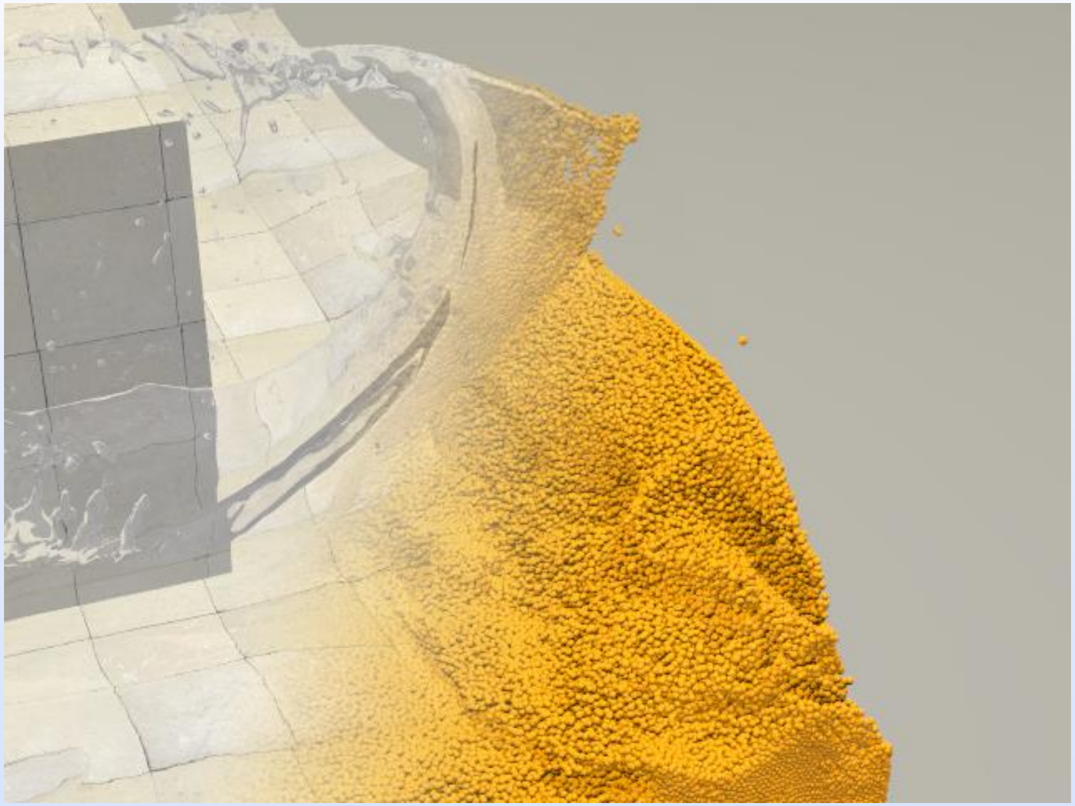
Requires systems of linear equations to be solved at every time step

Further Reading

- Bridson et al., Fluid Simulation Course, Siggraph Course Notes 2006/2007 http://www.cs.ubc.ca/~rbridson/fluidsimulation/
- Cline et al., Fluid Flow for the Rest of Us: Tutorial of the Marker and Cell Method in Computer Graphics http://people.sc.fsu.edu/~jburkardt/pdf/fluid_flow_for_the_rest_of_us.pdf

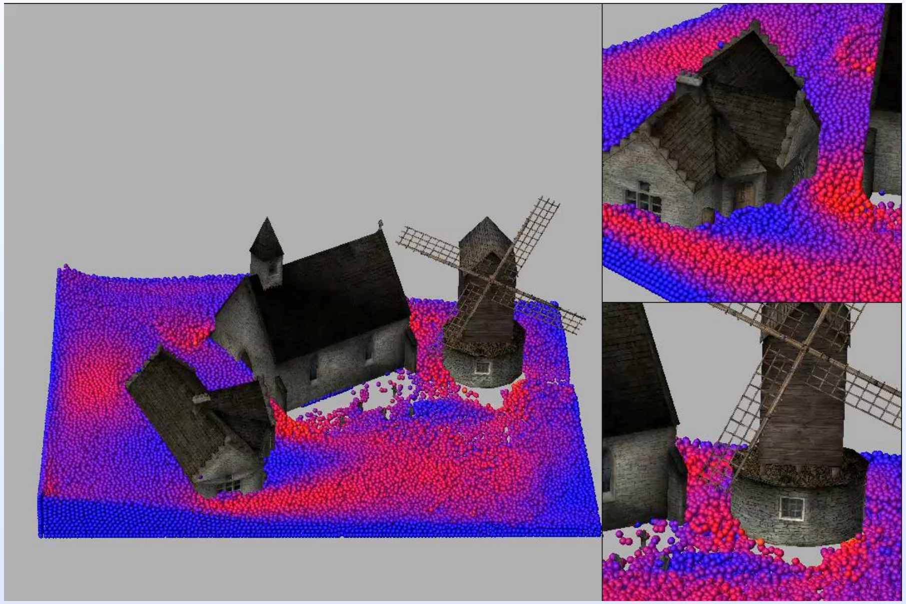
Spatial Discretization: particle-based methods

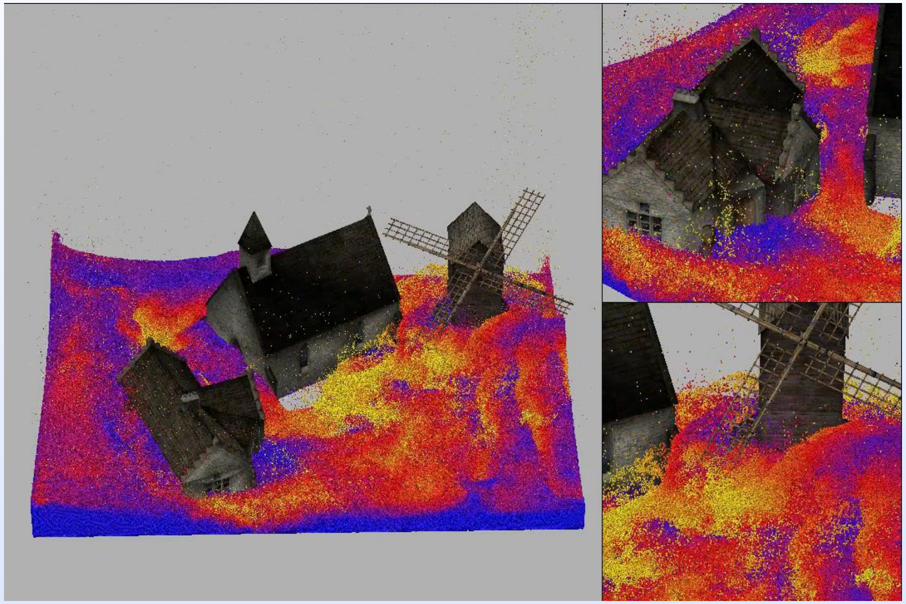
Smoothed Particle Hydrodynamics (SPH)



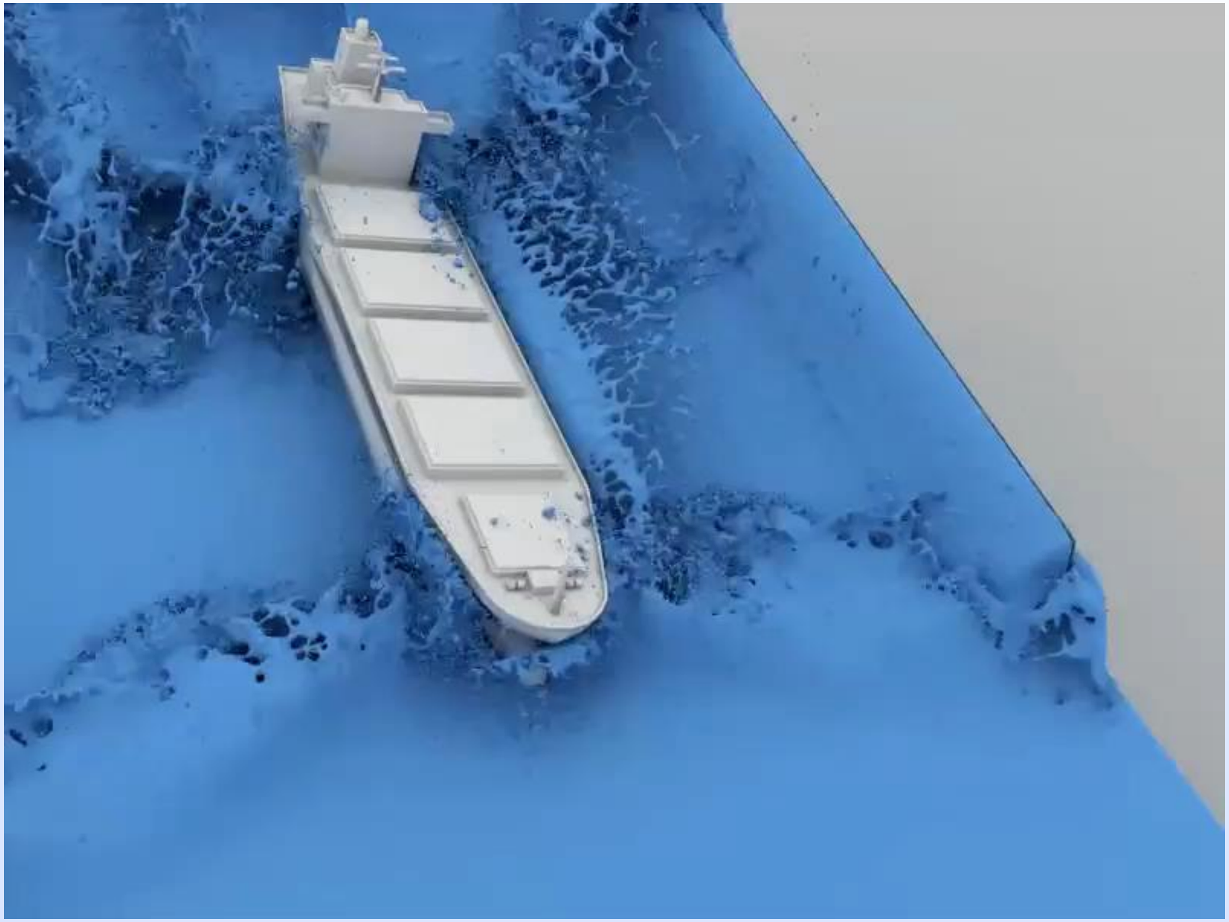
Particle Fluids

Resolution - 40K vs 4M



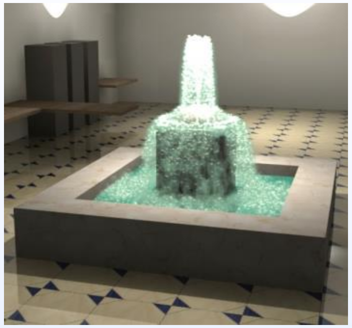


30M - Opaque Surface

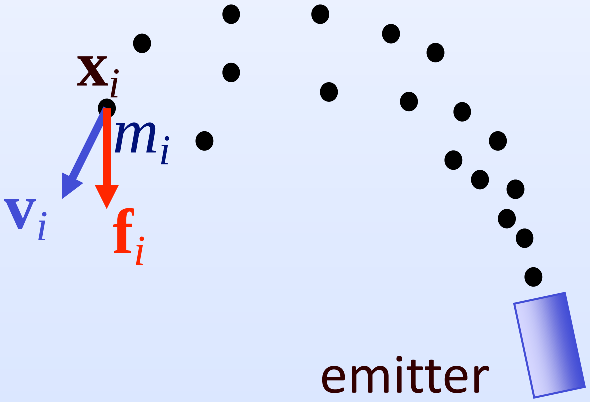


Simple Particle System

- Without particle-particle interaction
  - Simple and fast
  - Fuzzy objects like fire, clouds, smoke



- Particles generated by emitters, deleted when lifetime is exceeded
- Forces -> velocities -> positions



-> Smoothed Particle Hydrodynamics (SPH): With particle-particle interaction based on NS eqs

SPH Simulations in Films

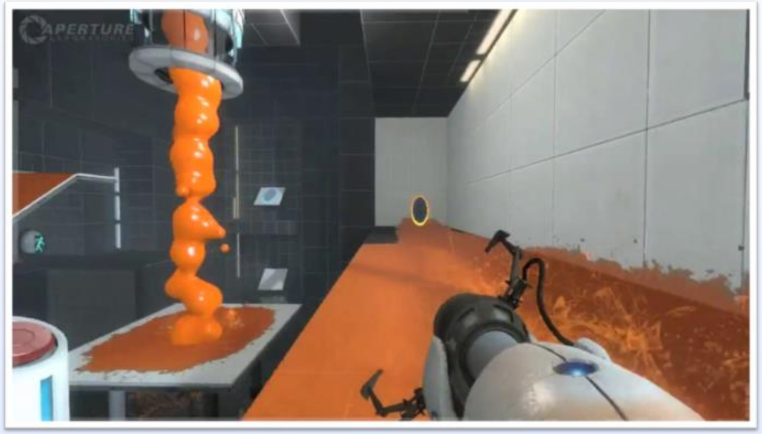
Lord of the Rings 3

Superman Returns

SPH Simulations in Games

Alice: Madness Returns

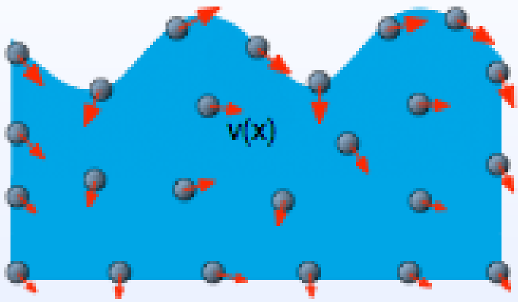
Portal 2



Epic Mickey

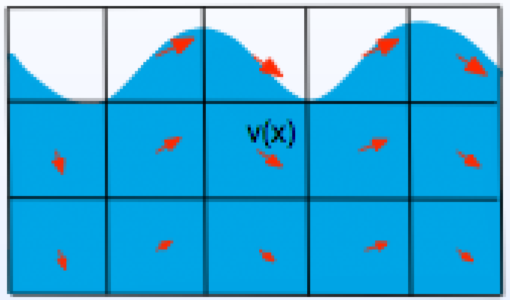
Spatial Discretization

Langrangian viewpoint



- Particles represent the fluid, carry quantities
- Fluid motion by moving particles

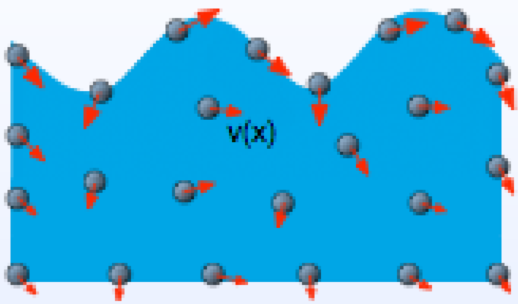
Eulerian viewpoint



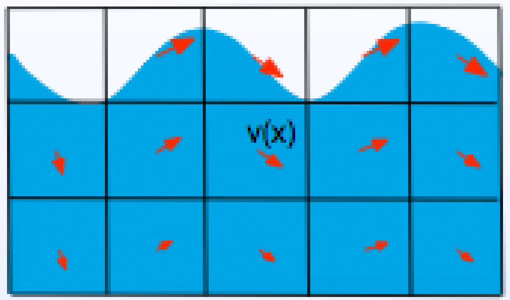
- Fixed spatial locations
- Measure quantities as it flows past

Spatial Discretization

Langrangian viewpoint



Eulerian viewpoint



Pros and Cons

Langrangian		Eulerian
−	**Neighbor search**	+
+	**Advection**	−
+	**Splashes, droplets**	−
−	**Smooth surfaces**	+
−	**Parameters**	+

Particles - Observations

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + \mathbf{g}$$

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

Advection → Body Force → Diffusion → Pressure Solve

- Advection: No advection needed
- Diffusion: Viscosity always needed
- Pressure Solve: One linear equation / particle

Neighborhood size 30-40, dynamically changing!

Many SPH solvers do not explicitly enforce incompressibility

NS in the Lagrangian Viewpoint

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u} + \mathrm{g}$$

// a=F/m

// Multiply by density

$$\rho\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \mu\Delta\mathbf{u} + \rho\mathrm{g}$$

// rho*a=F/V

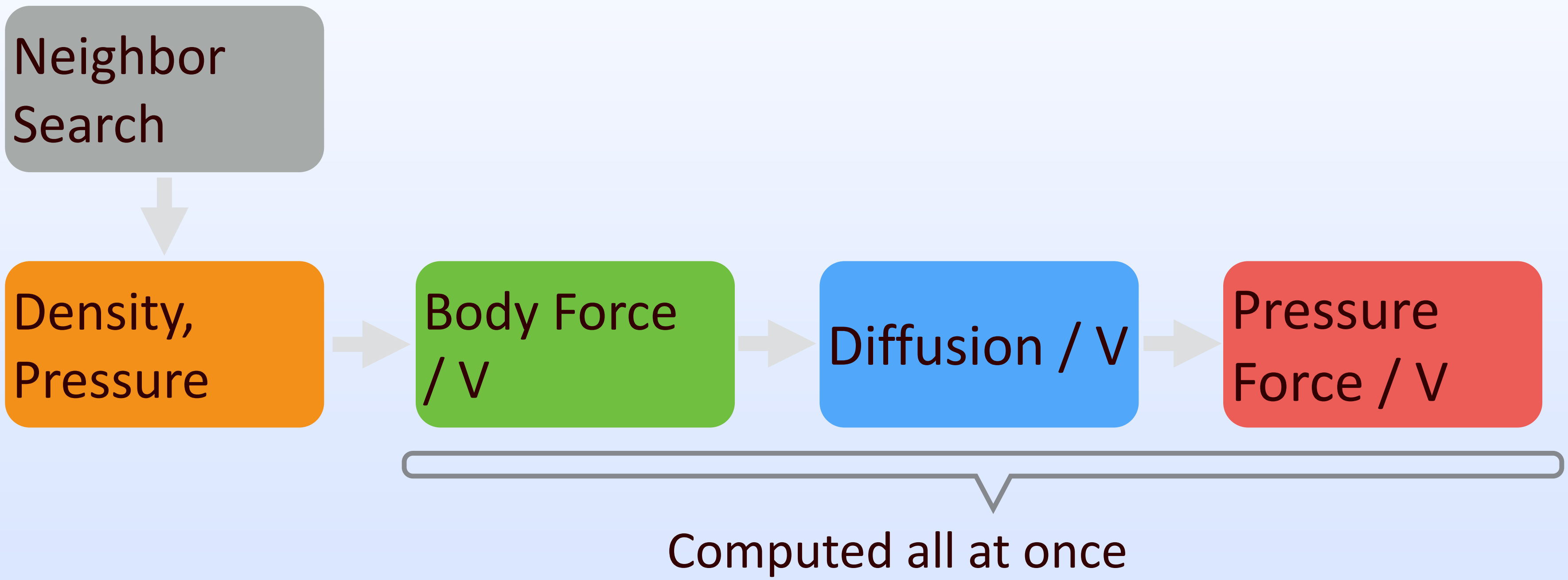
Force densities f (F/V)

SPH Literature in Graphics:

- [Müller03] *Particle-Based Fluid Simulation for Interactive Applications*
- [Bridson / Müller07] Fluid Simulation Course Notes, Siggraph 2007
- [Ihmsen13] State of the art report, *SPH Fluids in Computer Graphics*

Computation Order

$$\rho\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \mu\nabla^2\mathbf{u} + \rho \mathrm{g}$$



Required reading for next class

- Smoothed Particle Hydrodynamics

  “Particle-Based Fluid Simulation for Interactive Applications”, Muller et al., 2003

- Position-based Fluids

  “Position Based Fluids”, Macklin & Muller, 2014

Reminder

- Project Ideas
  - Send me brief description by next Tuesday (1 para)
    - Team info, topic, etc.