Fluid Techniques

Jim Van Verth
Senior Engine Programmer, Insomniac Games

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 5-9, 2012
EXPO DATES: MARCH 7-9
2012

Introductory Bits

- General summary with some details
- Not a fluids expert
- Theory and examples

What is a Fluid?

- Deformable
- Flowing
- Examples
  - Smoke
  - Fire
  - Water

What is a Fluid?

What is a Fluid?

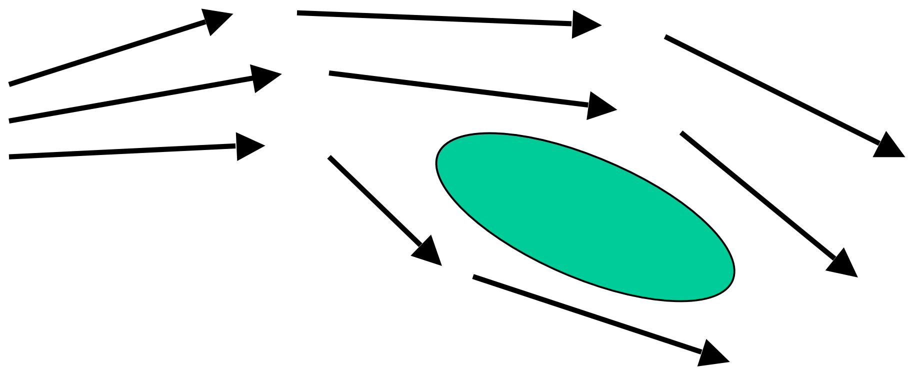
What is a Fluid?

Fluid Concepts

- Fluids have variable density
  - (Density field)

Fluid Concepts

- Fluids “flow”
  - (Vector field)

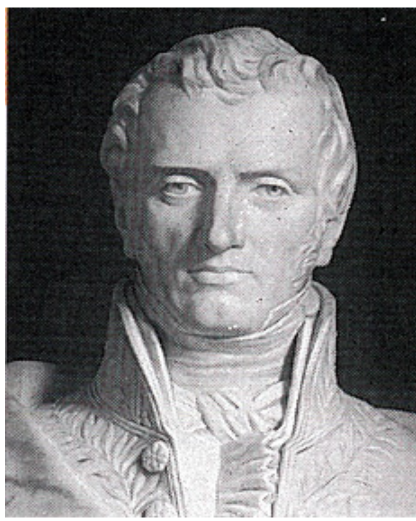


Fluid Concepts

- Need way to represent
  - Density ($\mathbf{x}$)
  - Velocity ($\mathbf{u}$)
  - Sometimes temperature

Fluid Concepts

- Our heroes:

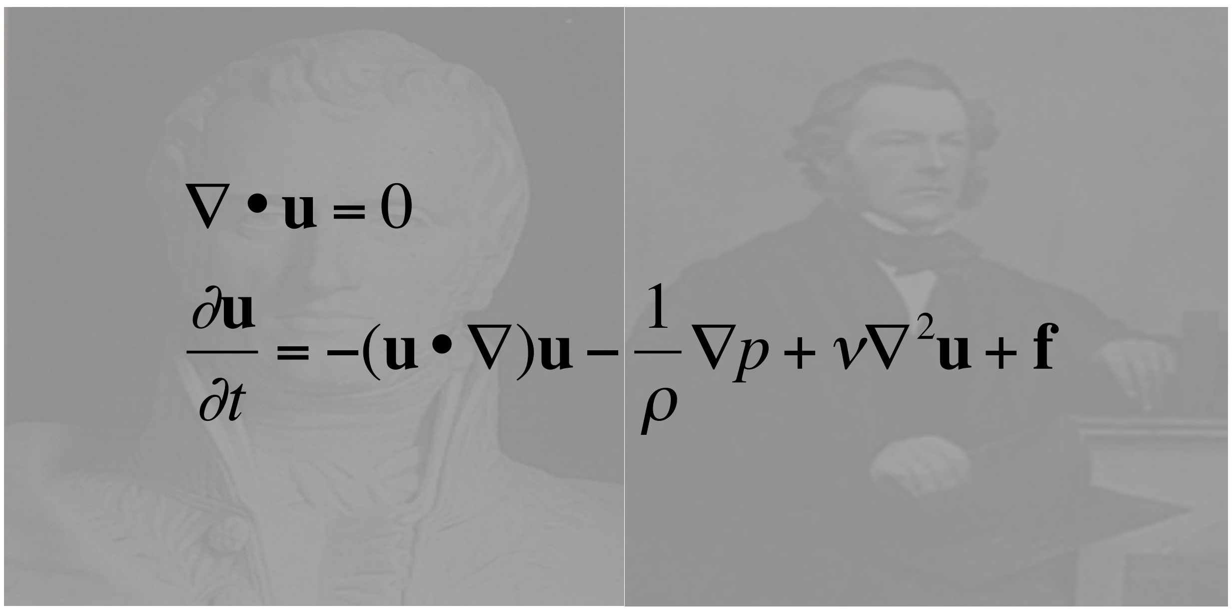


Navier

Stokes

Fluid Concepts

- Their creation:

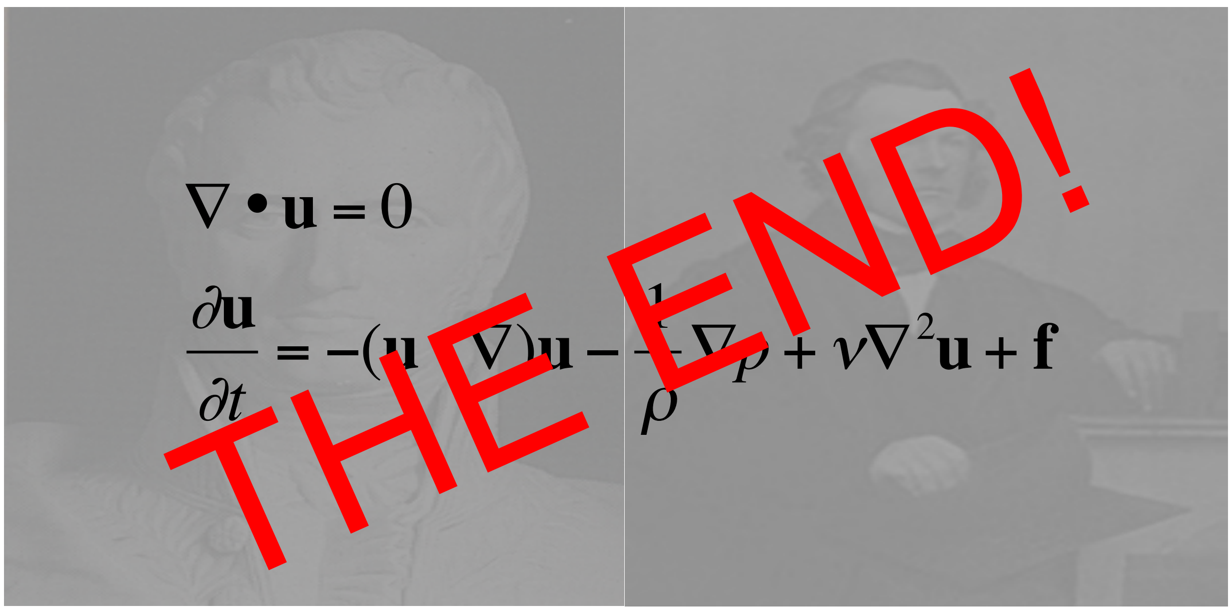


$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

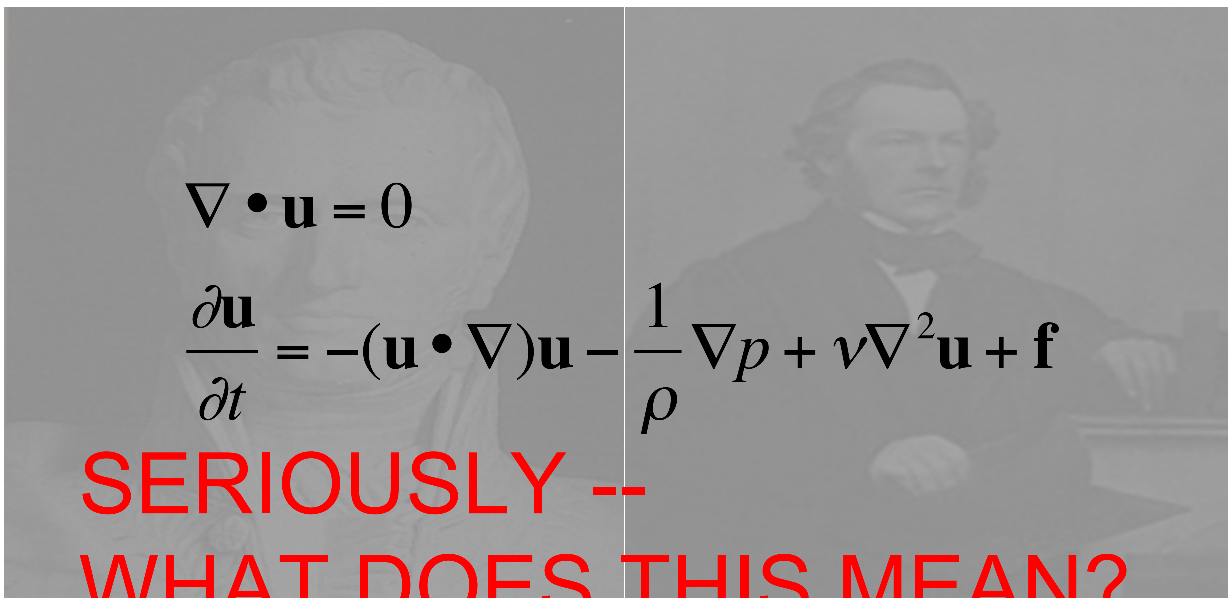
Fluid Concepts

- Their creation:



Fluid Concepts

- Their creation:



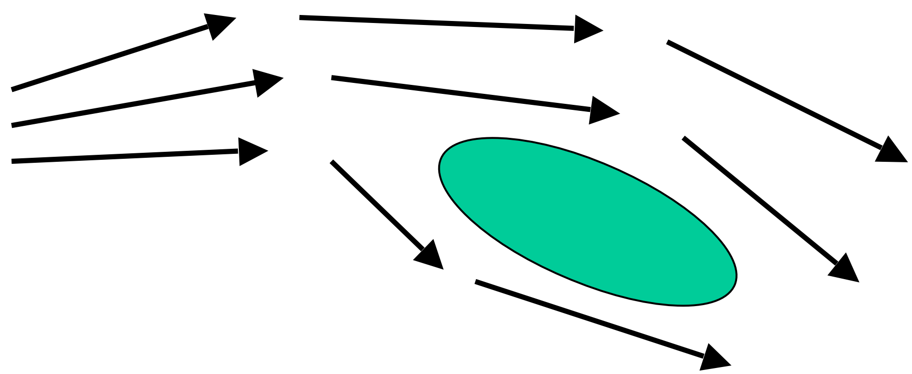
$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{f}$$

SERIOUSLY --
WHAT DOES THIS MEAN?

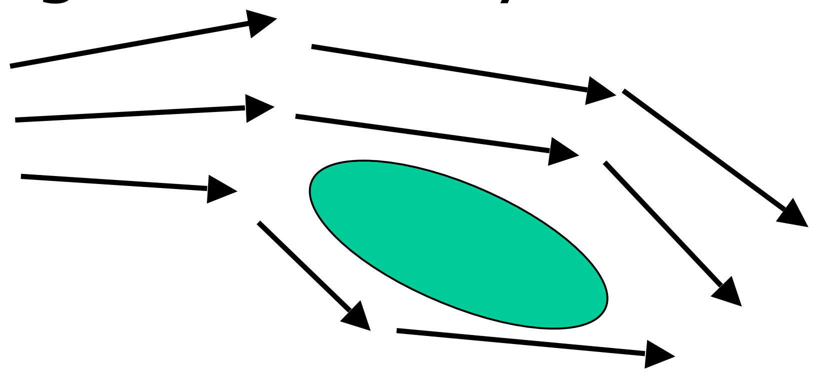
Fluid Concepts

- Want change in velocity field



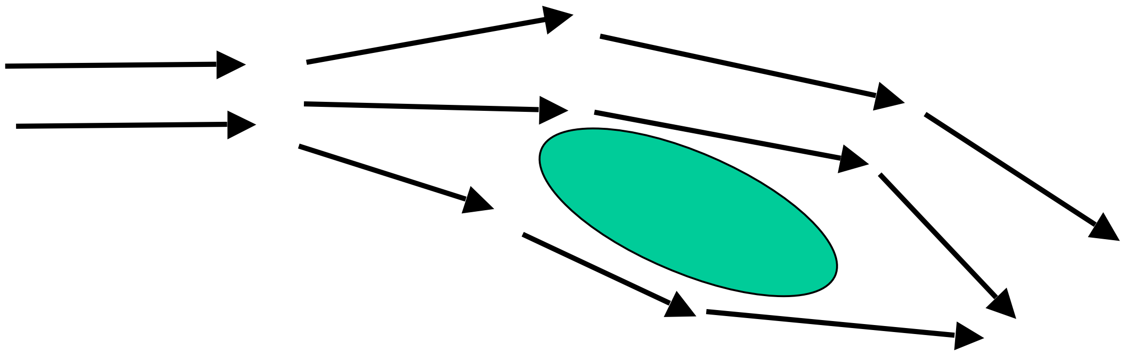
Fluid Concepts

- Want change in velocity field



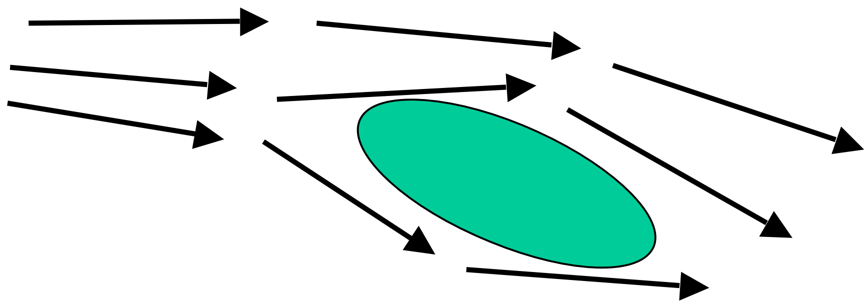
Fluid Concepts

- Want change in velocity field



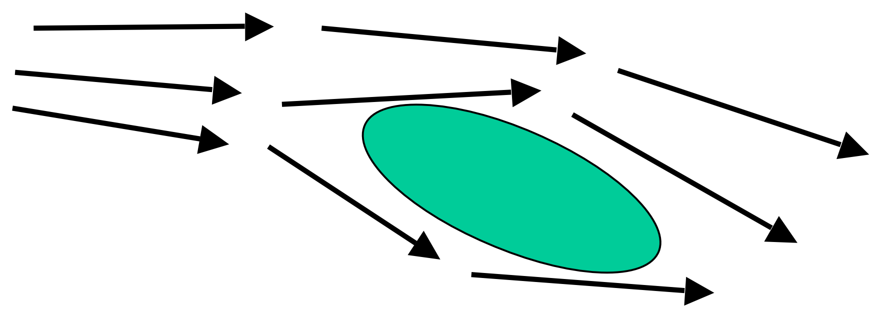
Fluid Concepts

- Want change in velocity field



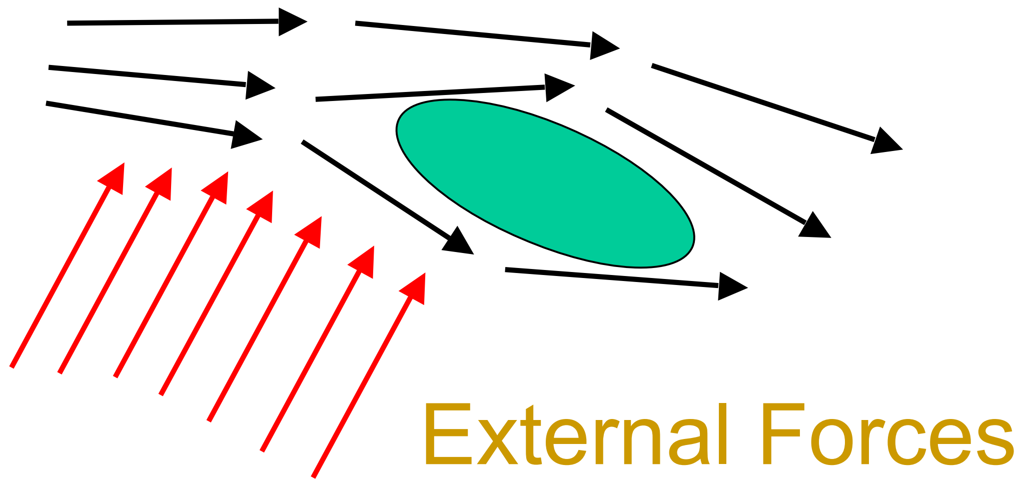
Fluid Concepts

- What affects it?



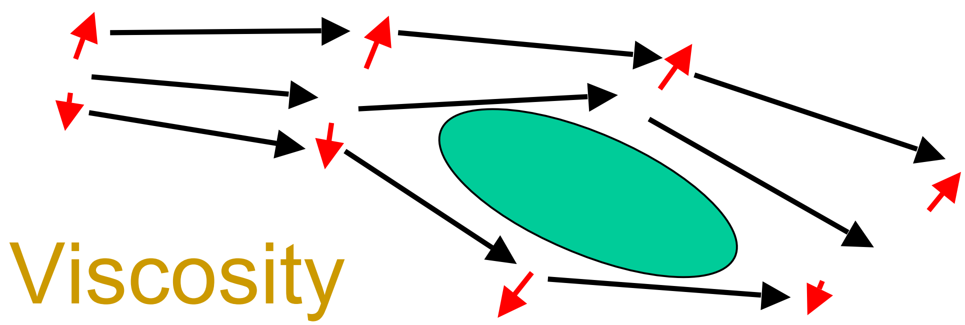
Fluid Concepts

- What affects it?



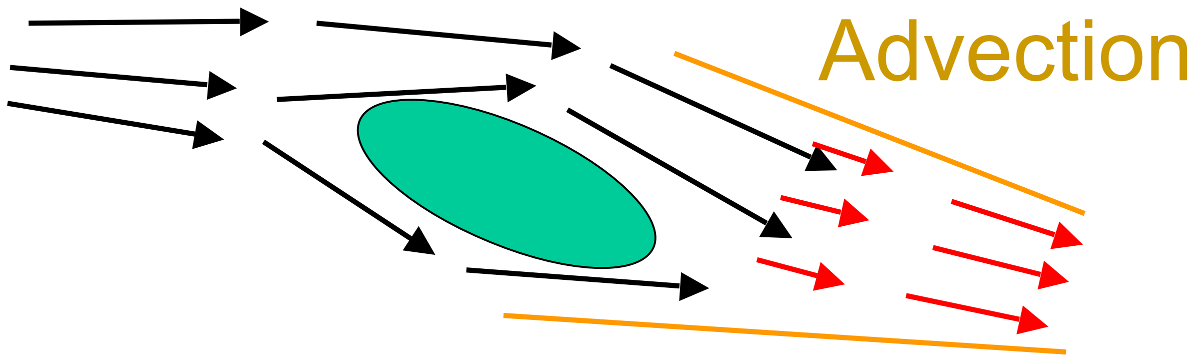
Fluid Concepts

- What affects it?



Fluid Concepts

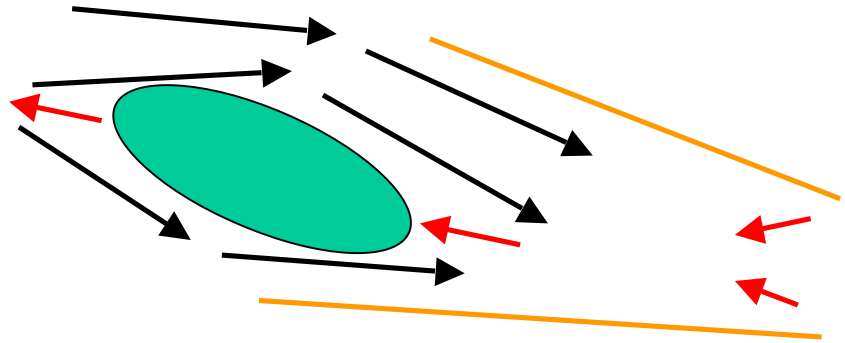
- What affects it?



Fluid Concepts

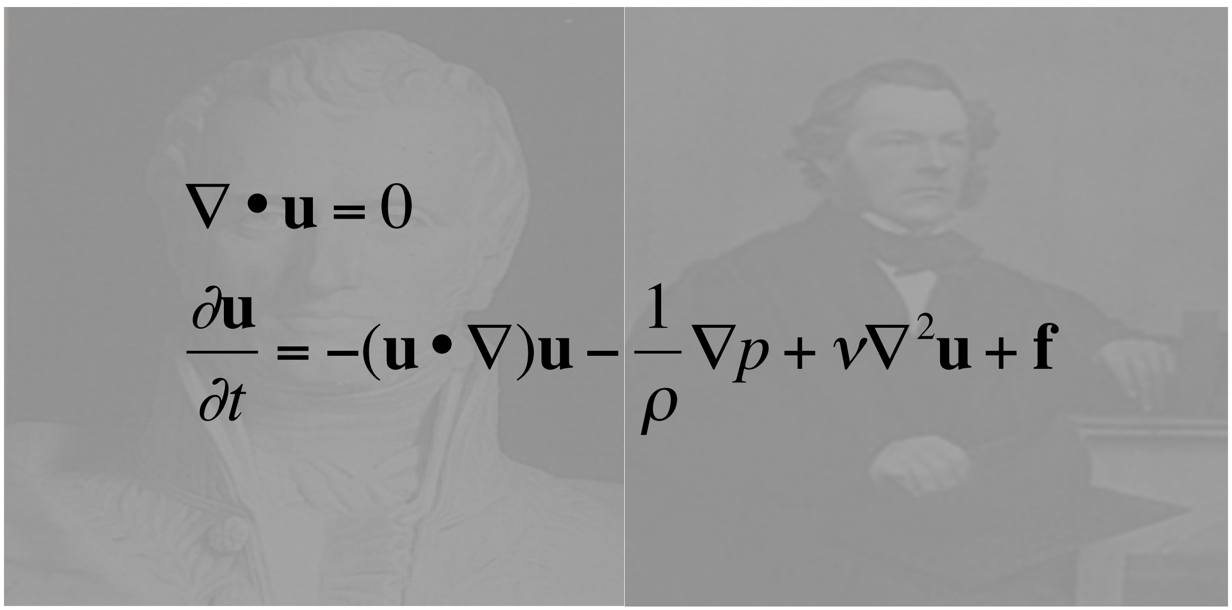
- What affects it?

Pressure



Fluid Concepts

- Back to Navier-Stokes

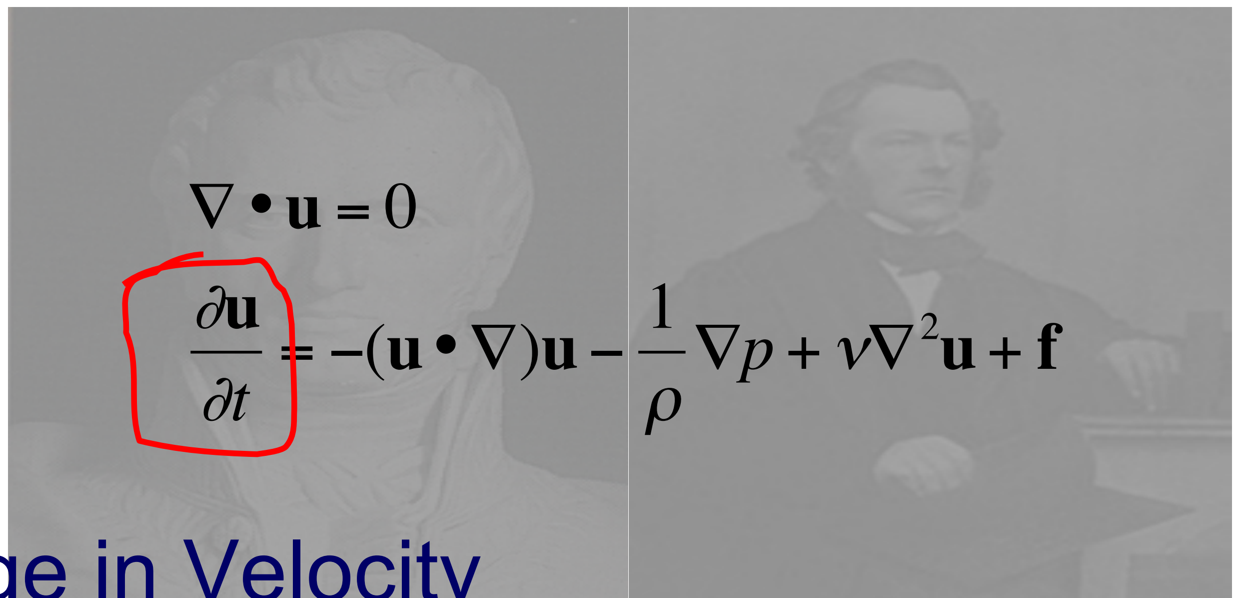


$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Fluid Concepts

- Back to Navier-Stokes



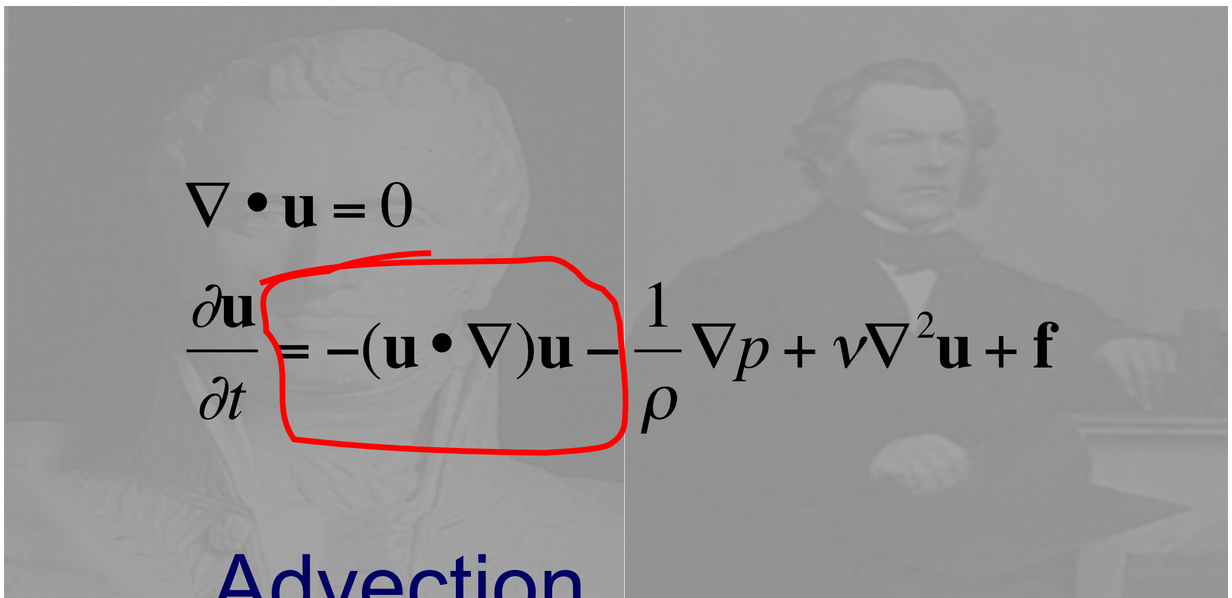
$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Change in Velocity

Fluid Concepts

- Back to Navier-Stokes



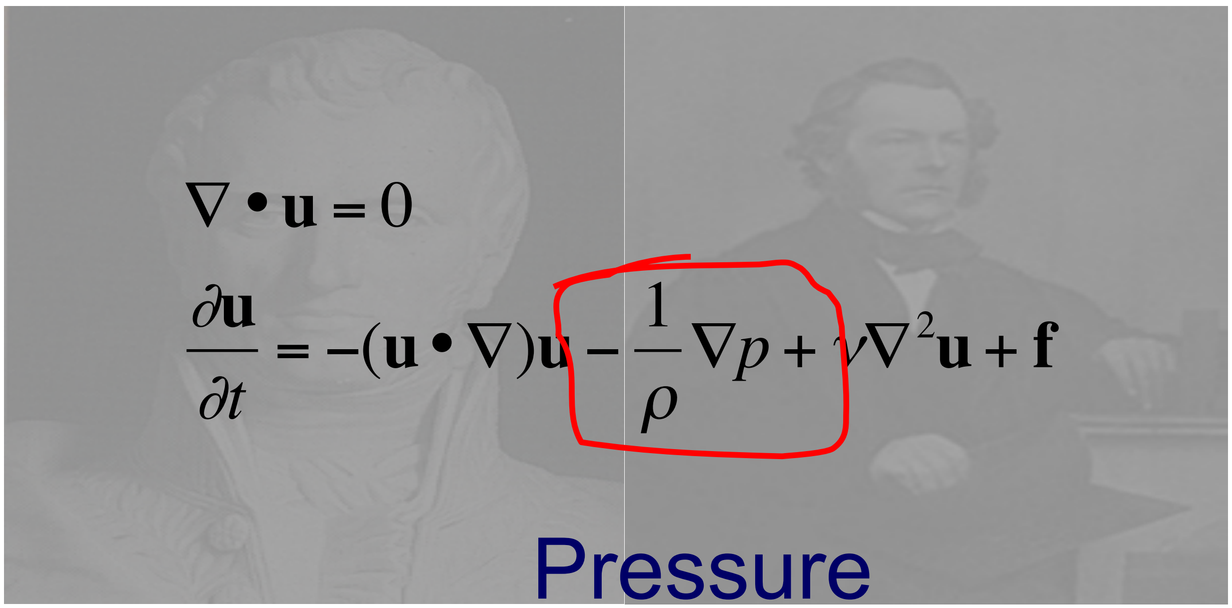
$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Advection

Fluid Concepts

- Back to Navier-Stokes

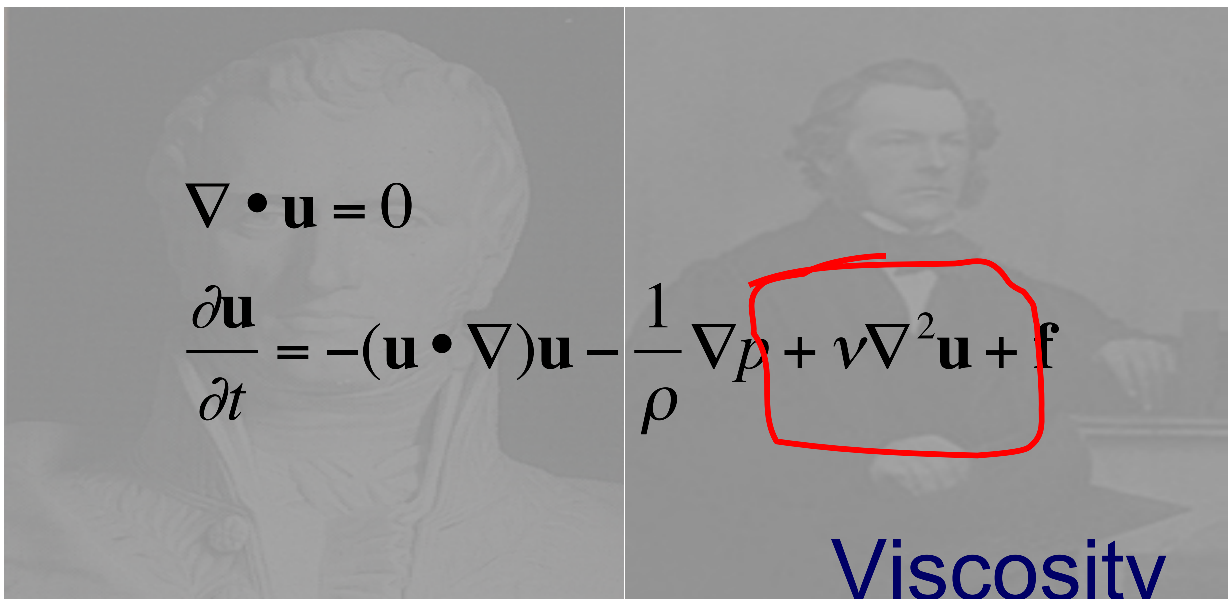


$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

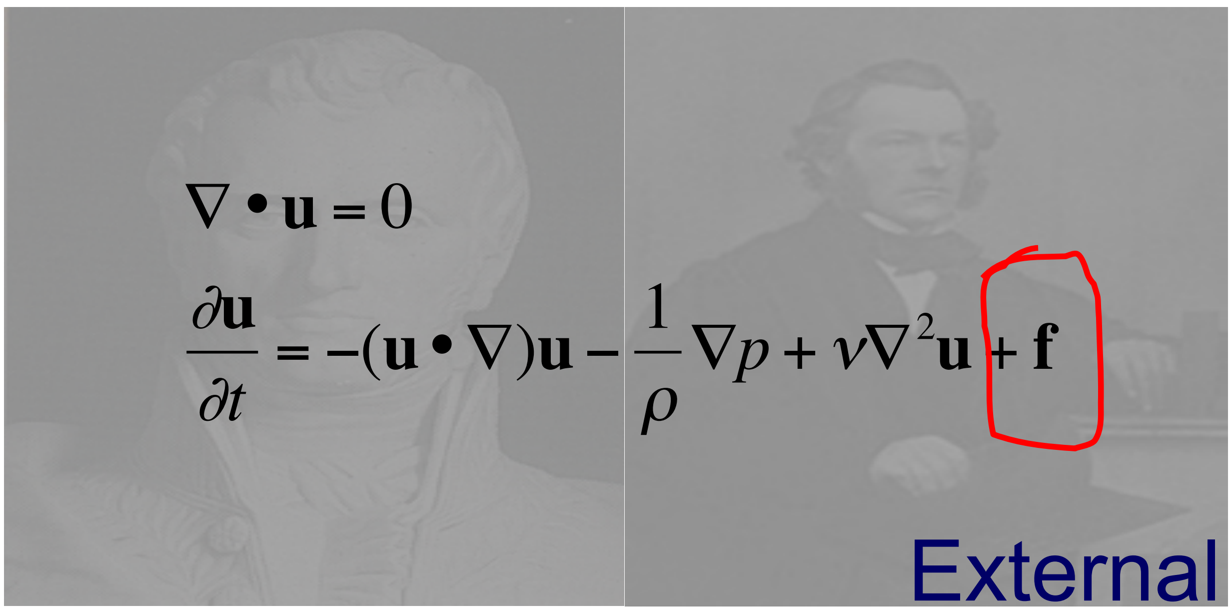
Fluid Concepts

- Back to Navier-Stokes



Fluid Concepts

- Back to Navier-Stokes



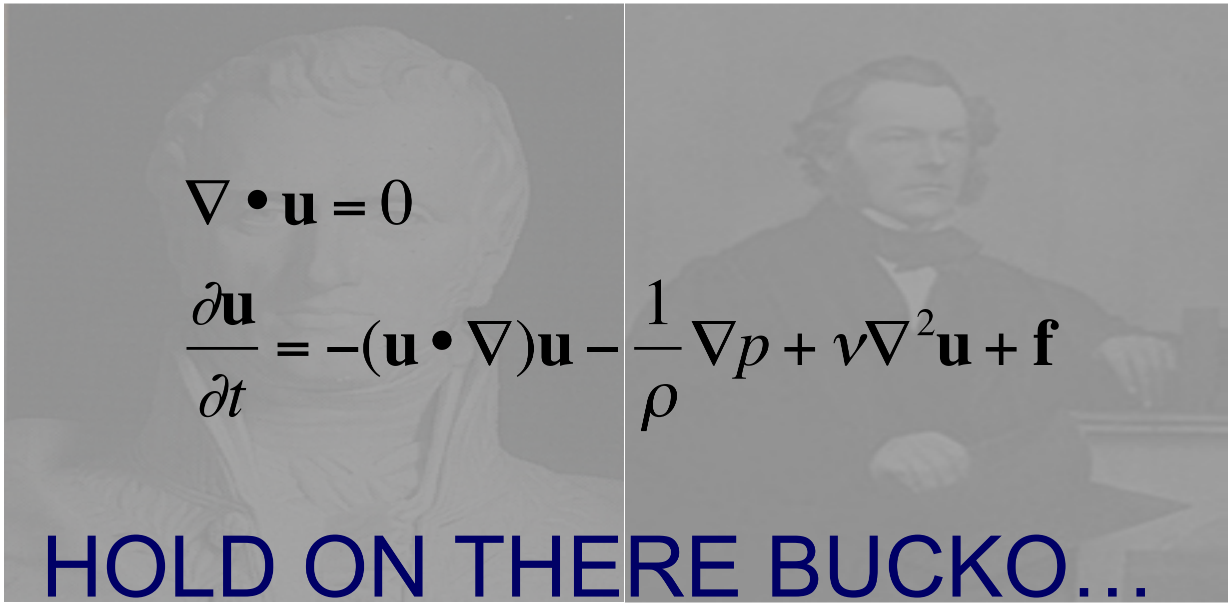
$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

External Forces

Fluid Concepts

- Back to Navier-Stokes

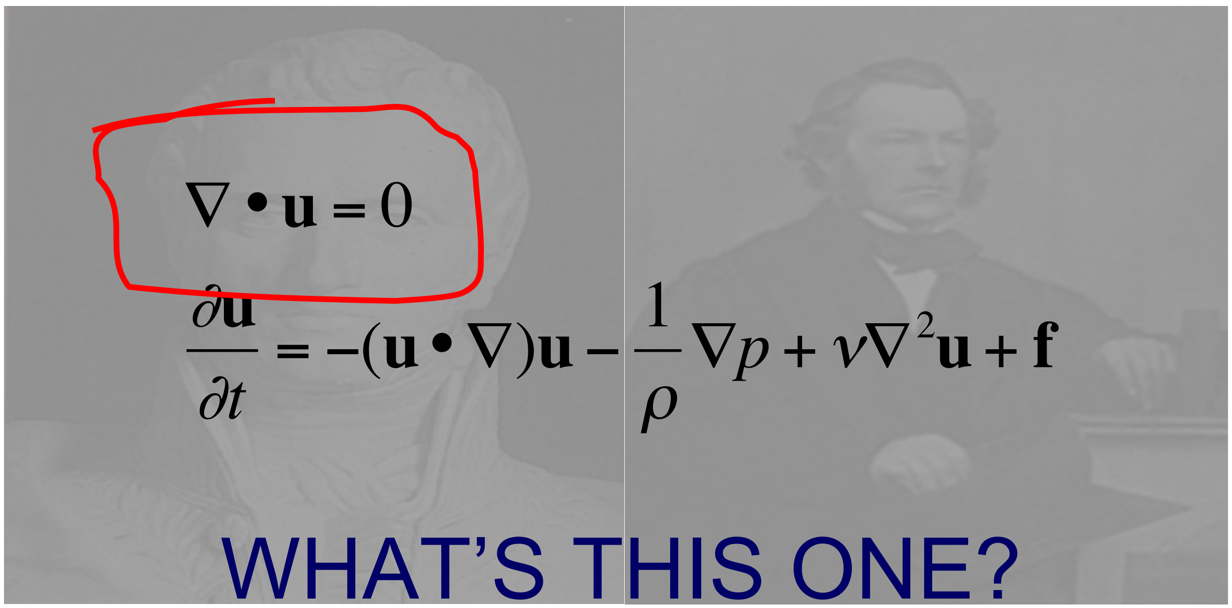


$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Fluid Concepts

- Back to Navier-Stokes



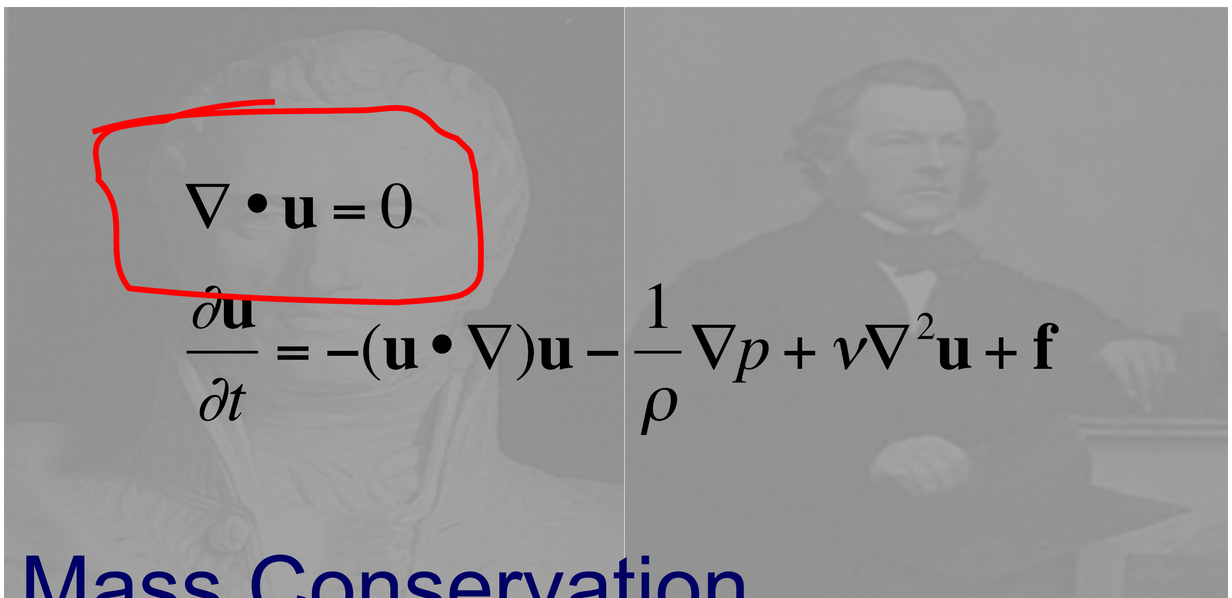
$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

WHAT'S THIS ONE?

Fluid Concepts

- Back to Navier-Stokes



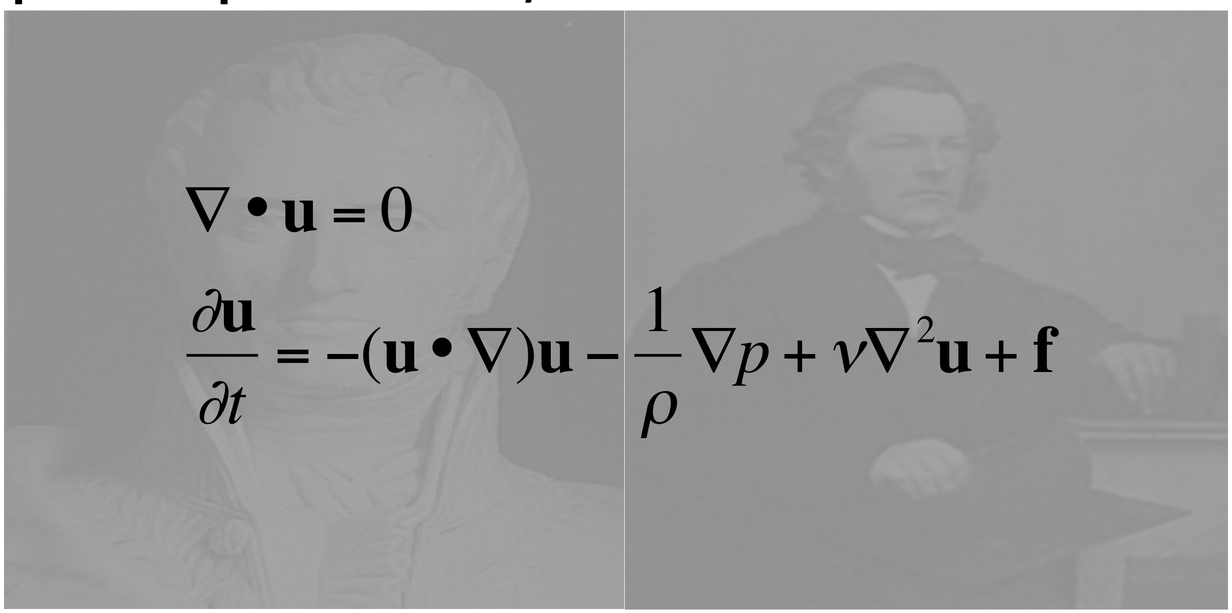
$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Mass Conservation

Fluid Concepts

- In principle then, Navier-Stokes is…

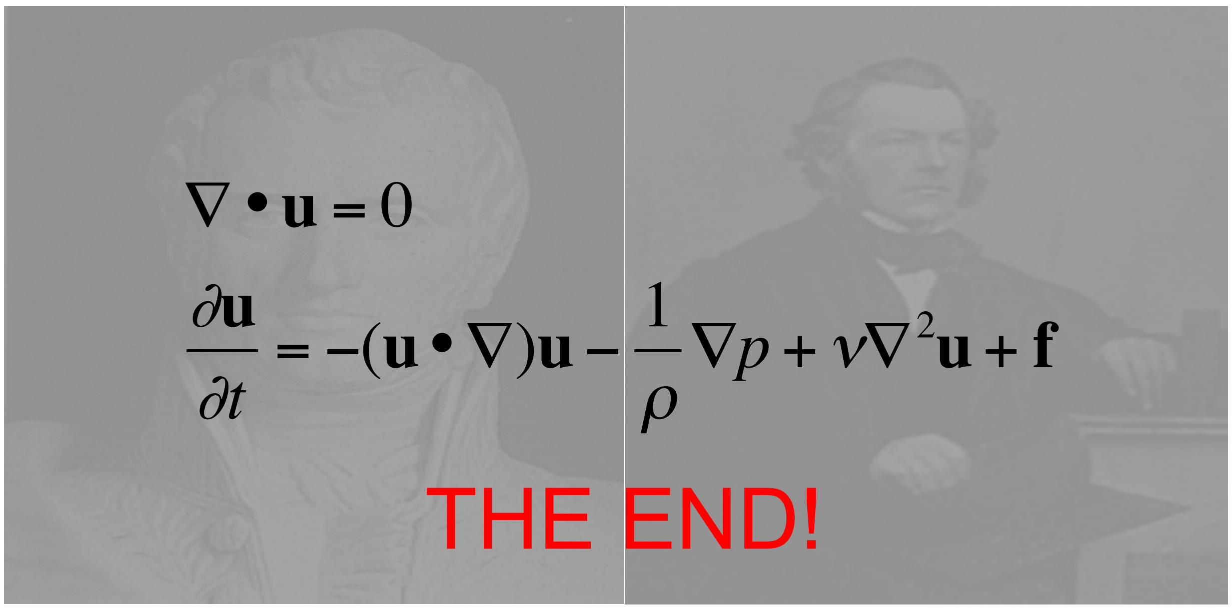


$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Fluid Concepts

- In principle then, Navier-Stokes is…



Fluid Concepts

- But not really, of course

Fluid Concepts

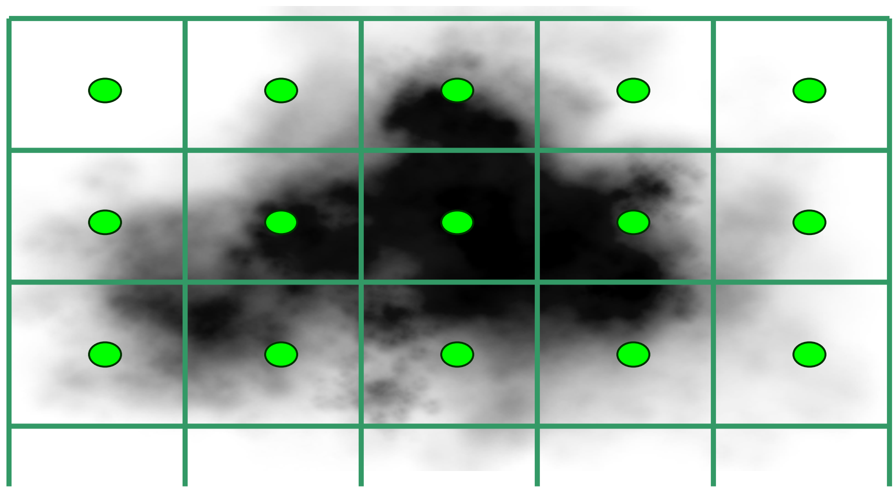
- But not really, of course
- Little tiny detail of implementation

Computational Fluid Types

- Grid-based/Eulerian (Stable Fluids)
- Particle-based/Lagrangian (Smoothed Particle Hydrodynamics)
- Surface-based (wave composition)

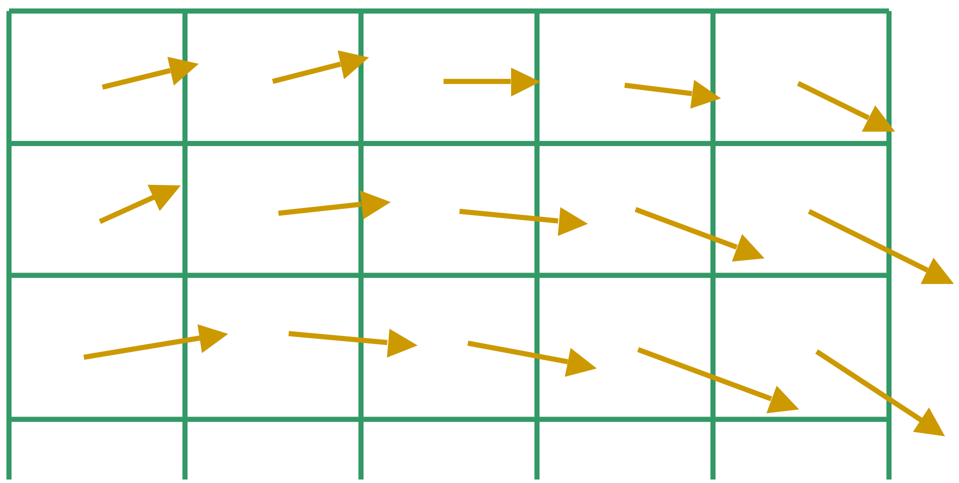
Grid-Based

- Store density, temp in grid centers



Grid-Based

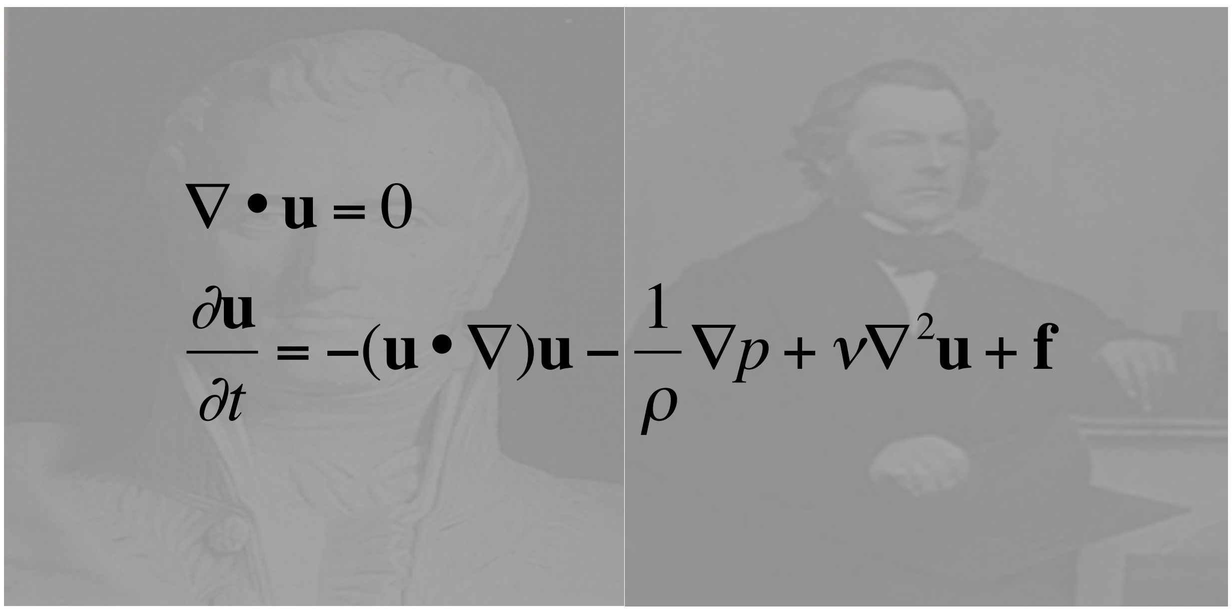
- Velocity (flow) from centers as well



- Could also do edges

Grid-Based

- How do we use this?



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

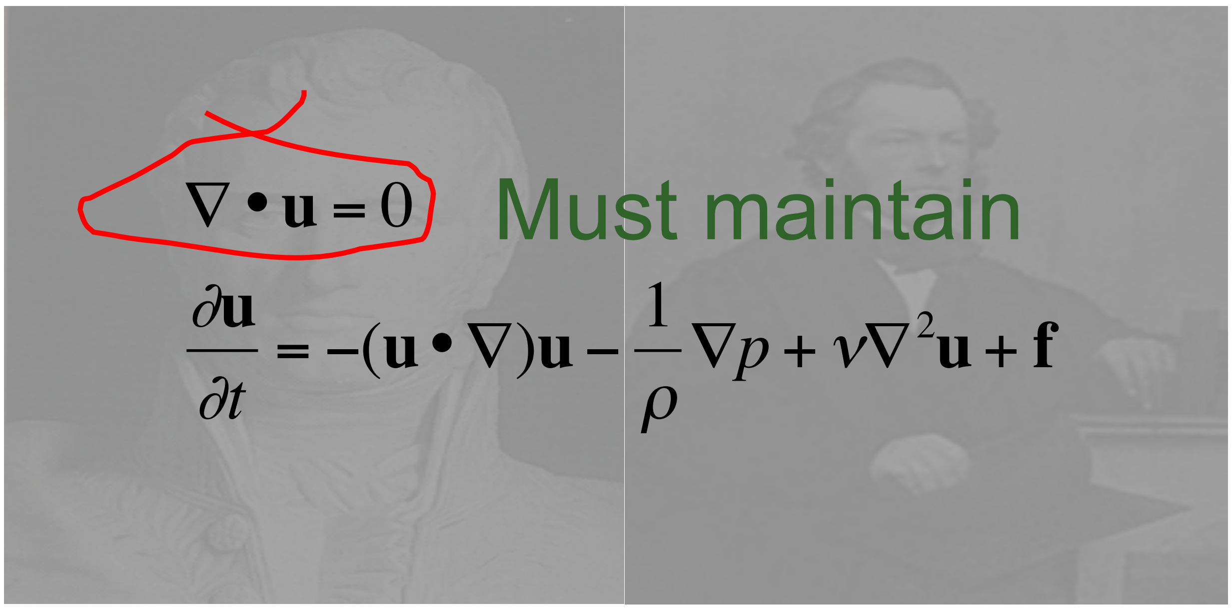
$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Grid-Based

- Jos Stam devised stable approximation: "Stable Fluids", SIGGRAPH '99

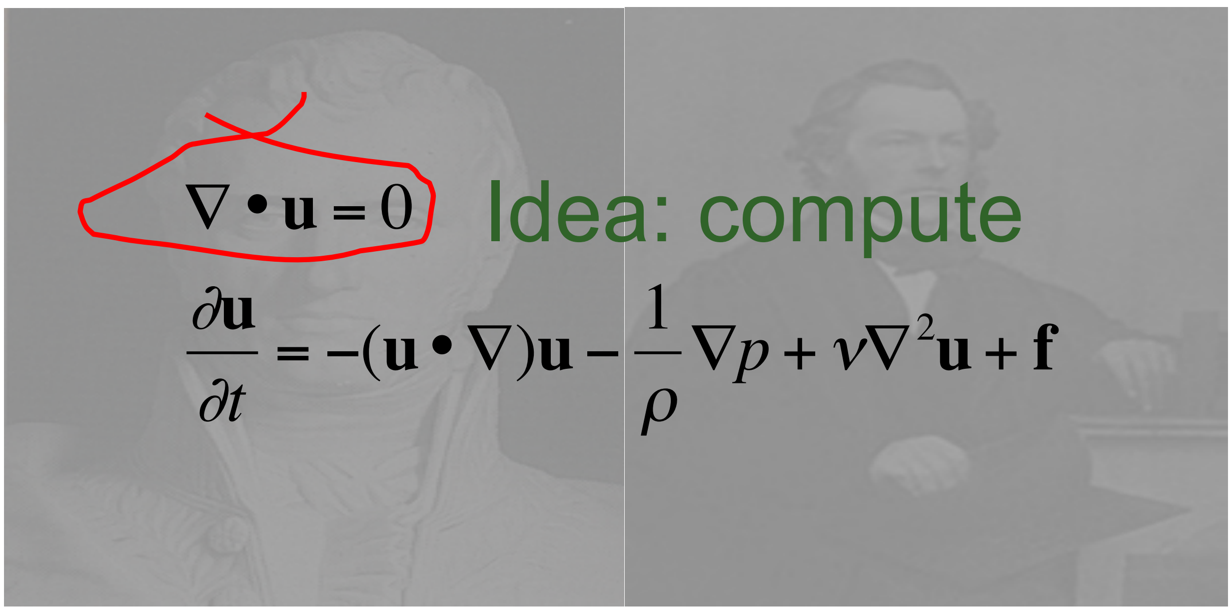
Grid-Based

- How do we use this?



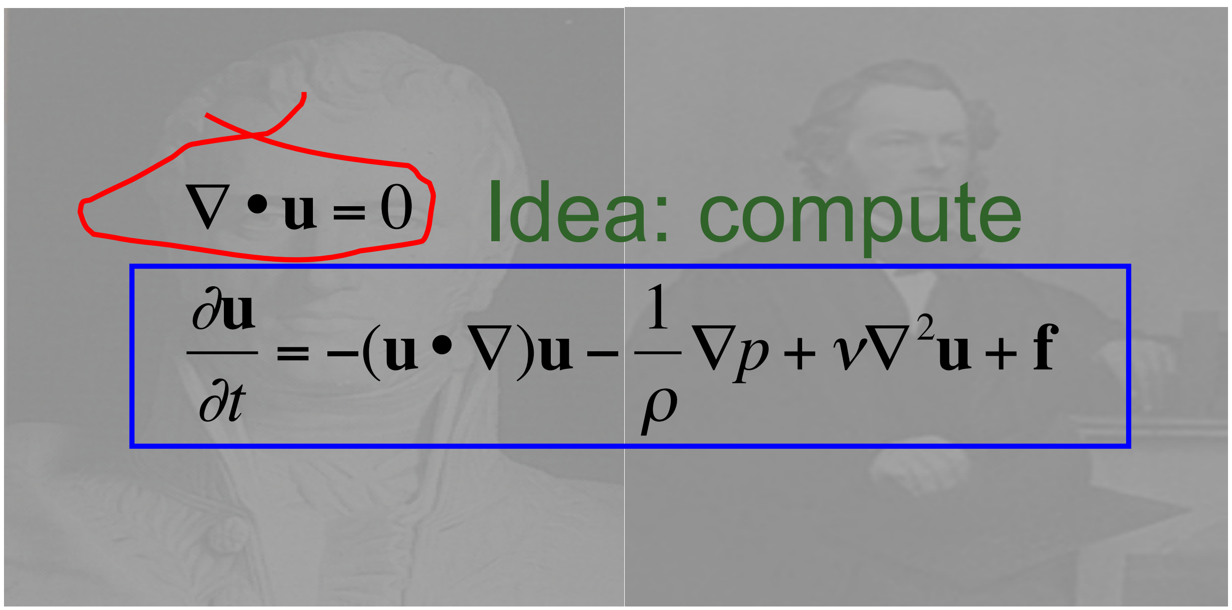
Grid-Based

- How do we use this?



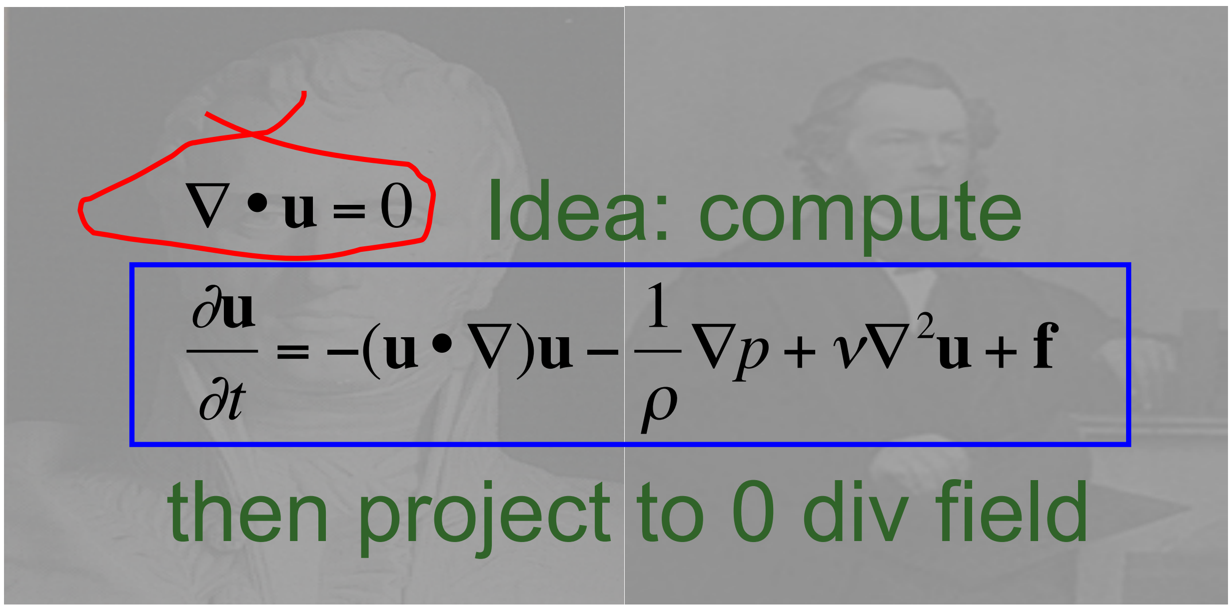
Grid-Based

- How do we use this?



Grid-Based

- How do we use this?



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

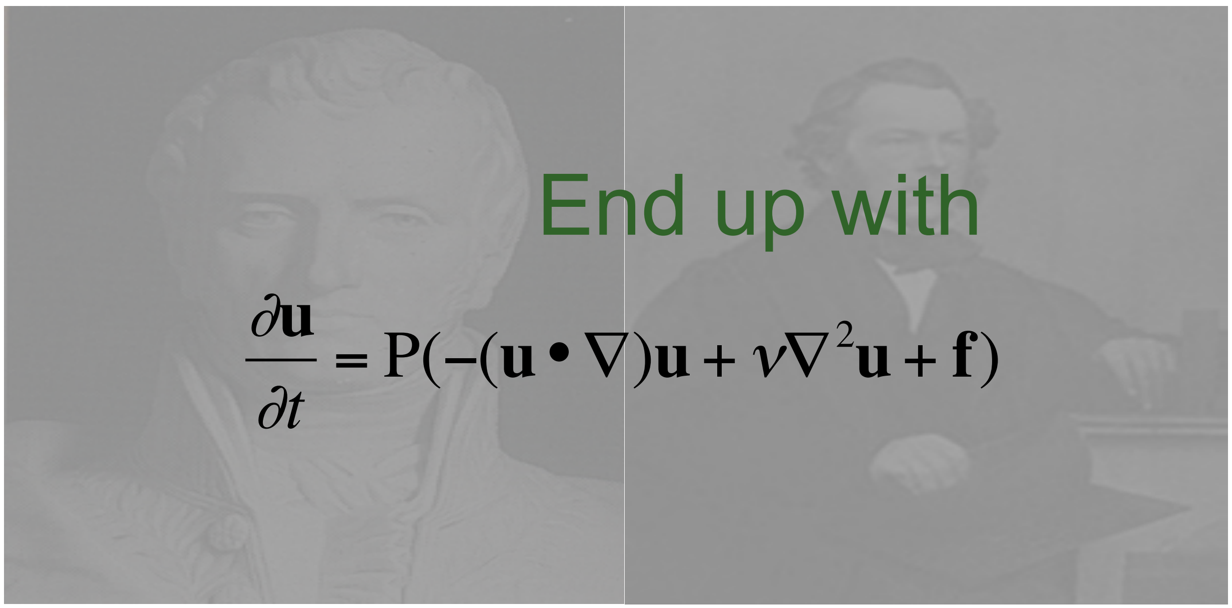
Idea: compute

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

then project to 0 div field

Grid-Based

- How do we use this?

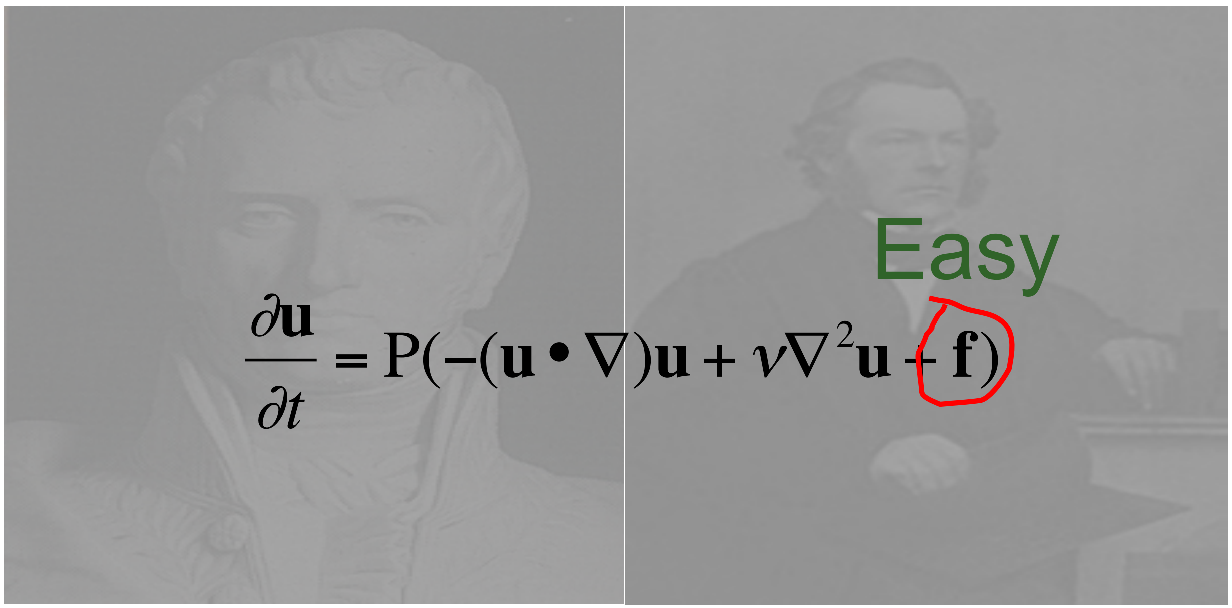


End up with

$$\frac{\partial \mathbf{u}}{\partial t} = \mathrm{P}(-(\mathbf{u} \bullet \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f})$$

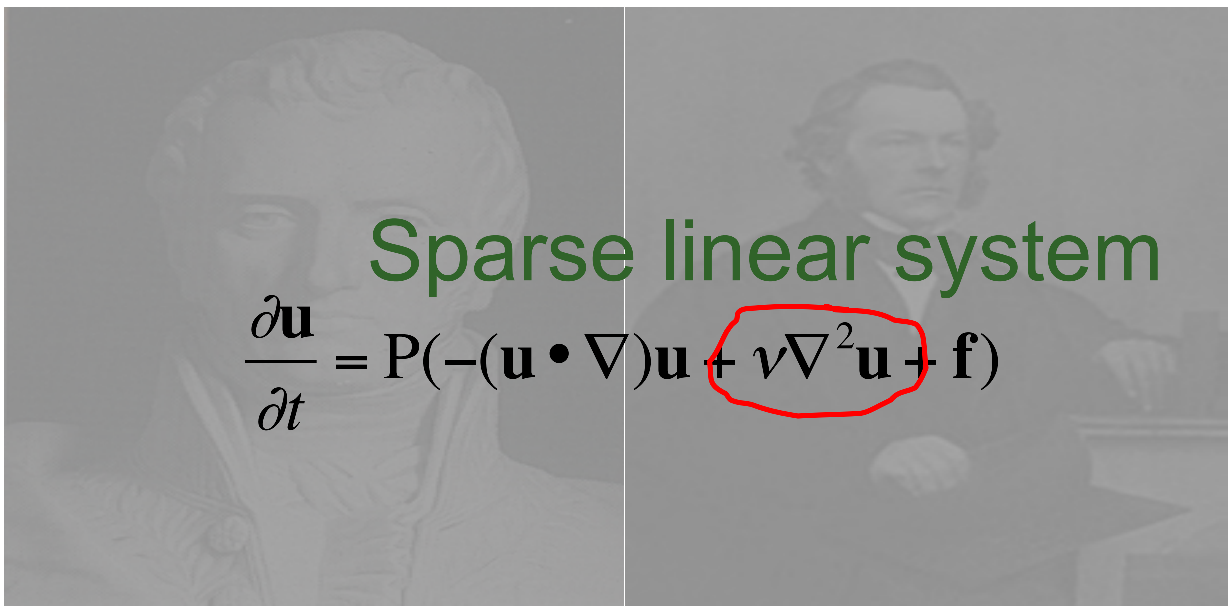
Grid-Based

- How do we use this?



Grid-Based

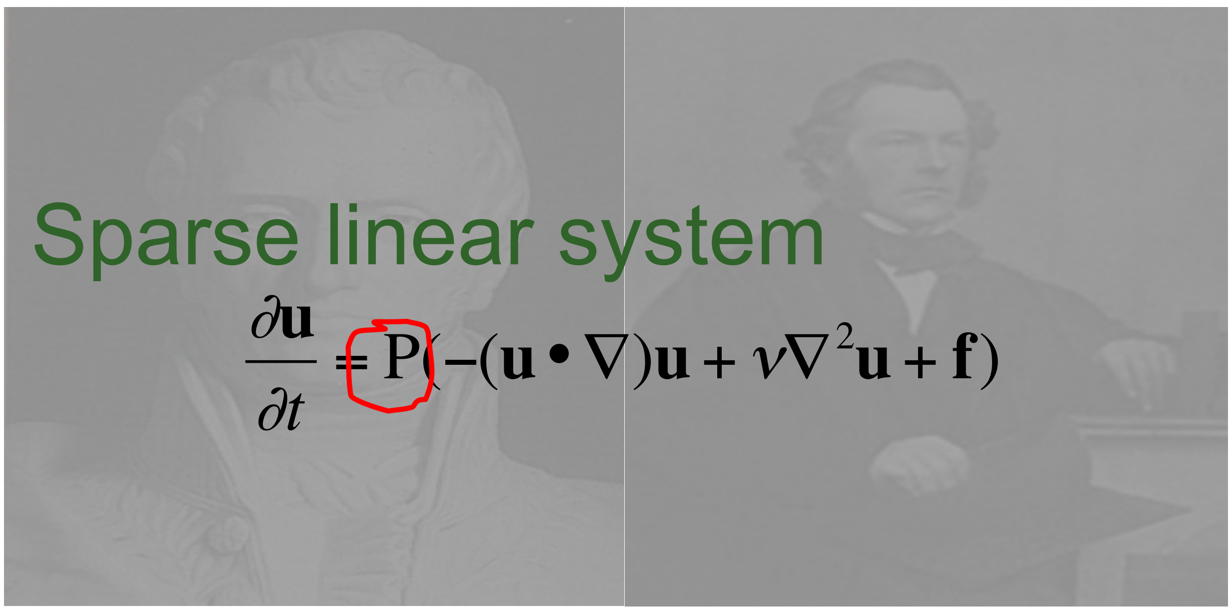
- How do we use this?



$$\frac{\partial \mathbf{u}}{\partial t} = \mathrm{P}(-(\mathbf{u} \bullet \nabla)\mathbf{u} + \nu\nabla^2\mathbf{u} + \mathbf{f})$$

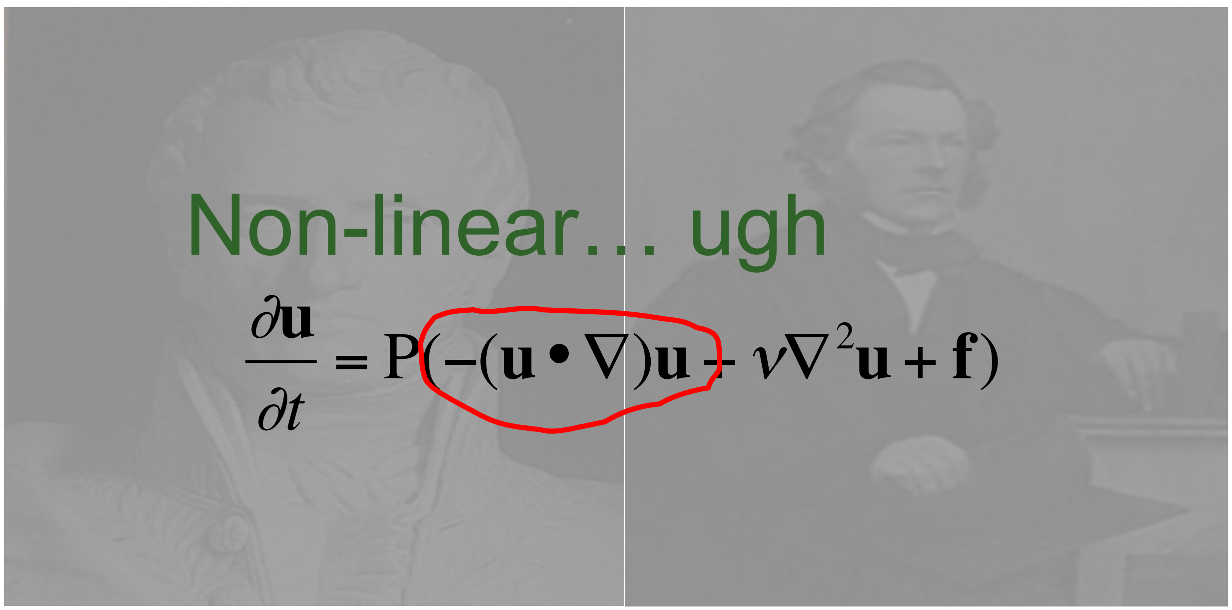
Grid-Based

- How do we use this?



Grid-Based

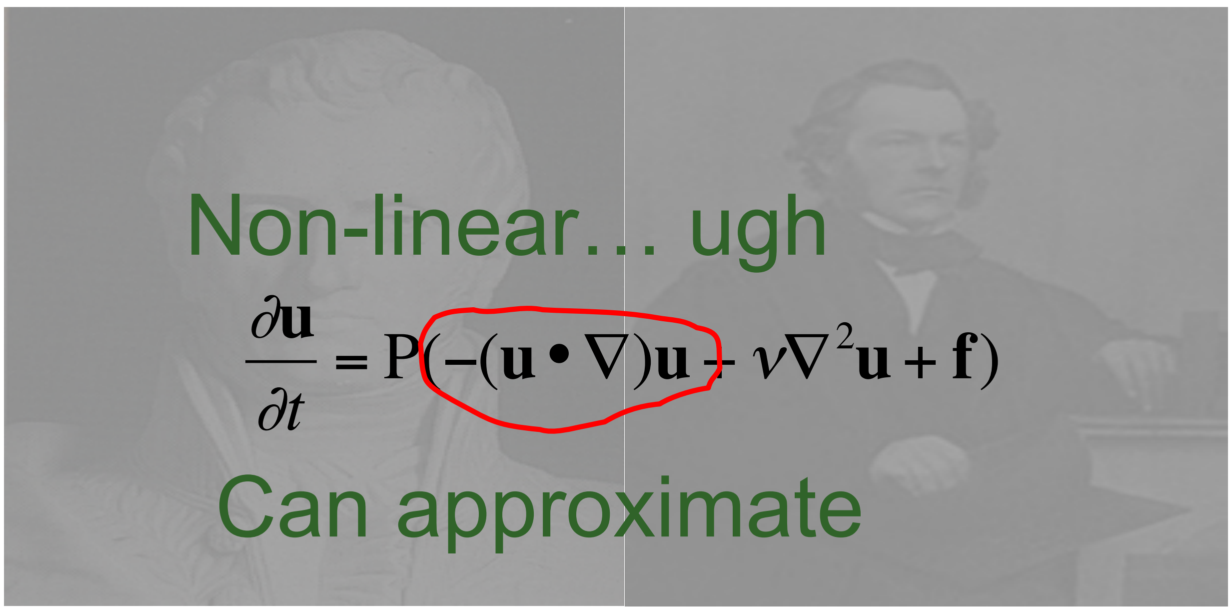
- How do we use this?



$$\frac{\partial \mathbf{u}}{\partial t} = \mathrm{P}\left(-(\mathbf{u} \bullet \nabla)\mathbf{u} + \nu\nabla^2\mathbf{u} + \mathbf{f}\right)$$

Grid-Based

- How do we use this?



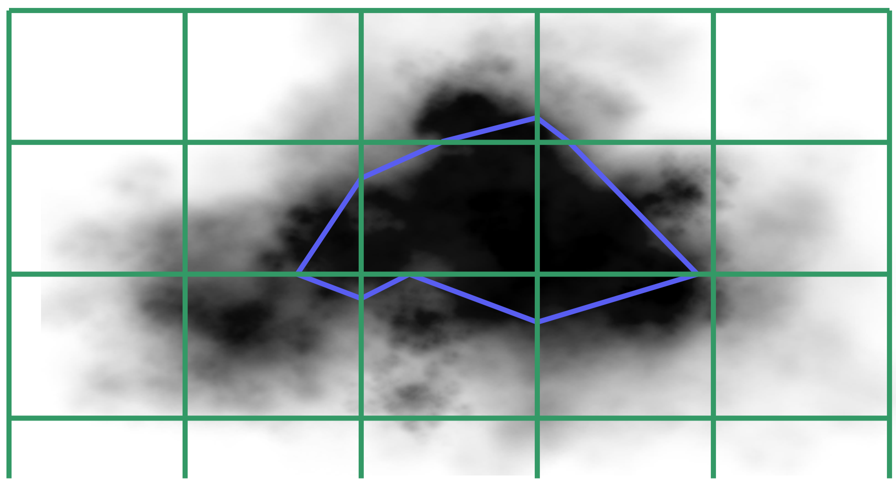
$$\frac{\partial \mathbf{u}}{\partial t} = \mathrm{P}\left(-(\mathbf{u} \bullet \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f}\right)$$

Grid-Based

- Overview
  - Update velocities based on
    - Forces, then
    - Advection, then
    - Viscosity
  - Project velocities to zero divergence
  - Update densities based on
    - Input sources
    - Velocity
    - Diffusion (similar to viscosity, sometimes not used)
  - Draw it

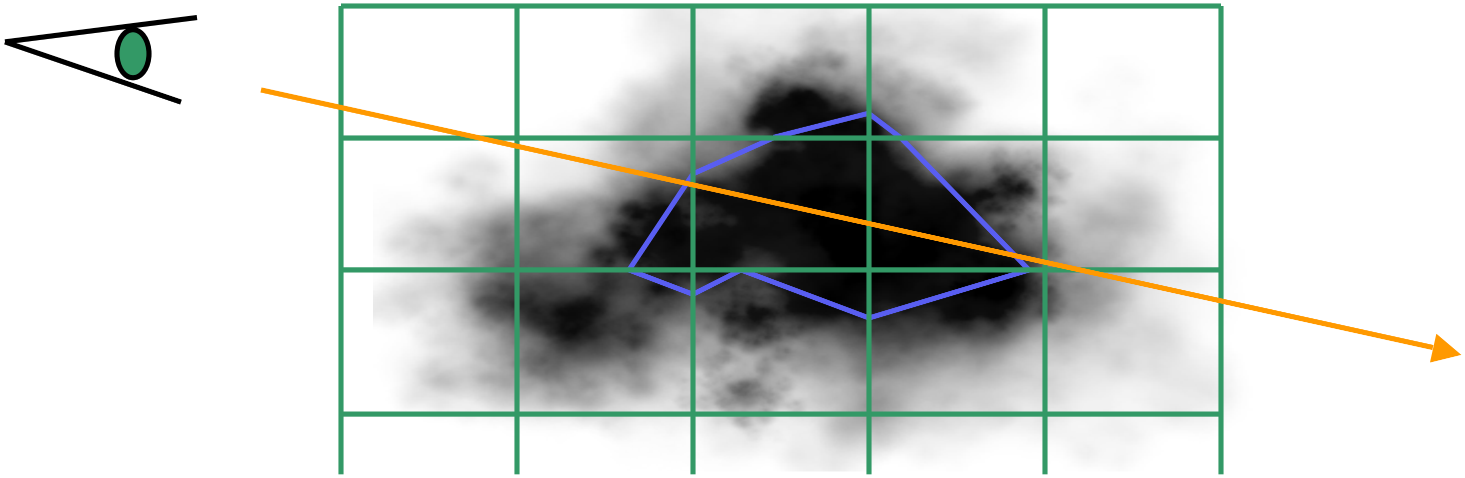
Rendering Grid-Based

- Build level surface



Rendering Grid-Based

- Determining color, transparency



Issues

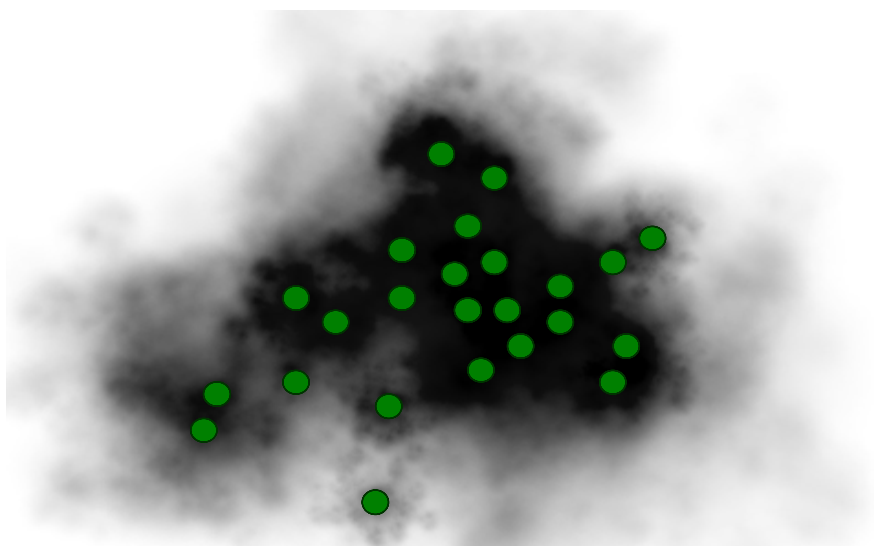
- Limited space
- Water "splashes" get lost
- Can be computationally expensive
- Dampens down
- But stable

Implementation

- Little Big Planet
  - “Death smoke”
  - Bubble pop
  - Other smoke effects
- Hellgate: London
- GDC09 NVIDIA demo

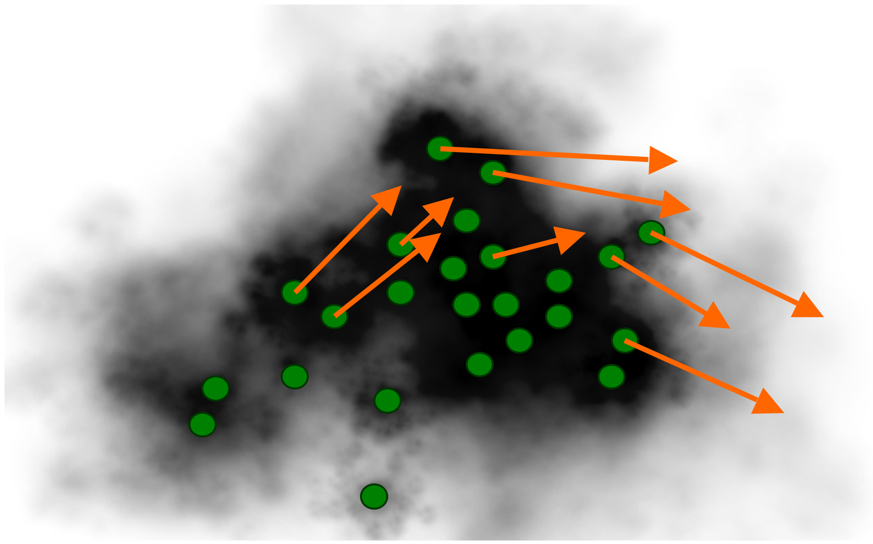
Smoothed Particle Hydrodynamics

- Approximate fluid with small(er) set of particles



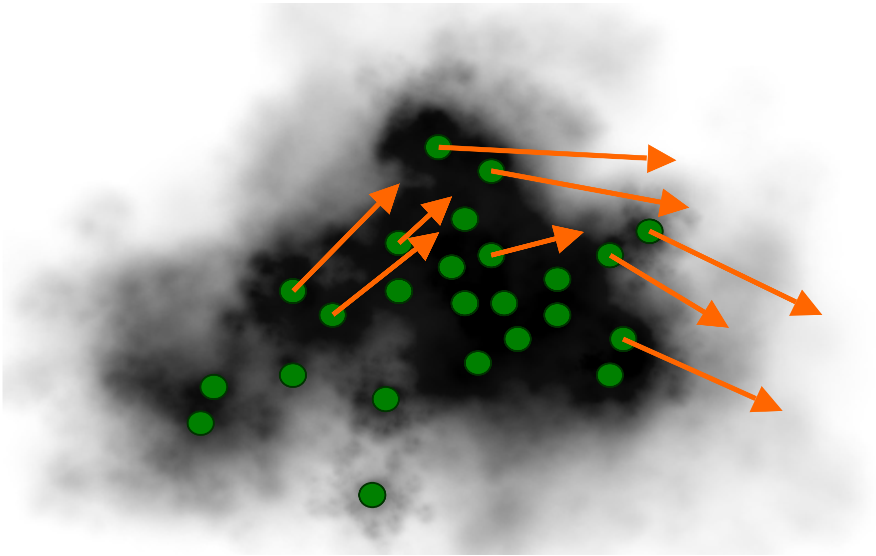
SPH

- Velocities at particles provide flow



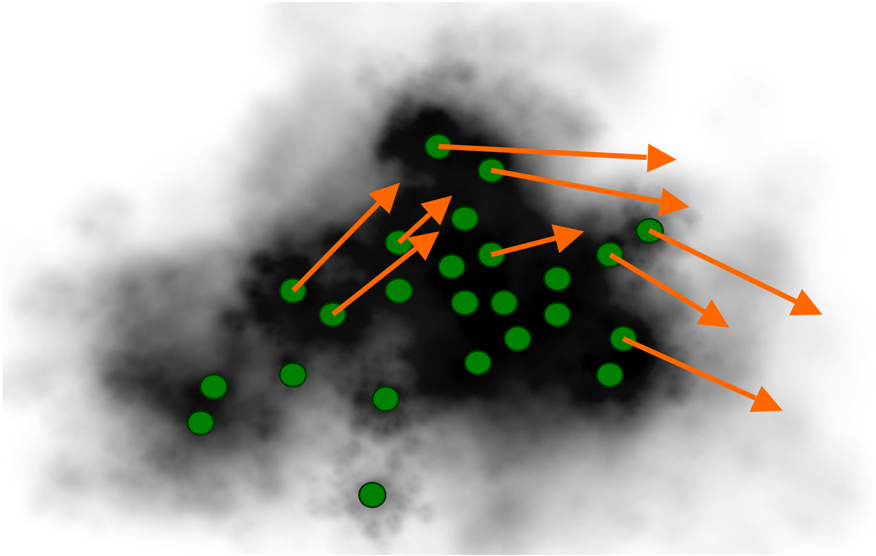
SPH

- Idea: treat as particle system



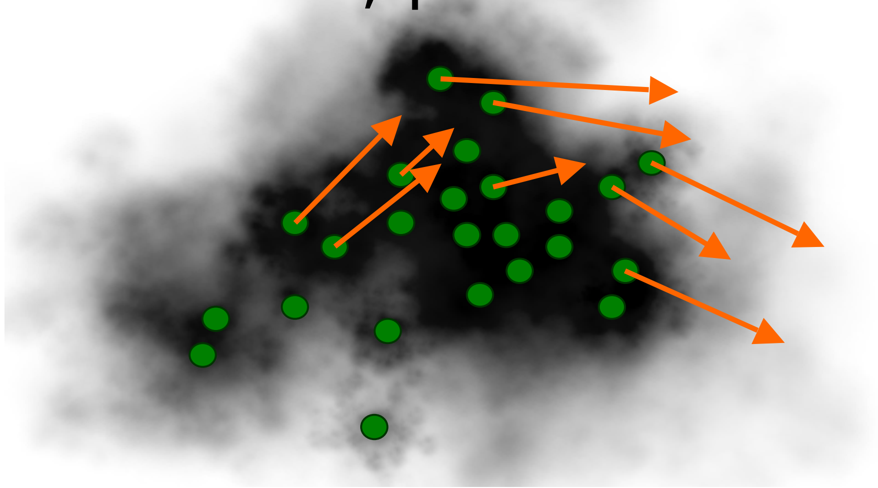
SPH

- Idea: treat as particle system
  - Determine forces



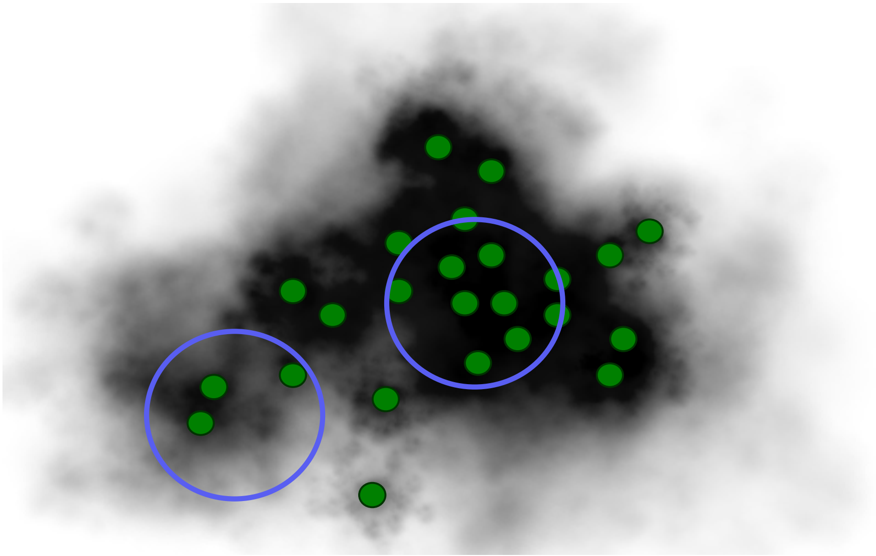
SPH

- Idea: treat as particle system
  - Determine forces
  - Update velocities, positions



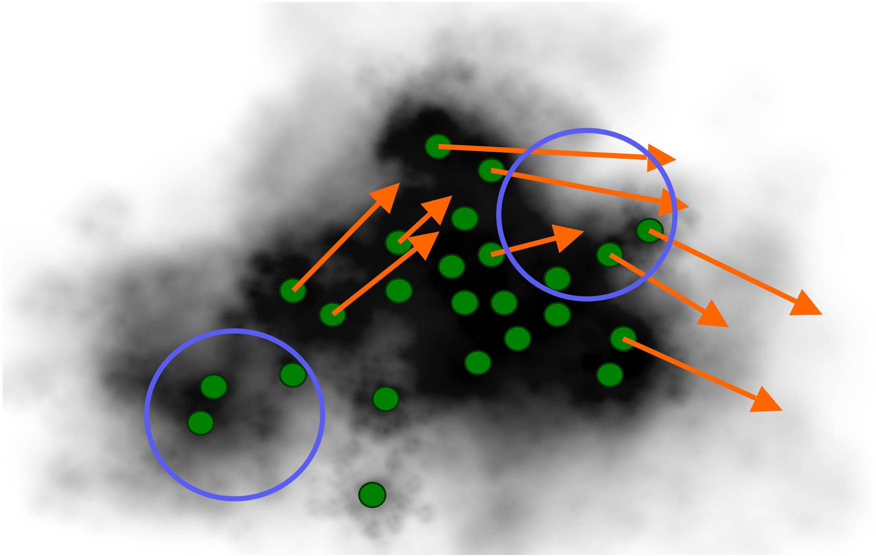
SPH

- Weighted average gives density (smoothing kernel)



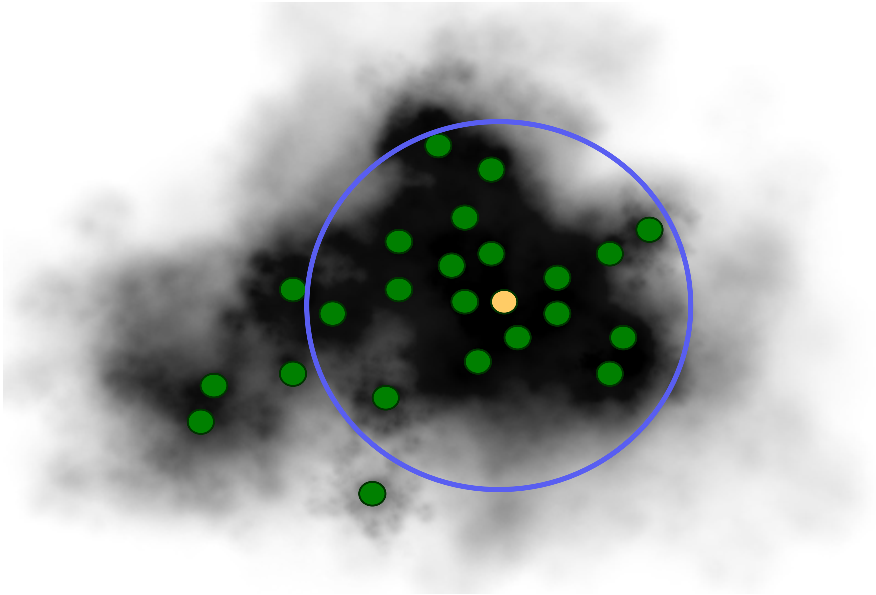
SPH

- Can also use kernel to get general velocity



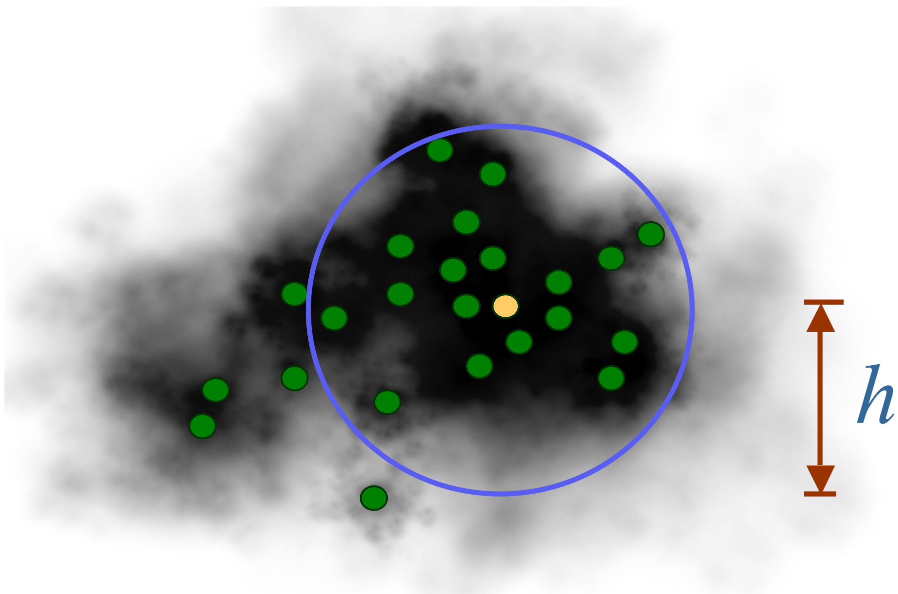
SPH

- Usually center at particle



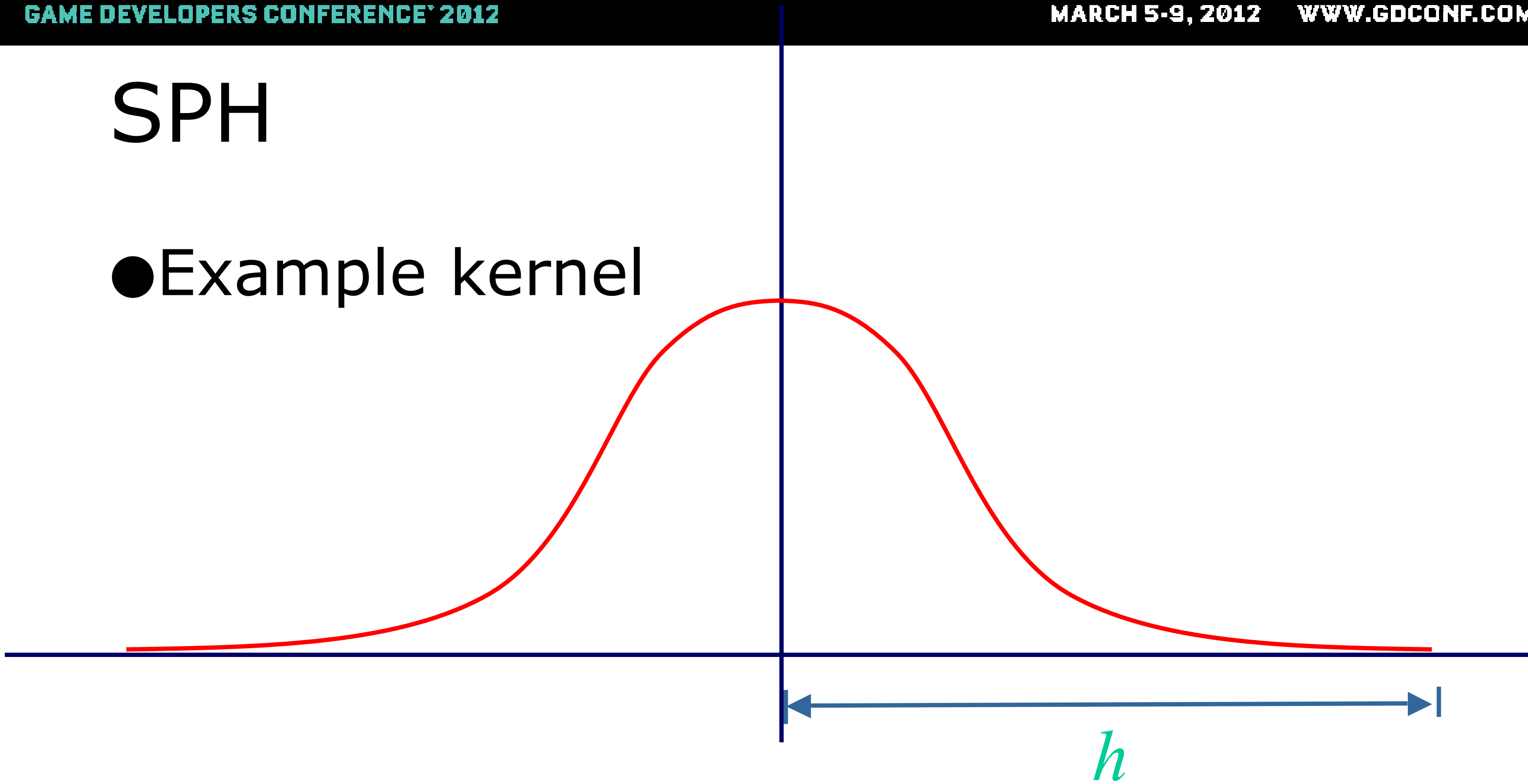
SPH

- Specify width by h



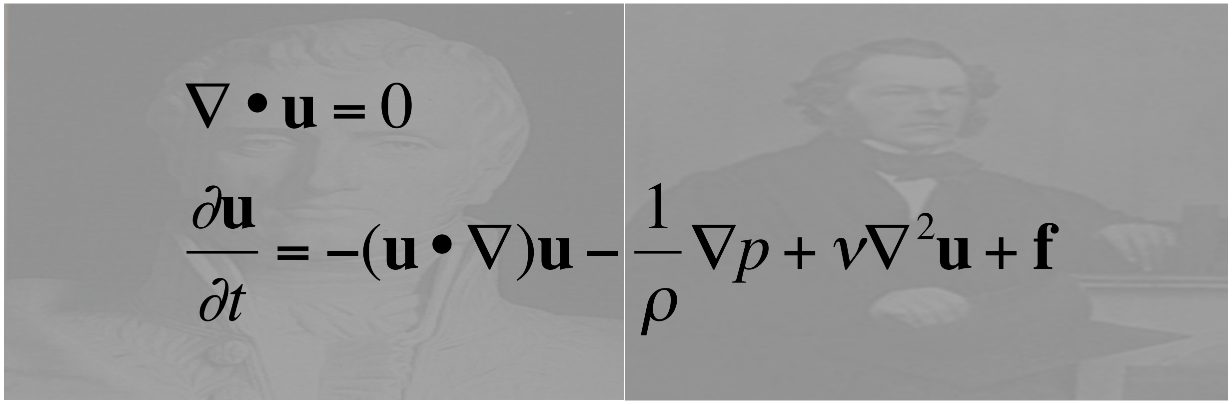
SPH

- Example kernel



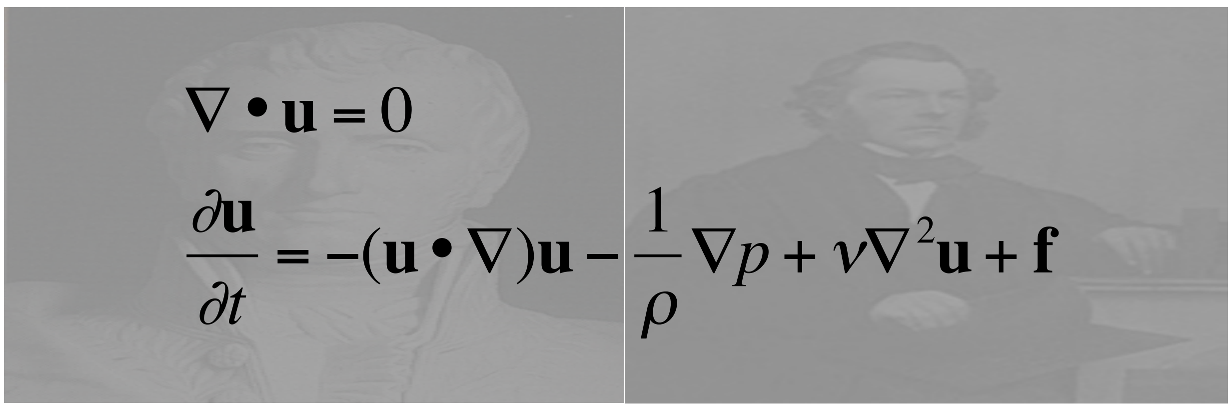
SPH

- Back to Navier-Stokes



SPH

- Back to Navier-Stokes



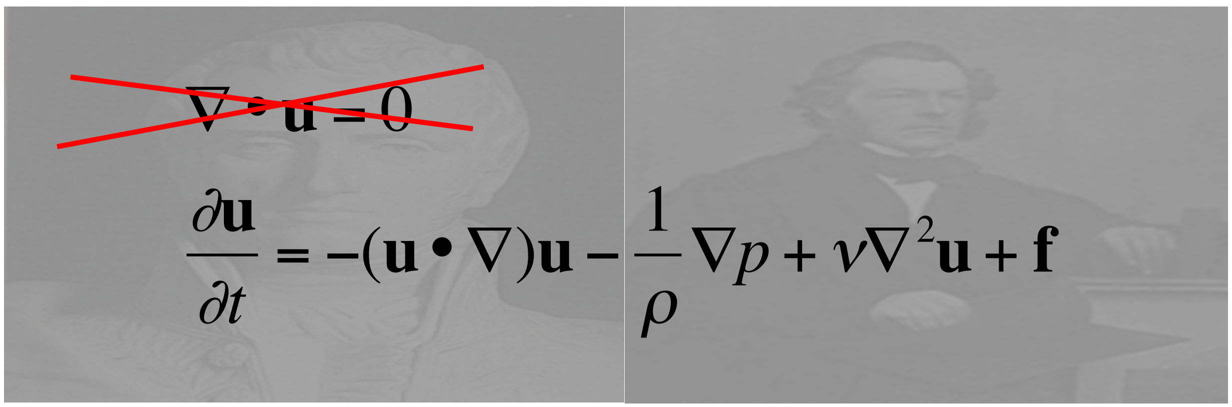
$$\nabla \bullet \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \bullet \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Have fixed # particles and mass, so…

SPH

- Back to Navier-Stokes



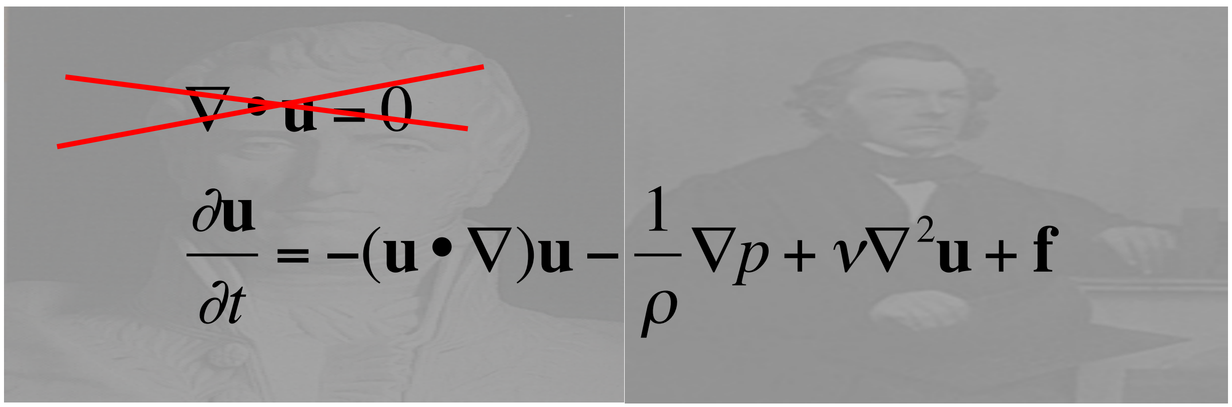
$$\nabla \cdot \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Have fixed # particles and mass, so… mass is automatically conserved

SPH

- Back to Navier-Stokes



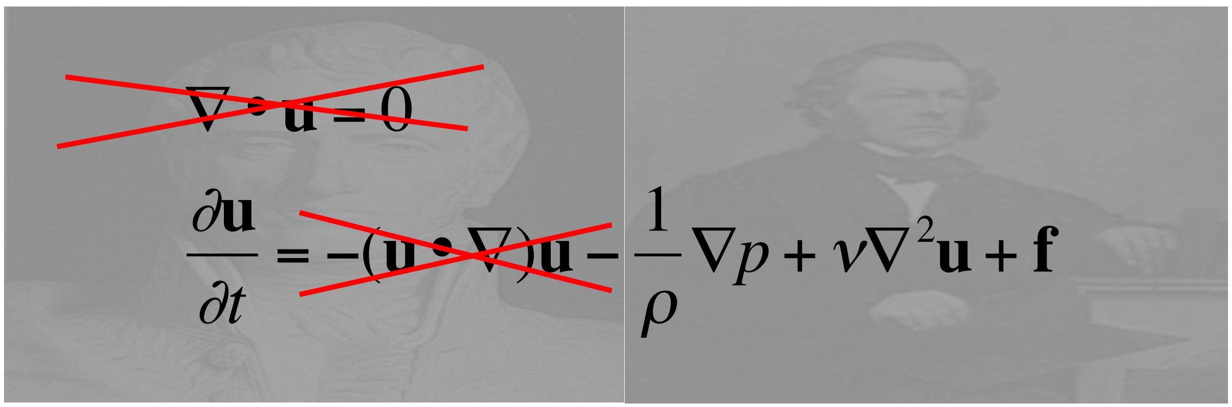
$$\nabla \cdot \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

Advection automagically handled by particle update, so…

SPH

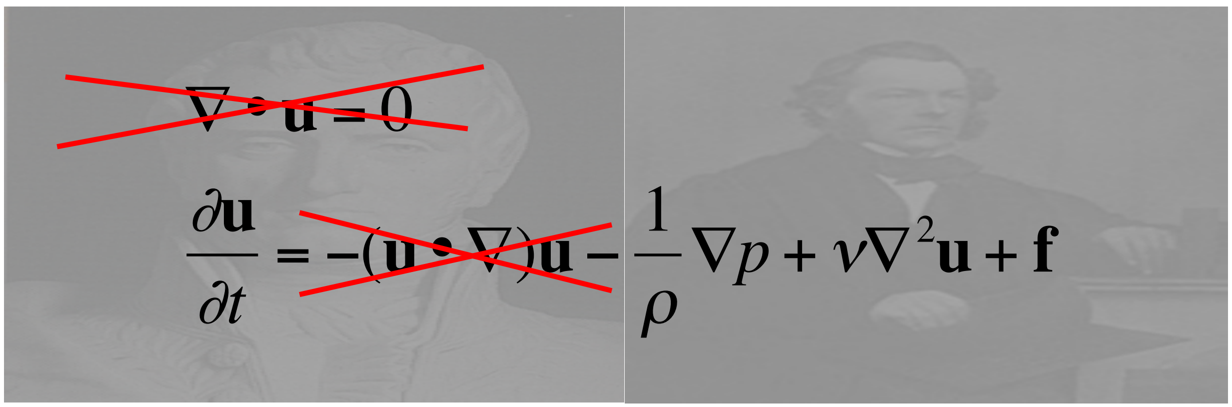
- Back to Navier-Stokes



Advection automagically handled by particle update, so…

SPH

- Simplifies to

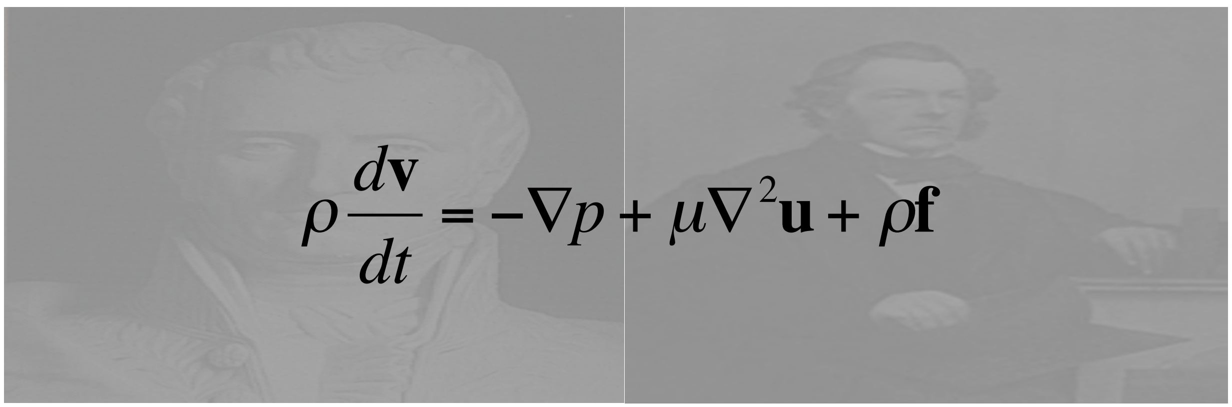


$$\cancel{\nabla \bullet \mathbf{u} = 0}$$

$$\frac{\partial \mathbf{u}}{\partial t} = \cancel{-(\mathbf{u} \bullet \nabla)\mathbf{u}} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

SPH

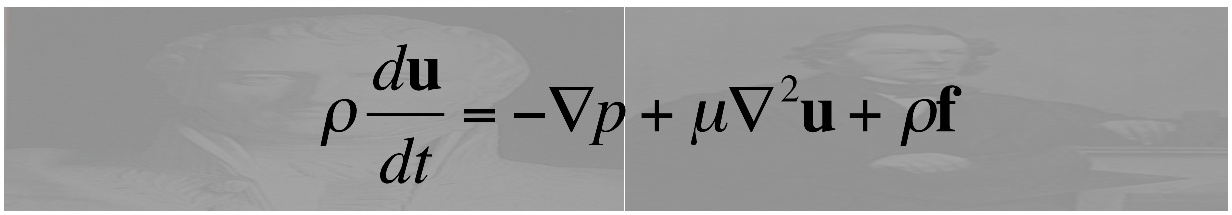
- Simplifies to



$$\rho \frac{d\mathbf{v}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f}$$

SPH

- Functional breakdown



$$\rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f}$$

SPH

- Functional breakdown

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f}$$

Change in velocity

SPH

- Functional breakdown

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f}$$

Pressure

SPH

- Functional breakdown

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f}$$

Viscosity

SPH

- Functional breakdown

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{f}$$

External forces

SPH

- Compute densities, local pressure
- Generate forces on particles
  - External
  - Pressure
  - Viscosity
- Update velocities, positions
- Render

SPH

- Rendering
  - Marching cubes (using smoothing kernel)
  - Blobs around particles/splatting

SPH Implementations

- Takahiro Harada
- Kees van Kooten (Playlogic)
- NVIDIA PhysX
- Rama Hoetzlein* (SPH Fluids 2.0)
- Takashi AMADA*

* Source code available

SPH Issues

- Need a *lot* of particles
- Computing level surface can be a pain
- Can be difficult to get stable simulation

SPH Improvements

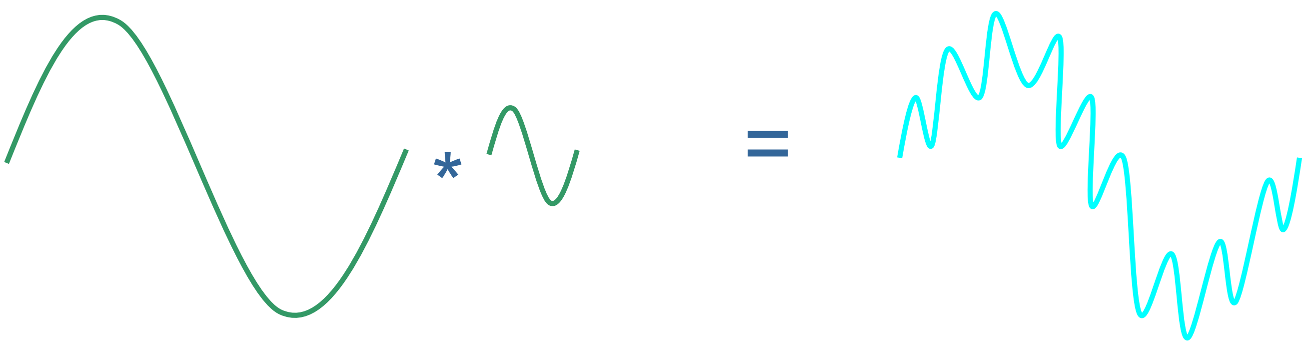
- Spatial hashing
- Variable kernel width
- CFD/SPH Hybrid
  - CFD manages general flow
  - SPH “splashes”

Surface Simulation

- Idea: for water, all we care about is the air-water boundary (level surface)
- Why simulate the rest?
- This is what Insomniac R20 system does

R20

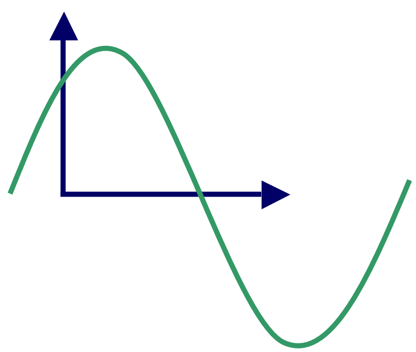
- Done by Mike Day, based on *Titanic* water
  - Basic idea: convolve sinusoids procedurally



  - Much cheaper to multiply in frequency domain and do FFT (assuming periodic)

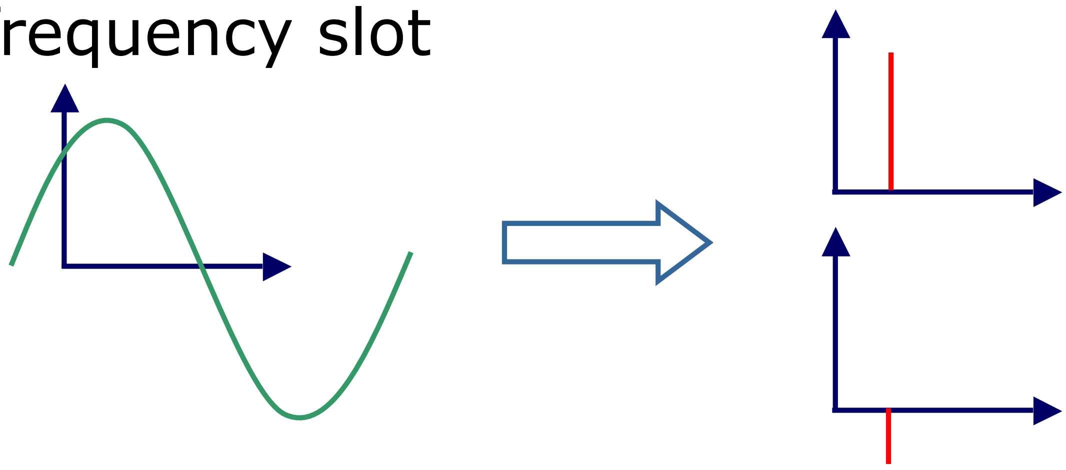
R2O

- Review
  - Sinusoid in spatial domain



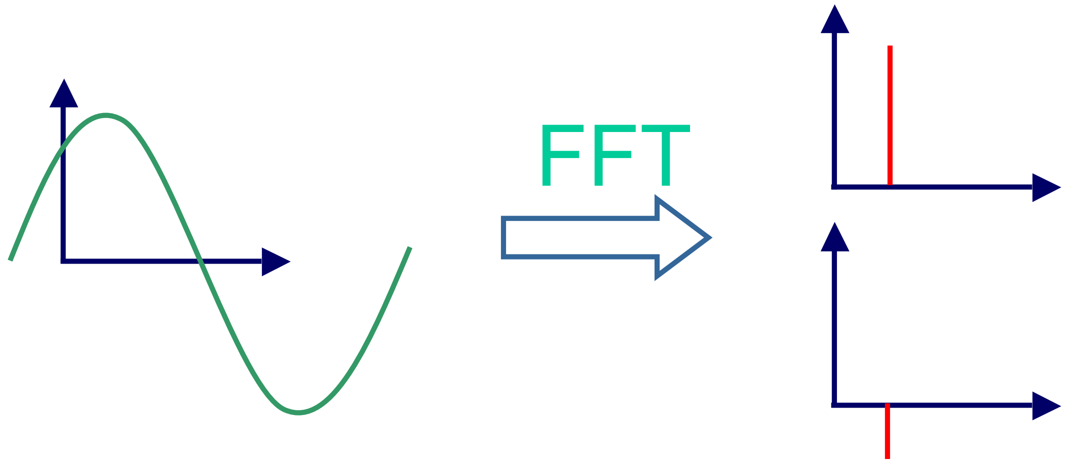
R2O

- Review
  - Can represent as magnitude+phase in frequency slot



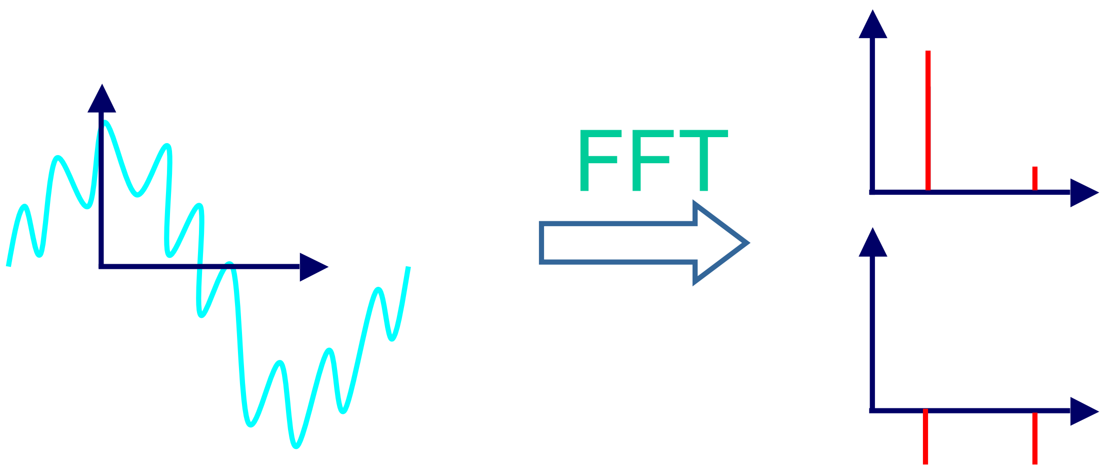
R2O

- Review
  - Requires periodic function



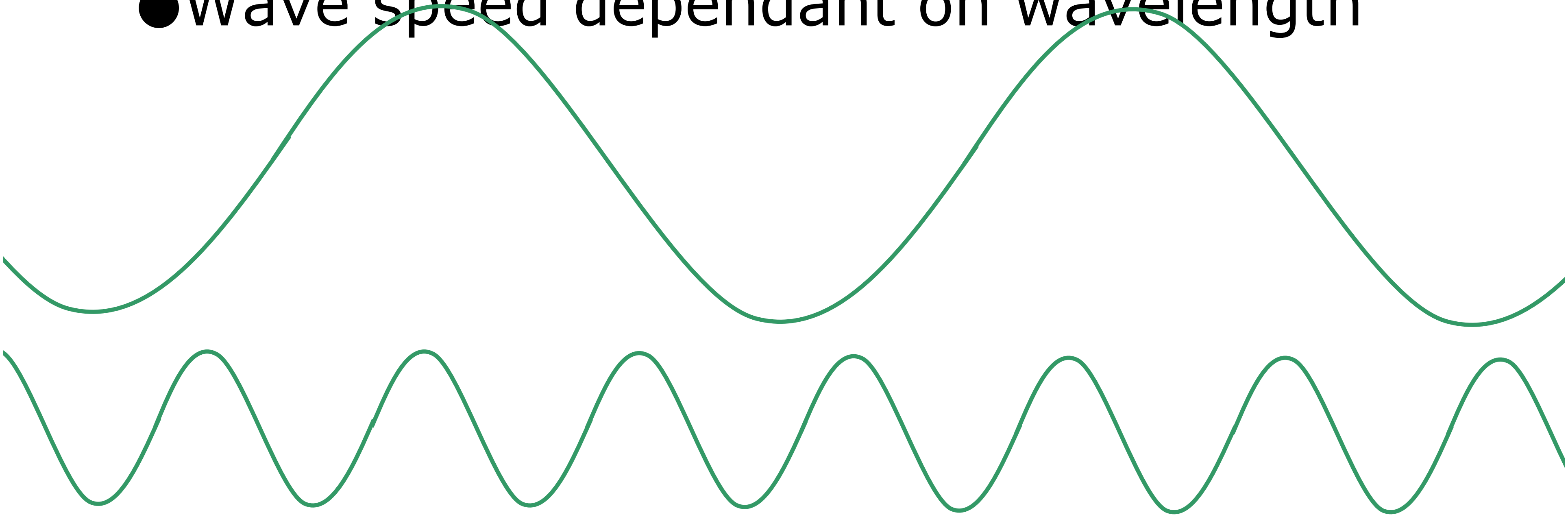
R2O

- Review
  - Multiple sinusoids end up at multiple entries



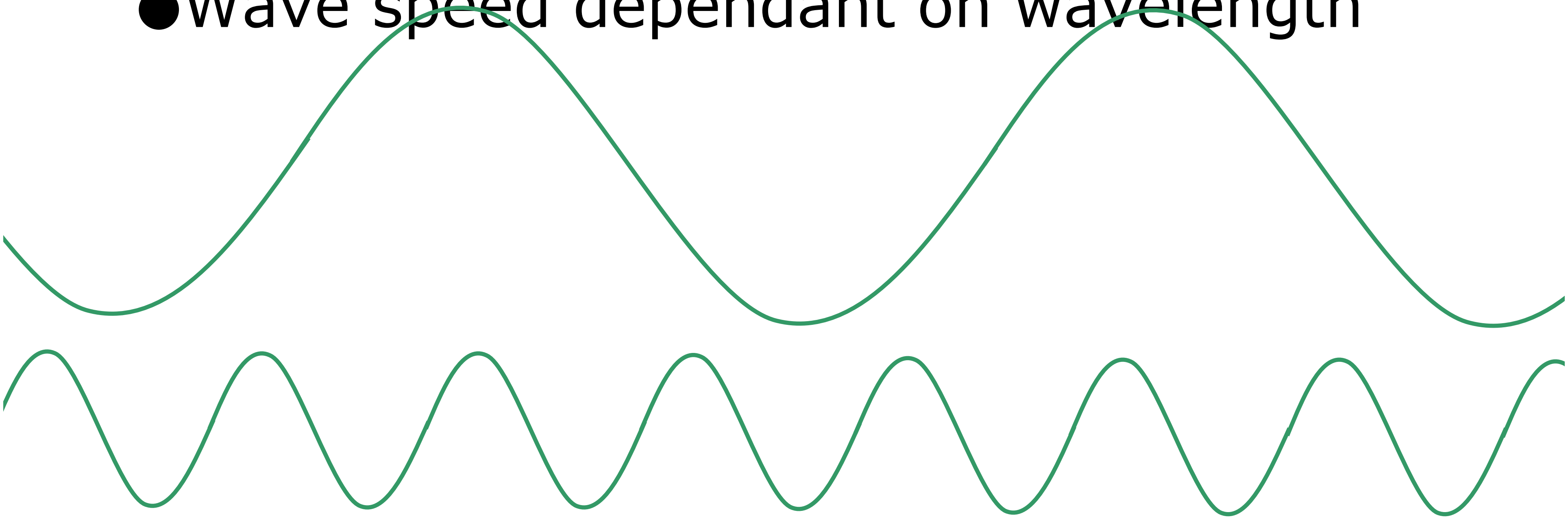
R2O

- Wave speed dependant on wavelength



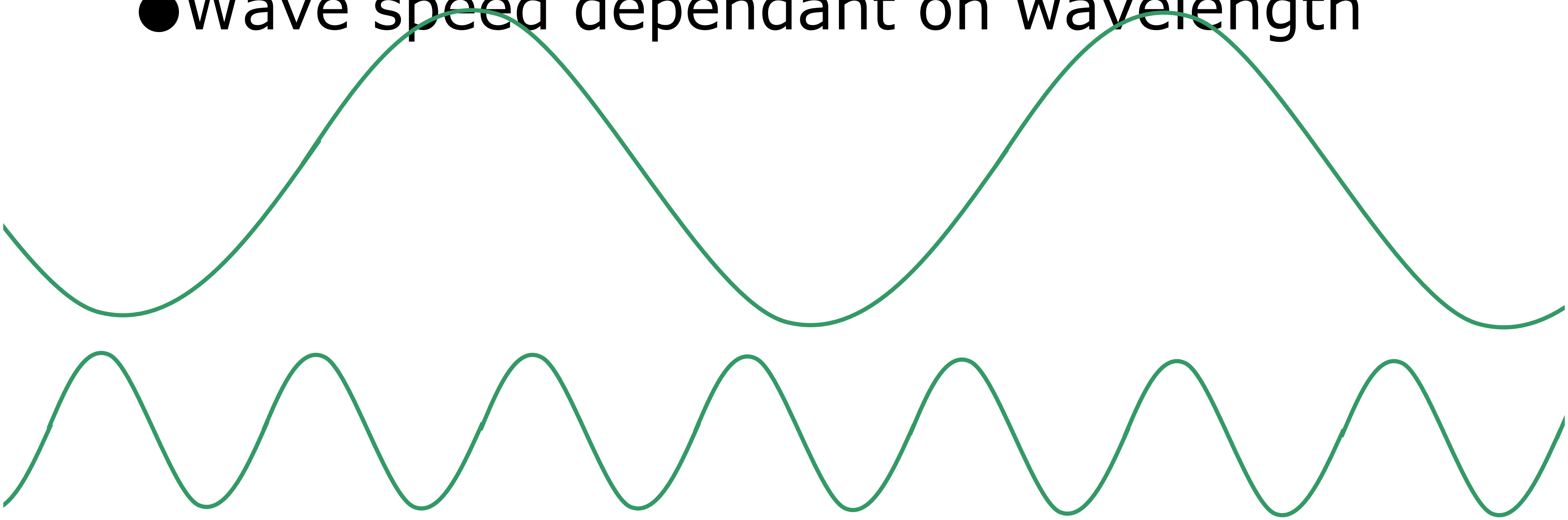
R2O

- Wave speed dependant on wavelength



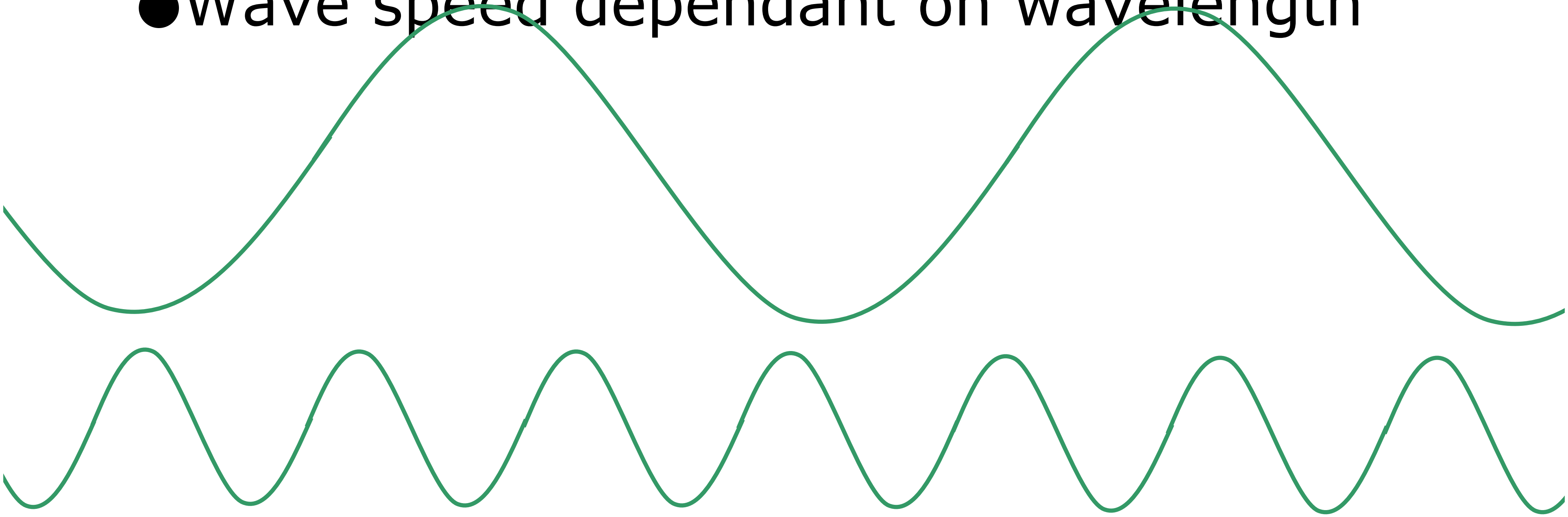
R2O

- Wave speed dependant on wavelength



R2O

- Wave speed dependant on wavelength



R2O

- Wave speed dependant on wavelength
  - I.e. phases update at different rates
  - AKA dispersion

R2O

- General procedure
  - Start with convolved data in (r,φ) form
  - Update phase angles for each sinusoid
    - Angular velocity*dt
    - Dependent on frequency
  - Do inverse FFT to get spatial result

R2O

- FFT kernel limited to 32x32
- Combine multiple levels via LOD height field scheme
  - Gives high detail close to camera

R2O

- Interactive waves
  - Just adding in splashes looks fake
  - Instead, do some more FFT trickery so all our work occurs in the same domain
  - Non-periodic, so have to manage edges
  - Gives nice dispersion effects

R2O

- Rendering
  - Rendered as height field mesh
  - Add normal map for detail
  - Cube map/frame buffer map for reflections
  - Distortion effect for refractions

R2O

- Nifty video

References

- Jos Stam, “Stable Fluids”, SIGGRAPH 1999
- Mattias Müller, et. al, “Particle-Based Fluid Simulation for Interactive Applications”, SIGGRAPH Symposium on Computer Animation 2003
- Jerry Tessendorf, “Simulating Ocean Water,” SIGGRAPH Course Notes.
- http://www.insomniacgames.com/tech