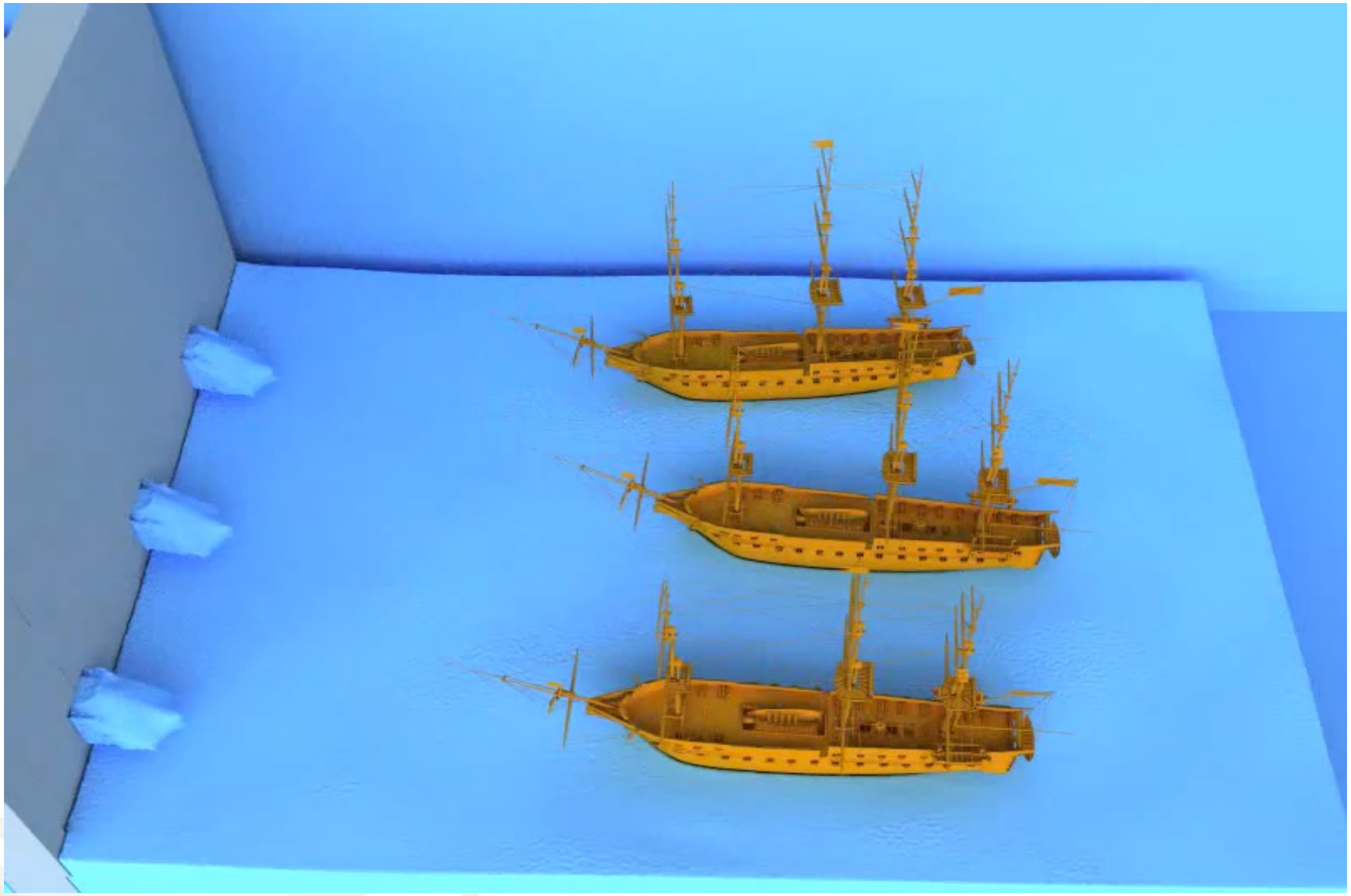
Animation for Computer Games
COMP 477/6311

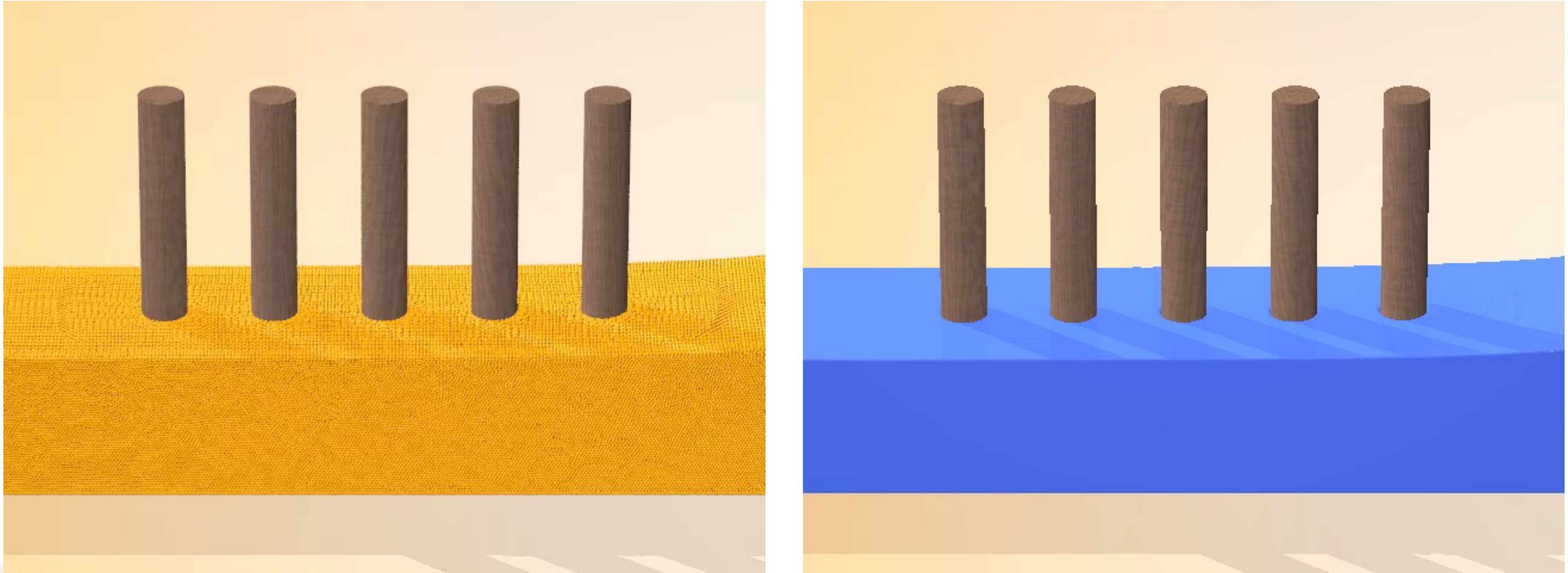
Prof. Tiberiu Popa

Fluid Animation

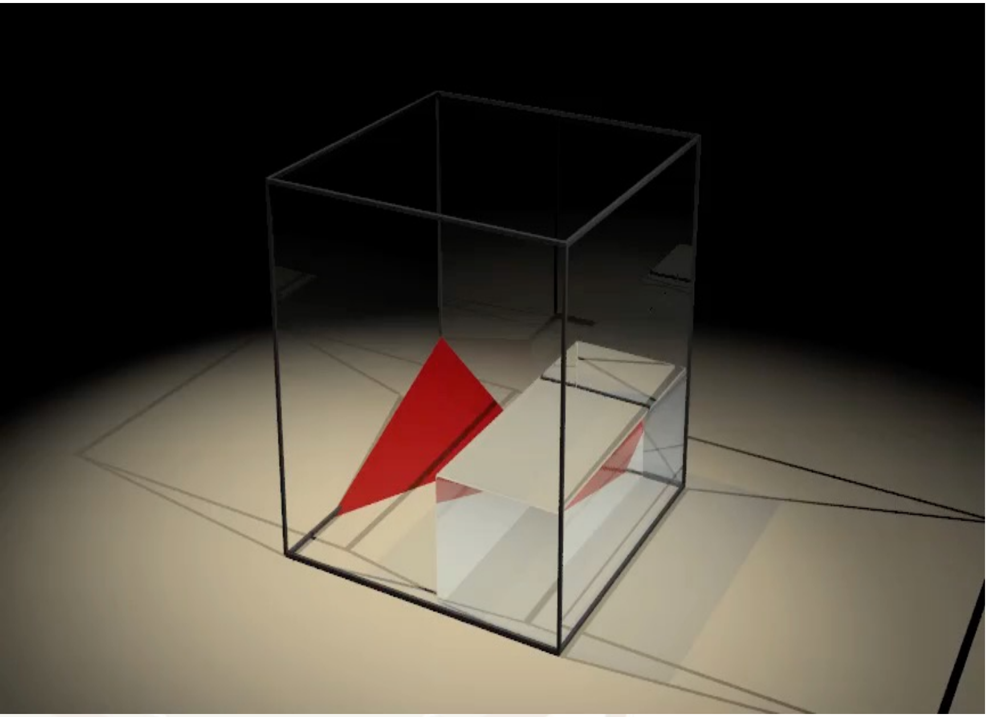
Fluid Animation

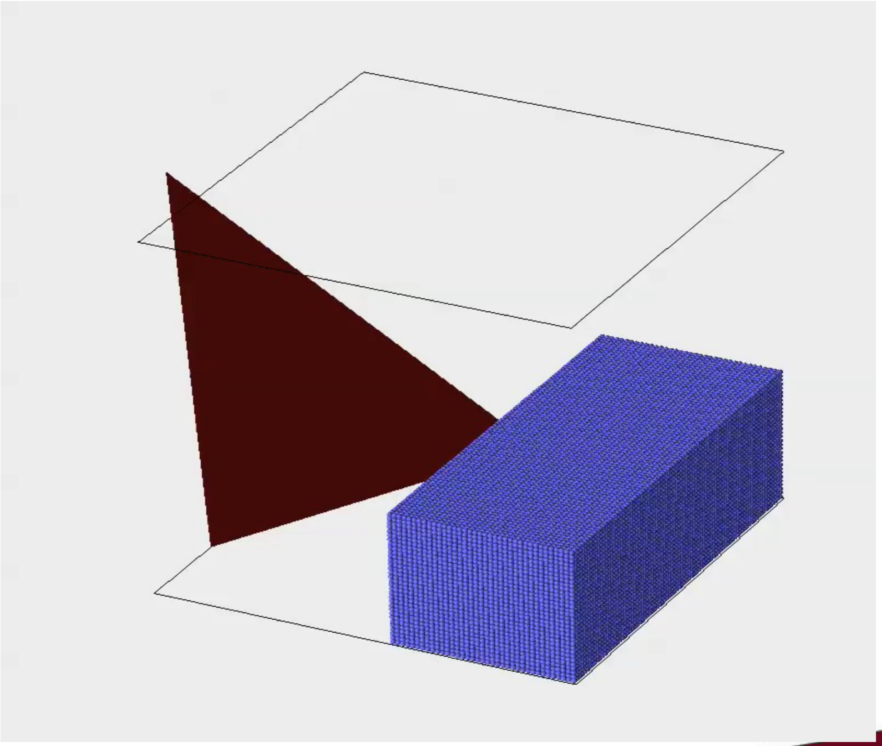


Fluid Animation



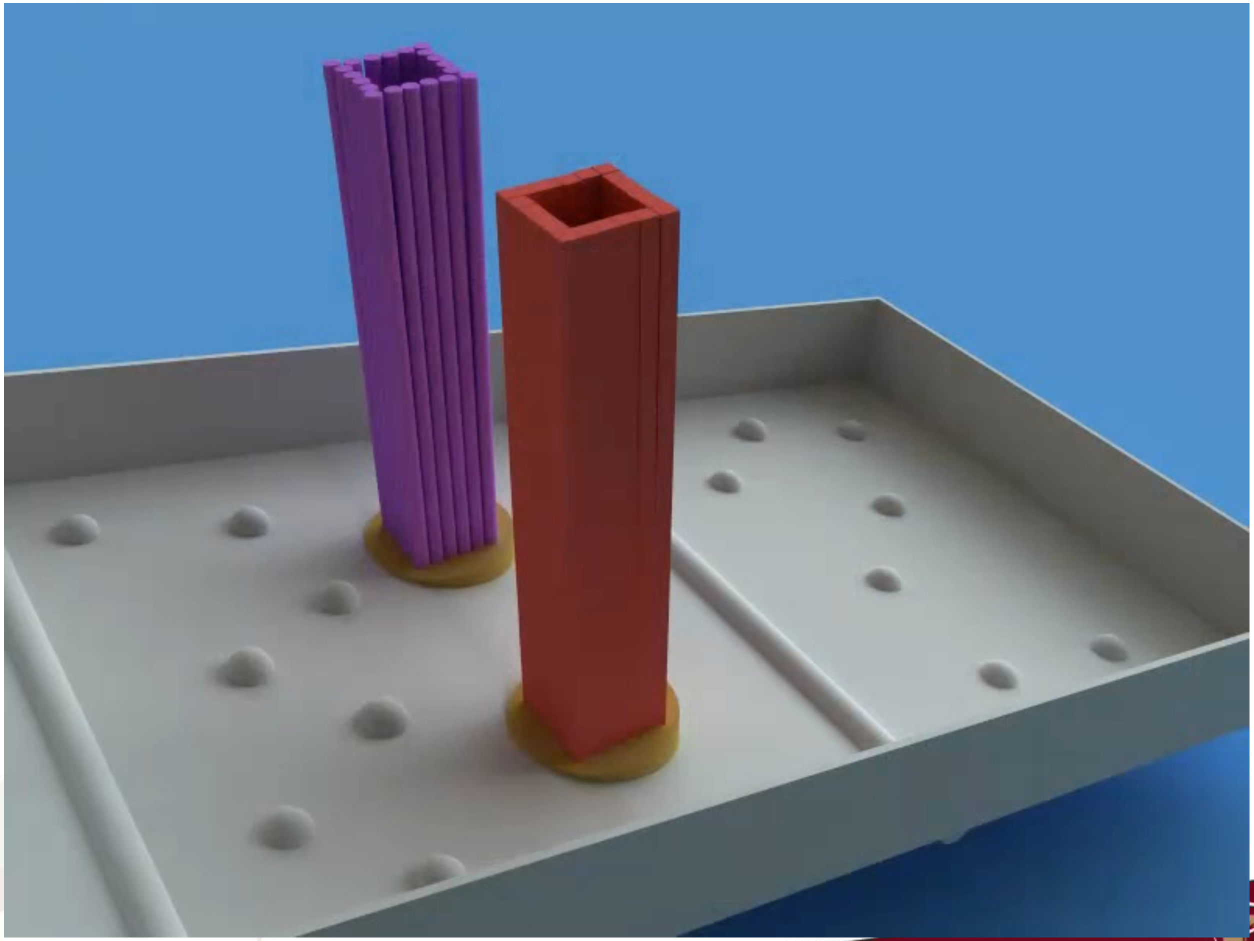
Particle Fluids





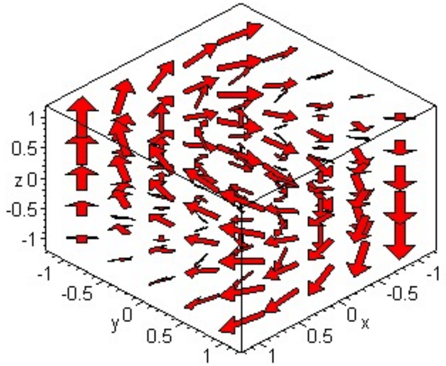
Université Concordia University

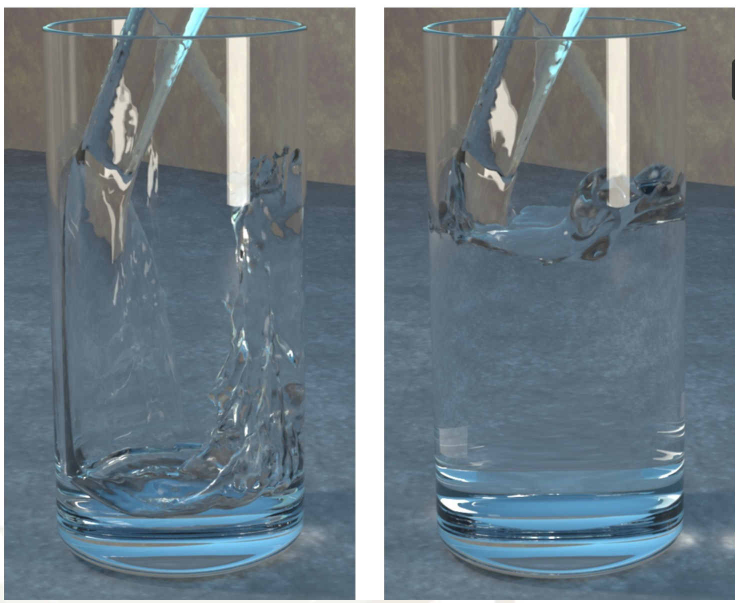
2.4M - Transparent Surface



Fluid Representation

- Eulerian view
  - Represent fluid as a 3D function
  - **What does this mean?**

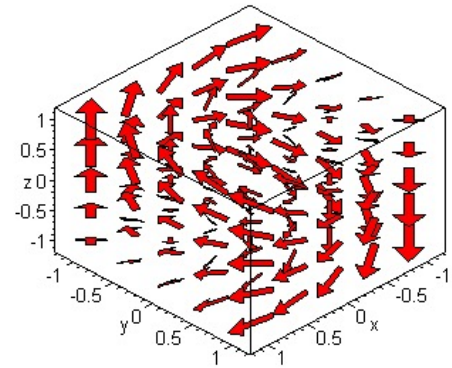


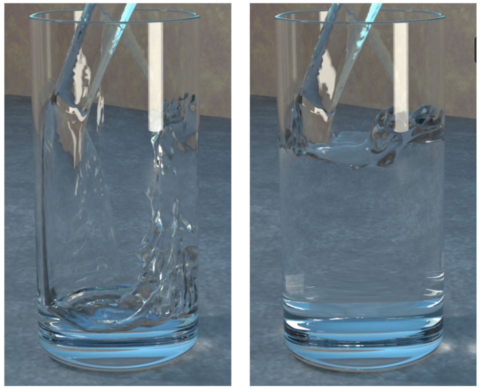


Enright, D., Marschner, S., & Fedkiw, R. (2002, July). Animation and rendering of complex water surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (pp. 736-744).

Fluid Representation

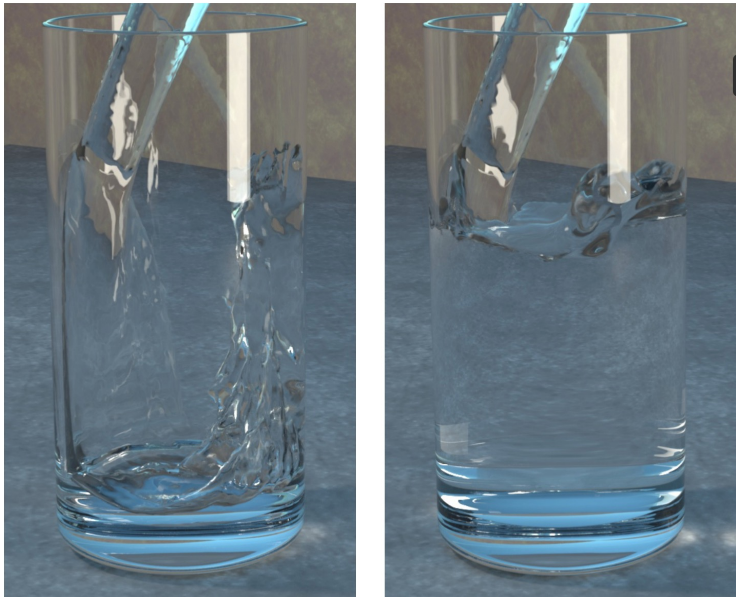
- Eulerian view
  - Represent fluid as a 3D function
  - **What does this mean?**
  - To render we need to know **where** is fluid and **where** does it move
  - Velocity $\vec{u}(x,t)$
  - $\phi(x,t)$→ a distance function that is <=0 where there is fluid, ==0 on the boundary between fluid and air or other object, >0 no fluid
  - $\phi(x,t)$ depends on $\vec{u}(x,t)$





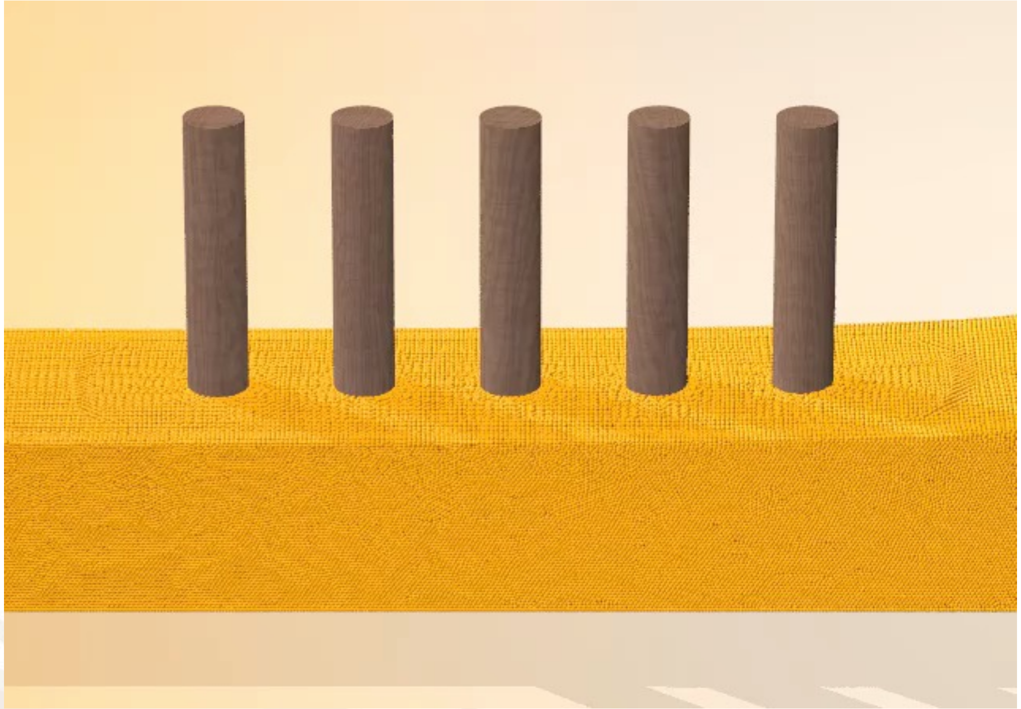
Fluid Representation

- Eulerian view
  - Represent fluid as a 3D function
  - **How do we render?**
  - Using ϕ
  - Volumetrically (Raytracing)
  - Extracting iso-surface (Level-set)



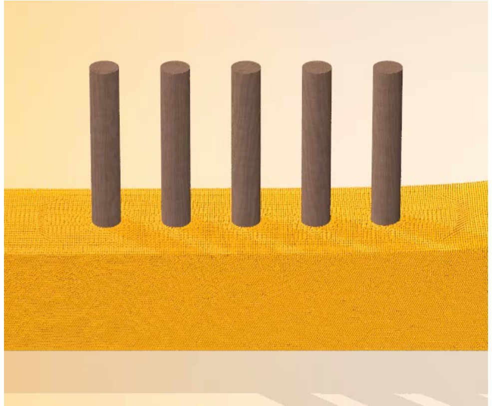
Fluid Representation

- Lagrangian view
  - Represent fluid explicitly as a set of particles



Fluid Representation

- Lagrangian view
  - Represent fluid explicitly as a set of particles
  - Compute position and velocity for each particle very similar to spring systems (different forces)
  - How do we render?
    - Raytracing
    - Extracting iso-surface



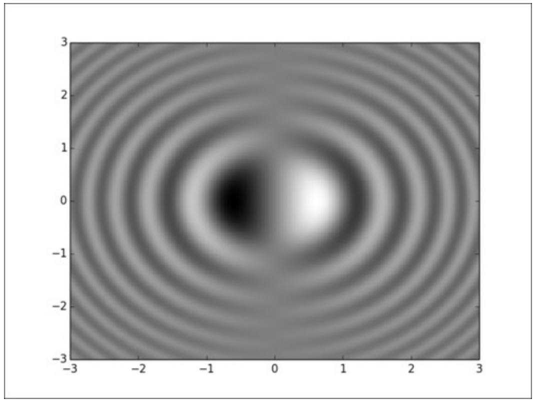
Internal Fluid Forces

$$\vec{F} = m\vec{a} \rightarrow \vec{a} \equiv \frac{D\vec{u}}{Dt}$$ - material derivative

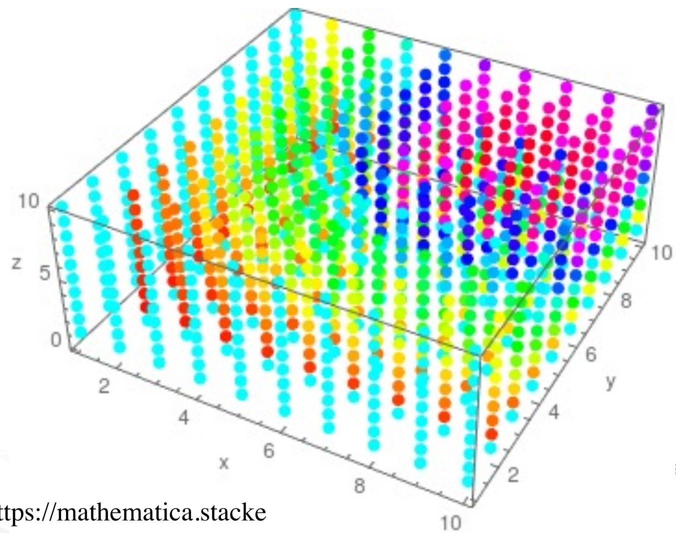
- Forces
  - Pressure forces
  - Viscosity forces
  - Surface tension forces
- These forces are expressed a differential operators
- A refresher on these

A little bit of Math

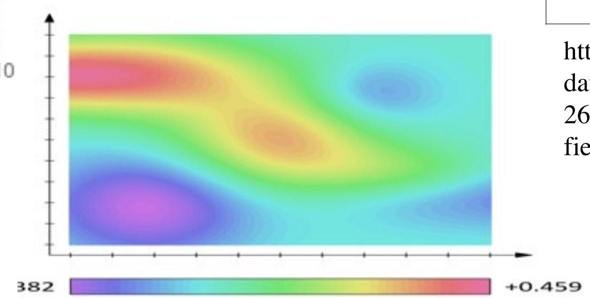
- Data sampled over compact domain (1D, 2D, 3D)
- $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$
- $f: \mathbb{R}^n \rightarrow \mathbb{R}$ - scalar field



https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781849513265/6/ch06lvl1sec68/visualizing-a-2d-scalar-field



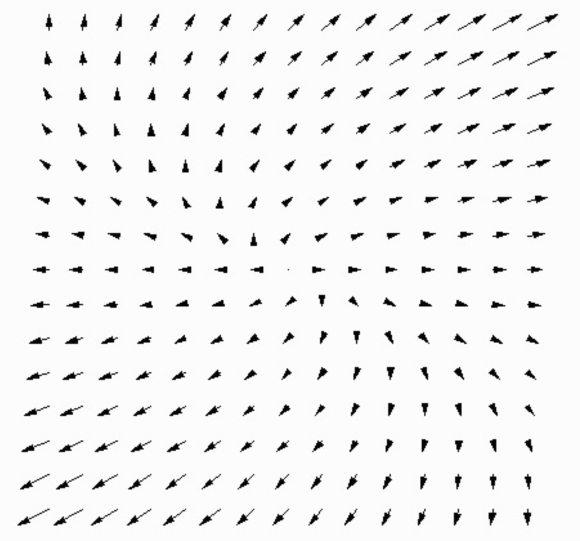
https://mathematica.stackexchange.com/questions/91233/visualize-scalar-field-fx-y-z-in-three-dimension

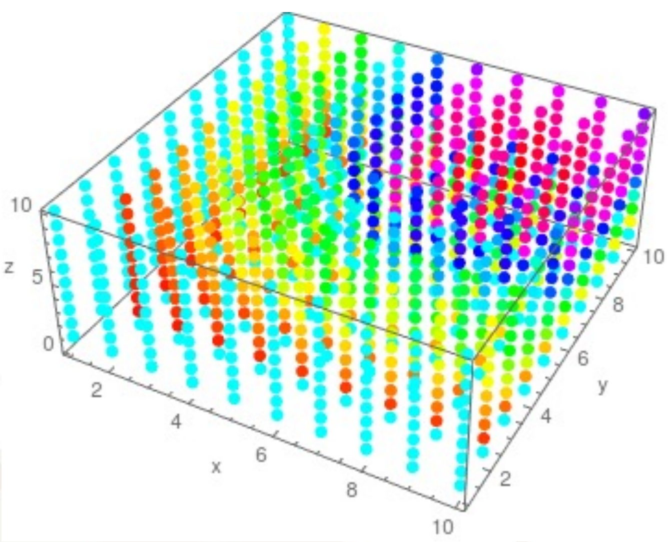


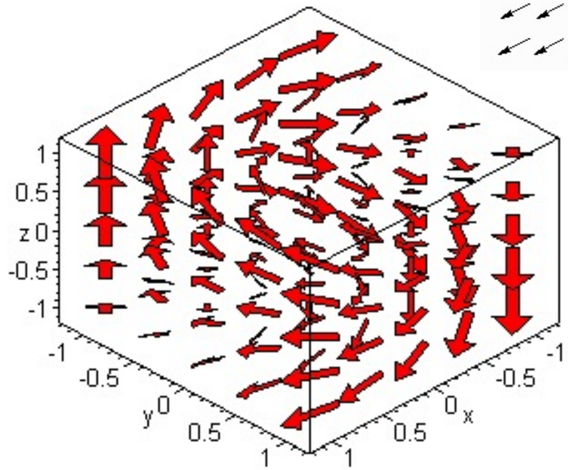
https://www.assignmentpoint.com/science/physics/scalar-field.html

A little bit of Math

- Data sampled over compact domain (1D, 2D, 3D)
- $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$
- $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$- vector field

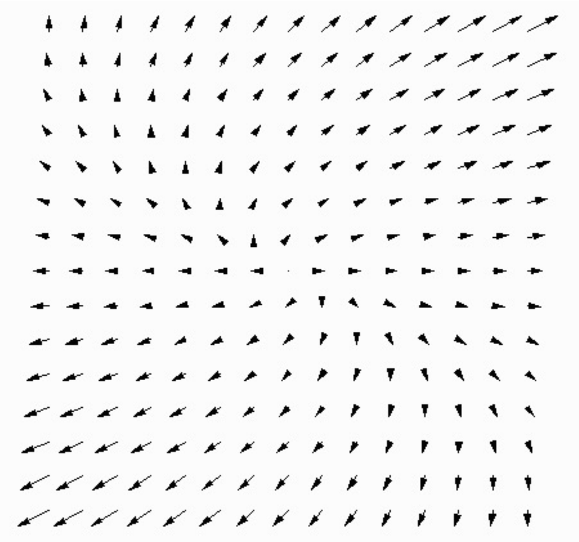


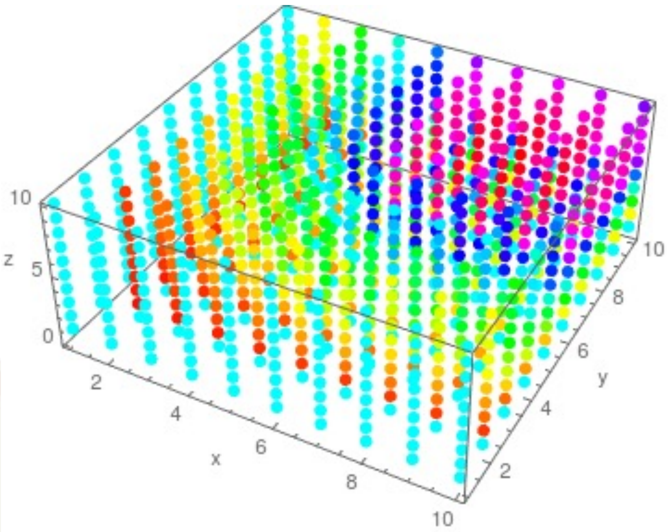


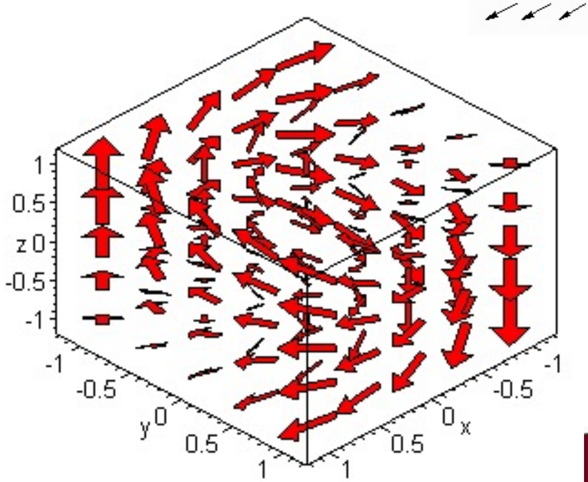


A little bit of Math

- Vector/Scalar field over compact domain (1D, 2D, 3D)
- Represent discretely
  - Grid (regular)
  - Grid (adaptive) (i.e. kd-trees, octrees, etc.)

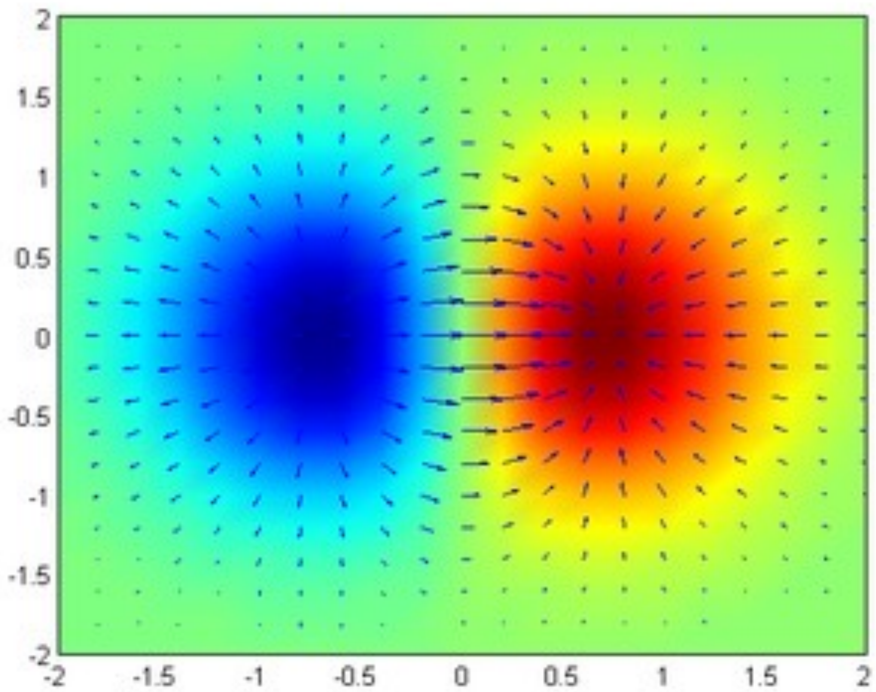






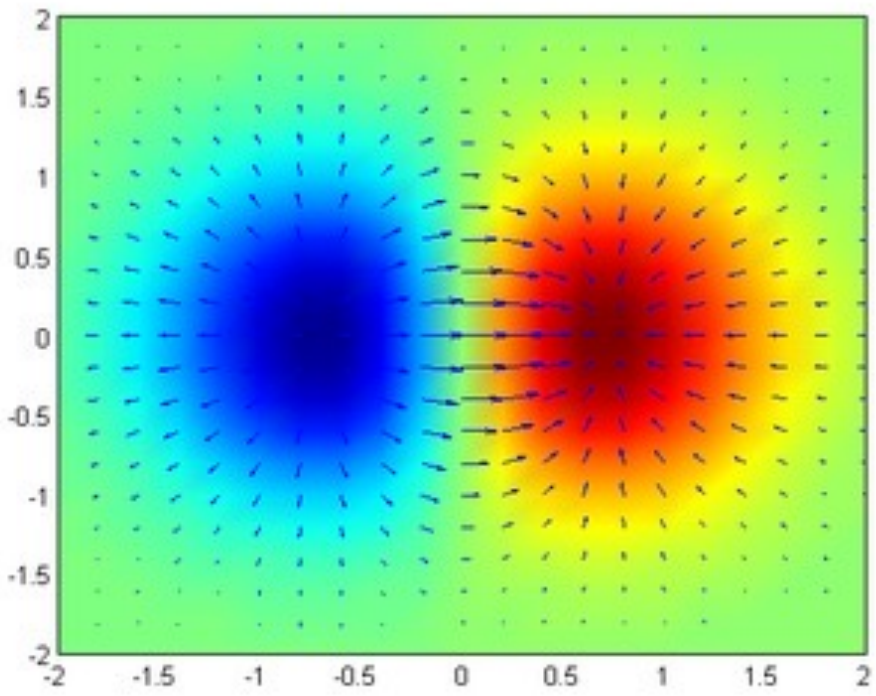
A little bit of Math

- Gradient
  - $f: \mathbb{R}^3 \rightarrow \mathbb{R}$
  - $\nabla f: \mathbb{R}^3 \rightarrow \mathbb{R}$
  - $\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$



A little bit of Math

- Discrete gradient
  - $f : \mathbb{R}^3 \rightarrow \mathbb{R}$
  - $\nabla f : \mathbb{R}^3 \rightarrow \mathbb{R}$
  - $\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$
  - Use finite differences



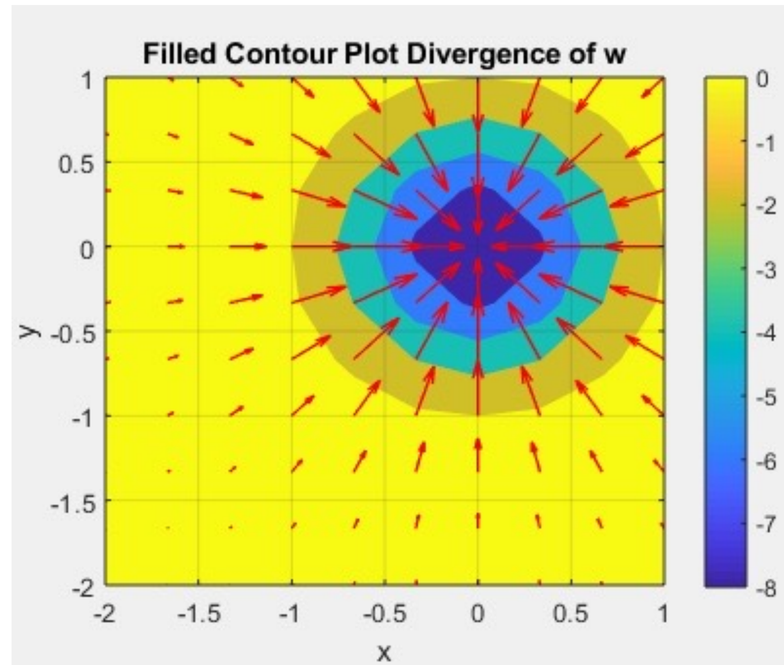
A little bit of Math

- Force expressed as gradient
  - p: $\mathbb{R}^3 \rightarrow \mathbb{R}$
  - ∇p: $\mathbb{R}^3 \rightarrow \mathbb{R}$

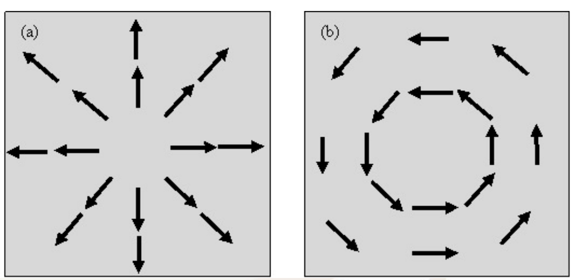
$$F_{pressure} = -\nabla p = -\begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$$

A little bit of Math

- Divergence: “flux density”—the amount of flux entering or leaving a point
  - $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$
  - $\nabla \cdot f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$
  - $\nabla \cdot f(x) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} + \frac{\partial f}{\partial z}$



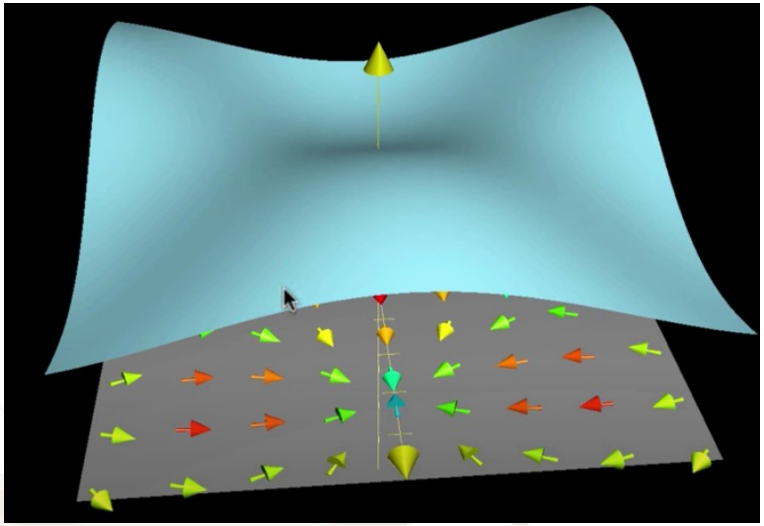
Roche de Guzman (2020). Graph of a 2D vector field equation and its divergence (https://www.mathworks.com/matlabcentral/fileexchange/69287-graph-of-a-2d-vector-field-equation-and-its-divergence), MATLAB Central File Exchange. Retrieved October 19, 2020.

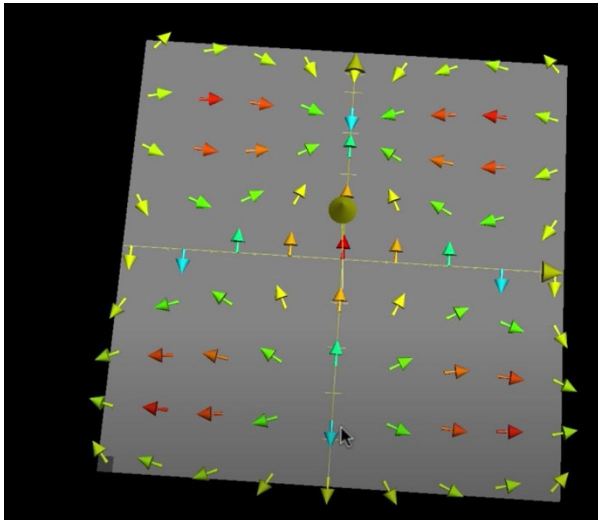


Incompressibility condition

A little bit of Math

- Laplacian
  - $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ $\quad \Delta f = \nabla^2 f = \nabla \cdot \nabla f: \mathbb{R}^3 \rightarrow \mathbb{R}$
  - $\Delta f = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2} + \dfrac{\partial^2 f}{\partial z^2}$

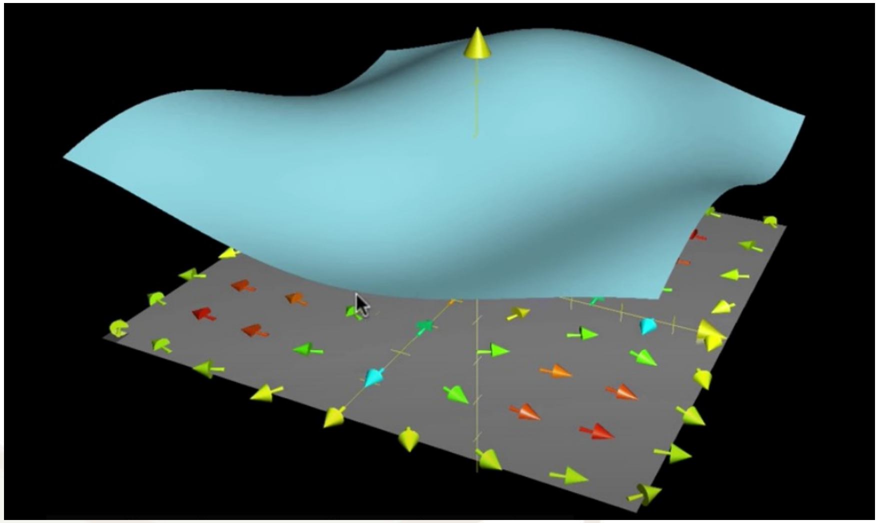


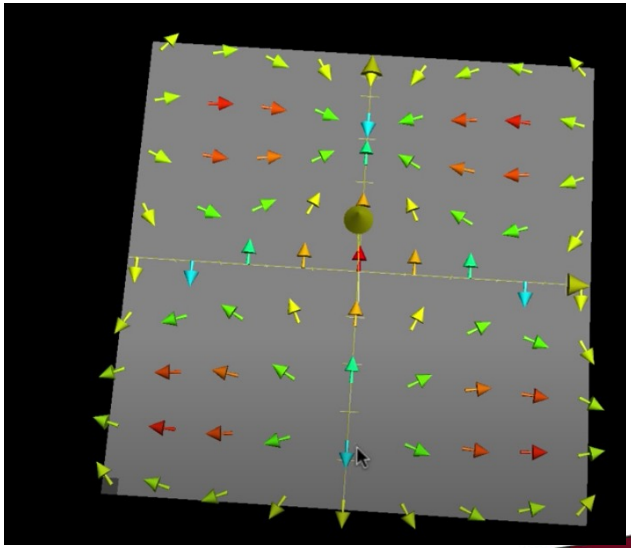


https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/laplacian/v/laplacian-intuition

A little bit of Math

- Laplacian
  - $f: \mathbb{R}^3 \to \mathbb{R}$ $\qquad \Delta f = \nabla^2 f = \nabla \cdot \nabla f: \mathbb{R}^3 \to \mathbb{R}$
  - $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$





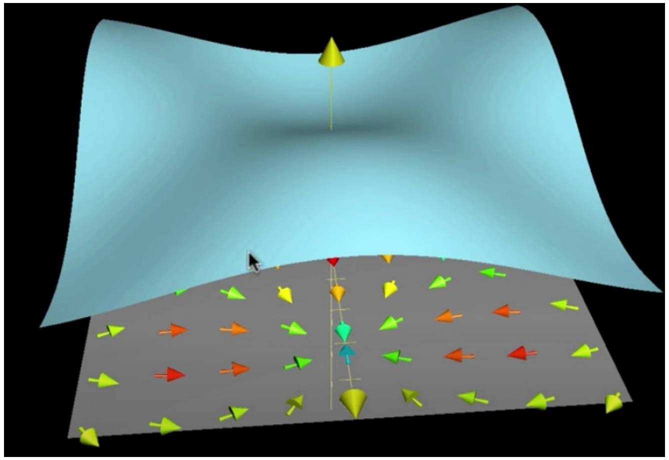
https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/laplacian/v/laplacian-intuition

A little bit of Math

- Laplacian
  - $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ $\quad \Delta f = \nabla^2 f = \nabla \cdot \nabla f: \mathbb{R}^3 \rightarrow \mathbb{R}$
  - $\Delta f = \dfrac{\partial^2 f}{\partial x^2} + \dfrac{\partial^2 f}{\partial y^2} + \dfrac{\partial^2 f}{\partial z^2}$

A little bit of Math

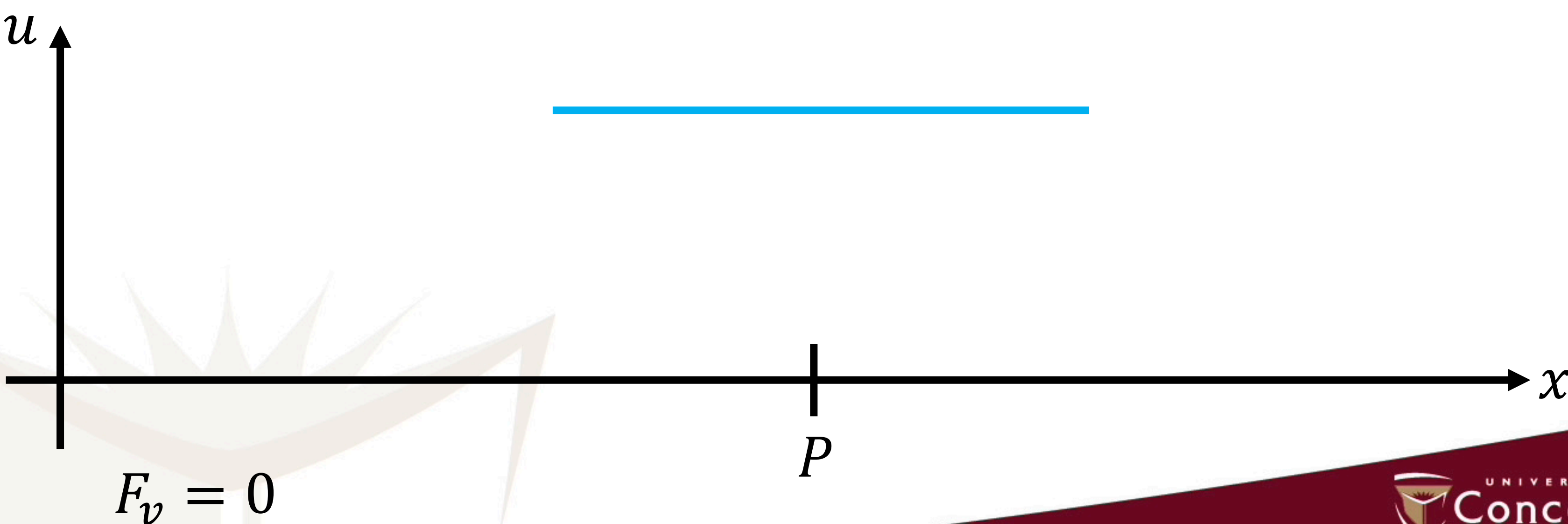
- Discrete Laplacian
  - $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$
  - $\Delta f \begin{pmatrix} x \\ y \\ z \end{pmatrix} \approx \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} x_i - x \\ \frac{1}{n}\sum_{i=1}^{n} y_i - y \\ \frac{1}{n}\sum_{i=1}^{n} z_i - z \end{pmatrix}$



https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/laplacian/v/laplacian-intuition

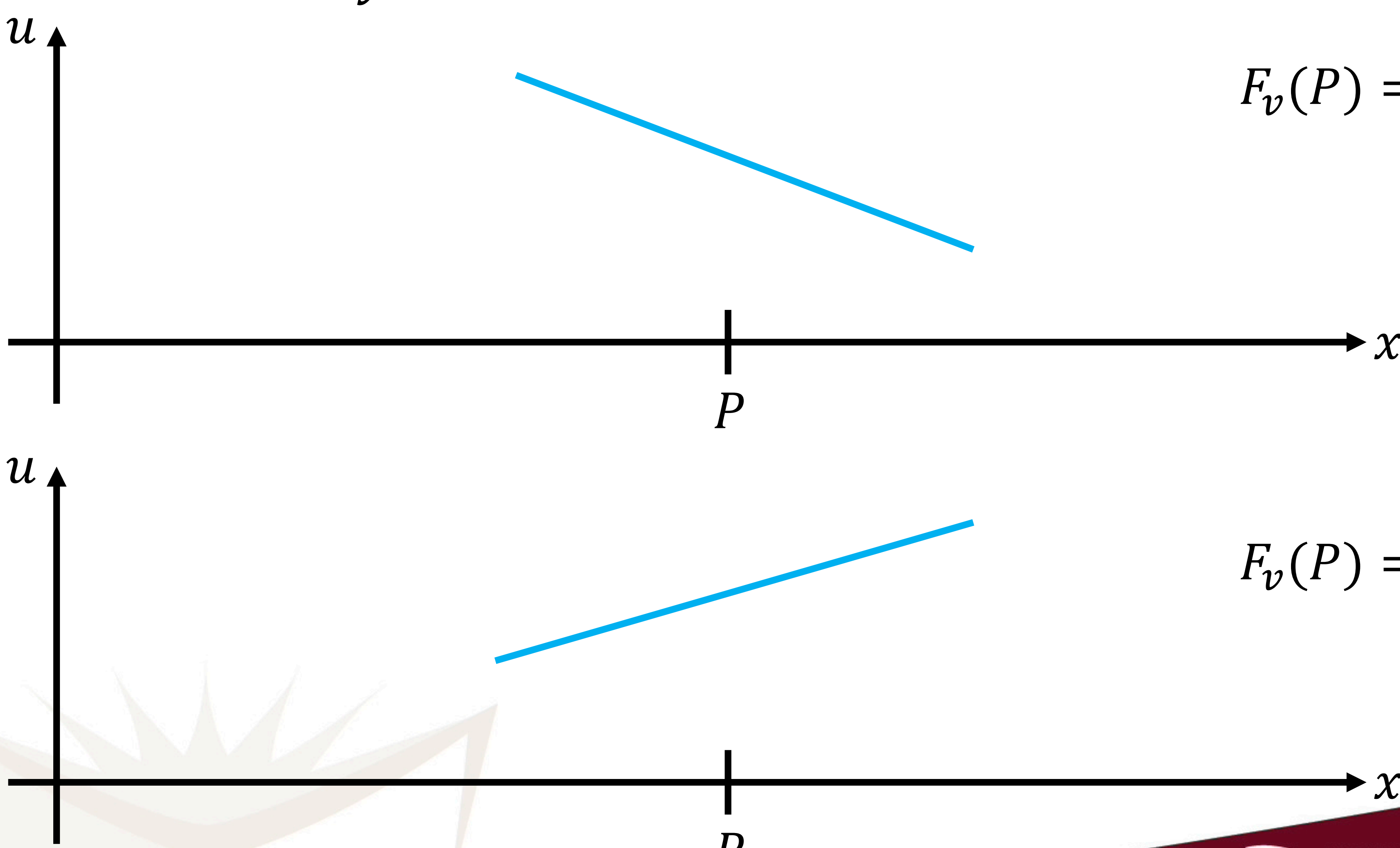
A little bit of Physics

- Force expressed as Laplacian
- Viscosity (friction)
  - $u: \mathbb{R}^3 \rightarrow \mathbb{R}^3$
  - $F_{viscosity} = F_v = \mu \nabla \cdot \nabla u: \mathbb{R}^3 \rightarrow \mathbb{R}^3$



A little bit of Physics

- $F_{viscosity} = F_v = \mu \nabla \cdot \nabla u : \mathbb{R}^3 \rightarrow \mathbb{R}^3$

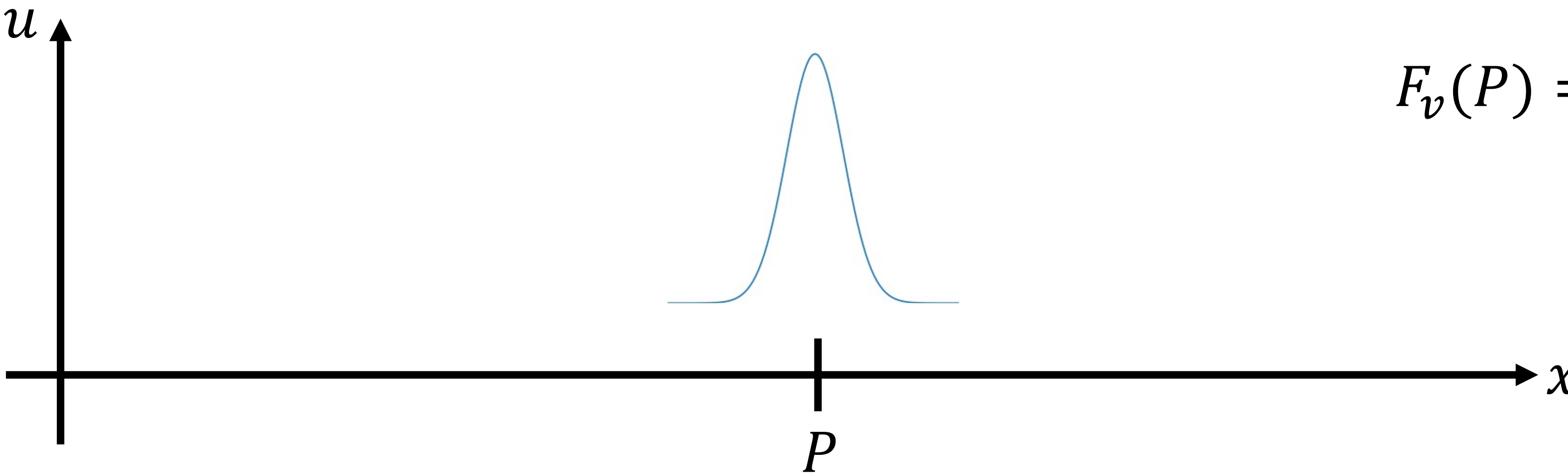


$F_v(P) =?$

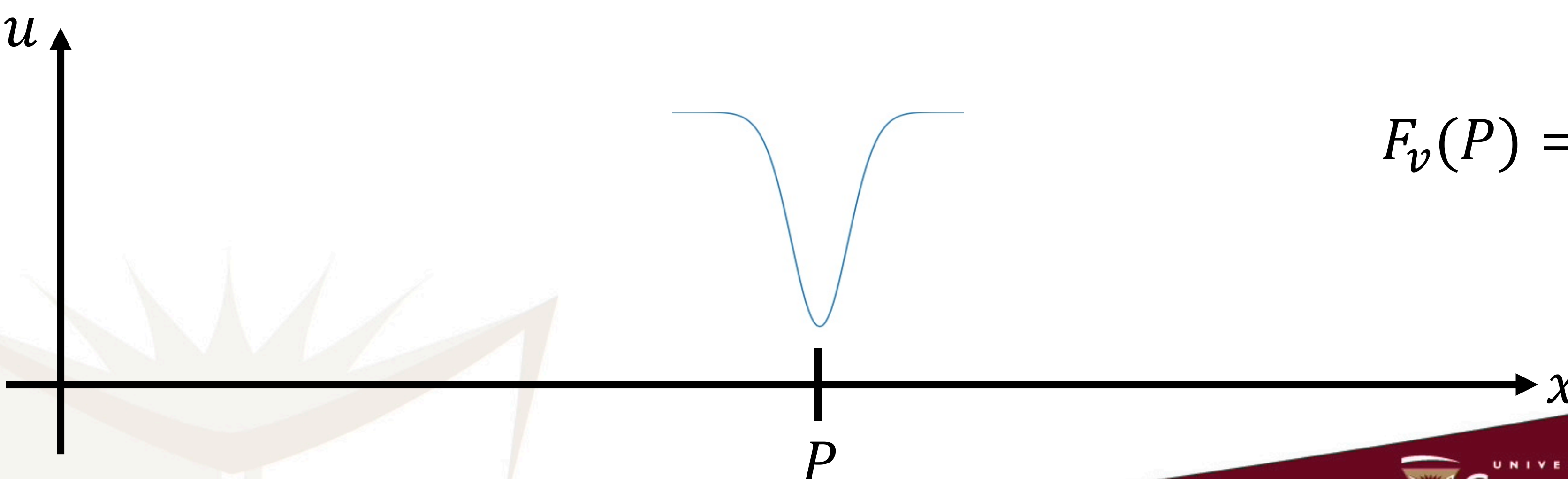
$F_v(P) =?$

A little bit of Physics

- $F_{viscosity} = F_v = \mu\nabla \cdot \nabla u \colon \mathbb{R}^3 \rightarrow \mathbb{R}^3$



$F_v(P) = ?$



$F_v(P) = ?$

Material Derivative

- $\vec{F} = m\vec{a}$ → $\vec{a} = \dfrac{D\vec{u}}{Dt}$ - material derivative

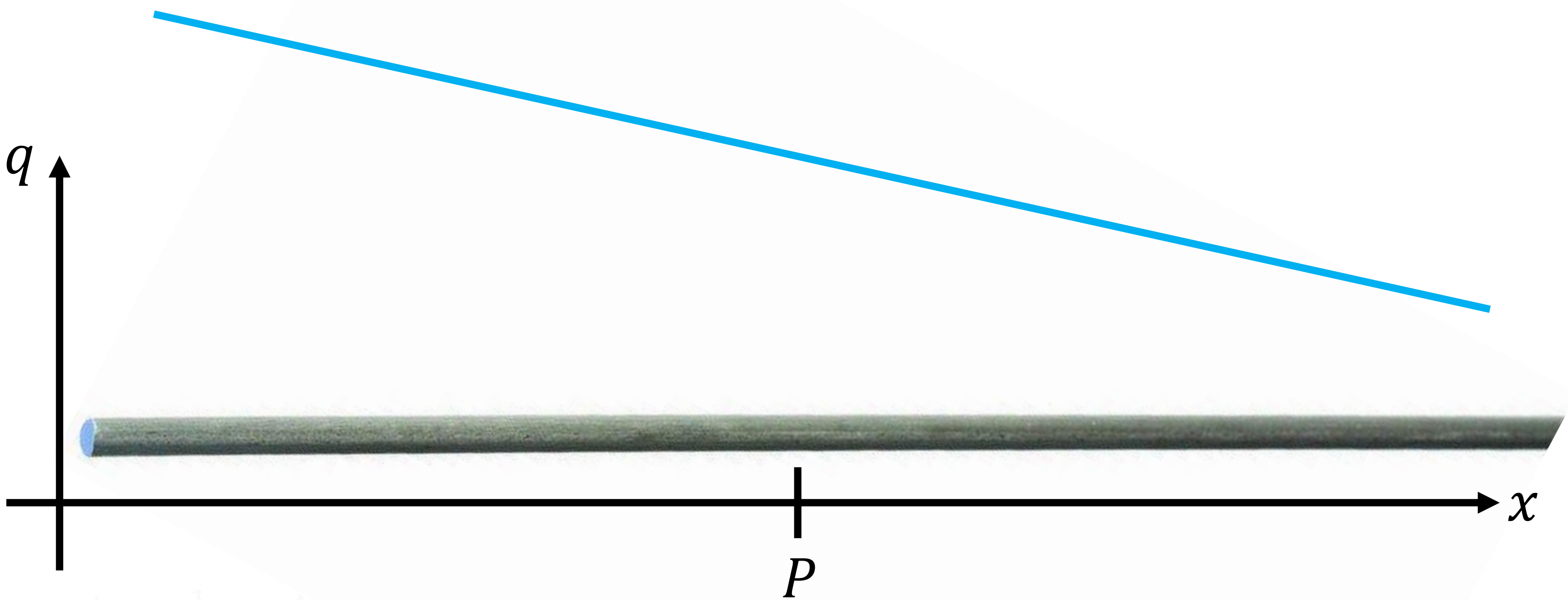
$\frac{d}{dt}q(t,\vec{x}) =?$ A good example to illustrate is temperature

$\frac{d}{dt}T(t,\vec{x}) =?$ What is the variation in temperature over time at some location in the fluid?

Q1: What dimension has $\frac{d}{dt}T(t,\vec{x})$ independent on the spatial dimension? A: one number → temperature is scalar and the derivative is w.r.t. to one variable t

A little bit of Physics

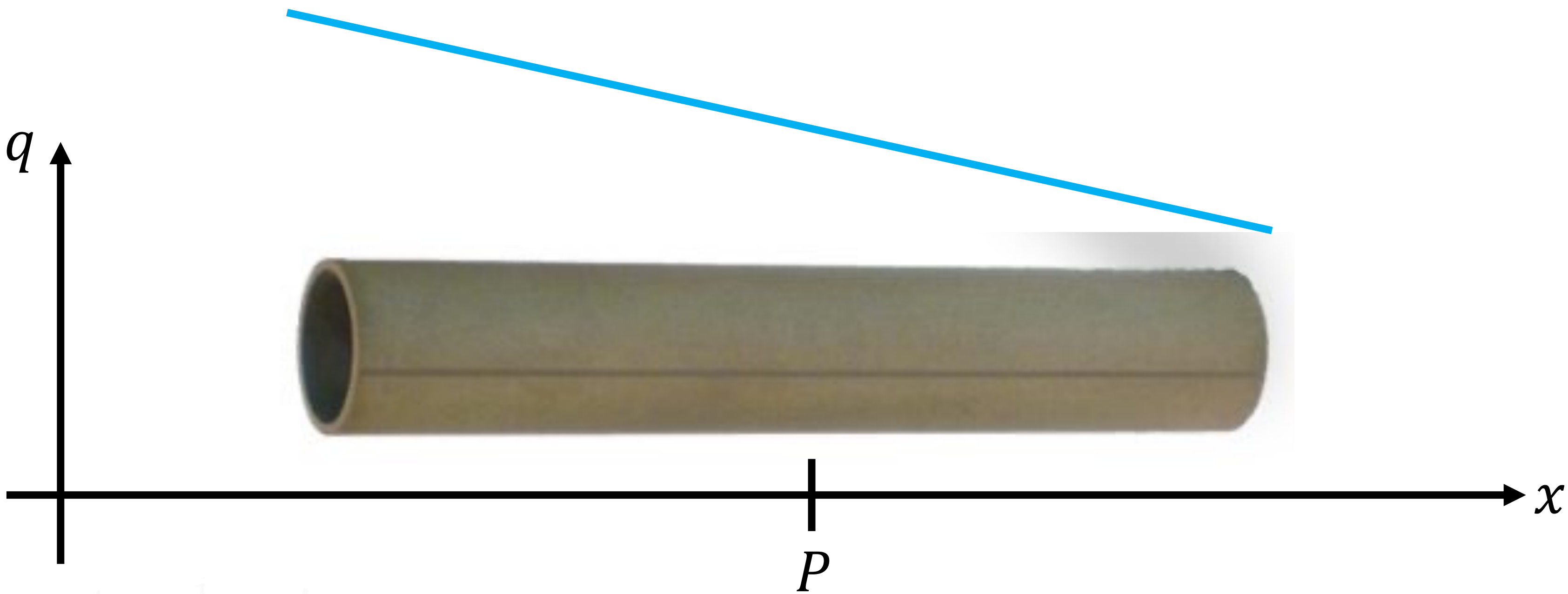
- Material derivative



$\frac{d}{dt}q(t,\vec{x}) = \frac{\partial q}{\partial t} \rightarrow$ because metal particles do not move $\rightarrow$ x does not depend on t

A little bit of Physics

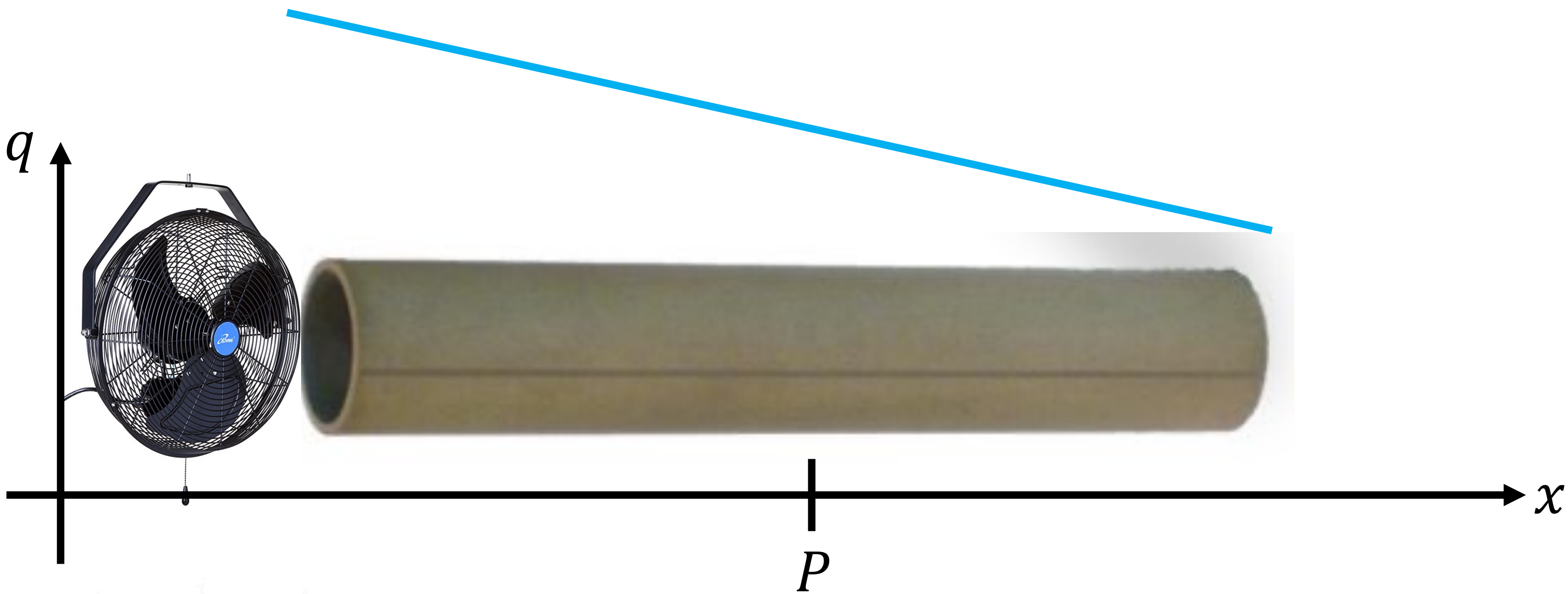
- Material derivative



$\frac{d}{dt} q(t, \vec{x}) = \frac{\partial q}{\partial t}$ → if no wind because air does not move inside tube → x does not depend on t

A little bit of Physics

- Material derivative



$\frac{d}{dt}q(t,\vec{x}) \neq \frac{\partial q}{\partial t}$ → wind moves air particles → x(t) non-constant
→ heat moves faster through the tube → convection

A little bit of Physics

- Material derivative:

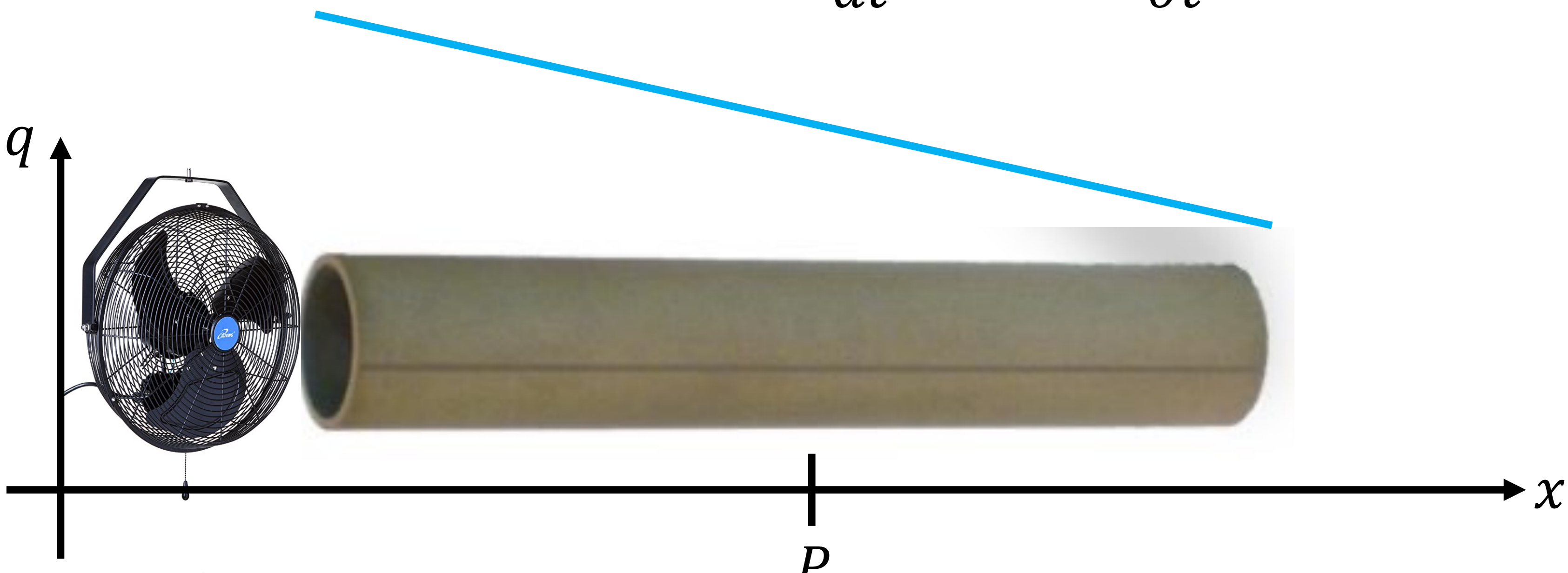
$$\frac{d}{dt}q(t,\vec{x}) = \frac{\partial q}{\partial t} + \frac{\partial q}{\partial x}\frac{dx}{dt} + \frac{\partial q}{\partial y}\frac{dy}{dt} + \frac{\partial q}{\partial z}\frac{dz}{dt} = \frac{\partial q}{\partial t} + \vec{u}\cdot\nabla q = \frac{Dq}{Dt}$$

$$\frac{d}{dt}u(t,\vec{x}) = \frac{\partial u}{\partial t} + \vec{u}\cdot\nabla u = \frac{Du}{Dt}$$

A little bit of Physics

- Material derivative

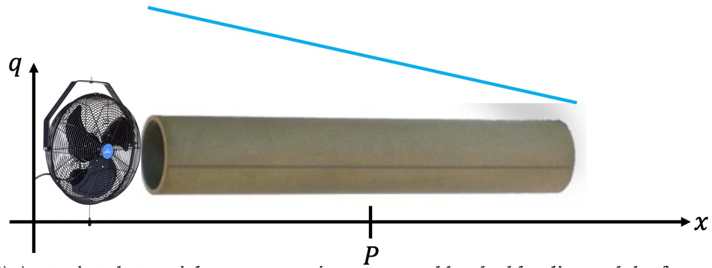
$$\frac{d}{dt}q(t,\vec{x}) = \frac{\partial q}{\partial t} + \vec{u}\cdot\nabla q$$



Q2) Assuming the spatial temperature is represented by the blue line and the fan on the left pushes the air particles away from the fan with some constant velocity, will the temperature a blob of air initially located at P increase faster or slower than if we turn off the fan? (assume $\frac{\partial q}{\partial t}$ at P >0) Justify both intuitively and formally.

A little bit of Physics

- Material derivative



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

Q2) Assuming the spatial temperature is represented by the blue line and the fan on the left pushes the air particles away from the fan with some constant velocity, will the temperature a blob of air initially located at P increase faster or slower than if we turn off the fan? (assume $\frac{\partial q}{\partial t}$ at P >0) Justify both intuitively and formally.

A2) Slower or may even decrease because the second term is negative (in 1D) $\vec{u} > 0, \cdot \nabla q < 0$
Intuitively, if a blob of air moves to the right where the temperature is smaller, the overall increase in temperature will be smaller or might even decrease

A little bit of Physics

- Material derivative:

$$\frac{d}{dt}q(t,\vec{x}) = \frac{\partial q}{\partial t} + \vec{u}\cdot\nabla q = \frac{Dq}{Dt}$$

Material derivative is the change in temperature at time t of a infinitesimally small blob of fluid located at $\vec{x}$ (also at time t)

$\frac{\partial q}{\partial t}$ → represents the change in temperature at a fixed constant location $\vec{x}$ at time t

Navier-Stokes equations

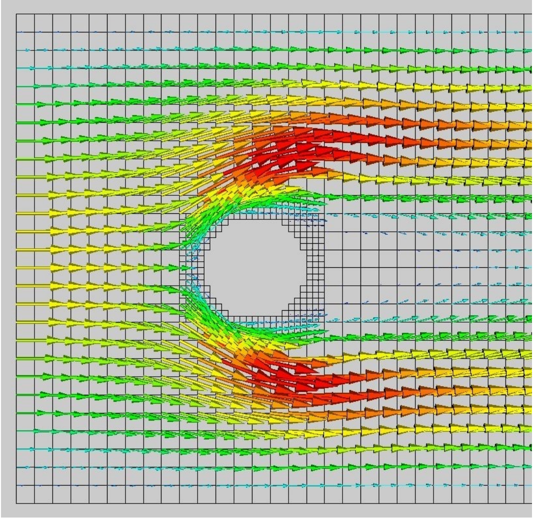
- Eulerian view

$$\vec{F} = m\vec{a} \Leftrightarrow m\vec{g} - V\nabla p + V\mu\nabla \cdot \nabla\vec{u} = \mathrm{m}(\frac{\partial\vec{u}}{\partial t} + \vec{u} \cdot \nabla u) \Leftrightarrow$$

$$\vec{g} - \frac{V}{m}\nabla p + \frac{V}{m}\mu\nabla \cdot \nabla\vec{u} = \frac{\partial\vec{u}}{\partial t} + \vec{u} \cdot \nabla u \Leftrightarrow$$

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

$\nabla \cdot \vec{u} = 0$ (incompressibility conditions)



Represent all functions on a grid

All differential operators can be estimated reliably and easily

Q3) How do we compute pressure?

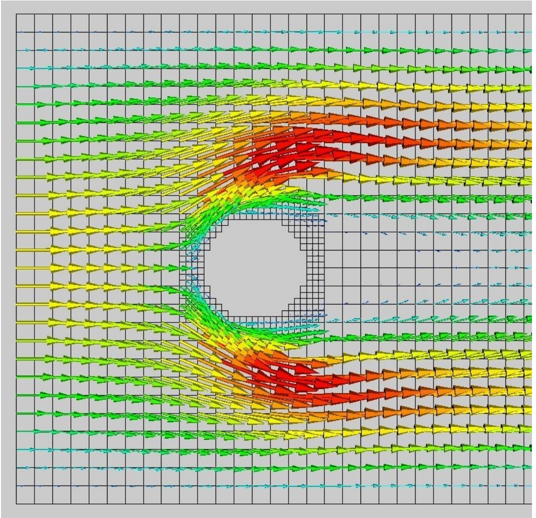
Foster, N., & Fedkiw, R. (2001, August). Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (pp. 23-30).
http://physbam.stanford.edu/~fedkiw/papers/stanford2001-02.pdf

Navier-Stokes equations

- Eulerian view

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

$\nabla \cdot \vec{u} = 0$ (incompressibility conditions)



Q3) How do we compute pressure?

Use incompressibility equation

Need another equation connecting p and u

$\Delta p = \rho\nabla \cdot \vec{u}/\Delta t$

This can be used to solve for pressure (linear system)

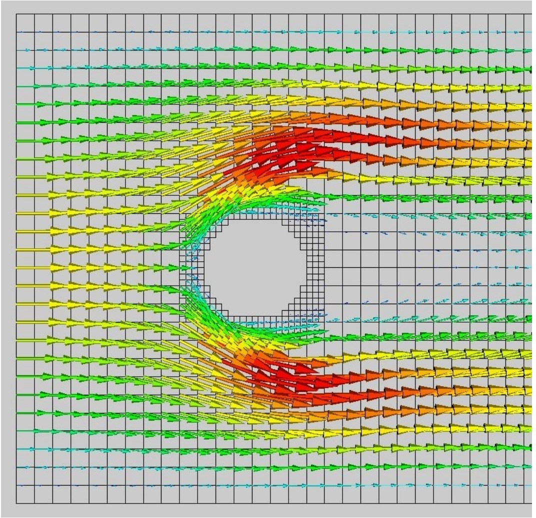
Navier-Stokes equations

- Eulerian view

$$\frac{\partial \vec{u}}{\partial t} = \vec{g} - \frac{1}{\rho}\nabla p + \frac{1}{\rho}\mu\nabla\cdot\nabla\vec{u} - \vec{u}\cdot\nabla\vec{u} \quad (1)$$

$\nabla\cdot\vec{u} = 0$ (incompressibility conditions)

$$\Delta p = \rho\nabla\cdot\vec{u}/\Delta t \quad (2)$$



1. Compute $\vec{u}_{t+1}$ from (1) wo pressure force
2. Solve (2) (linear system)
3. Update $\vec{u}_{t+1}$ with the pressure force

Navier-Stokes equations

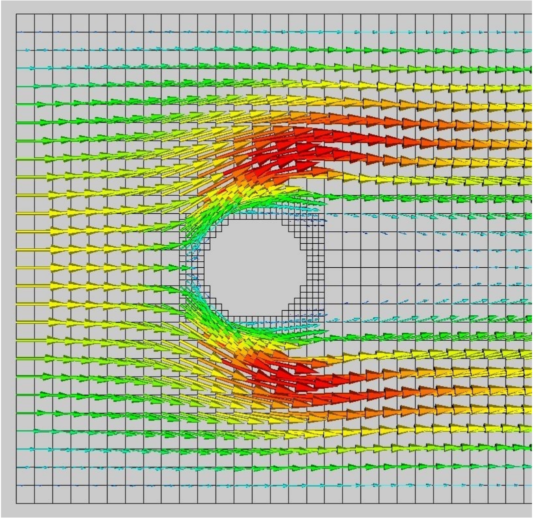
- Eulerian view

Pros:

- Simple(-ish)
- Efficient (grids are computationally fast

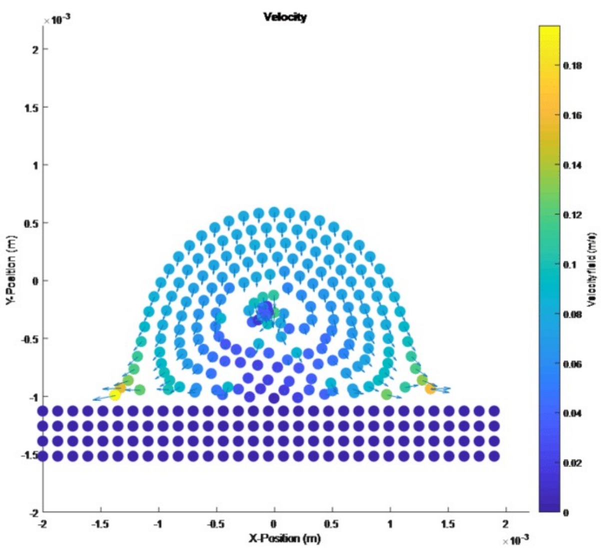
Cons:

- Difficult to model droplets
- Difficult to model multiple fluid densities



Navier-Stokes equations

- Lagrangian view:
  - Use explicit particles to represent the fluid

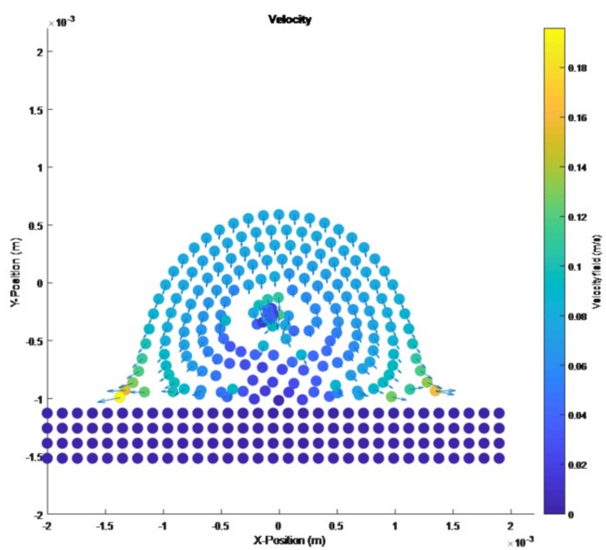


https://www.researchgate.net/post/Why_particles_clustering_in_the_center

Navier-Stokes equations

- Lagrangian view: good news and bad news

$$\frac{\partial u}{\partial t} + \cancel{\vec{u}\cdot\nabla u} + \frac{1}{\rho}\nabla p = \vec{g} + \nu\nabla\cdot\nabla\vec{u}$$
$$\cancel{\nabla\cdot\vec{u} = 0}$$



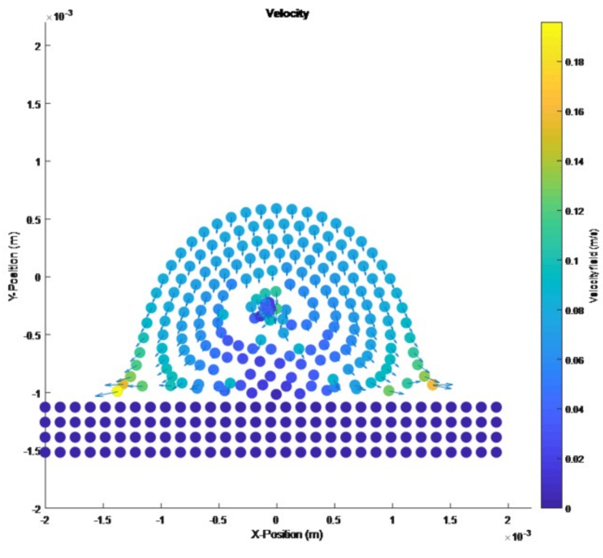
- We can remove some terms
- Do not know how to compute operators

https://www.researchgate.net/post/Why_particles_clustering_in_the_center

Simulation

- Input: discretization of particles, initial state (position/velocity)
- Output: per particle velocity and position

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

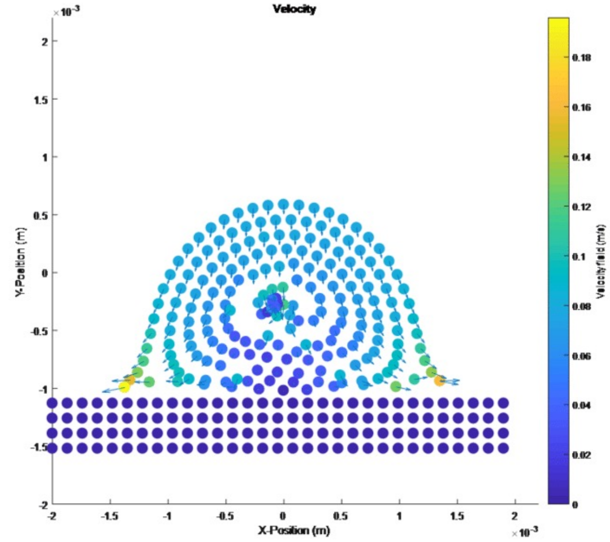


Simulation

- Input: discretization of particles, initial state (position/velocity)
- Output: per particle velocity and position

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

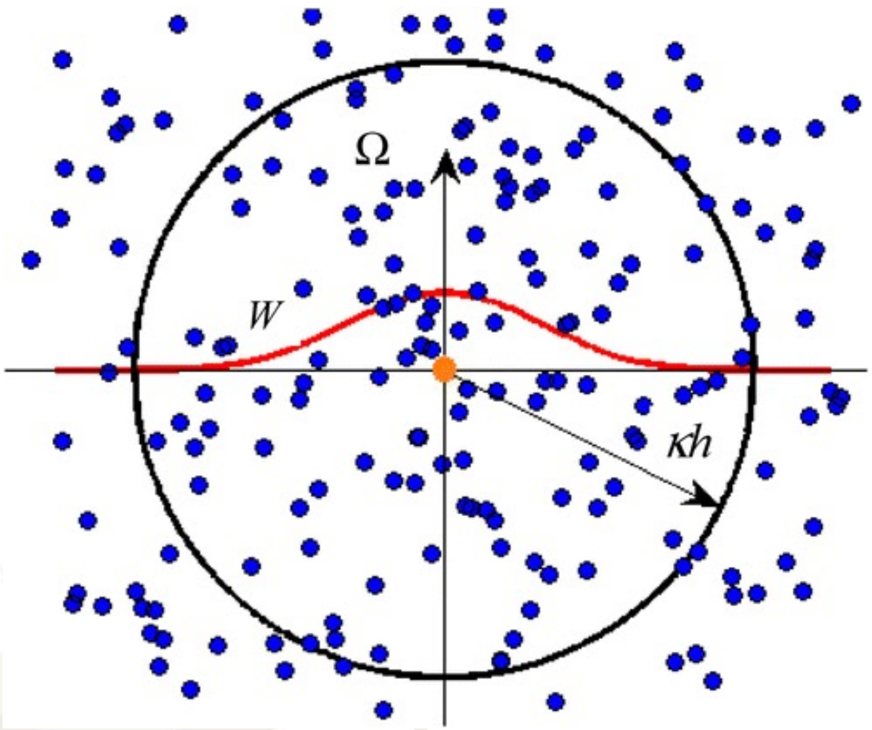
- Particles are
  - NOT Molecules, atoms, etc.
  - Volume of fluid
  - **Q4) No collision with each other, why?**



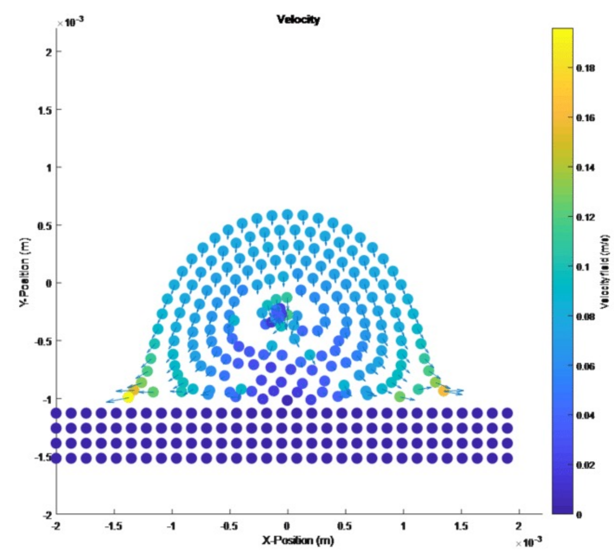
SPH

- How do we compute the differential operators?
- Smooth Particle Hydrodynamics

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

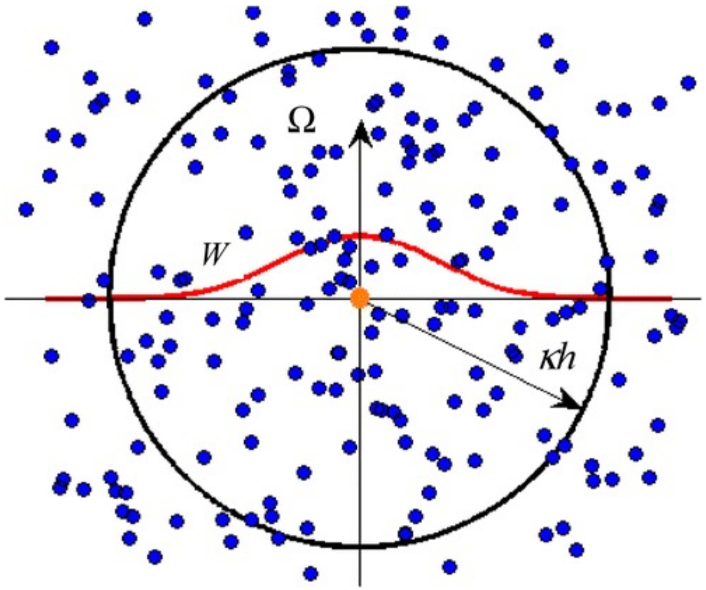


https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29GM.1943-5622.0000149



SPH

- How do we represent smooth functions $A(x)$?
  - We store values at the particles A_j
  - What is in-between points?
- Scattered data interpolation problem
- Need to find the spatial derivatives of these functions
- Use a Kernel function (red) W with compact support to smooth the data

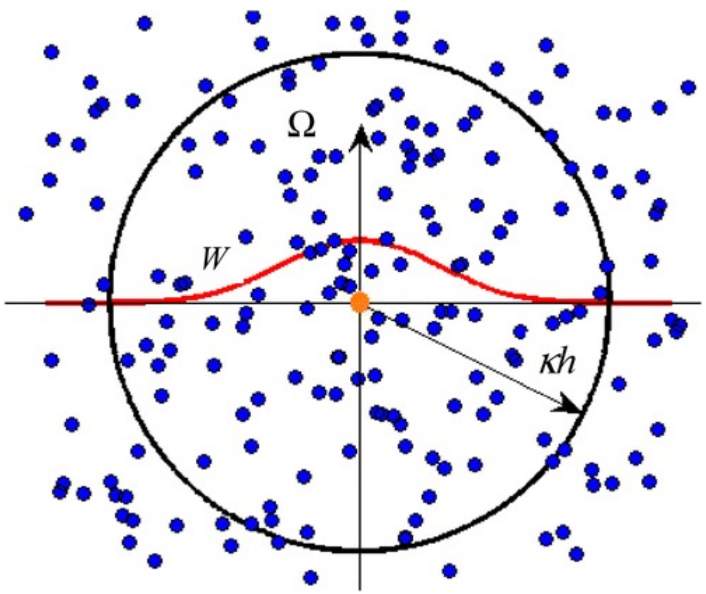


- $A(x) = \sum_j \frac{m_j}{\rho_j} A_j W(\|x_j - x\|\ h)$

- Q5) Is $A(x_j) = A_j$
- Q6) How about "h"

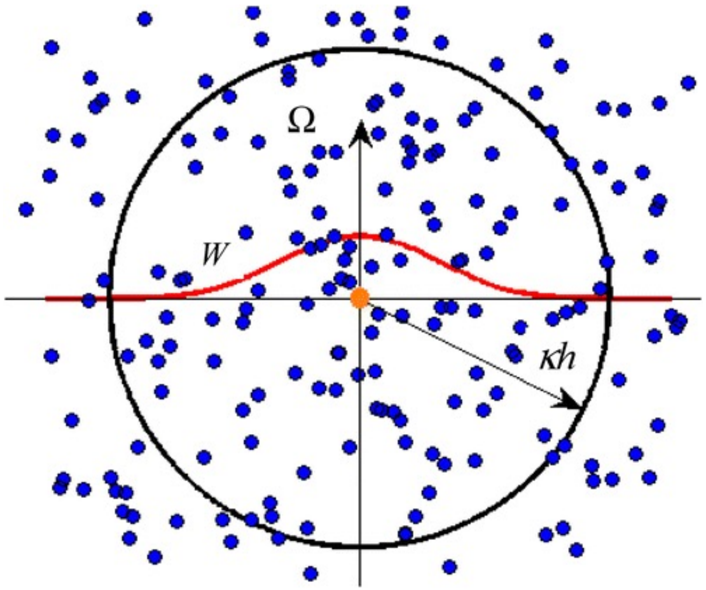
SPH

- $A(x) = \sum_j \frac{m_j}{\rho_j} A_j W(\|x_j - x\| \, h)$
- Q5) Is $A(x_j) = A_j$
- NO, hence the "smoothed"
- But $A(x)$ is infinitely differentiable with good kernel
- Q6) How about "h"
- h – kernel size → area to smooth
- The larger the h the smoother the function
- Too small h → does not include enough neighbors



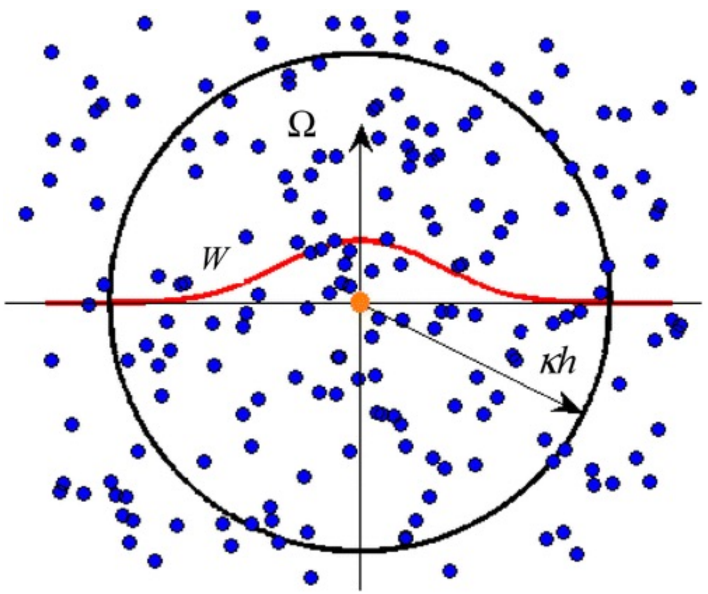
SPH

- $A(x) = \sum_j \frac{m_j}{\rho_j} A_j W(\|x_j - x\|, h)$
- Kernel $W(d, h)$
  - Distance function: depends ONLY on the distance to the center and h
  - Compact support: $W(d, h) = 0 \text{ iff d} \geq \text{h}$
  - $W(d, h) \geq 0$
  - $W(h, h) = 0$
  - $W(d, h)$ strictly monotonically decreasing $\forall\, d \leq h$
  - $W(d, h)$ differentiable everywhere
  - Common to scale $W\ s.t. W(0, h) = 1$
  - (not necessary)



SPH

- $A(x) = \sum_j \frac{m_j}{\rho_j} A_j W(\|x_j - x\|\ h)$
- What is ∇A?
- Everything except for $W(\|x_j - x\|\ h)$ are constants
- $\nabla A(x) = \sum_j \frac{m_j}{\rho_j} A_j \nabla W(\|x_j - x\|\ h)$
- $\Delta A(x) = \sum_j \frac{m_j}{\rho_j} A_j \Delta W(\|x_j - x\|\ h)$
- W analytical function → gradient, Laplacian can be computed analytically



SPH

$\rho \frac{\partial \vec{u}}{\partial t} = -\nabla p + \rho \vec{g} + \mu \nabla^2 \vec{u}$ (NS-equation)

$A(x) = \sum_j \frac{m_j}{\rho_j} A_j W(\|x_j - x\| \, h)$ (SPH)

Compute density: $\rho_i = \sum_j \frac{m_j}{\rho_j} \rho_j W(\|x_j - x\| \, h)$= $\sum_j m_j W(\|x_j - x\| \, h)$

Compute pressure – "spring" $p_i = k(\rho_i - \rho_0)$, ρ_0 - rest fluid density

SPH Algorithm

1. For all particles
    1. Find particles in the neighborhood at time t
2. For all particles
    1. Compute density at time t
    2. Compute pressure at time t
3. For all particles
    1. Compute right side of NS equation (i.e. Forces)

$$\rho \frac{\partial \vec{u}}{\partial t} = -\nabla p + \rho \vec{g} + \mu \nabla^2 \vec{u} \text{ (NS-equation)}$$

SPH Algorithm

4. For all particles

   1. Estimate the new position and velocity (time integration)
   2. Collision handling (with external objects ONLY)

SPH Algorithm

Questions:

7. Why not a huge for loop?

8. Why for the second step we can combine computing density and pressure?

9. Why the collision handling does not have to be in a separate loop?

10. How do we find the neighbors?

SPH Algorithm

Neighborhood computation:

- Normally trees perform well (i.e. kd-tree, oct-tree)
  - Construction time: O(nlog(n))
  - Look-up time: O(log(n))
  - Total: O(nlog(n)) assuming constant radius

SPH Algorithm

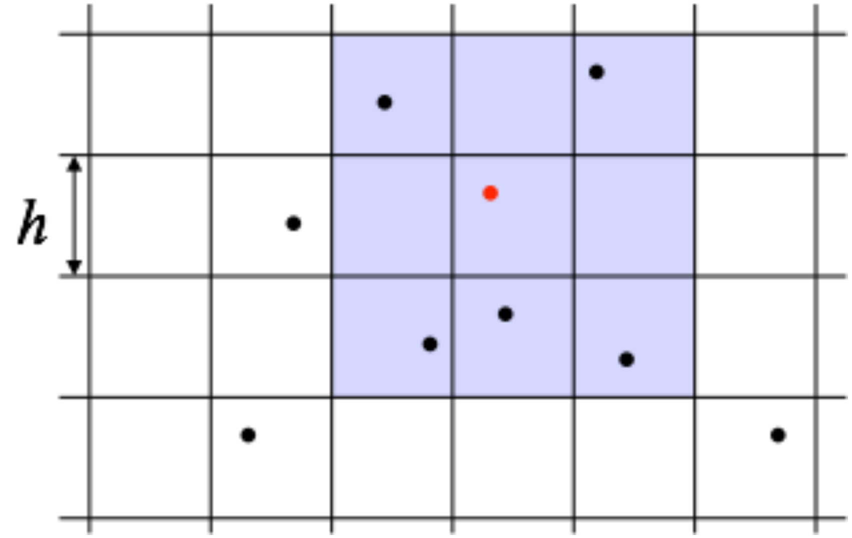
Neighborhood computation:

- Observations:
  - Need to do this at each time frame
  - Strictly speaking it is not a nearest neighbour search, but a radius search, and radius is constant!!!

SPH Algorithm

Neighborhood computation:

- Uniform grid:
  - Constructing: O(n)
  - Look-up: O(1)
- Faster!
- Why?



State of the Art

– Problem is in fact more complex and SPH has limitations

- Interface problems

- Assumes smooth data → not true at boundaries

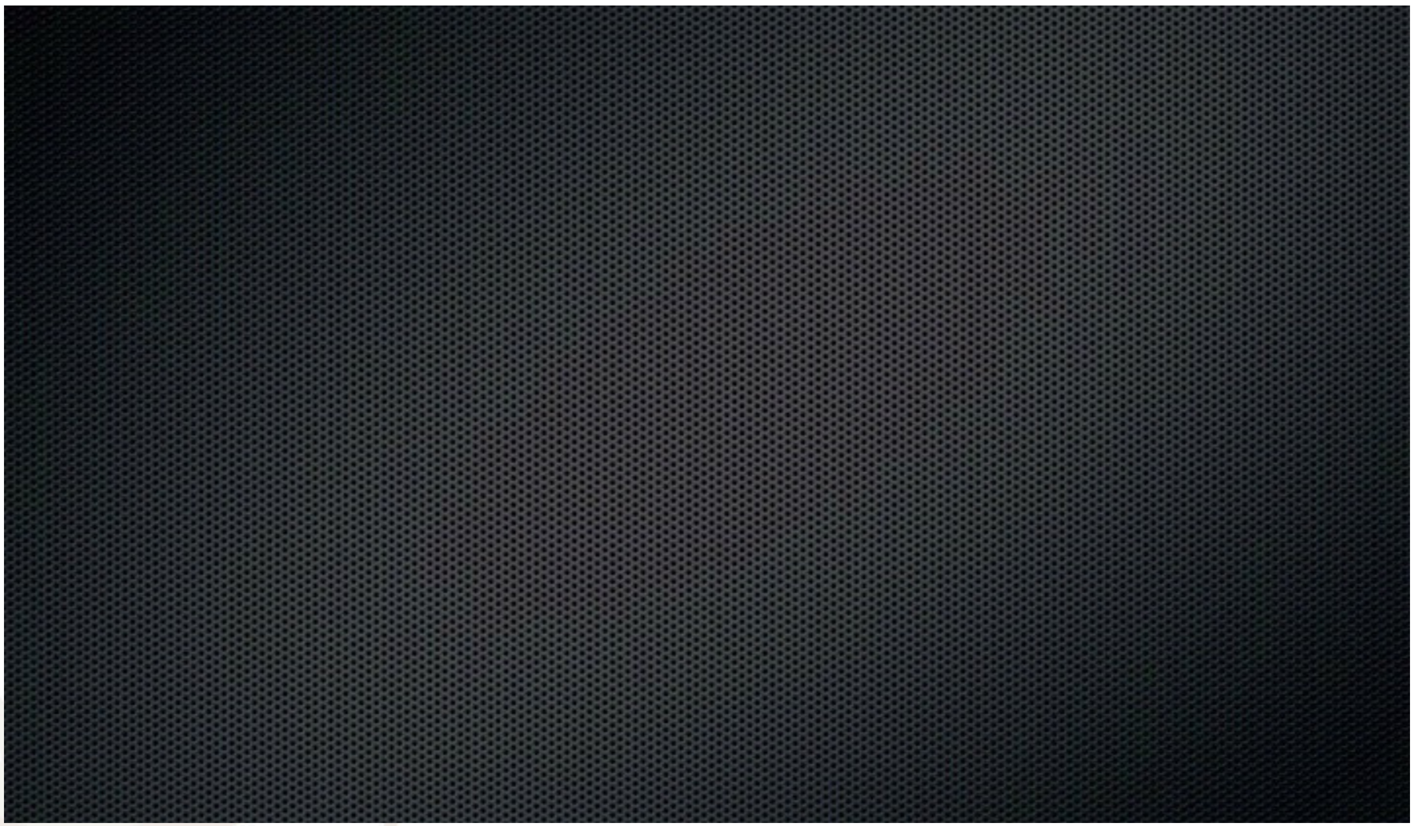
  – Does not take into account surface tension forces

State of the Art

– Problem is in fact more complex and SPH has limitations
  - Mixing fluid densities
  - State changes (solid → fluid) (i.e. welding)

State of the Art

- Unified architecture



SPH

- Kernel examples:

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

$$W_{spiky}(\mathbf{x}_{ij}, h) = \frac{15}{\pi h^6} \begin{cases} (h - x_{ij})^3 & 0 \le x_{ij} \le h \\ 0 & \text{otherwise.} \end{cases}$$

$$W_{viscous}(\mathbf{x}_{ij}, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{x_{ij}^3}{2h^3} + \frac{x_{ij}^2}{h^2} + \frac{h}{2x_{ij}} - 1 & 0 \le x_{ij} \le h \\ 0 & \text{otherwise,} \end{cases}$$