

Lecture 10

Fluid Animation

A brief introduction

**FUNDAMENTALS OF COMPUTER GRAPHICS
Animation & Simulation**

Stanford CS248B, Fall 2023

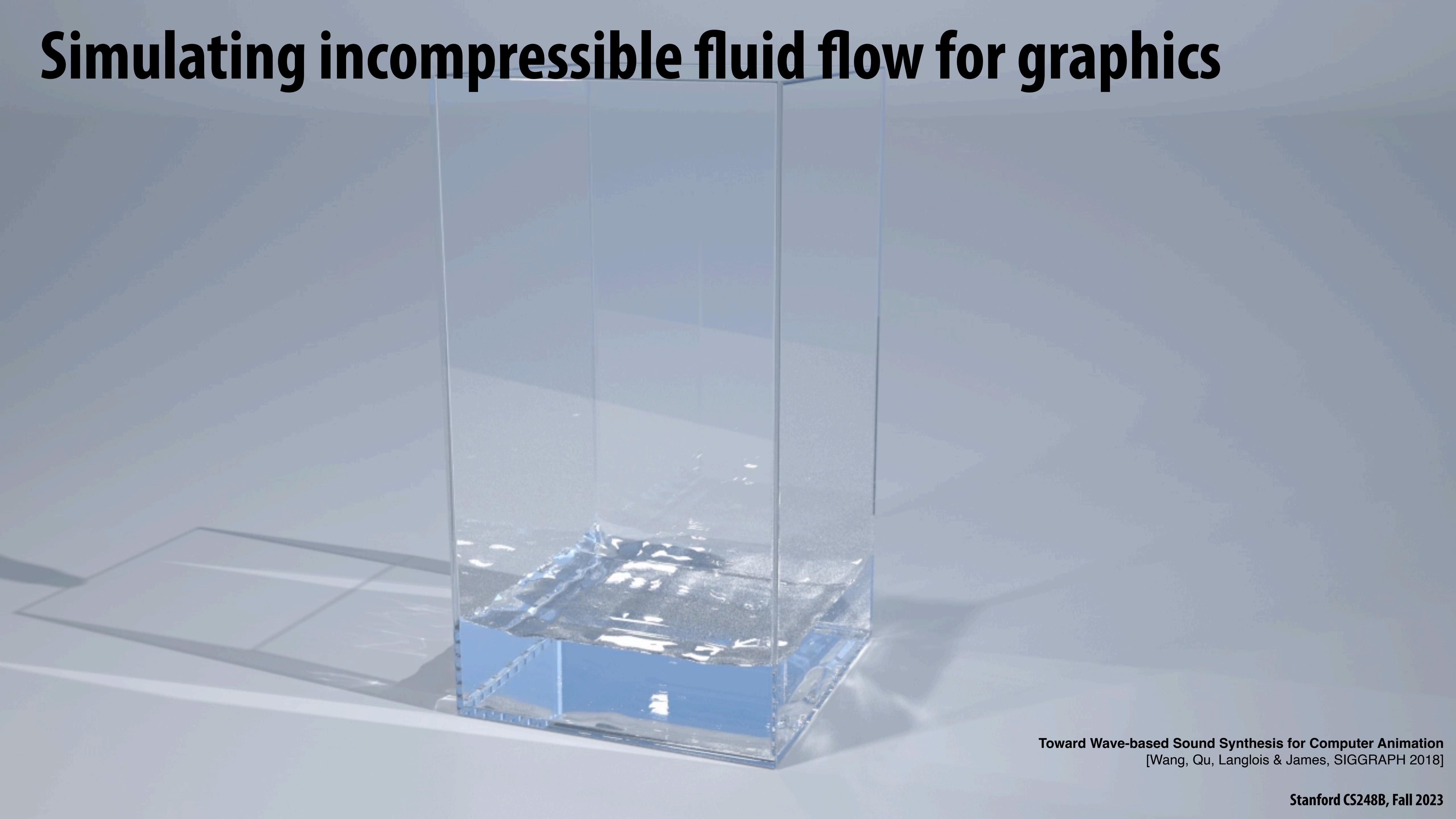
PROFS. KAREN LIU & DOUG JAMES

Fluid Animation

Reference:

Robert Bridson and Matthias Müller-Fischer
Fluid Simulation, SIGGRAPH 2007 Course Notes
https://www.cs.ubc.ca/~rbridson/fluidsimulation/fluids_notes.pdf

Simulating incompressible fluid flow for graphics



Toward Wave-based Sound Synthesis for Computer Animation
[Wang, Qu, Langlois & James, SIGGRAPH 2018]

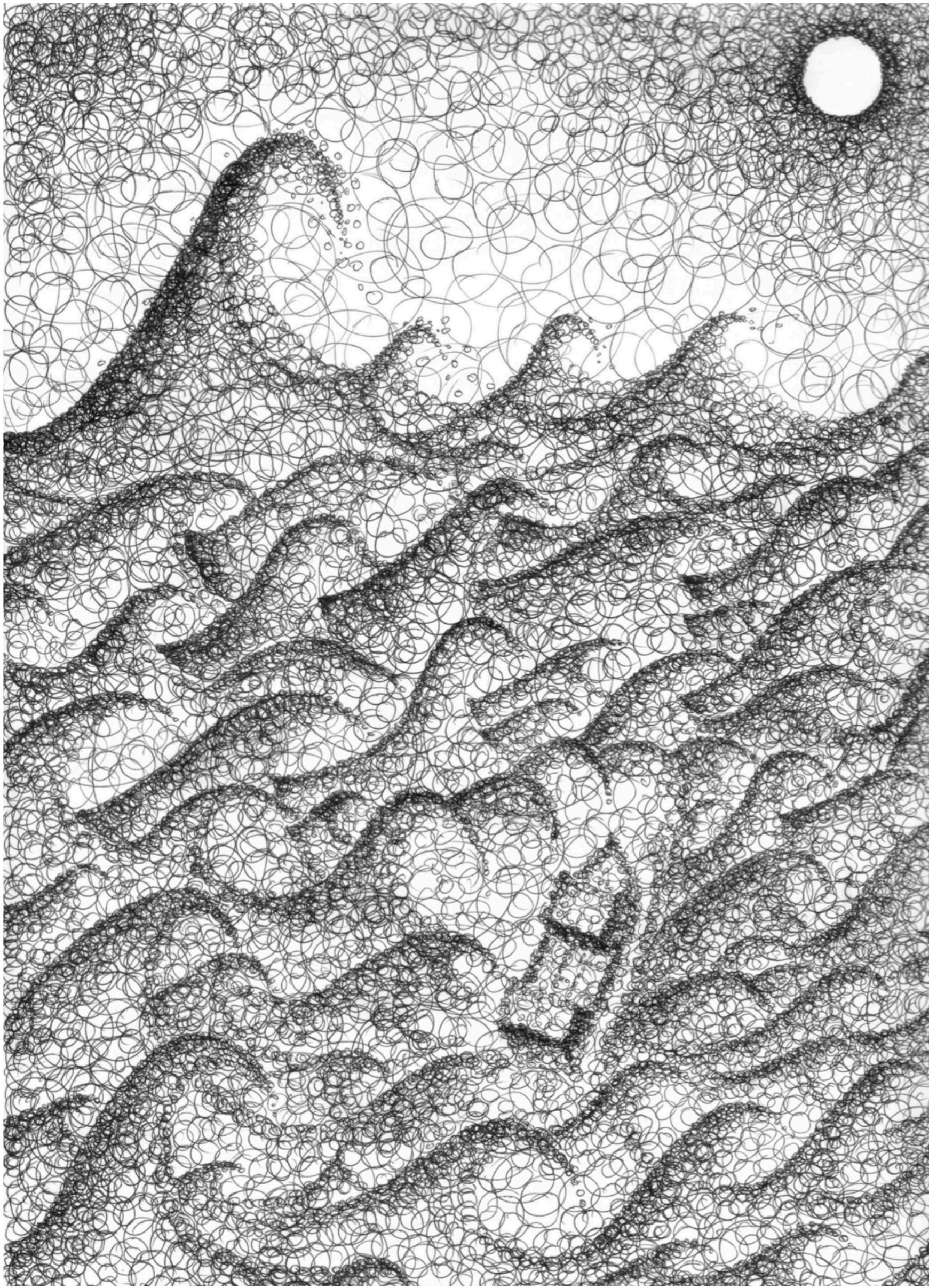
Navier-Stokes Equations

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

Momentum Equation ("a=f/m")

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

Incompressibility Constraint



The Foamy Brine, Robert Bridson. An artistic rendition of particle-based fluid simulation.

Particle-based Fluids

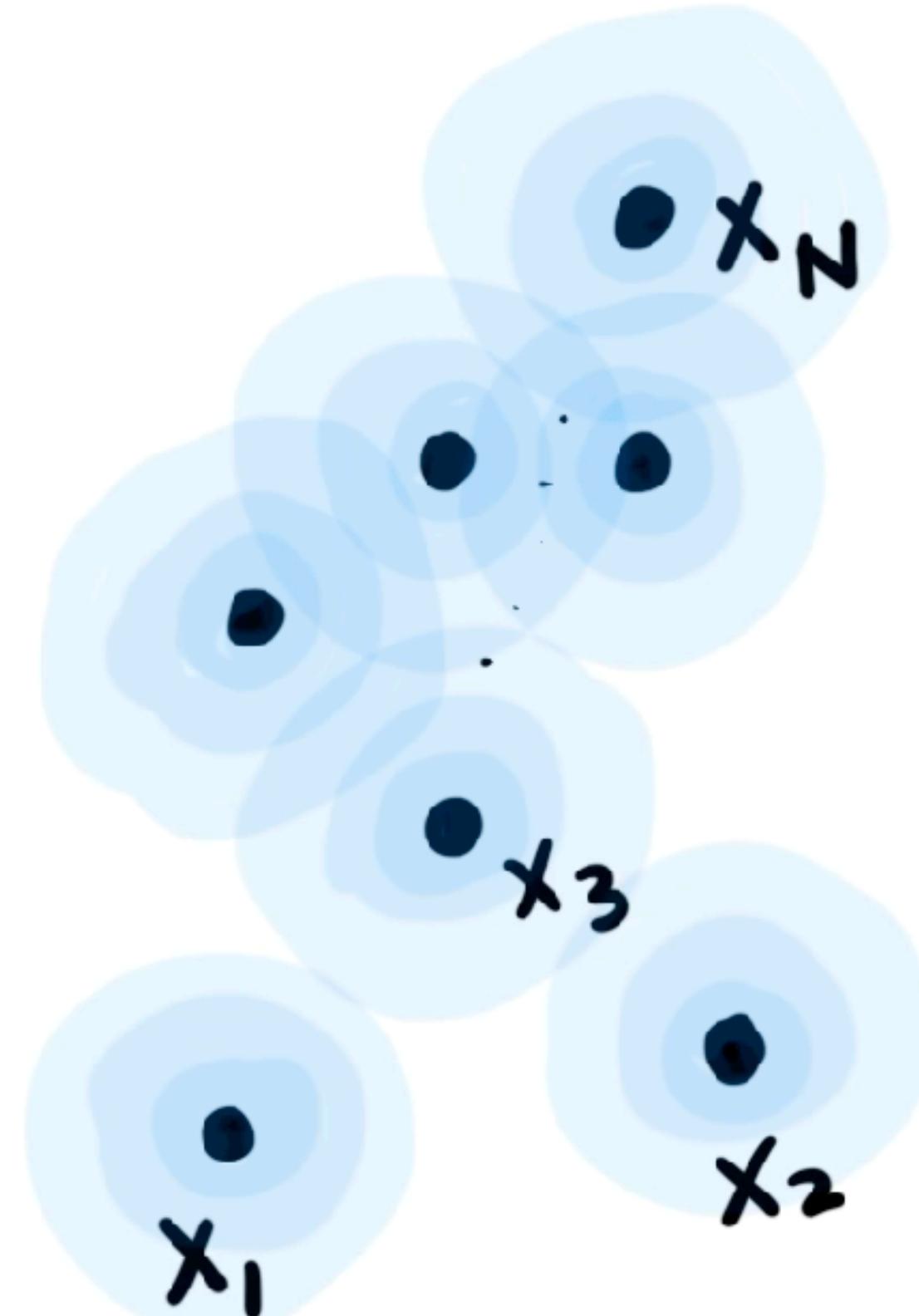
Lagrangian viewpoint: Data lives on particles

Smoothed Particle Hydrodynamics (SPH)

Smoothed Particle Hydrodynamics (SPH)

- Introduced in astrophysics by [Monagan 1992] for modeling stars and galaxies.
- SPH models everything using "blobby particles"
- Particles interact via Gaussian-like smoothing kernels, $W(\|\mathbf{x}_i - \mathbf{x}_j\|)$ of radius d .
- Unit normalized such that $\iiint W(\|\mathbf{x}\|) d^3\mathbf{x} = 1$
- Example: poly6 kernel
$$W_{\text{poly6}}(r) = \frac{315}{64\pi d^9} \begin{cases} (d^2 - r^2)^3 & 0 \leq r \leq d \\ 0 & \text{otherwise,} \end{cases}$$

SPH concepts: Mass density field, $\rho(x)$



Kernel $W(r)$ models density of unit mass blob, radius d .
⇒ $m_j W(\|x - x_j\|)$ is density of blob
with mass m_j at position x_j .

Density field = $\rho(x) = \sum_{j=1}^N m_j W(\|x - x_j\|)$
of N blobs

⇒ density at particle i : $\rho_i = \rho(x_i)$.

$$\begin{aligned}\text{TOTAL MASS} &= \int \rho(x) d^3x = \int \left(\sum_j m_j W(\|x - x_j\|) \right) d^3x = \sum_j m_j \boxed{\int W(\|x - x_j\|) d^3x} \\ &= \sum_j m_j \circlearrowleft\end{aligned}$$

SPH concepts: Any field, $A(x)$, from particle values, A_j

Generalize density interpolation formula:

$$\rho(x) = \sum_j m_j W(\|x - x_j\|) = \sum_j \frac{m_j}{s_j} s_j W(\|x - x_j\|)$$

"VOLUME"
OF PARTICLE j

So, for any attributes A_j on particles,

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

$$\Rightarrow \nabla A(x) = \sum_j \frac{m_j}{s_j} A_j \nabla W(\|x - x_j\|)$$

SPH force density, f_i (force per volume)

- Generalize Newton's second law, " $\mathbf{f} = m \mathbf{a}$," to force densities?
- Force density at particle i : $\mathbf{f}_i = \rho_i \mathbf{a}_i$
- Particle acceleration: $\mathbf{a}_i = \frac{\mathbf{f}_i}{\rho_i}$
- Force densities:
 - Gravitational force on a particle: $\mathbf{f}_i = \rho_i \mathbf{g} \longrightarrow \mathbf{a}_i = \mathbf{g}$
 - Compression force due to pressure gradient: $-\nabla p(\mathbf{x})$

SPH pressure force maintains fluid density

- Pressure forces due to gradient: $-\nabla p(\mathbf{x})$

$$f_i^{\text{pressure}} = -\nabla p(x_i) \xleftarrow{\text{SPH Trick}} = -\sum_j \frac{m_j}{\rho_j} p_j \nabla W(\|\mathbf{x}_i - \mathbf{x}_j\|)$$

Problem: f_i^{pressure} forces are not symmetric. ☹

Symmetrized pressure forces:

$$f_i^{\text{pressure}} = -\sum_j \frac{m_j}{\rho_j} \frac{p_i + p_j}{2} \nabla W(\|\mathbf{x}_i - \mathbf{x}_j\|)$$

What is $p_i = p(x_i)$?

SPH pressure models

- Pressure is used to enforce incompressibility, $\rho(\mathbf{x}) \approx \rho_0 = \text{liquid density}$
- Popular: Penalty-based pressure models:
 - Gas state equations:
 - Linear model (ideal gas): $p(\mathbf{x}) = k(\rho(\mathbf{x}) - \rho_0)$
 - Nonlinear model:
$$p(\mathbf{x}) = \frac{k\rho_0}{\gamma} \left(\left(\frac{\rho(\mathbf{x})}{\rho_0} \right)^\gamma - 1 \right)$$
 - Pros: Simple
 - Cons:
 - Fluid compresses (bounces) unless a stiff spring constant, k , is used
 - May need implicit integrators
 - More parameter tuning

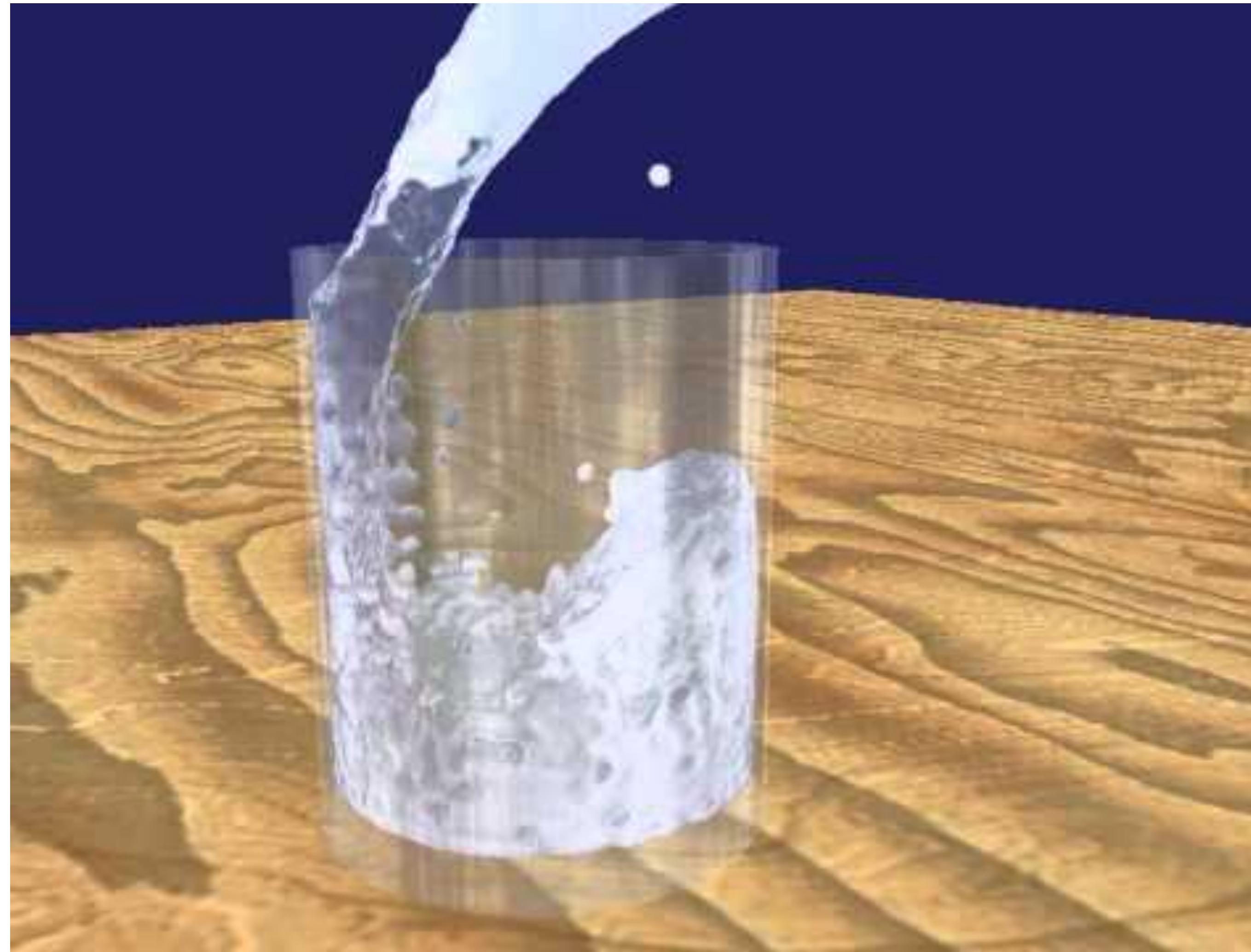
Typical SPH Implementation

Algorithm 1 SPH / WCSPH

```
1 while animating do
2   for all  $i$  do
3     find neighborhoods  $N_i(t)$ 
4   for all  $i$  do
5     compute density  $\rho_i(t)$ 
6     compute pressure  $p_i(t)$ 
7   for all  $i$  do
8     compute forces  $\mathbf{F}^{p,v,g,ext}(t)$ 
9   for all  $i$  do
10    compute new velocity  $\mathbf{v}_i(t + 1)$ 
11    compute new position  $\mathbf{x}_i(t + 1)$ 
```

[Solenthaler and Pajarola 2009]

Example: Interactive SPH [Mueller et al. 2003]



Example: Implicit SPH integration for stiffer compression forces

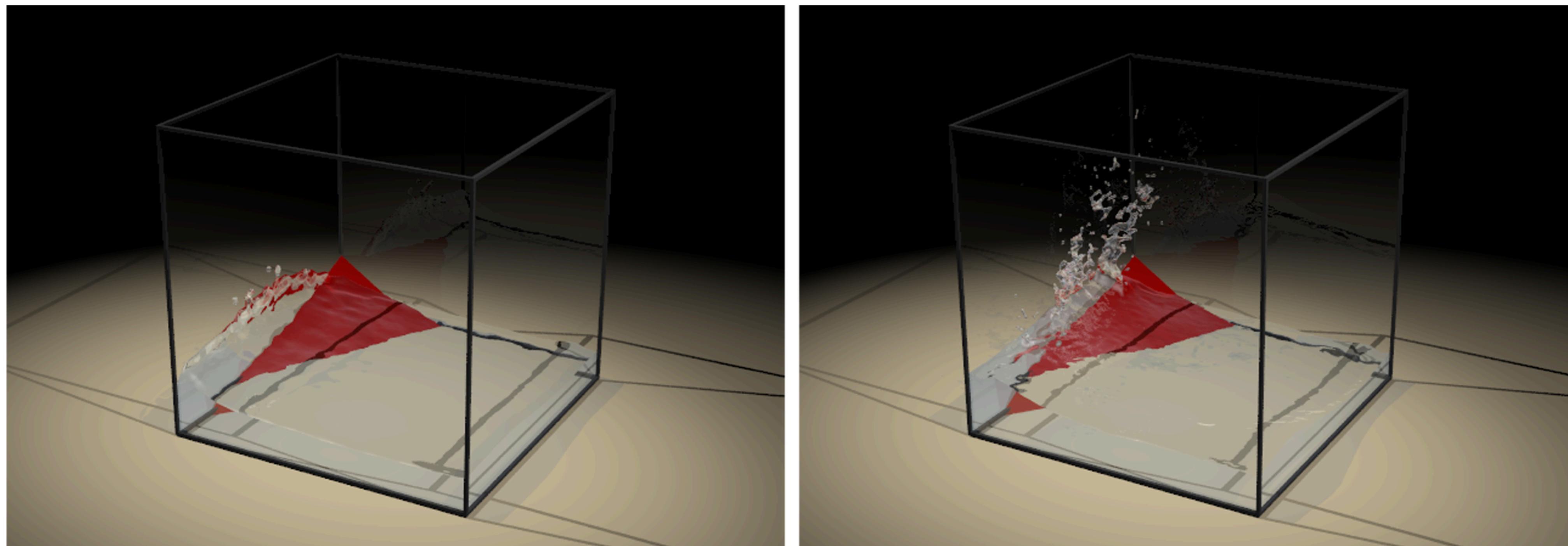


Figure 4: Comparison of WCSPH (left, 17k particles) and PCISPH (right, 100k particles) with equal computation times.

[Solenthaler and Pajarola 2009]

<https://people.inf.ethz.ch/~sobarbar/papers/Sol09/Sol09.pdf>

Implicit SPH: Predictive-Corrective Incompressible SPH (PCISPH)

Algorithm 2 PCISPH

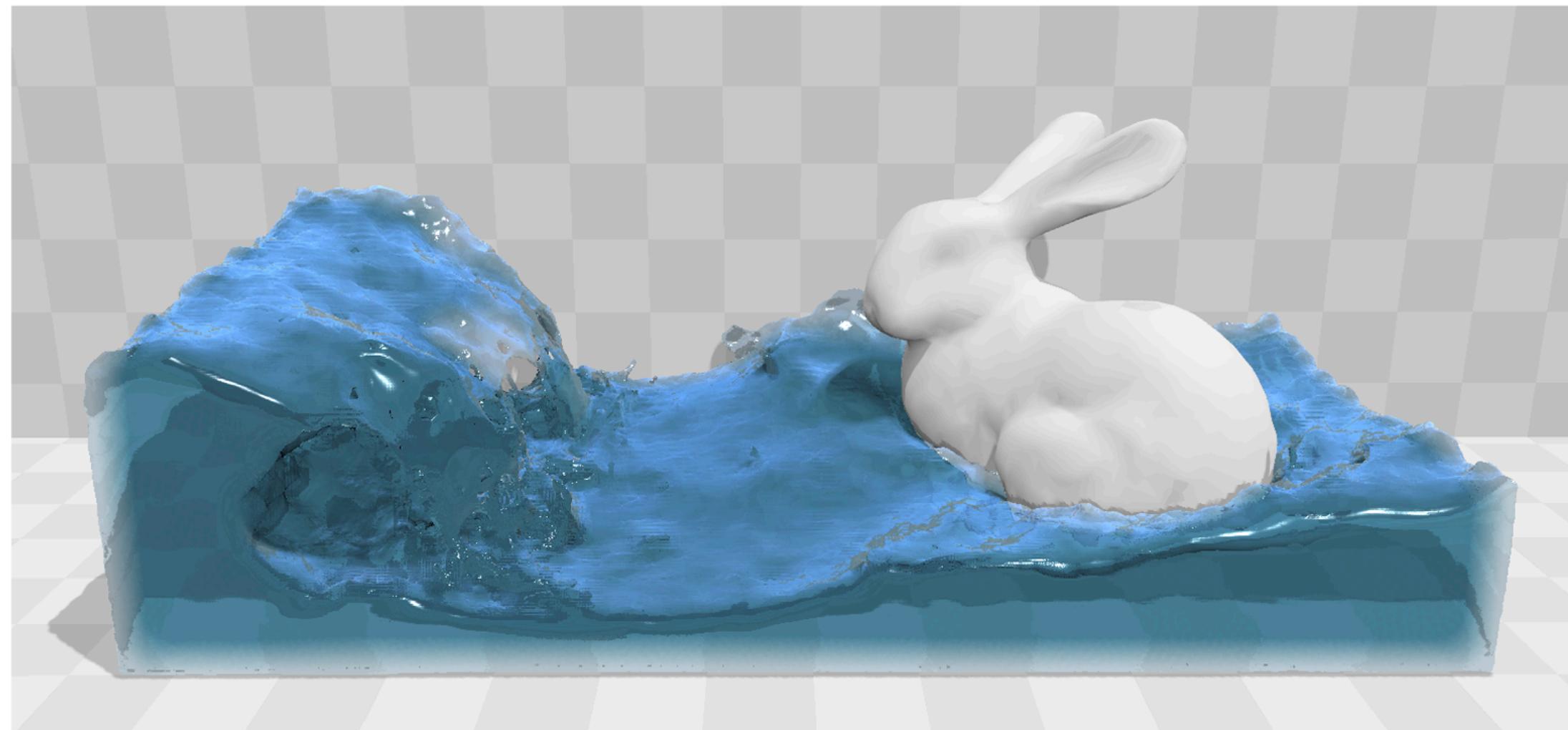
```
1 while animating do
2   for all  $i$  do
3     find neighborhoods  $N_i(t)$ 
4     for all  $i$  do
5       compute forces  $\mathbf{F}^{v,g,ext}(t)$ 
6       initialize pressure  $p(t) = 0.0$ 
7       initialize pressure force  $\mathbf{F}^p(t) = 0.0$ 
8     while  $(\rho_{err}^*(t + 1) > \eta) \parallel (iter < minIterations)$  do
9       for all  $i$  do
10      predict velocity  $\mathbf{v}_i^*(t + 1)$ 
11      predict position  $\mathbf{x}_i^*(t + 1)$ 
12      for all  $i$  do
13        predict density  $\rho_i^*(t + 1)$ 
14        predict density variation  $\rho_{err}^*(t + 1)$ 
15        update pressure  $p_i(t) += f(\rho_{err}^*(t + 1))$ 
16      for all  $i$  do
17        compute pressure force  $\mathbf{F}^p(t)$ 
18      for all  $i$  do
19        compute new velocity  $\mathbf{v}_i(t + 1)$ 
20        compute new position  $\mathbf{x}_i(t + 1)$ 
```

Implicit SPH: Predictive-Corrective Incompressible SPH (PCISPH)

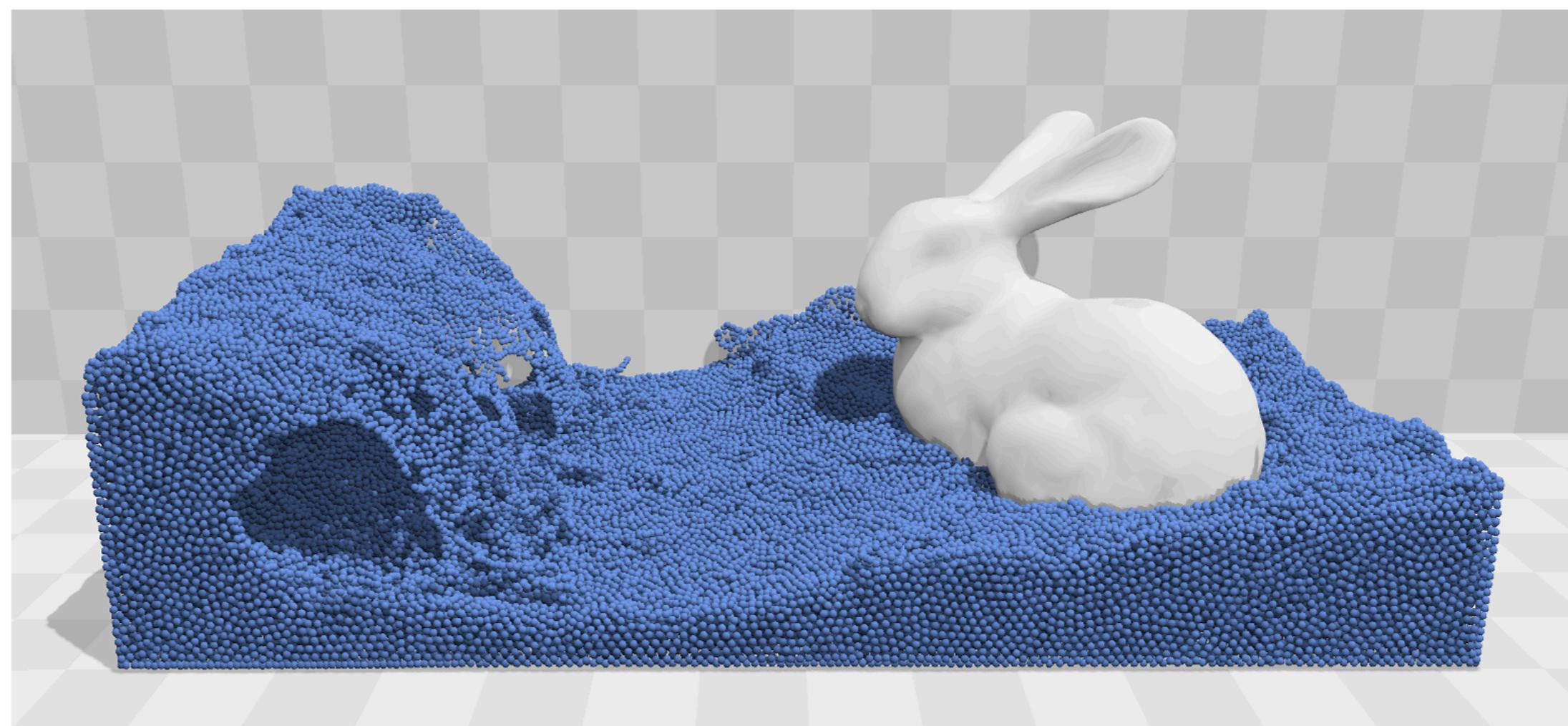


[Solenthaler and Pajarola 2009]

Example: Position Based Fluids [Mueller and Macklin 2013]



(a) Real-time rendered fluid surface using ellipsoid splatting



(b) Underlying simulation particles

Example: Position Based Fluids [Mueller and Macklin 2013]



https://www.youtube.com/watch?v=F5KuP6qEuew&ab_channel=MilesMacklin

Stanford CS248B, Fall 2023

Example: Position Based Fluids [Mueller and Macklin 2013]

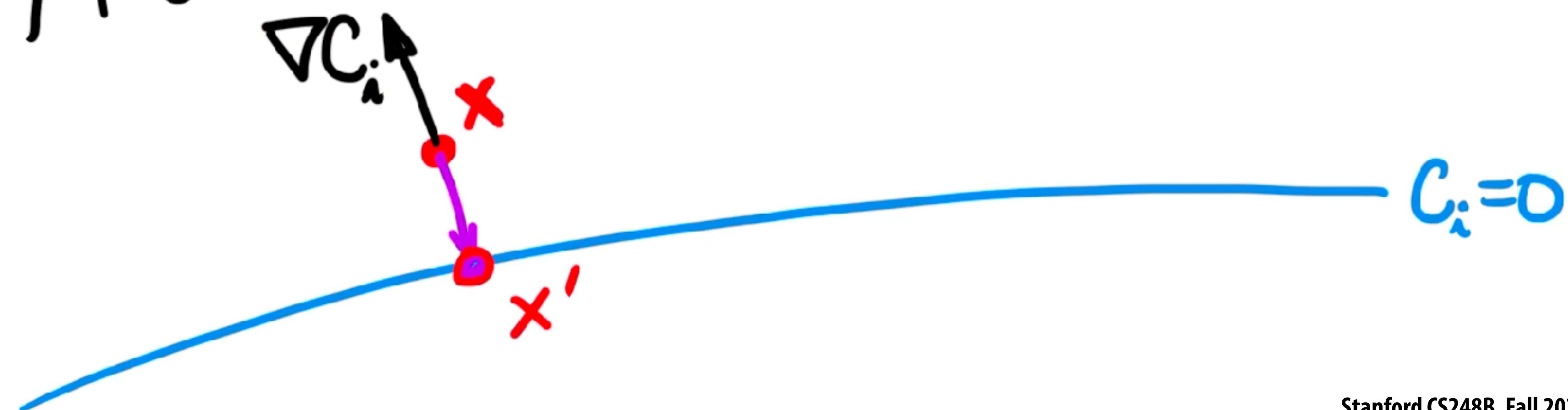
Want $\rho_i \approx \rho_0$ at each particle, i , where

$$\rho_i = \sum_j m_j W(\|x_i - x_j\|).$$

Impose constraint at each particle:

$$0 = C_i(x) = \frac{\rho_i(x)}{\rho_0} - 1$$

Idea: Iteratively project positions, x , onto constraints



Example: Position Based Fluids [Mueller and Macklin 2013]

Solve for position update Δx using linearization (Newton's method):

$$0 = C_i(x + \Delta x)$$

$$\approx C_i(x) + \nabla C_i(x)^T \Delta x = C_i(x) + \underbrace{J_i(x)}_{\text{Jacobian}} \Delta x$$

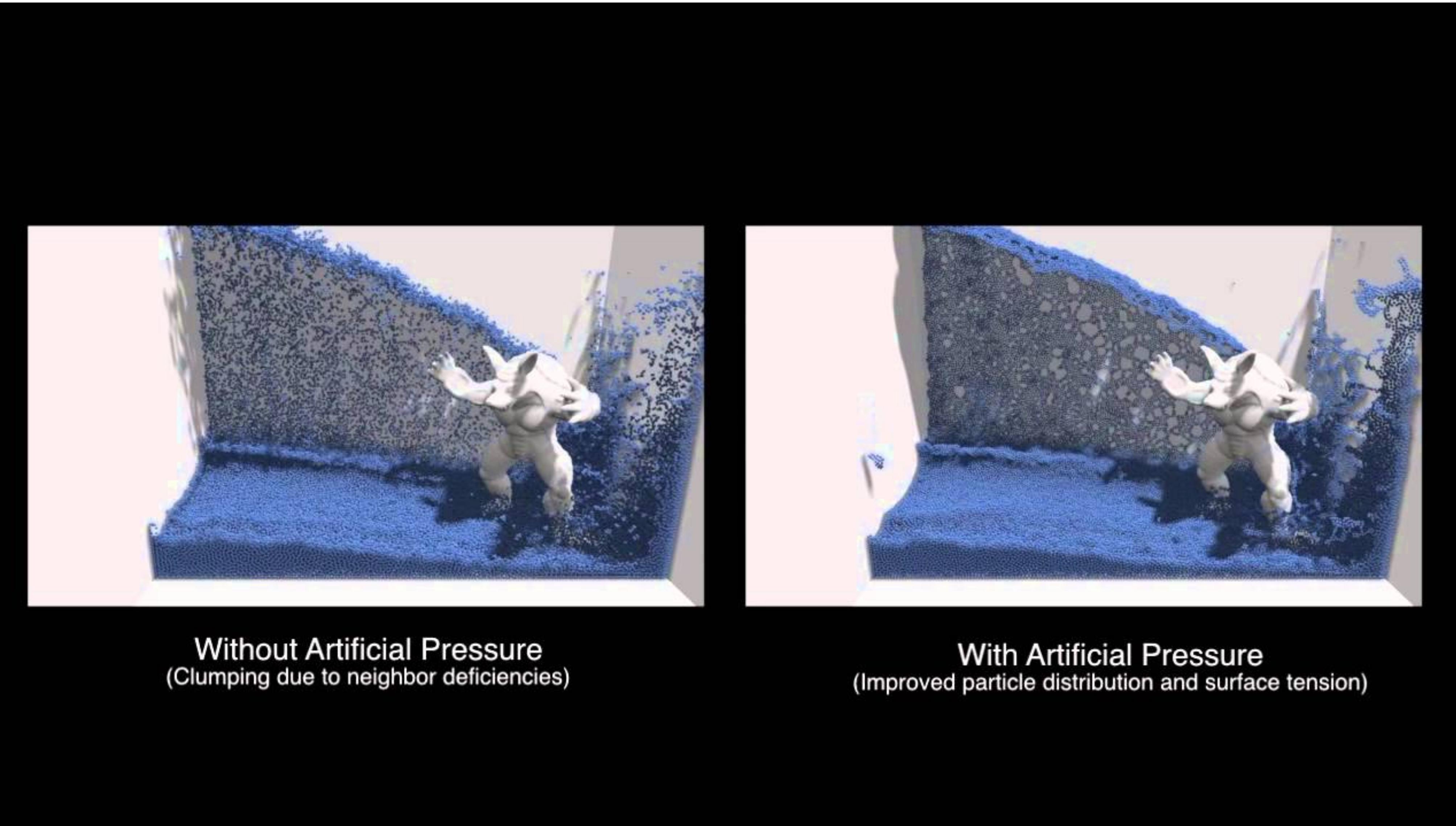
let $\Delta x = J_i^T \lambda_i = \begin{bmatrix} \square \\ \vdots \\ \square \end{bmatrix}$

$$= C_i + J_i J_i^T \lambda_i \longrightarrow \lambda_i = \frac{-C_i}{J_i J_i^T + \epsilon} = \frac{\square}{\boxed{\square} + \square}$$

$$\Rightarrow \boxed{\Delta x = -\frac{C_i J_i}{J_i J_i^T + \epsilon}}$$

Iterate over all constraints until error is small, or max-iterations.

Example: Position Based Fluids [Mueller and Macklin 2013]



Without Artificial Pressure
(Clumping due to neighbor deficiencies)

With Artificial Pressure
(Improved particle distribution and surface tension)

Particle-based rigid-fluid coupling

From "Versatile Rigid-Fluid Coupling for Incompressible SPH" [Akinci et al. 2012]

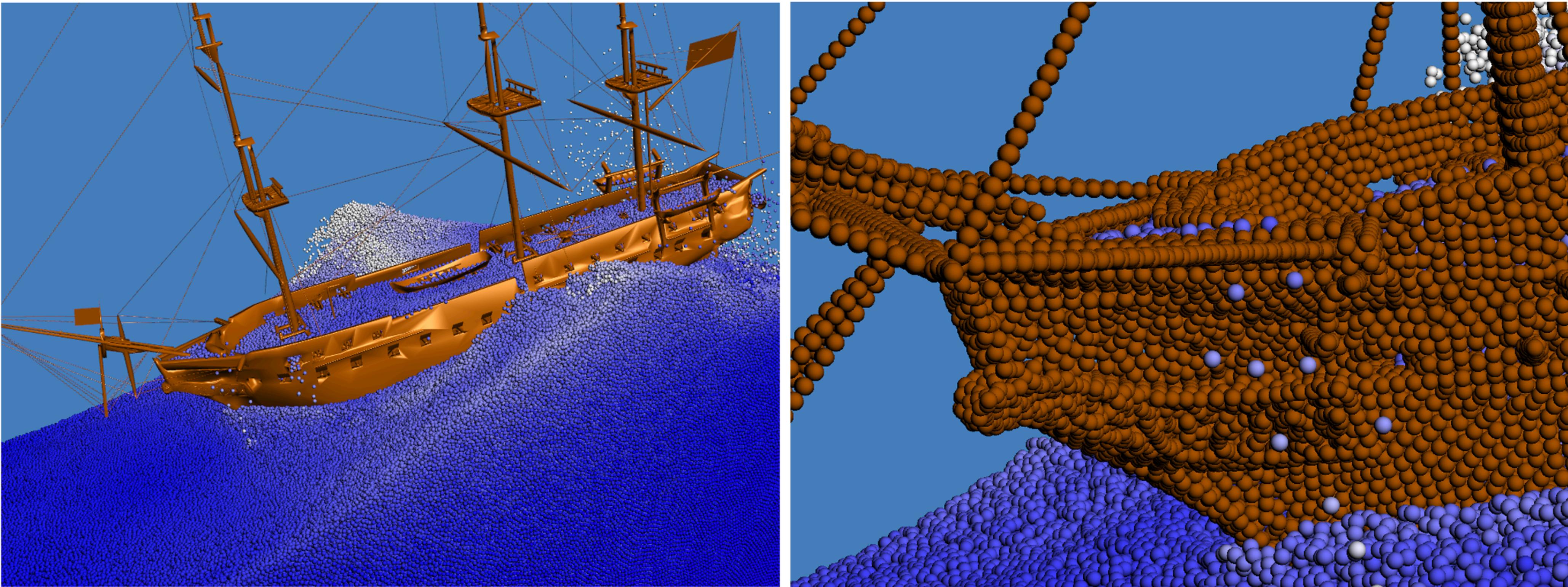


Figure 3: A frigate is sailing on wavy sea. The right image shows the irregular sampling of boundary particles.

Particle-based rigid-fluid coupling

From "Versatile Rigid-Fluid Coupling for Incompressible SPH" [Akinci et al. 2012]

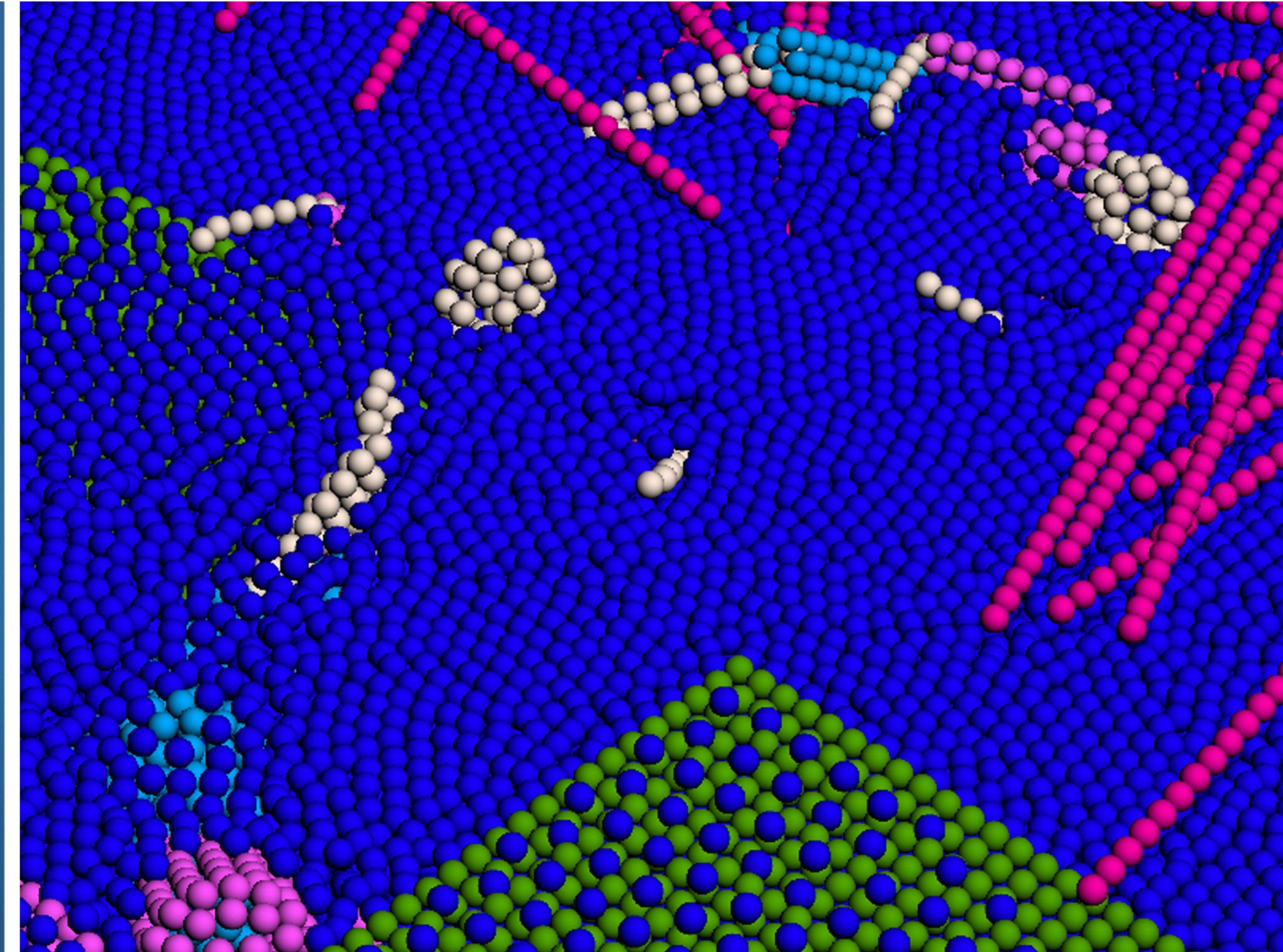
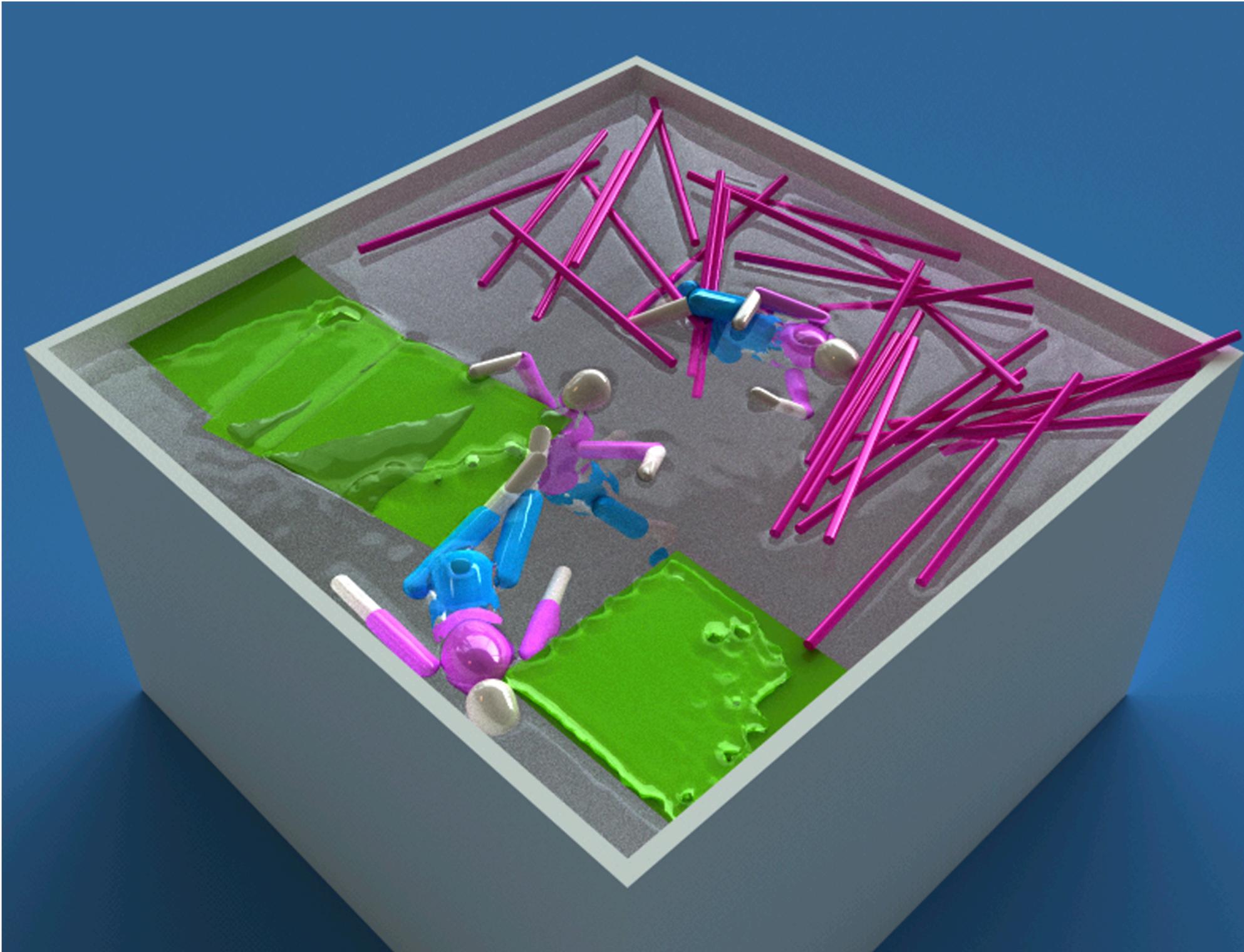
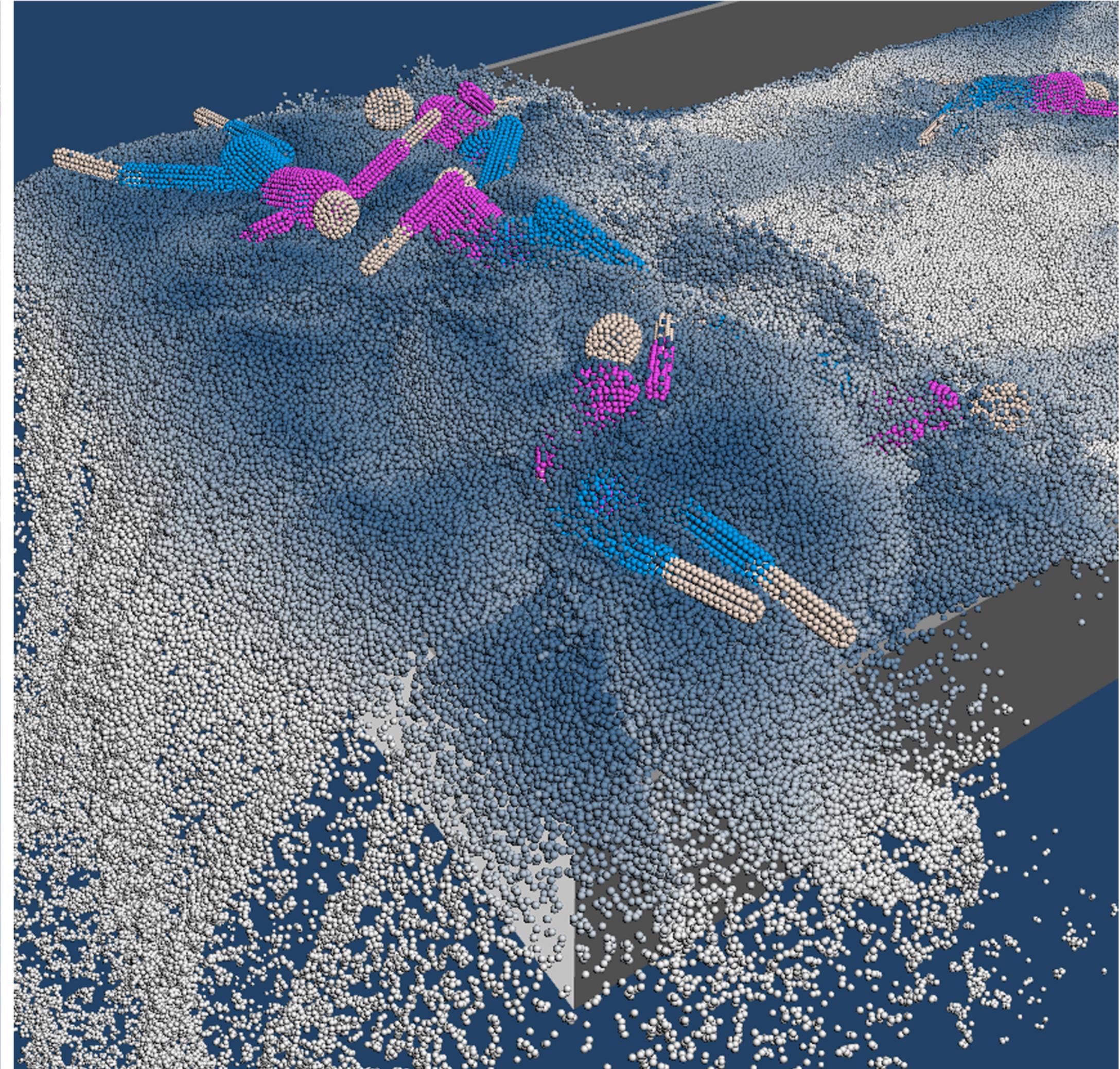
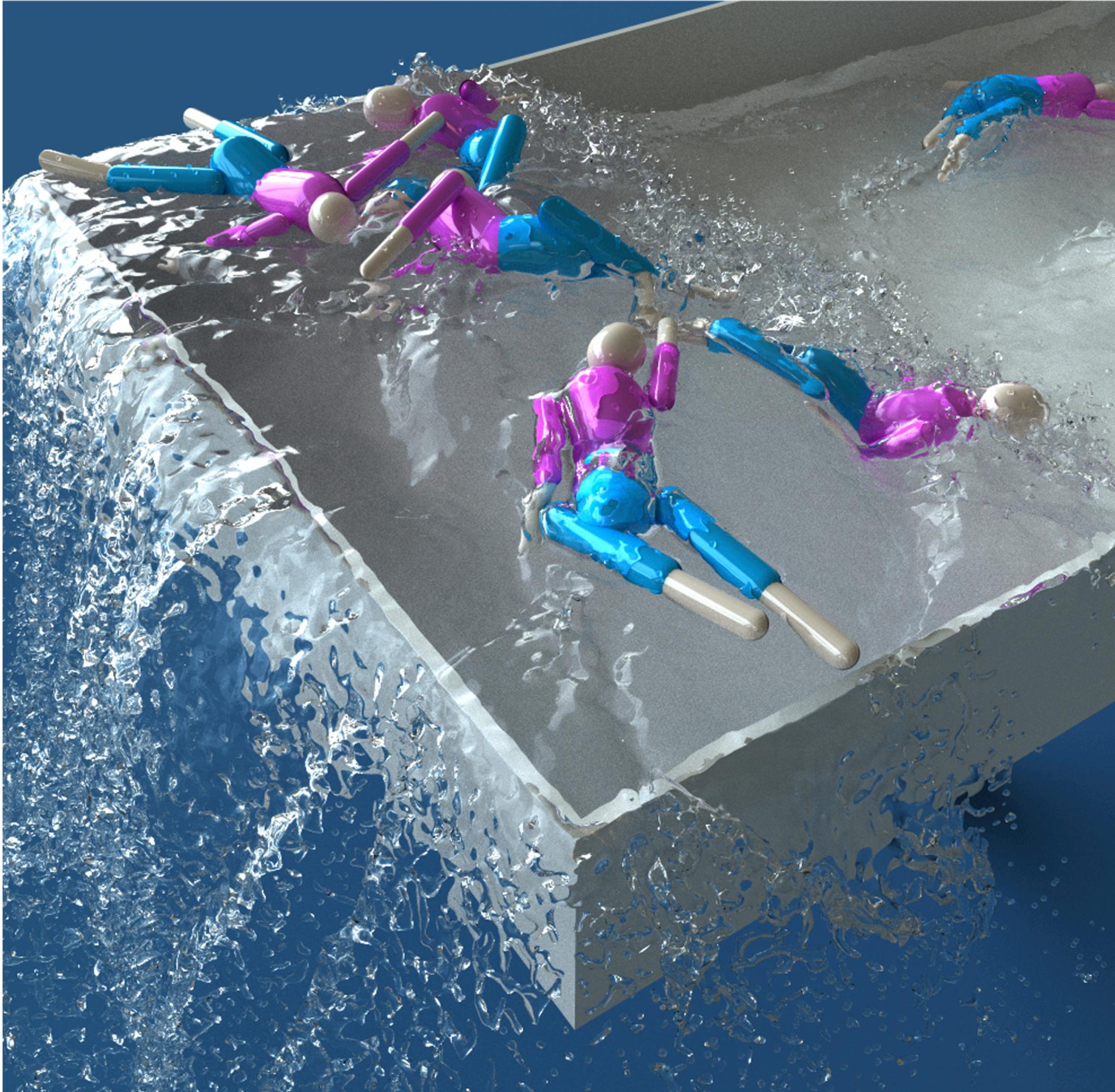


Figure 8: *Handling of lower dimensional objects.* The right image shows the underlying particles. Note that planes and rods are modeled with single particle layers.

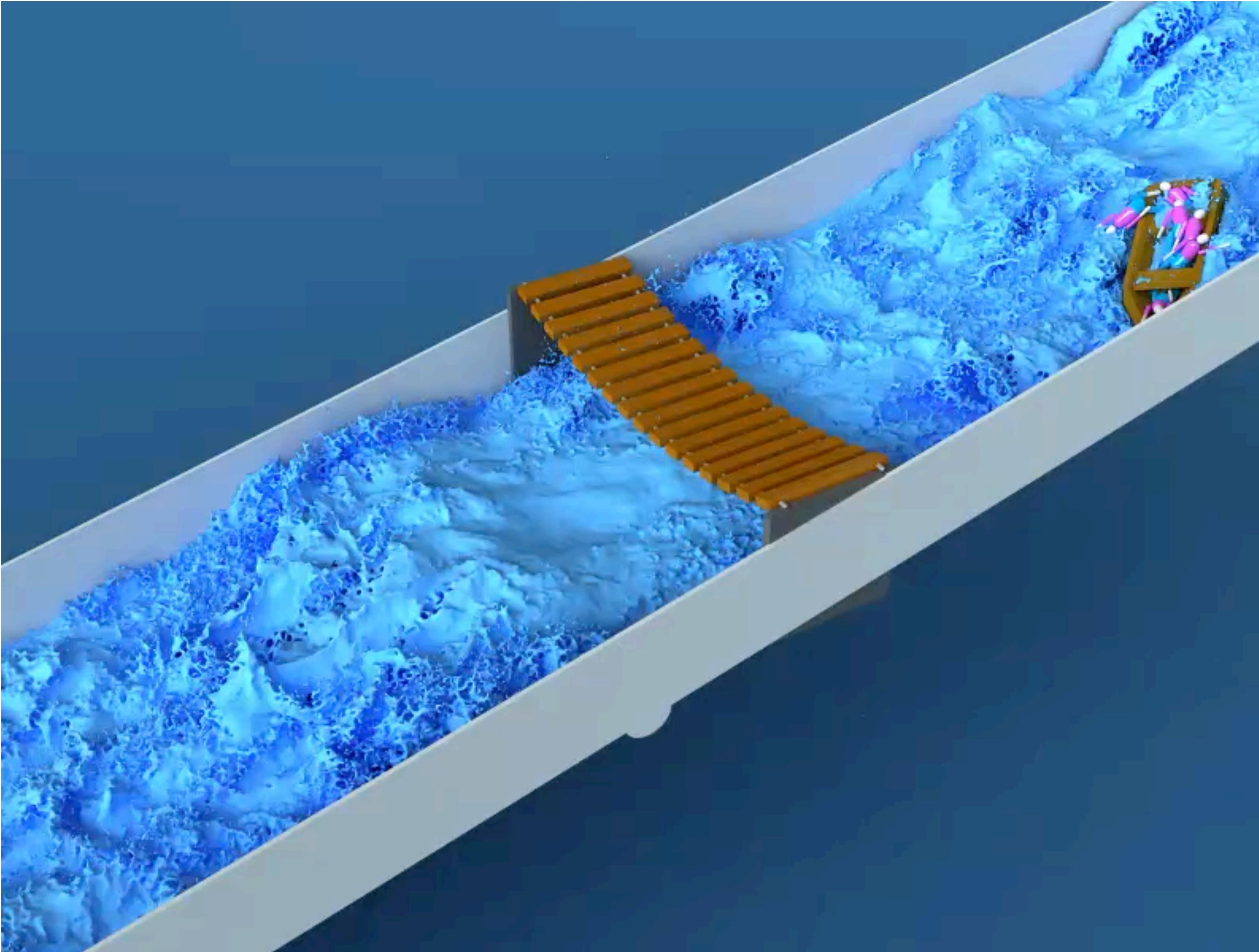
Particle-based rigid-fluid coupling

From "Versatile Rigid-Fluid Coupling for Incompressible SPH" [Akinci et al. 2012]



Particle-based rigid-fluid coupling

From "Versatile Rigid-Fluid Coupling for Incompressible SPH" [Akinci et al. 2012]



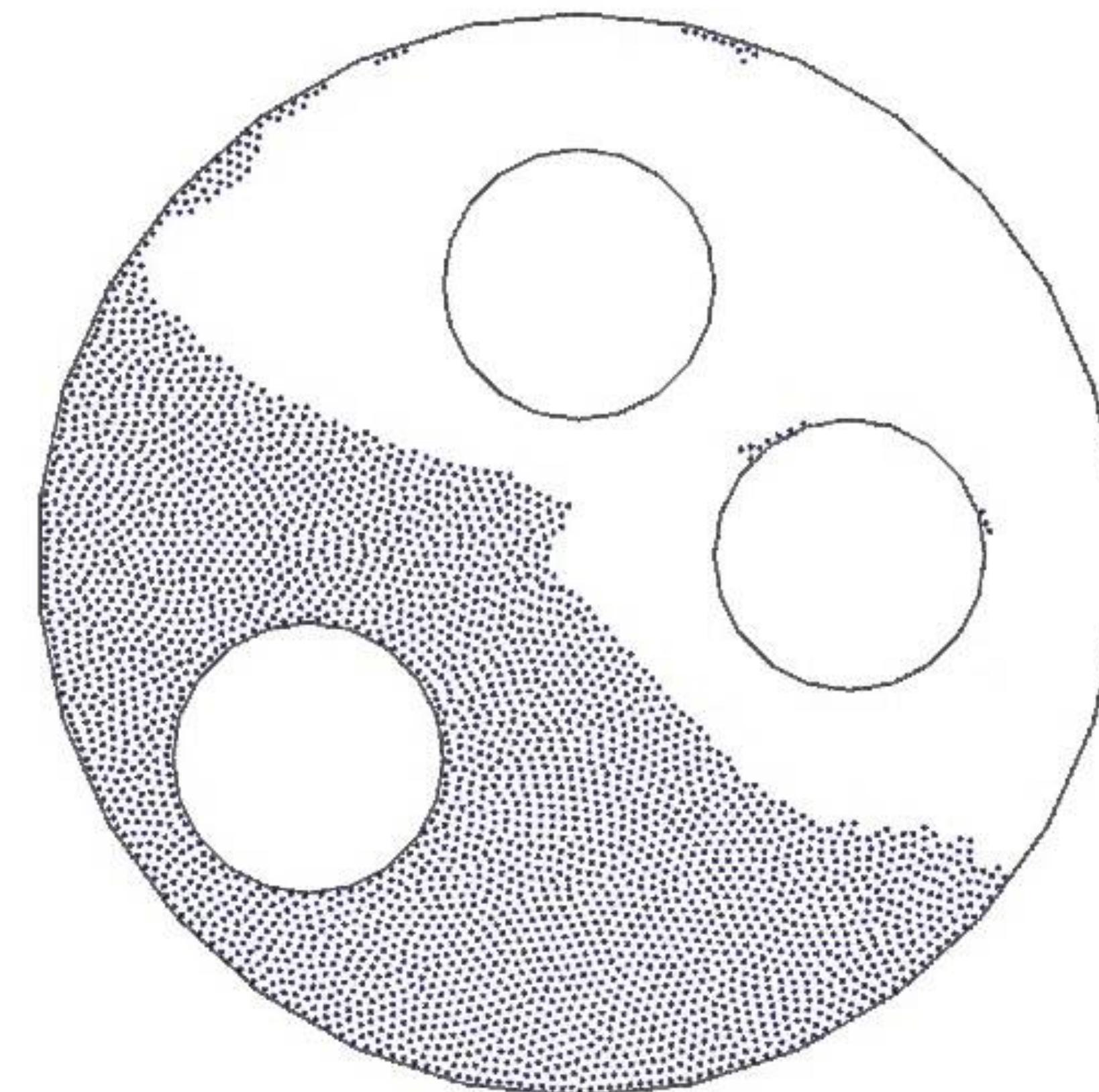
Example: Power Particles [de Goes et al. 2015]

Fluid against multiple obstacles

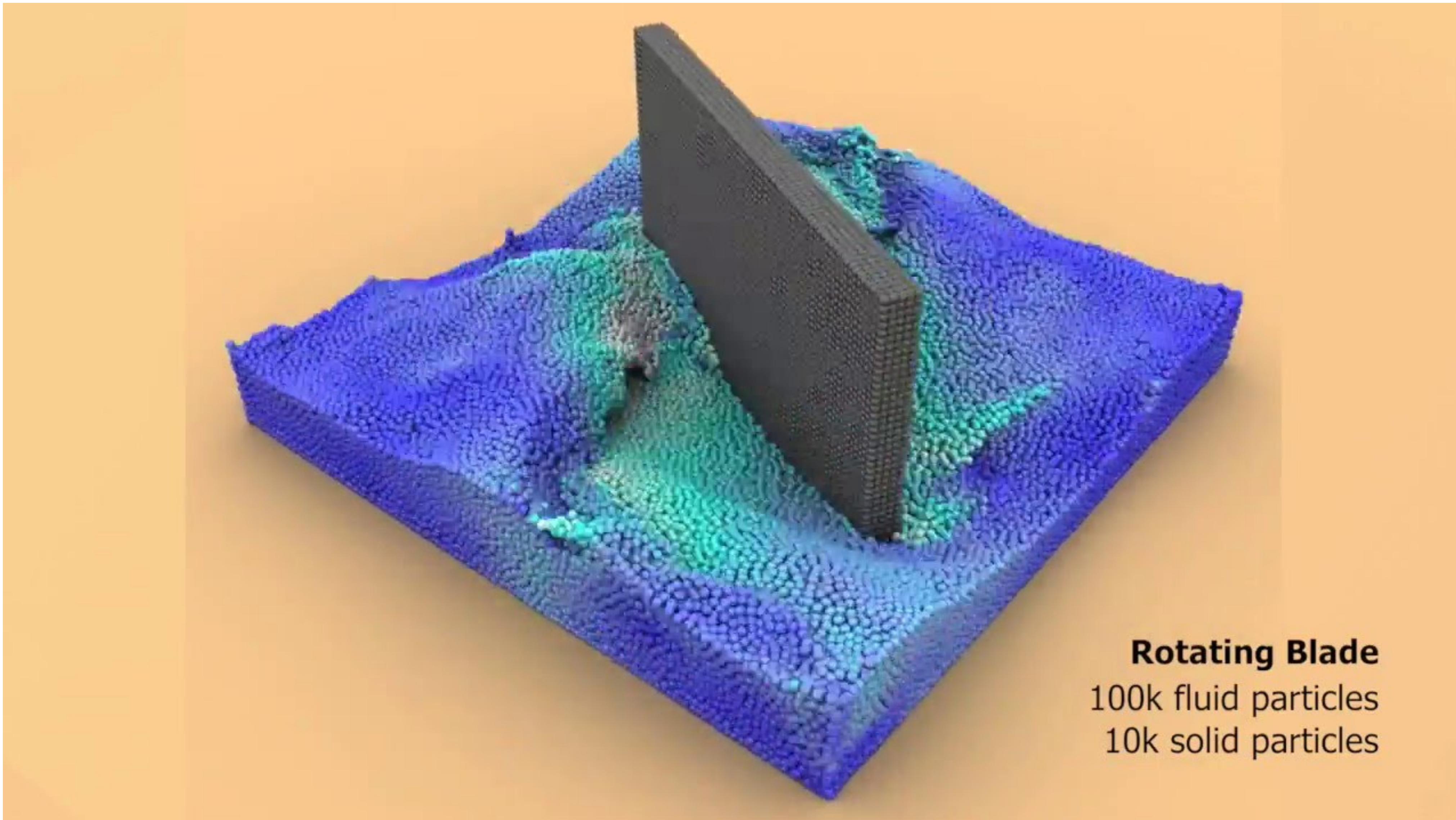
3.2k fluid particles

viscosity = 0.01

POWER CELLS CLIPPED VIA VORO++



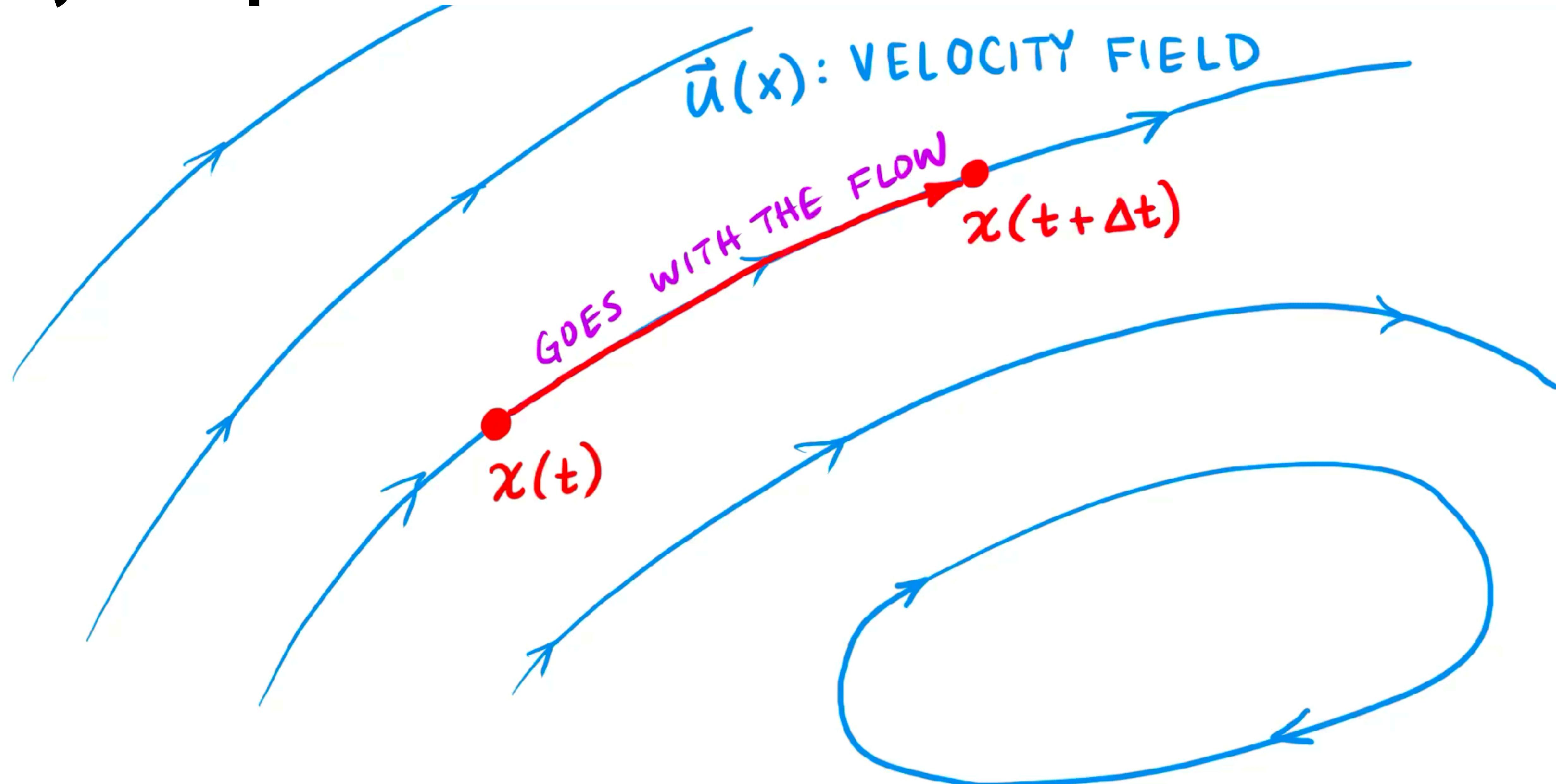
Example: Power Particles [de Goes et al. 2015]

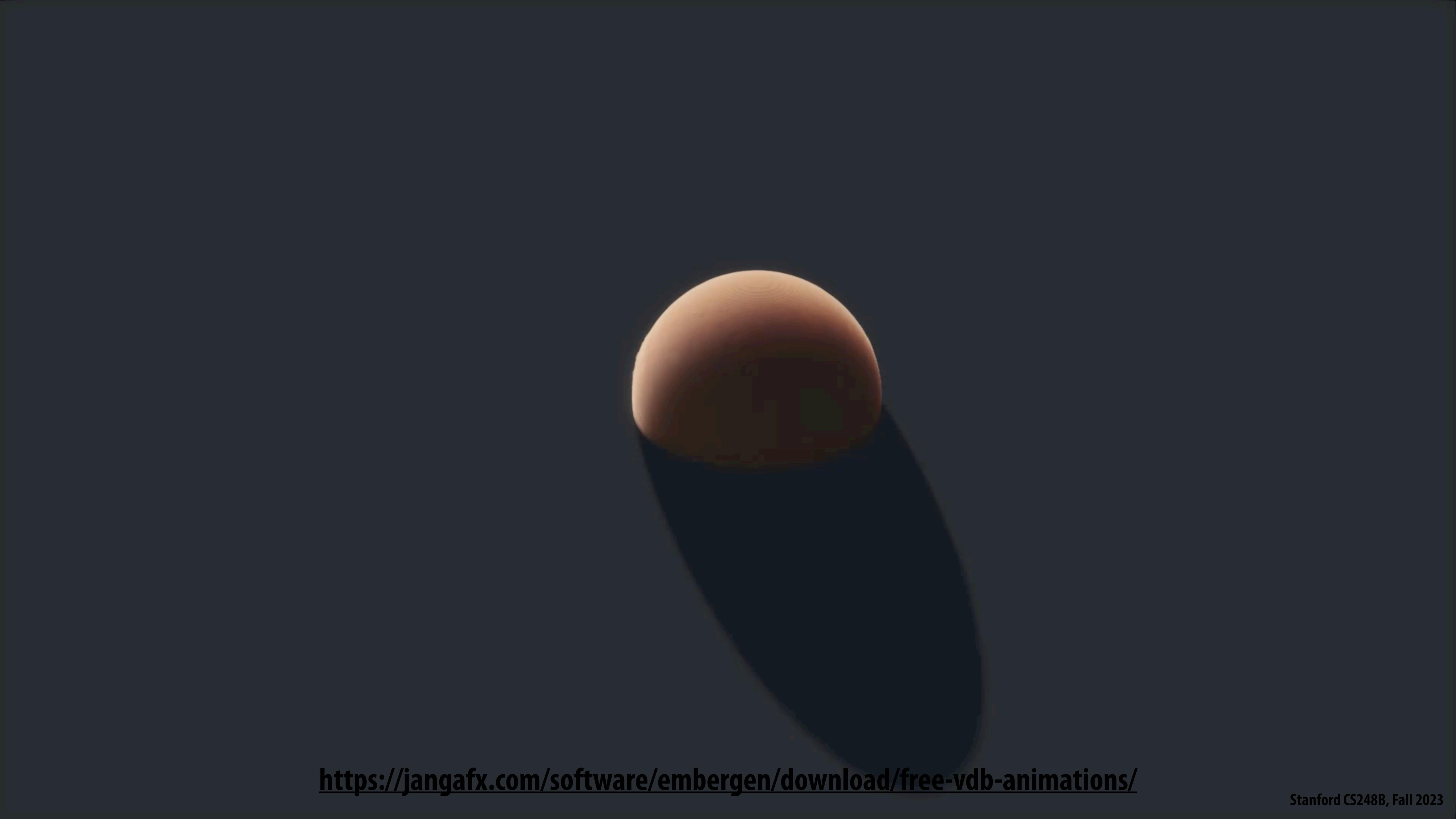


Grid-based Fluids

Eulerian viewpoint: Velocity & fields live at grid points.

Key concept: Advection





<https://jangafx.com/software/embergen/download/free-vdb-animations/>

Example: Bubble advection w/ buoyancy and drag forces



[Kim et al. 2010]

<https://player.vimeo.com/video/24967820?playsinline=true&autoplay=1>

Example: Advectiong smoke density through fluid velocity



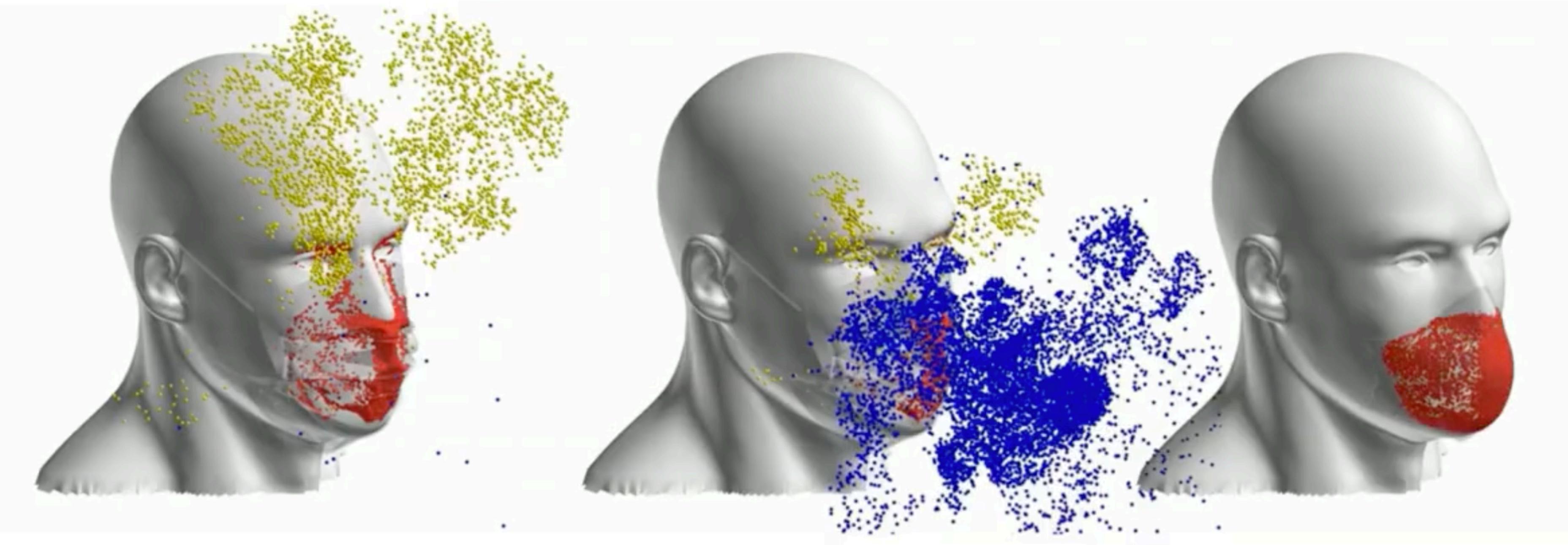
<https://jangafx.com/software/embergen/download/free-vdb-animations/>

Example: Advectiong smoke density through fluid velocity



<https://jangafx.com/software/embergen/download/free-vdb-animations/>

Example: Aerosol and droplet advection (RIKEN, Japan)



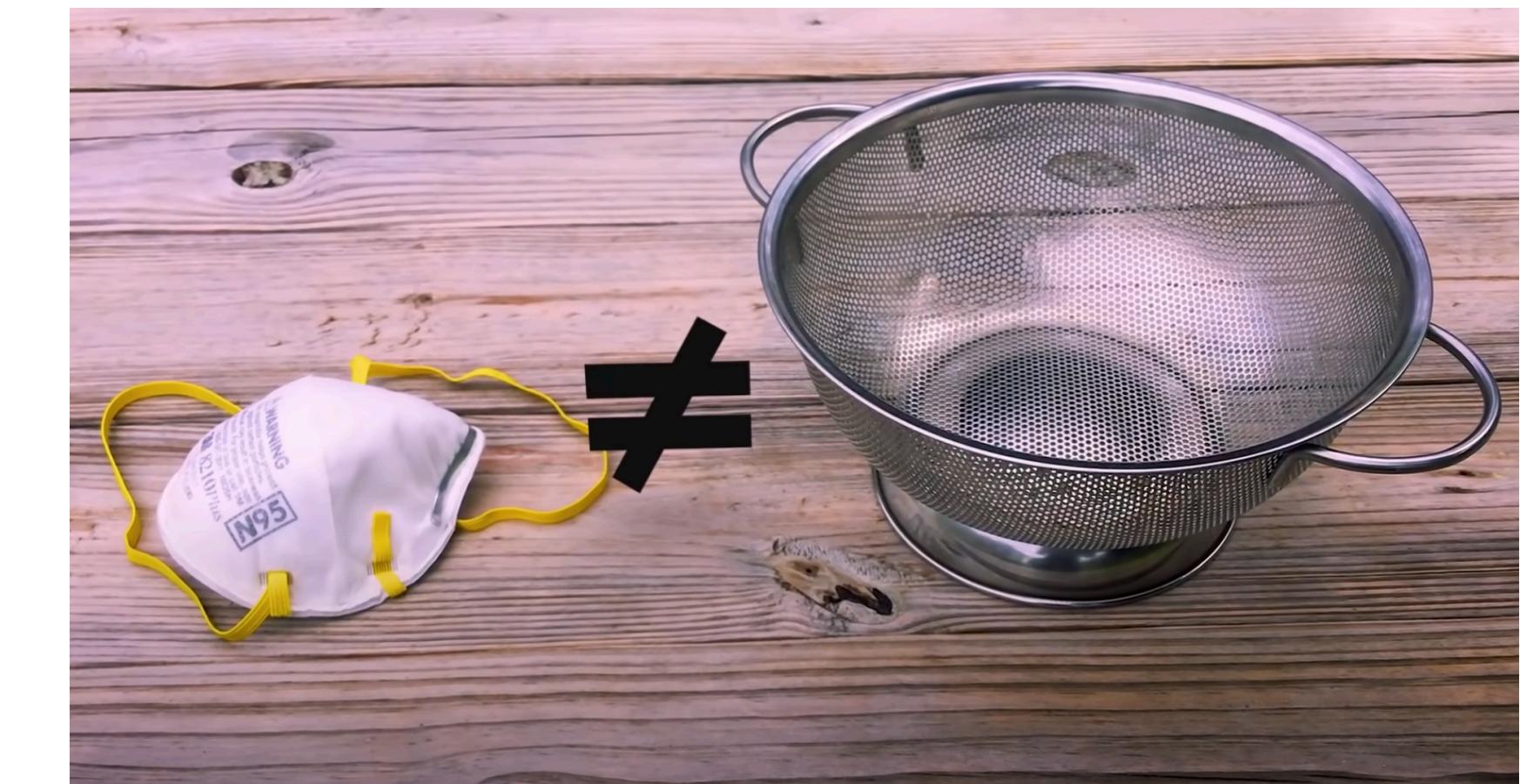
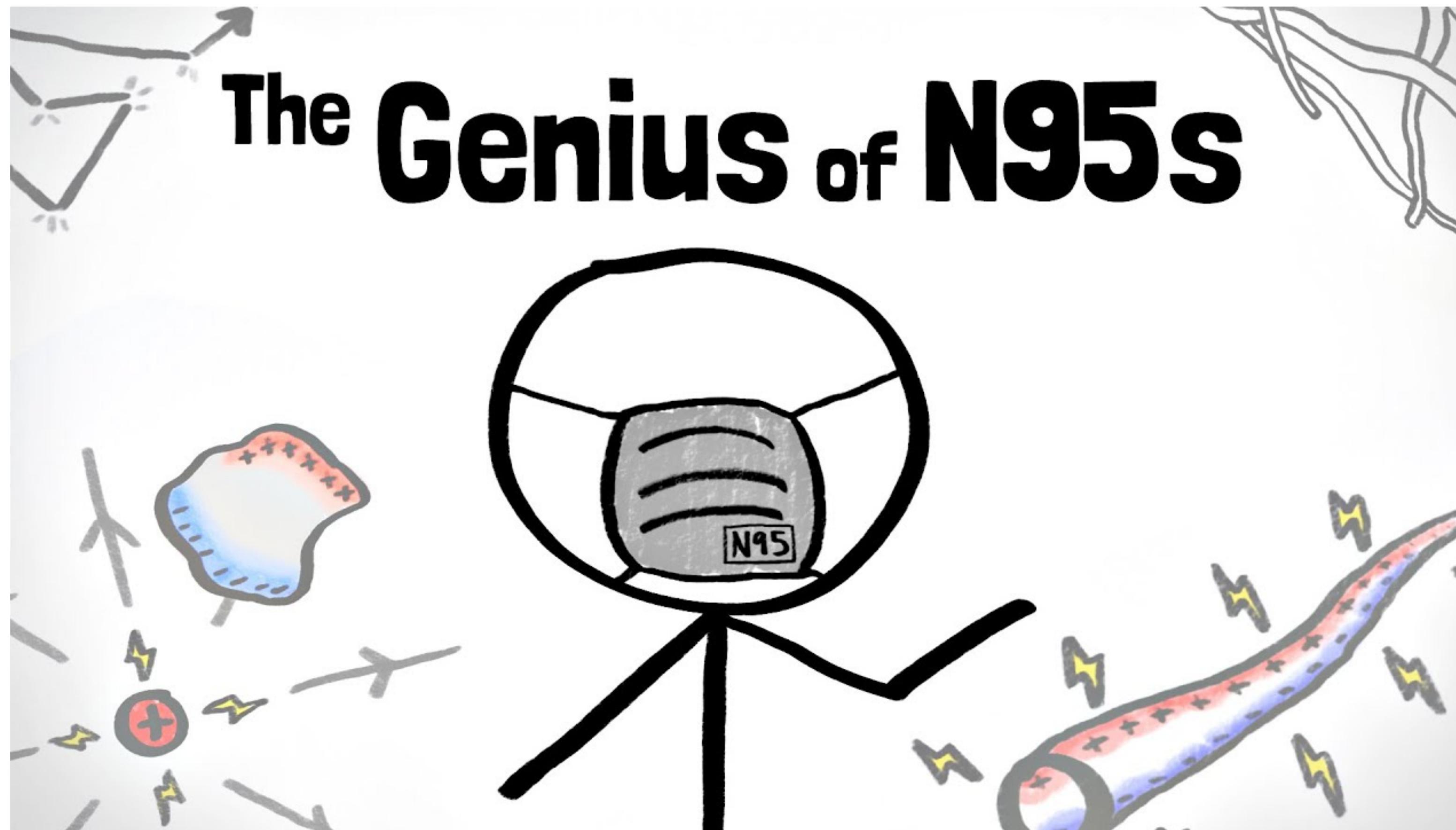
Non-woven mask

T-shirt material

N95 mask

<https://twitter.com/VCSTX/status/1582033089425190912>

Aside: The astounding physics of how N95 masks work



https://www.youtube.com/watch?v=eAdanPfQdCA&ab_channel=minutephysics

Key concept: The Material Derivative

ADVECTION OF SPACE-TIME QUANTITIES, $q(x, t)$

QUANTITIES DON'T CHANGE ALONG PATH, SO

$$0 = \frac{Dq(x,t)}{Dt} = \text{"material derivative"}$$

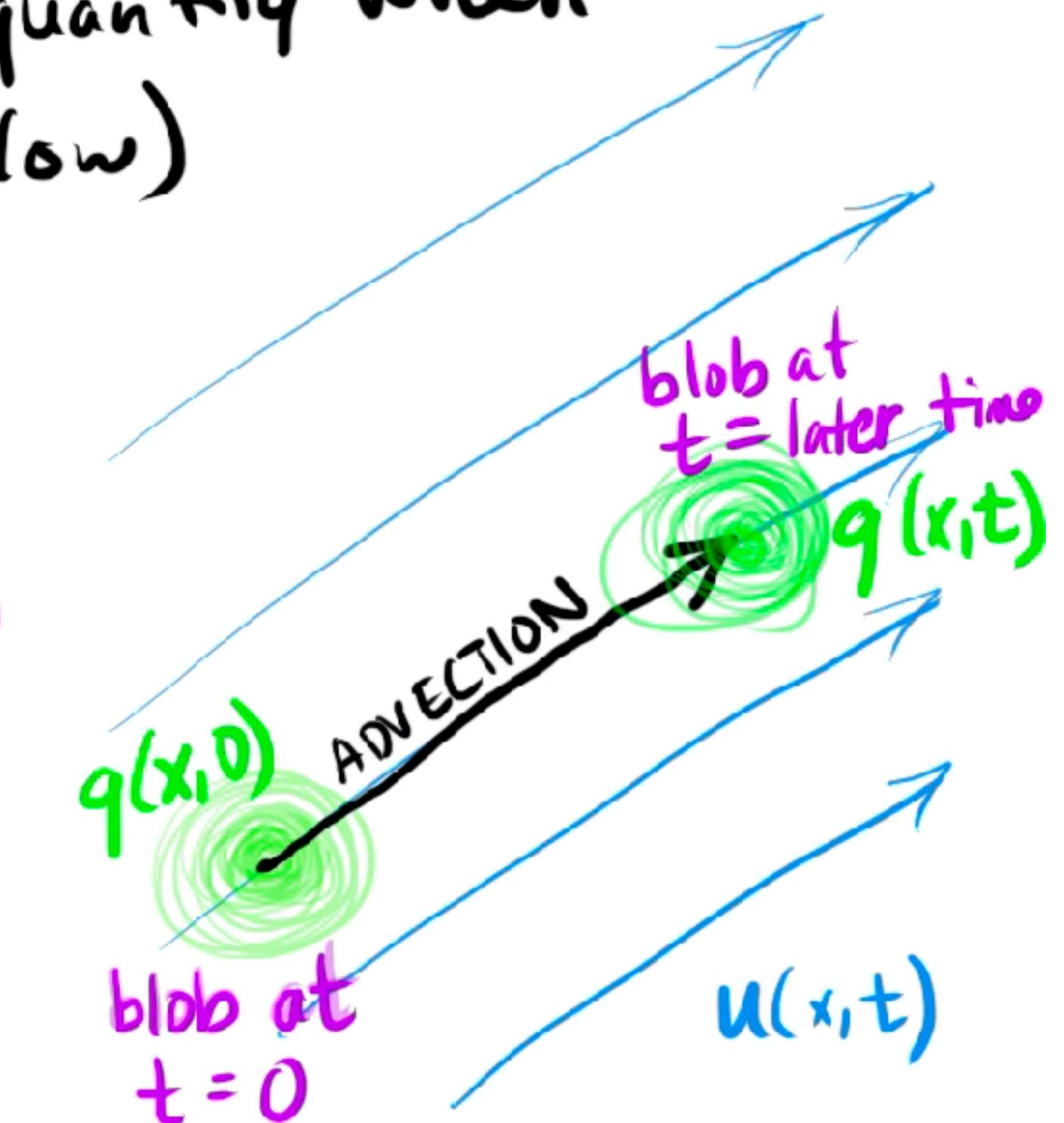
(the rate of change of the quantity when moving with the flow)

$$= \frac{\partial q}{\partial t} + \nabla_x q \cdot \frac{dx}{dt}$$

chain rule

fluid velocity, $u(x, t)$

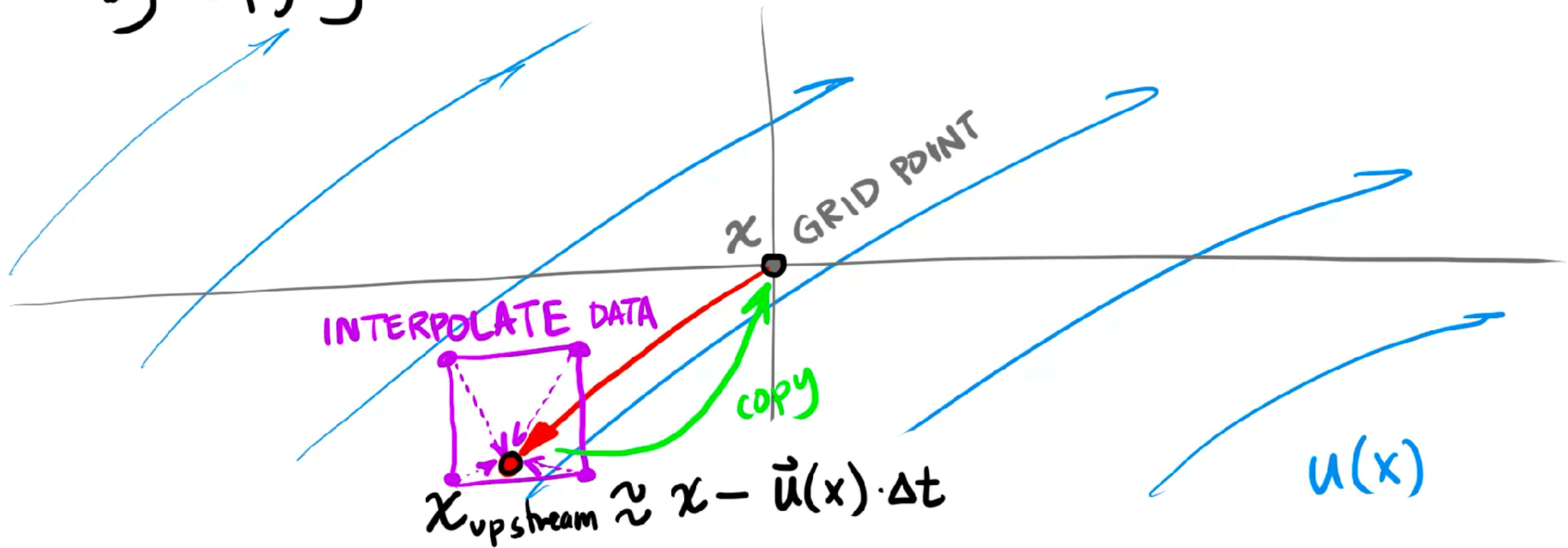
$$= \frac{\partial q}{\partial t} + \underbrace{(u \cdot \nabla) q}_{\text{directional derivative}}$$



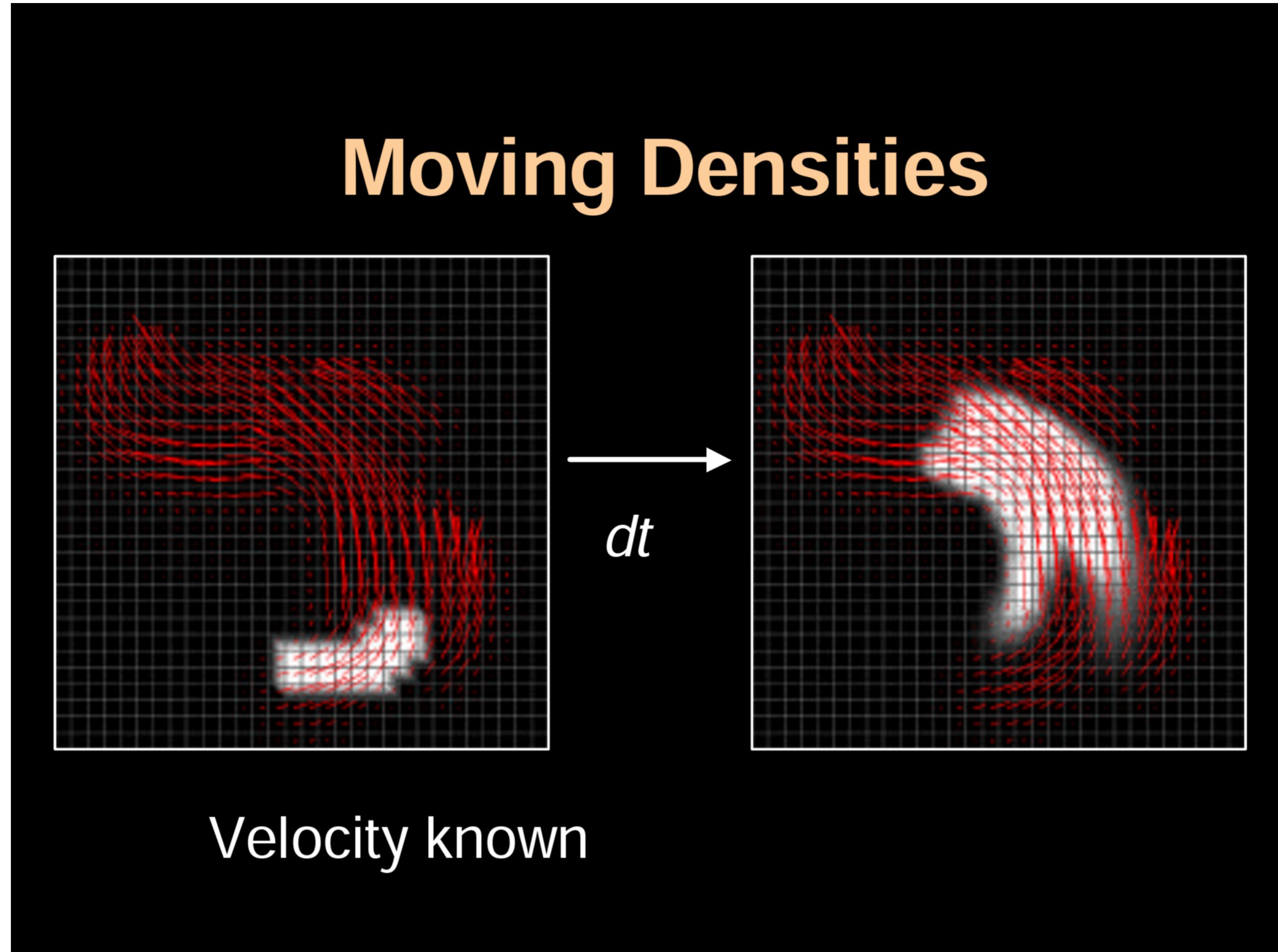
Important method: Semi-Lagrangian advection

Quantities stored at gridpoints.

Semi-Lagrangian advection updates values at time t to time $t + \Delta t$ by copying values from upstream.



Important method: Semi-Lagrangian advection

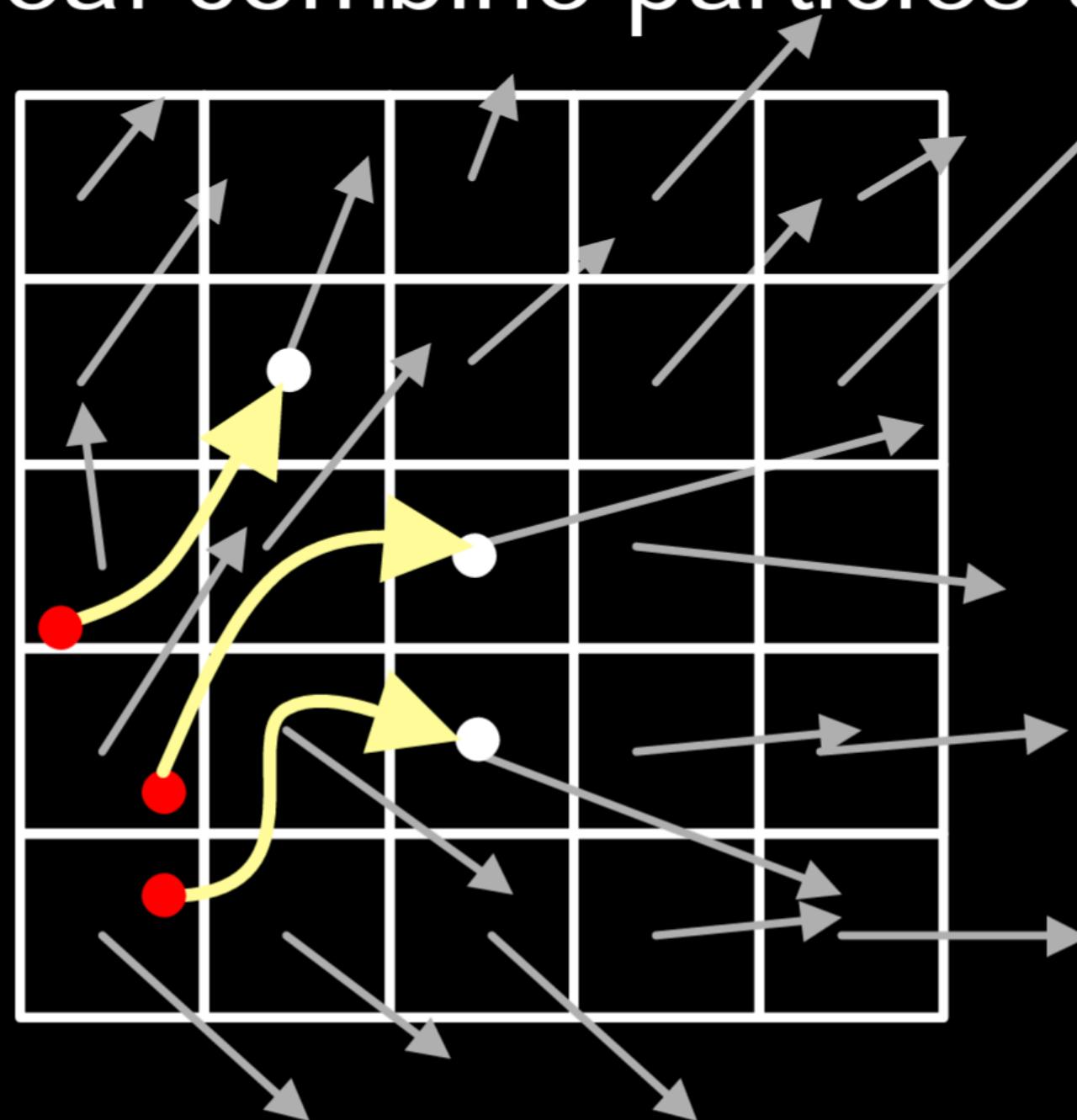


From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection

Moving Densities

Key Idea: combine particles and grids

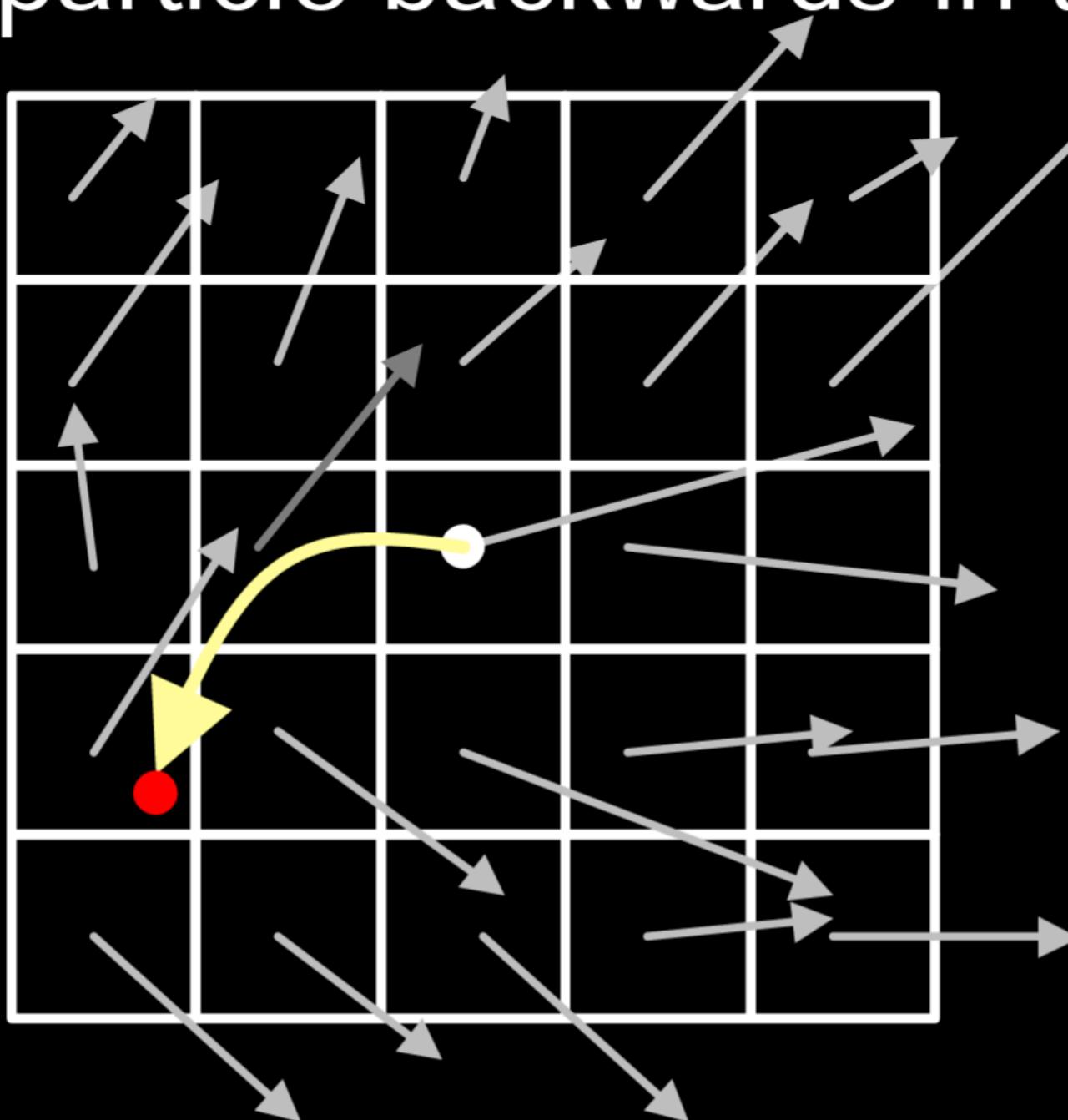


From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection

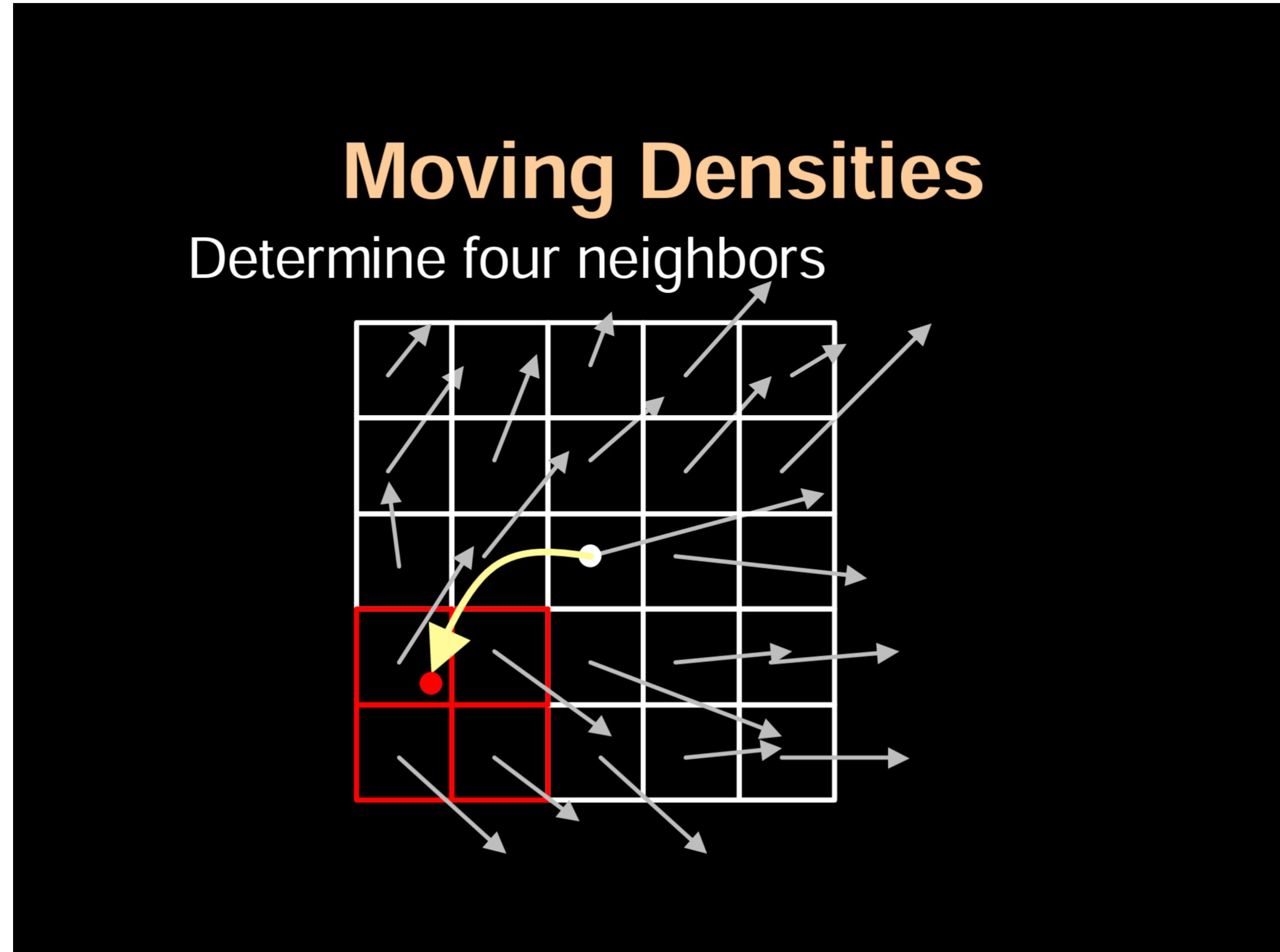
Moving Densities

Trace particle backwards in time



From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection

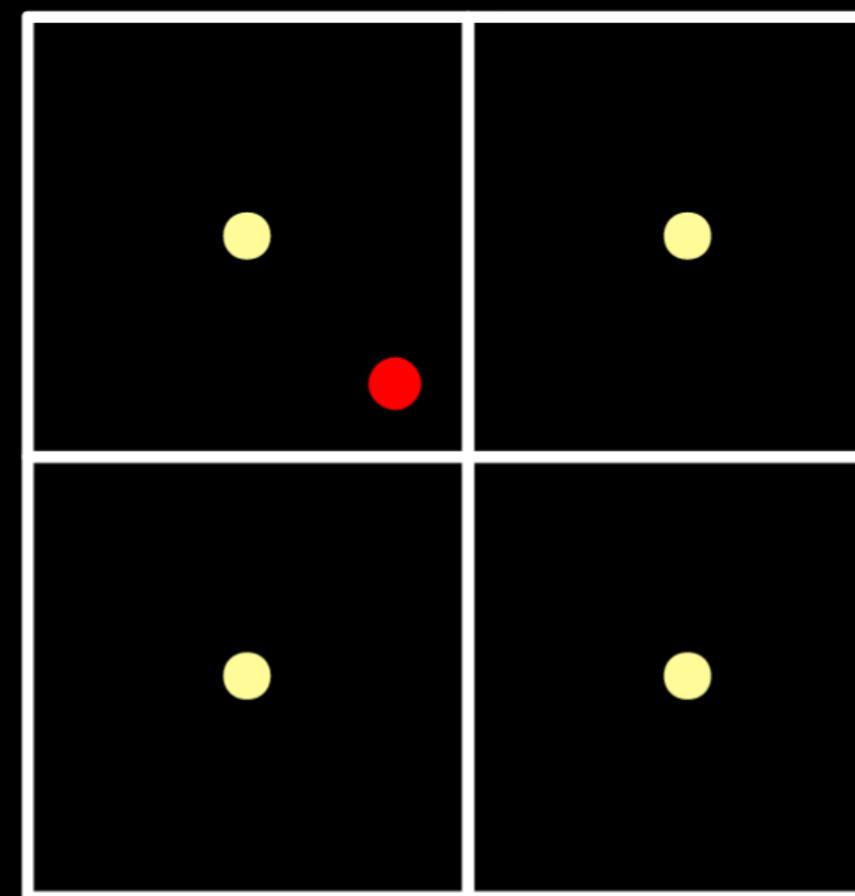


From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection

Moving Densities

Interpolate the density at new location



From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection

Moving Densities

This scheme is unconditionally stable:

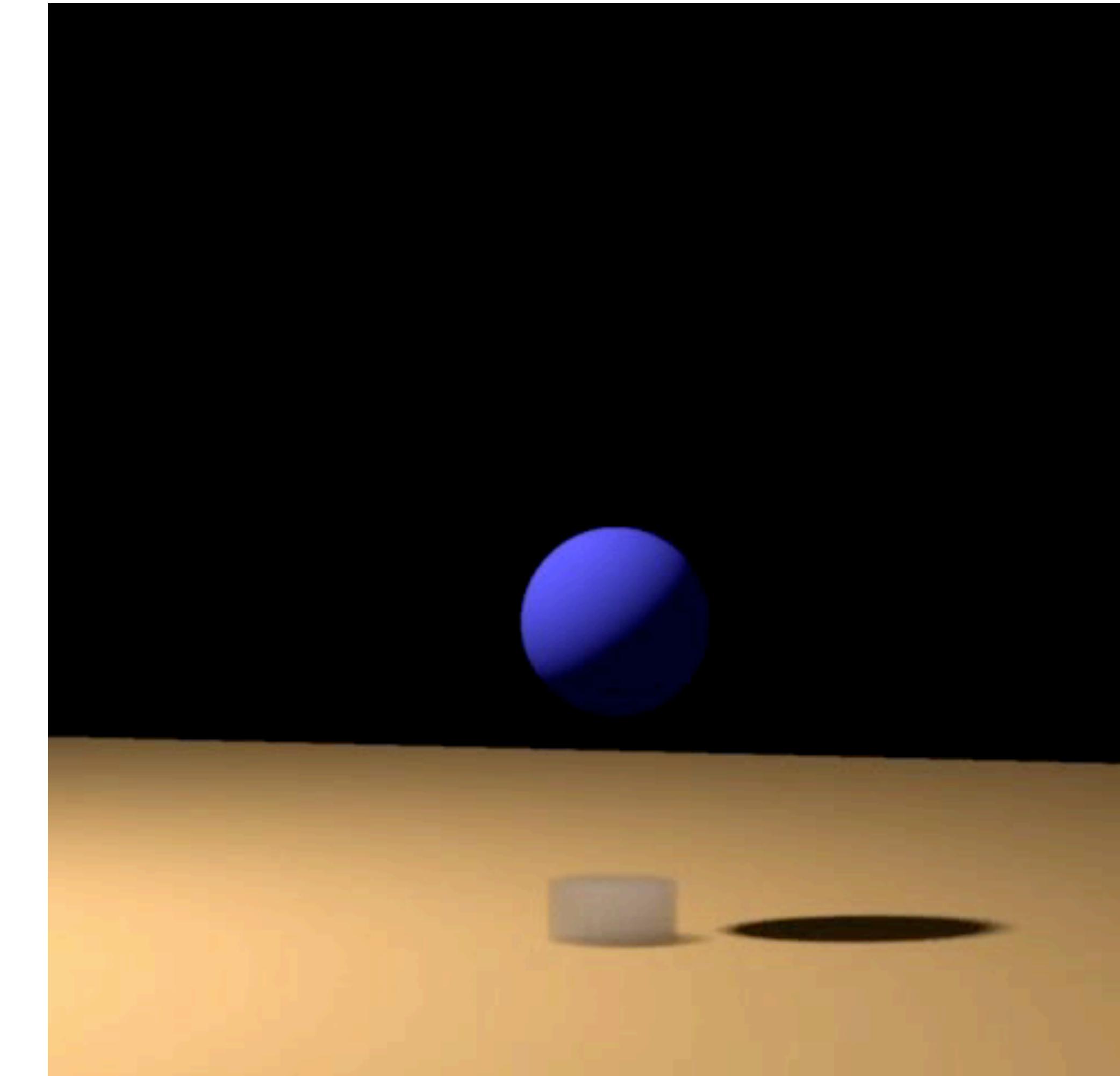
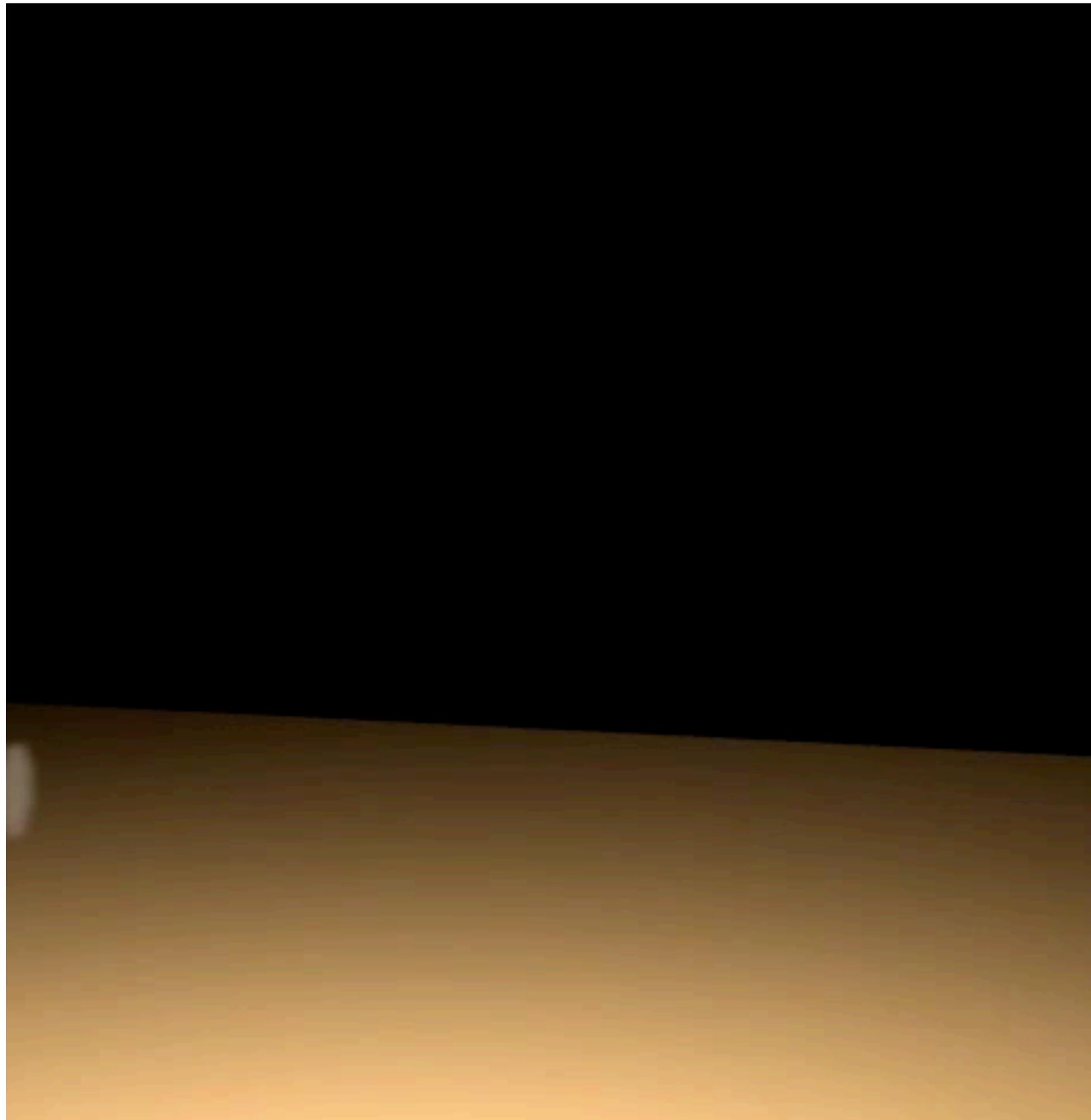
$$\rho_{int} = (1 - s)\rho_0 + s\rho_1$$

$$\rho_0, \rho_1 \leq \rho_{max}$$

$$\rho_{int} \leq (1 - s + s)\rho_{max} \leq \rho_{max}$$

→ density is always bounded

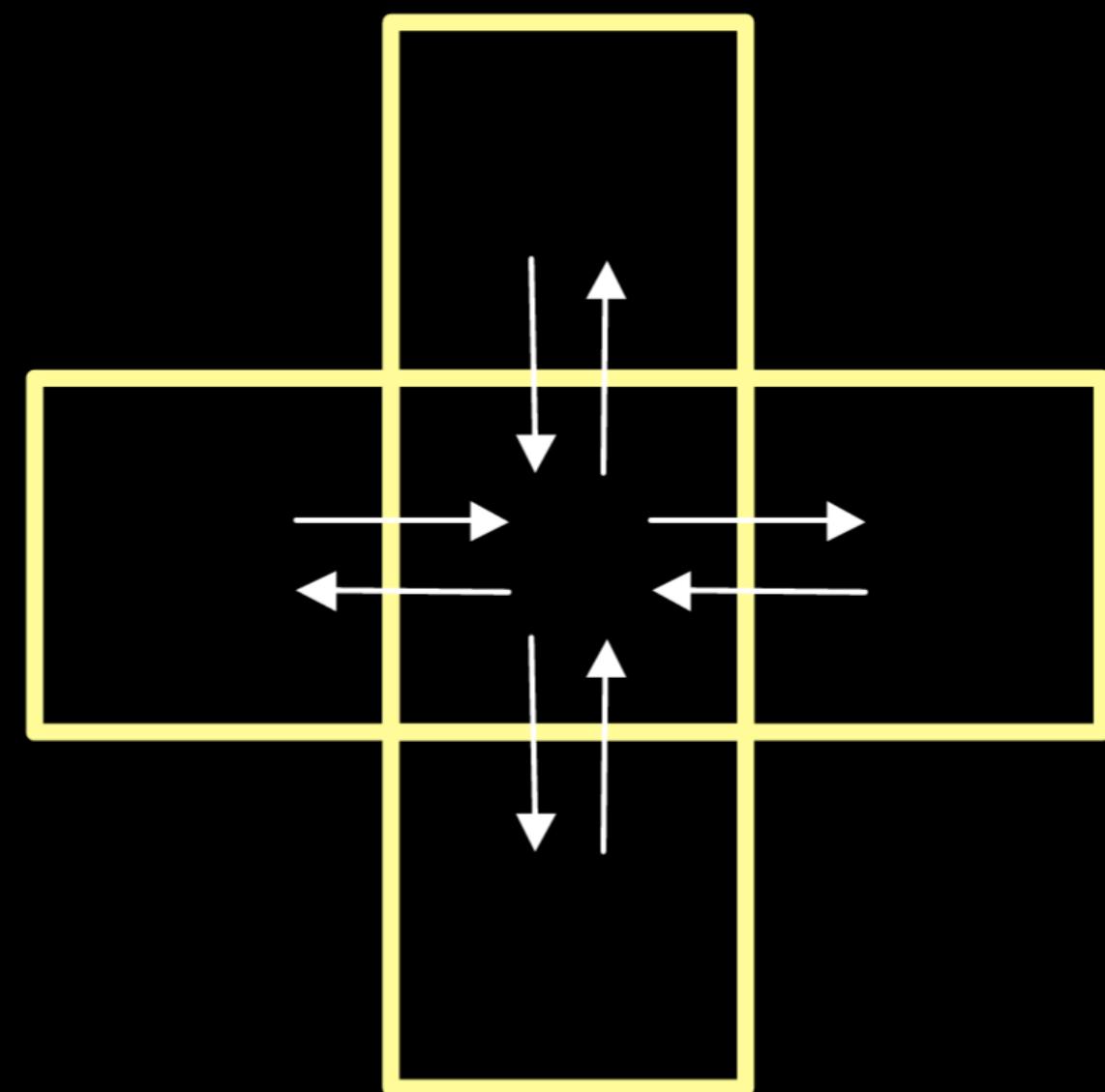
Semi-Lagrangian advection [Fedkiw et al. 2001]



Enforcing Incompressibility via Projection

Key idea: Helmholtz-Hodge Decomposition

Conservation of Mass

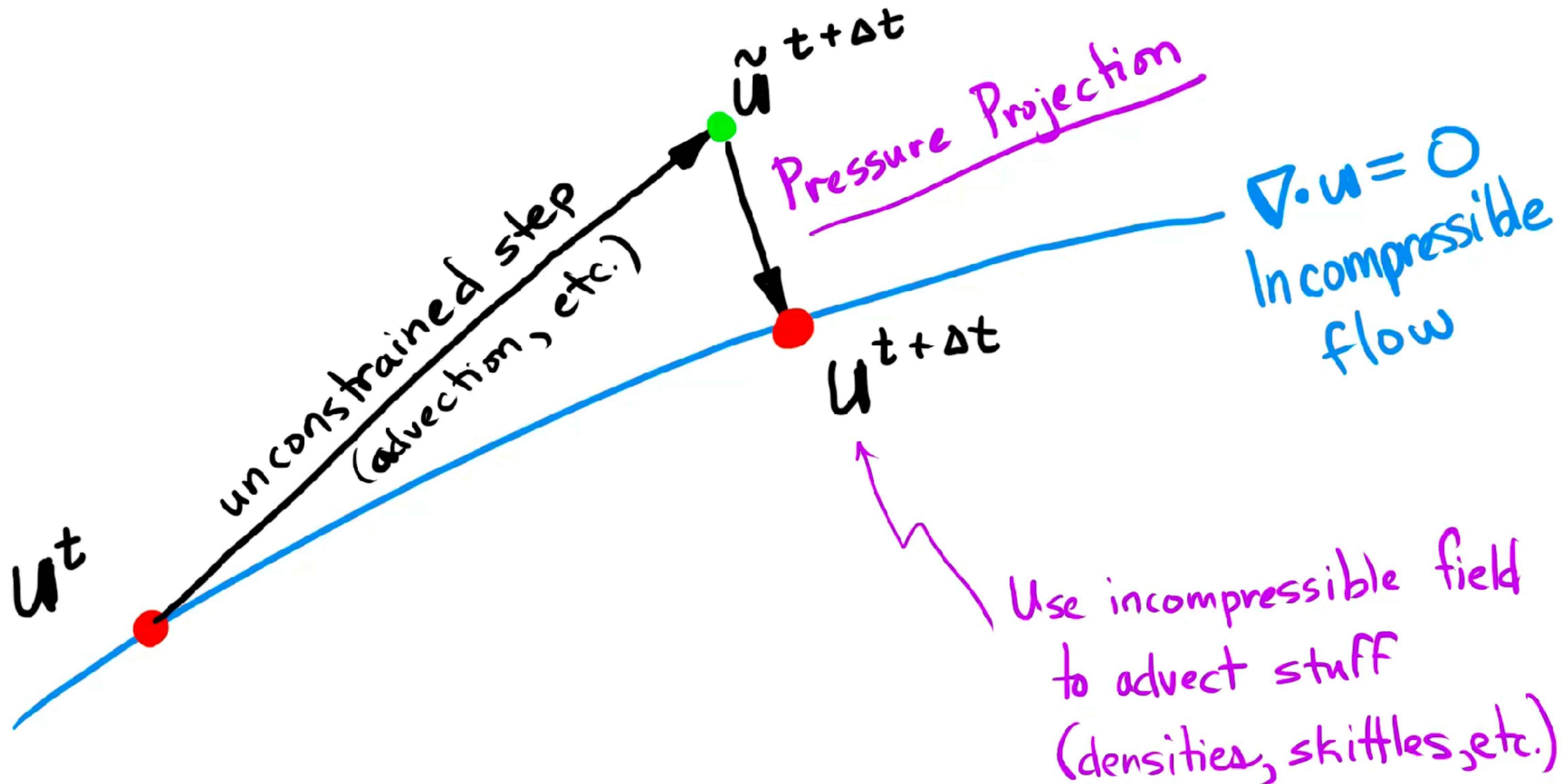


Flow into cell = Flow out of the cell

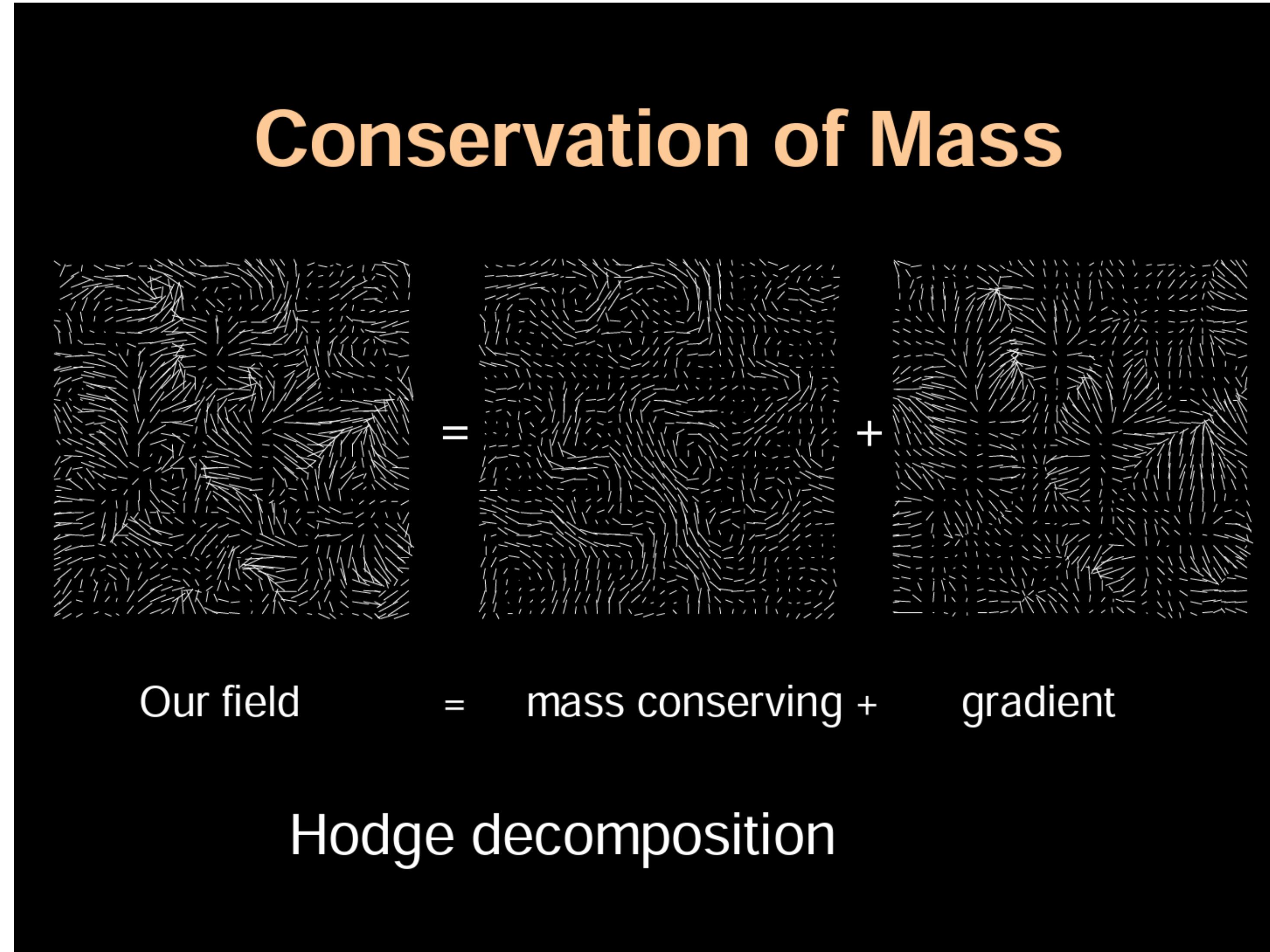
$$U_{i+1,j} - U_{i-1,j} + V_{i,j+1} - V_{i,j-1} = 0$$

From "Stable Fluids" [Stam 1999]

Geometric interpretation: Pressure projection

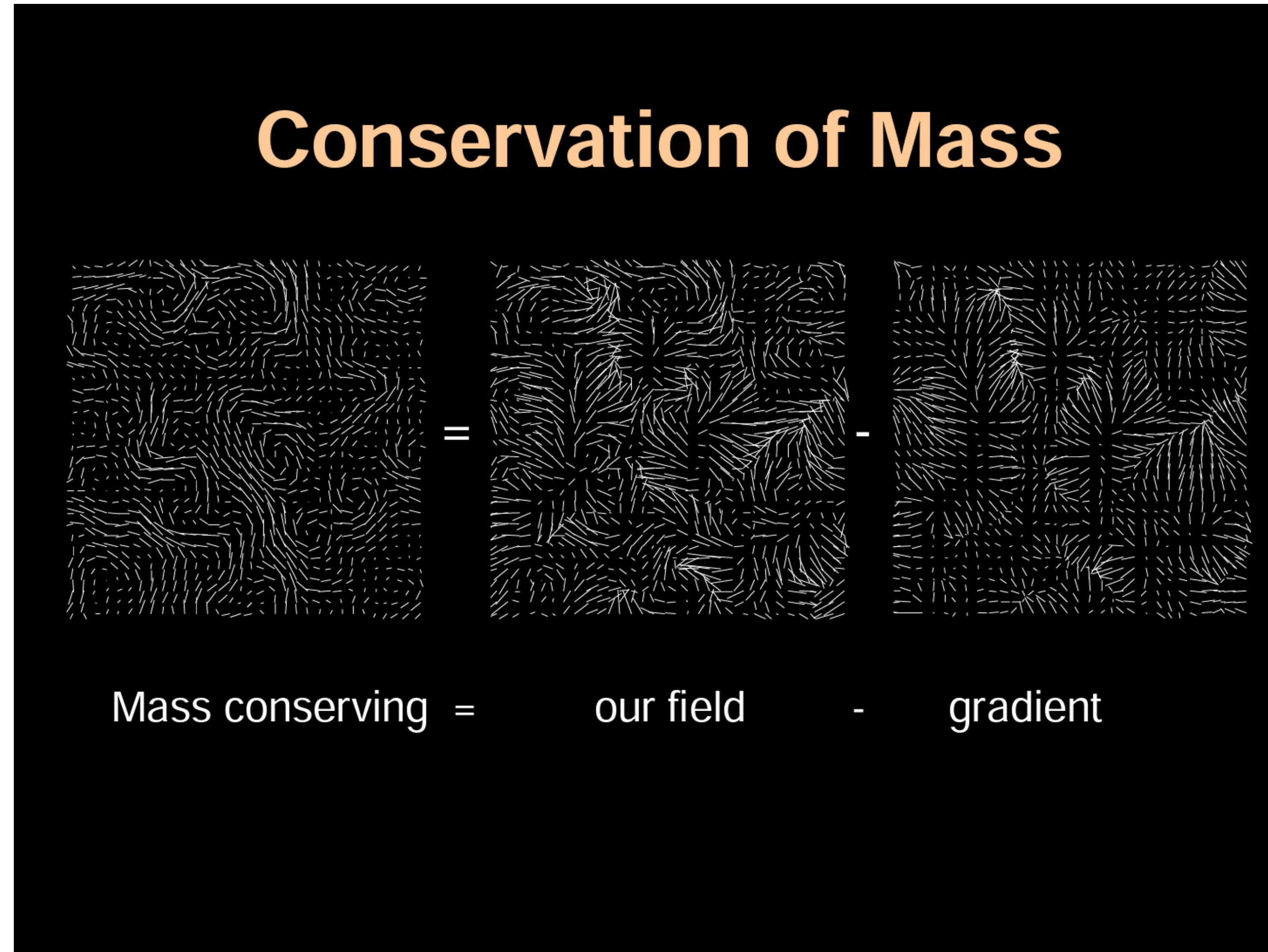


Key idea: Helmholtz-Hodge Decomposition (of $\tilde{u}^{t+\Delta t}$)



From "Stable Fluids" [Stam 1999]

Key idea: Helmholtz-Hodge Decomposition



From "Stable Fluids" [Stam 1999]

Math of pressure projection

$$\tilde{u} = u + \nabla\phi$$

(where $\nabla \cdot u = 0$)

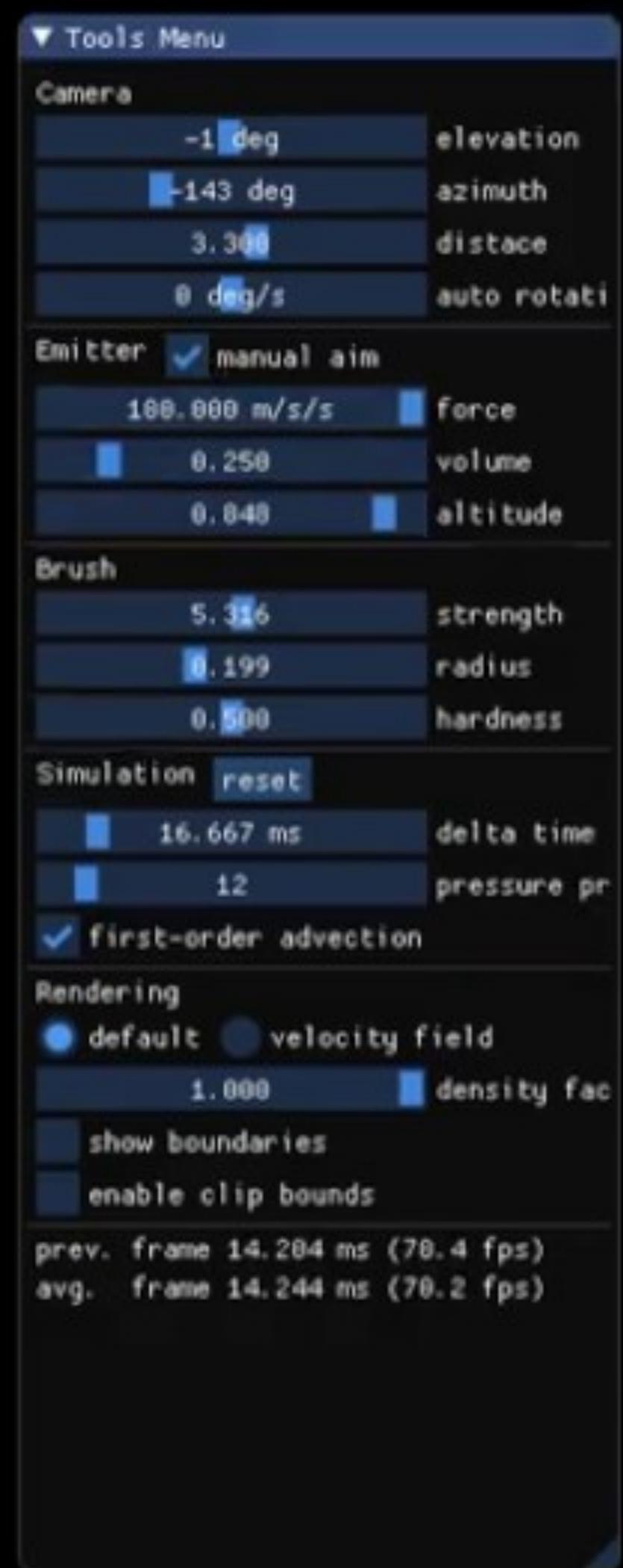
$$\nabla \cdot \tilde{u} = \cancel{\nabla \cdot u}^0 + \underbrace{\nabla \cdot \nabla \phi}_{\nabla^2}$$

$$\nabla \cdot \nabla = (\partial_x, \partial_y, \partial_z) \begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix} = \partial_x^2 + \partial_y^2 + \partial_z^2 = \nabla^2$$

SOLVE $\nabla^2 \phi = (\nabla \cdot \tilde{u})$ for ϕ ↪ ϕ is a pressure-like variable
• Linear system "A ϕ =b"

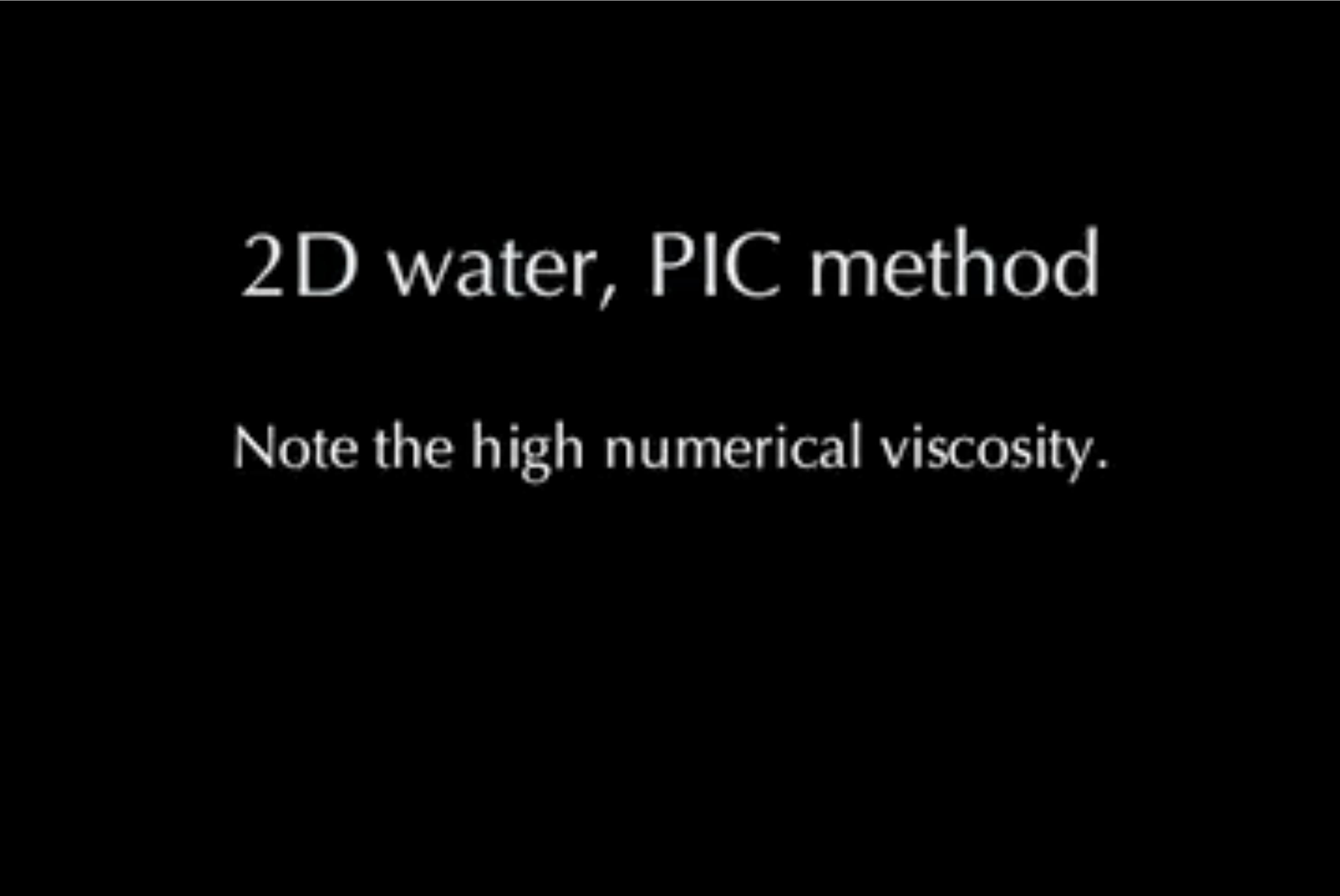
$$u = \tilde{u} - \nabla\phi$$

(Lagrange multiplier
for the constraint)



Hybrid schemes

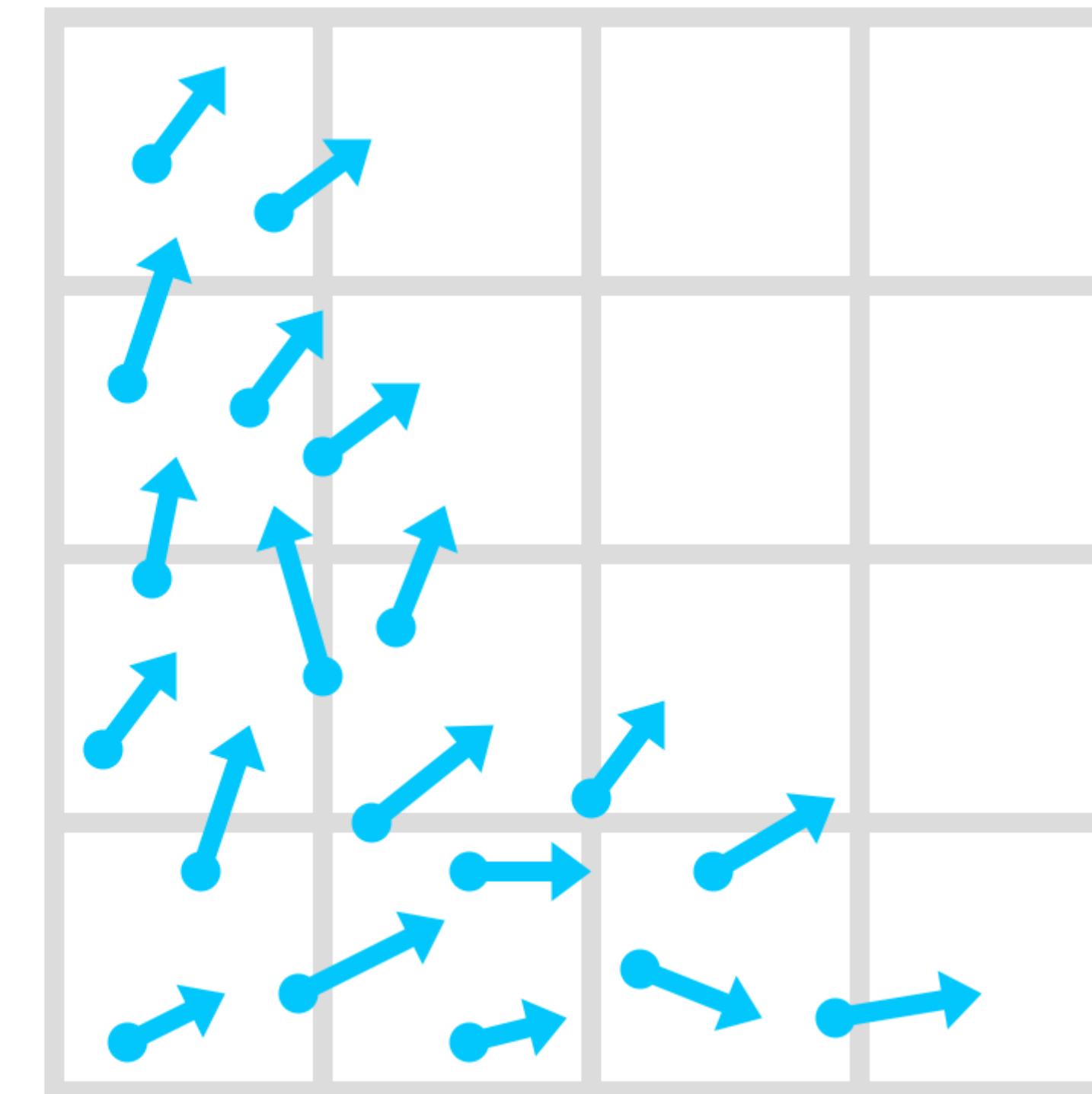
Particle-In-Cell (PIC)



2D water, PIC method

Note the high numerical viscosity.

- Start with particles
- Transfer to grid
- Resolve forces on grid
 - Gravity, boundaries, pressure, etc.
- Transfer velocity back to particles
- Advect: move particles

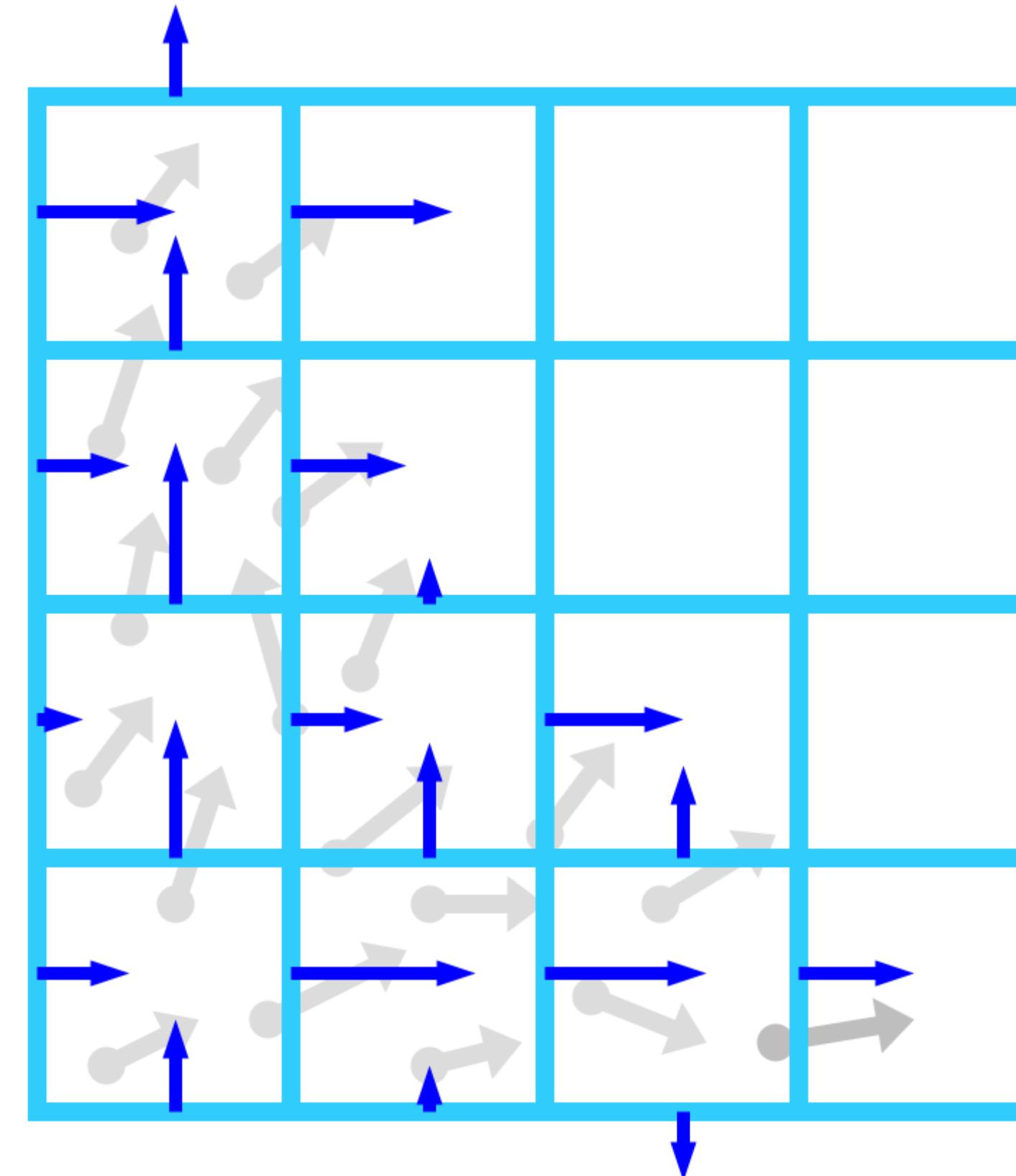


PIC

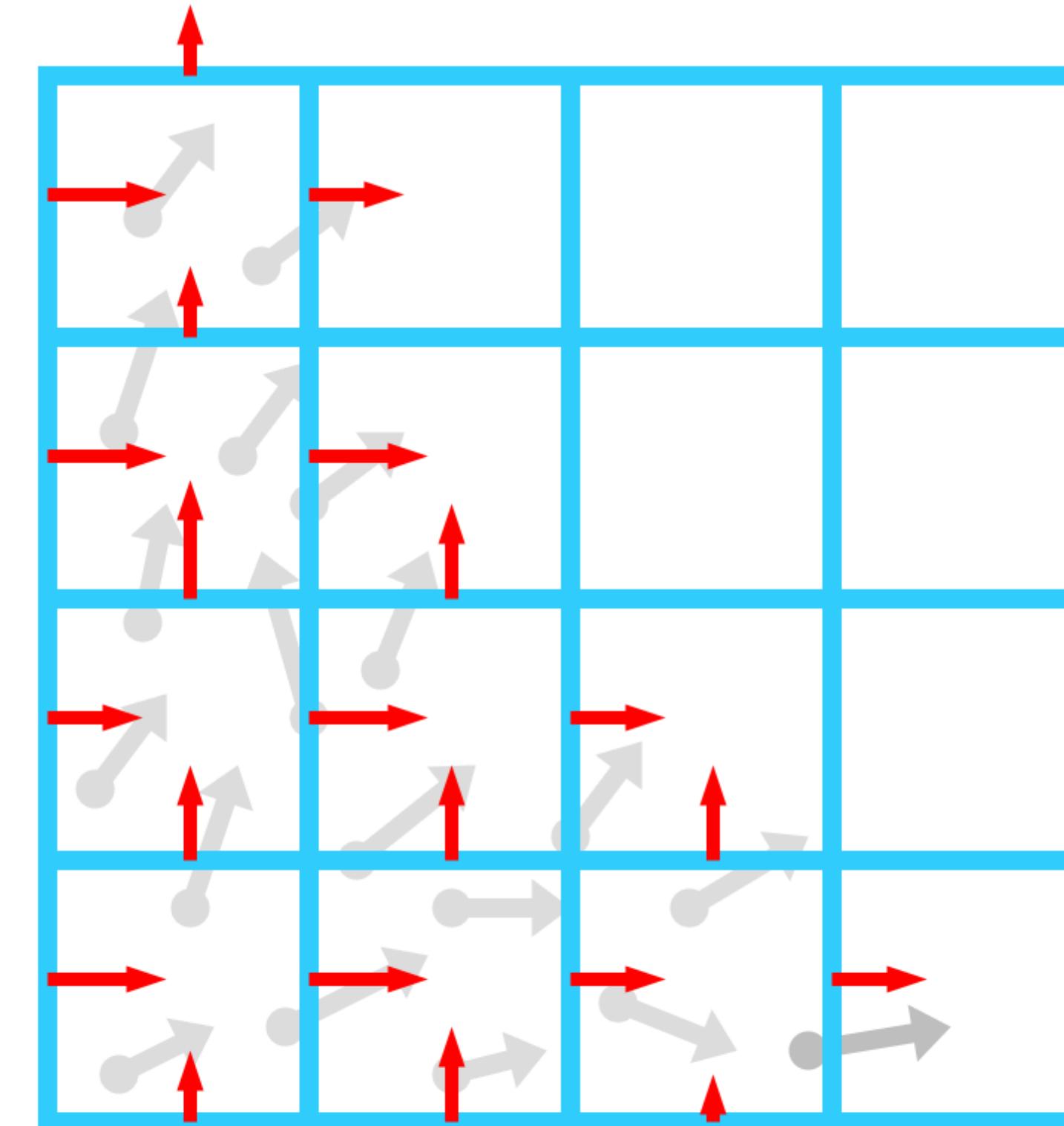


SIGGRAPH2006

- Start with particles
- Transfer to grid
- Resolve forces on grid
 - Gravity, boundaries, pressure, etc.
- Transfer velocity back to particles
- Advect: move particles



- Start with particles
- Transfer to grid
- Resolve forces on grid
 - Gravity, boundaries, pressure, etc.
- Transfer velocity back to particles
- Advect: move particles

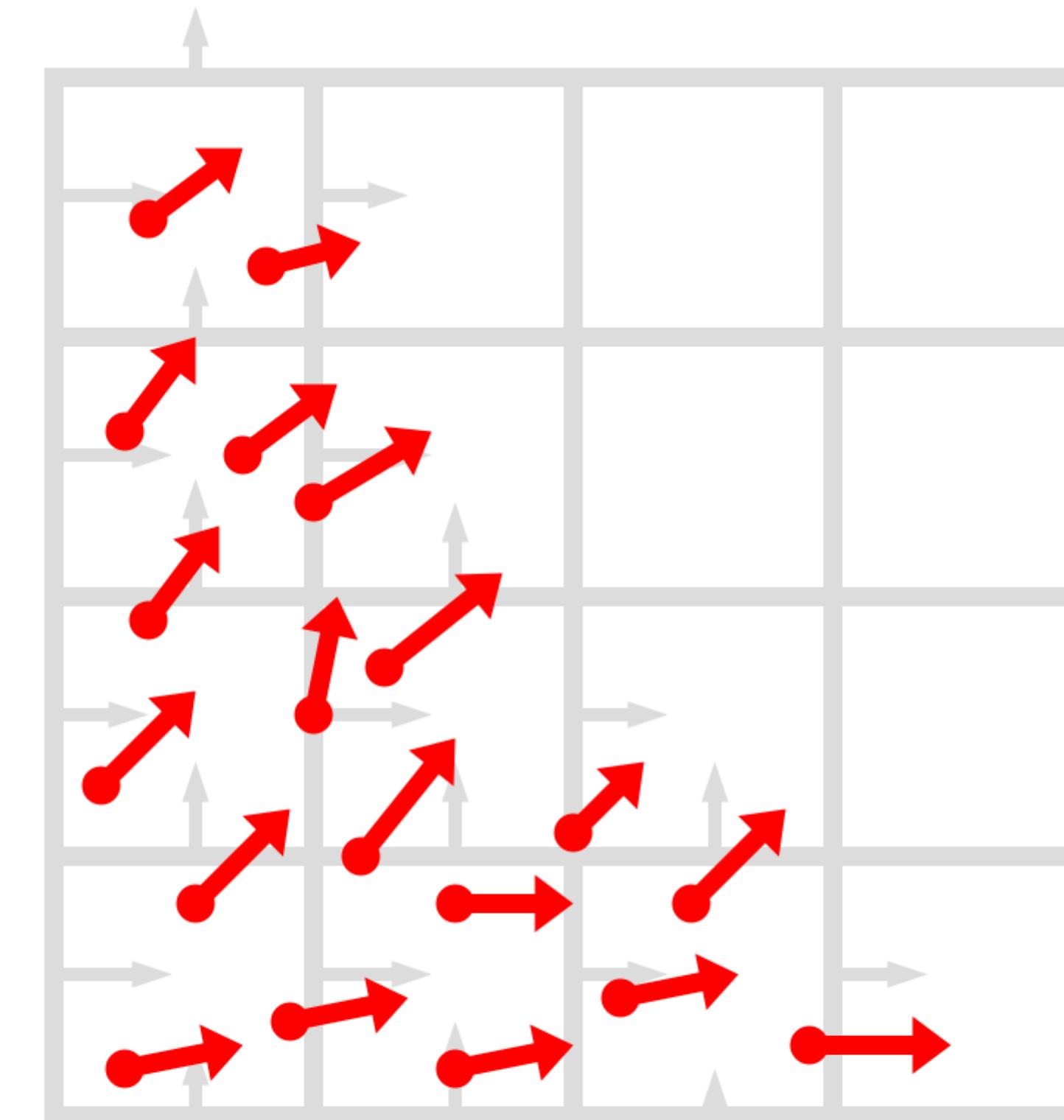


PIC



SIGGRAPH2006

- Start with particles
- Transfer to grid
- Resolve forces on grid
 - Gravity, boundaries, pressure, etc.
- Transfer velocity back to particles
- Advect: move particles

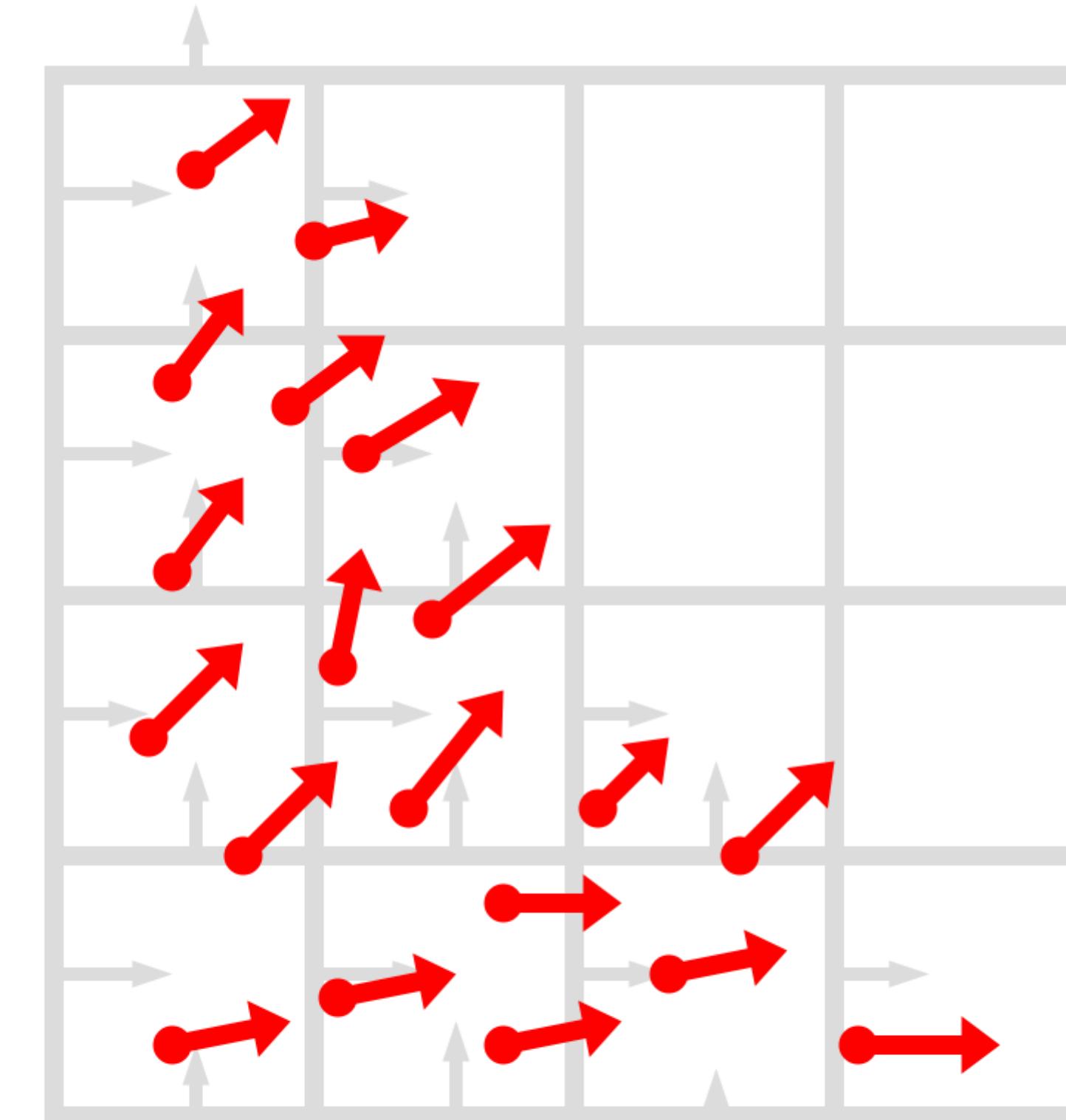


PIC



SIGGRAPH2006

- Start with particles
- Transfer to grid
- Resolve forces on grid
 - Gravity, boundaries, pressure, etc.
- Transfer velocity back to particles
- Advect: move particles



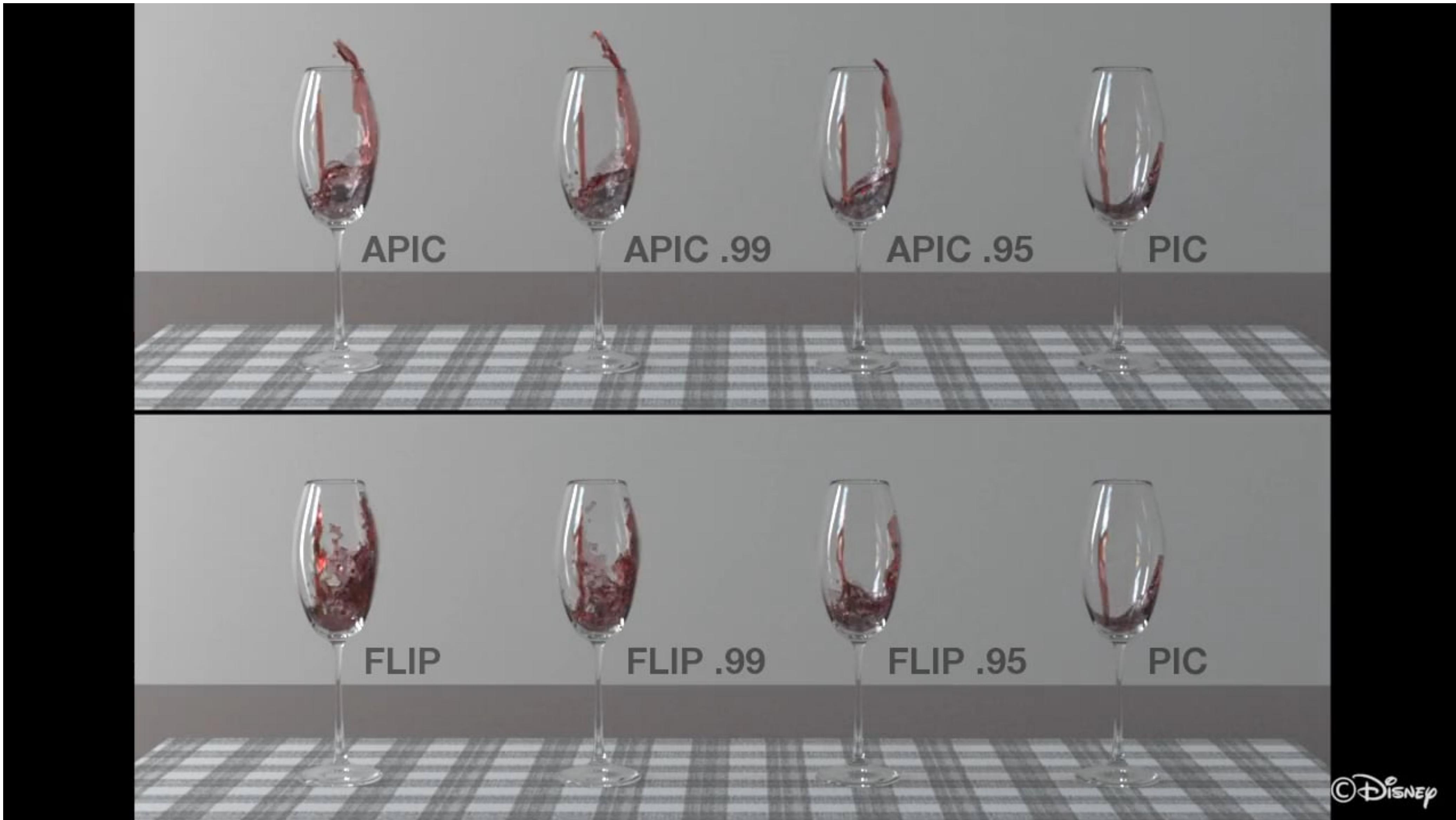
FLiquid-Implicit-Particle (FLIP): Less dissipation than PIC

Idea: Transfer back the change of a quantity from grid to particles, not the quantity itself

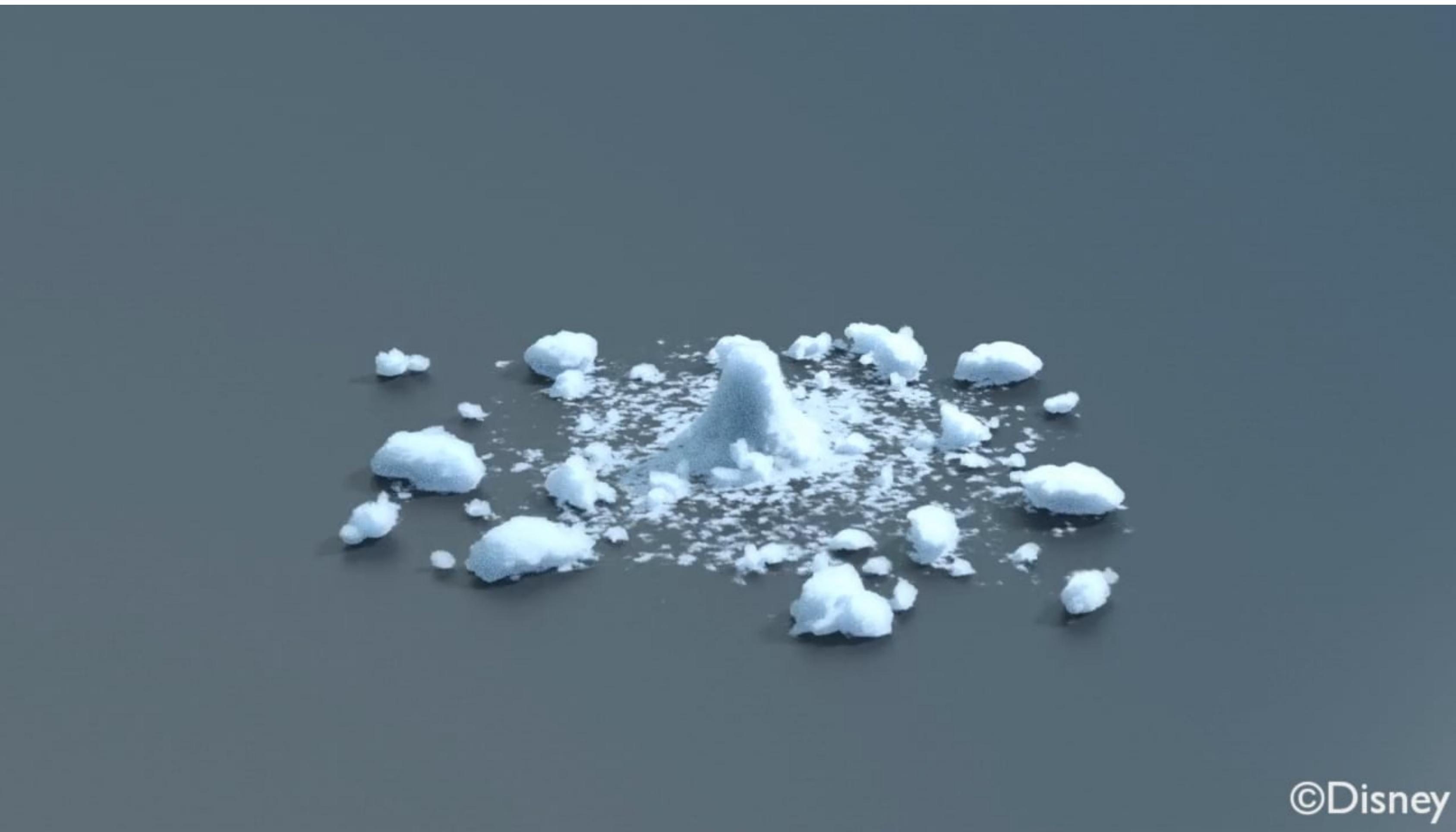


[Zhu and Bridson 2005]

Affine Particle in Cell (APIC)



Material Point Method (MPM)



A. Stomakhin, C. Schroeder, L. Chai, J. Teran, A. Selle, **A Material Point Method for Snow Simulation**,
ACM Transactions on Graphics (SIGGRAPH 2013), 32(4), pp. 102:1-102:10, 2013.

<https://vimeo.com/160322962>

Lattice Boltzmann Methods

Efficient Kinetic Simulation of Two-Phase Flows

WEI LI, Inria

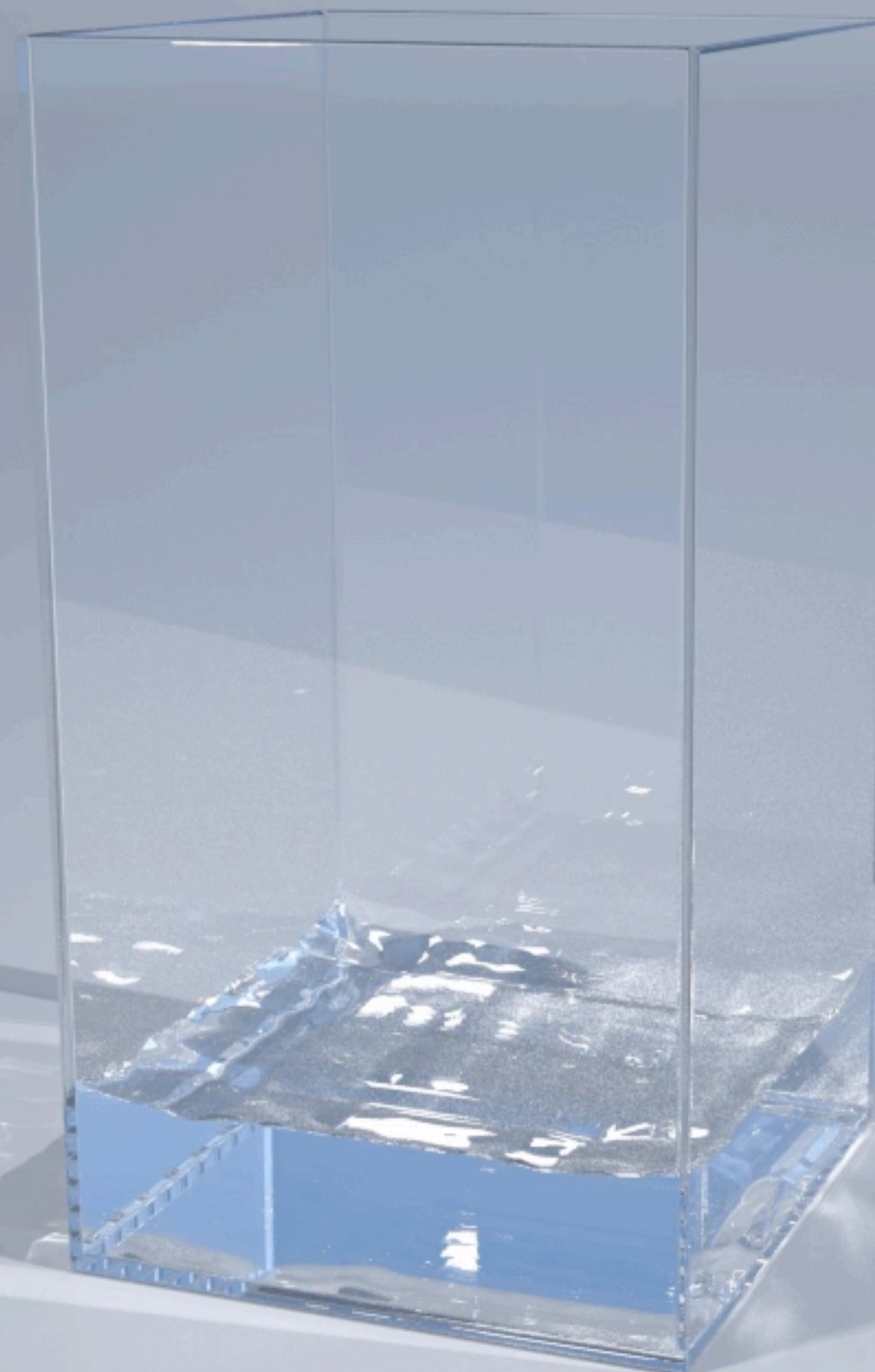
YIHUI MA, ShanghaiTech University

XIAOPEI LIU, ShanghaiTech University

MATHIEU DESBRUN, Inria / Ecole Polytechnique

ACM Transactions on Graphics (SIGGRAPH 2022)

Sound synthesis



Toward Wave-based Sound Synthesis for Computer Animation
[Wang, Qu, Langlois & James, SIGGRAPH 2018]

FLIP: Glass Pour CPU Wavesolver



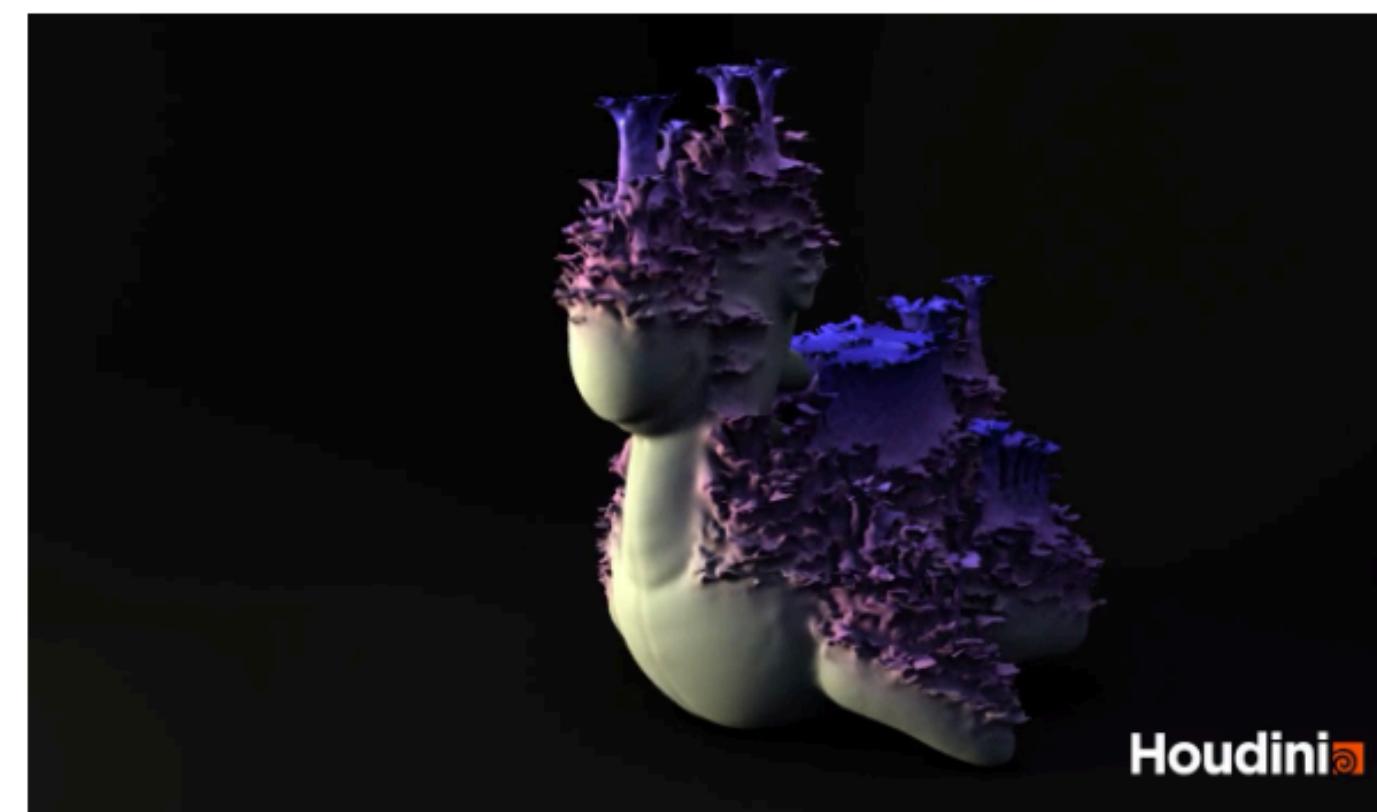
Improved Water Sound Synthesis using Coupled Bubbles
[Kangrui Xue et al. 2023]

Just scratching the surface

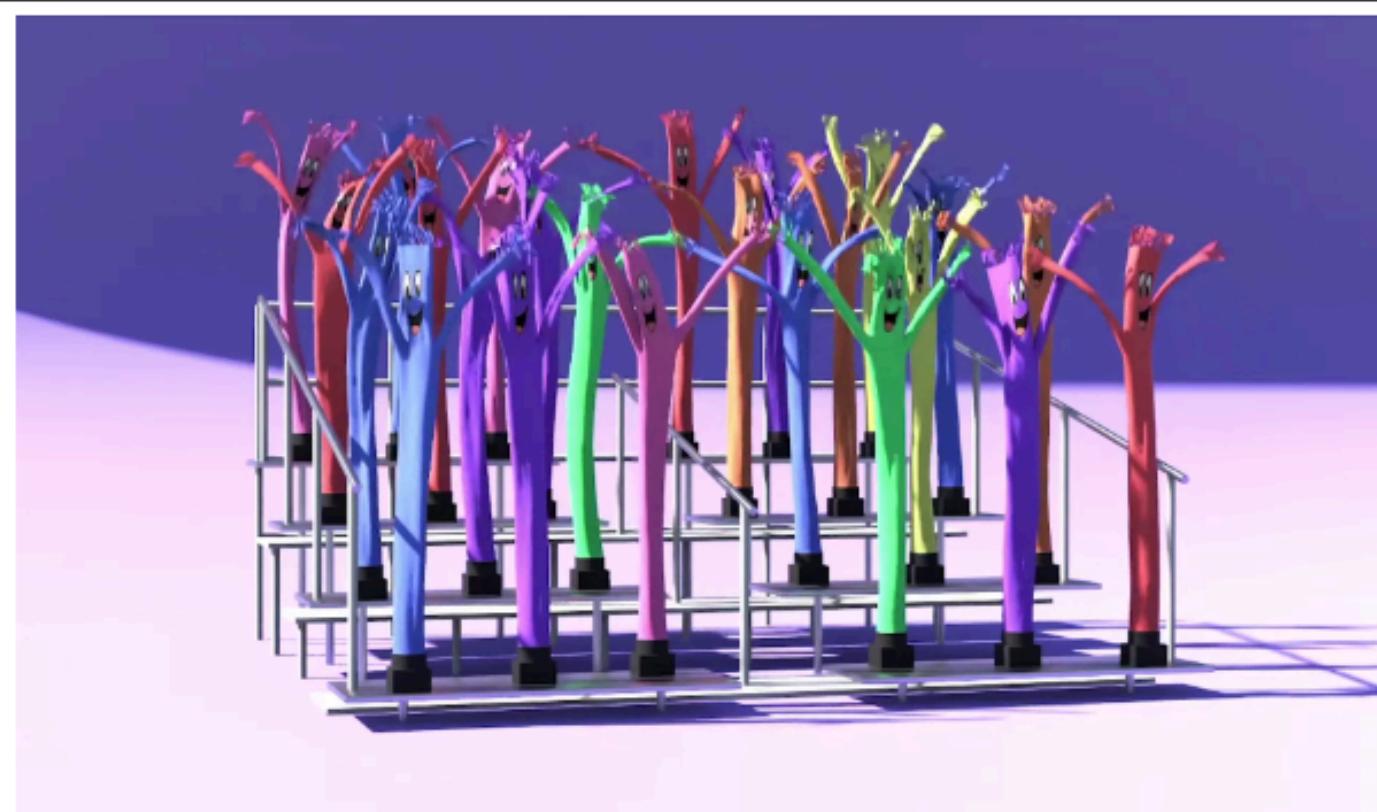
Explore more 3D fluid and solid animation techniques in CS 348C next quarter!



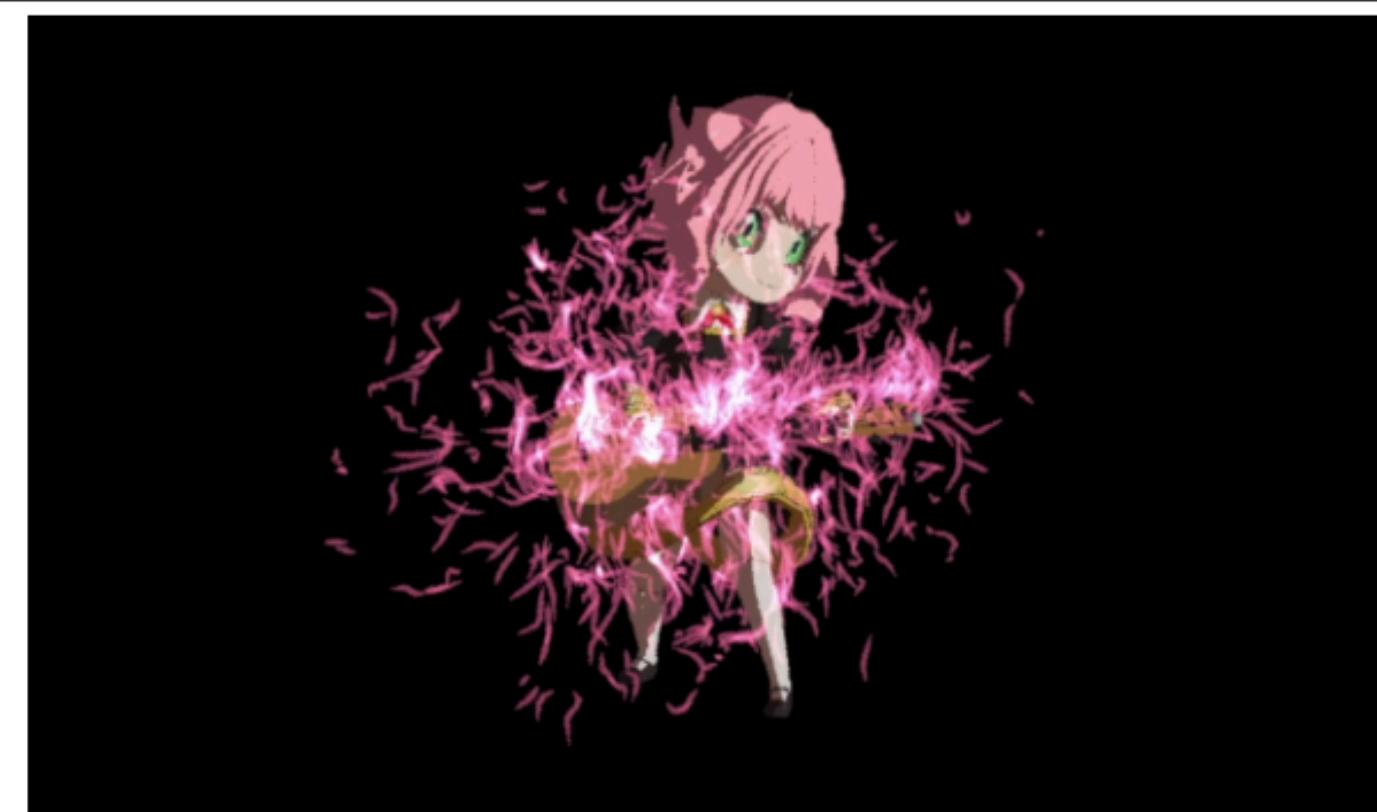
HW #1: Hello Houdini



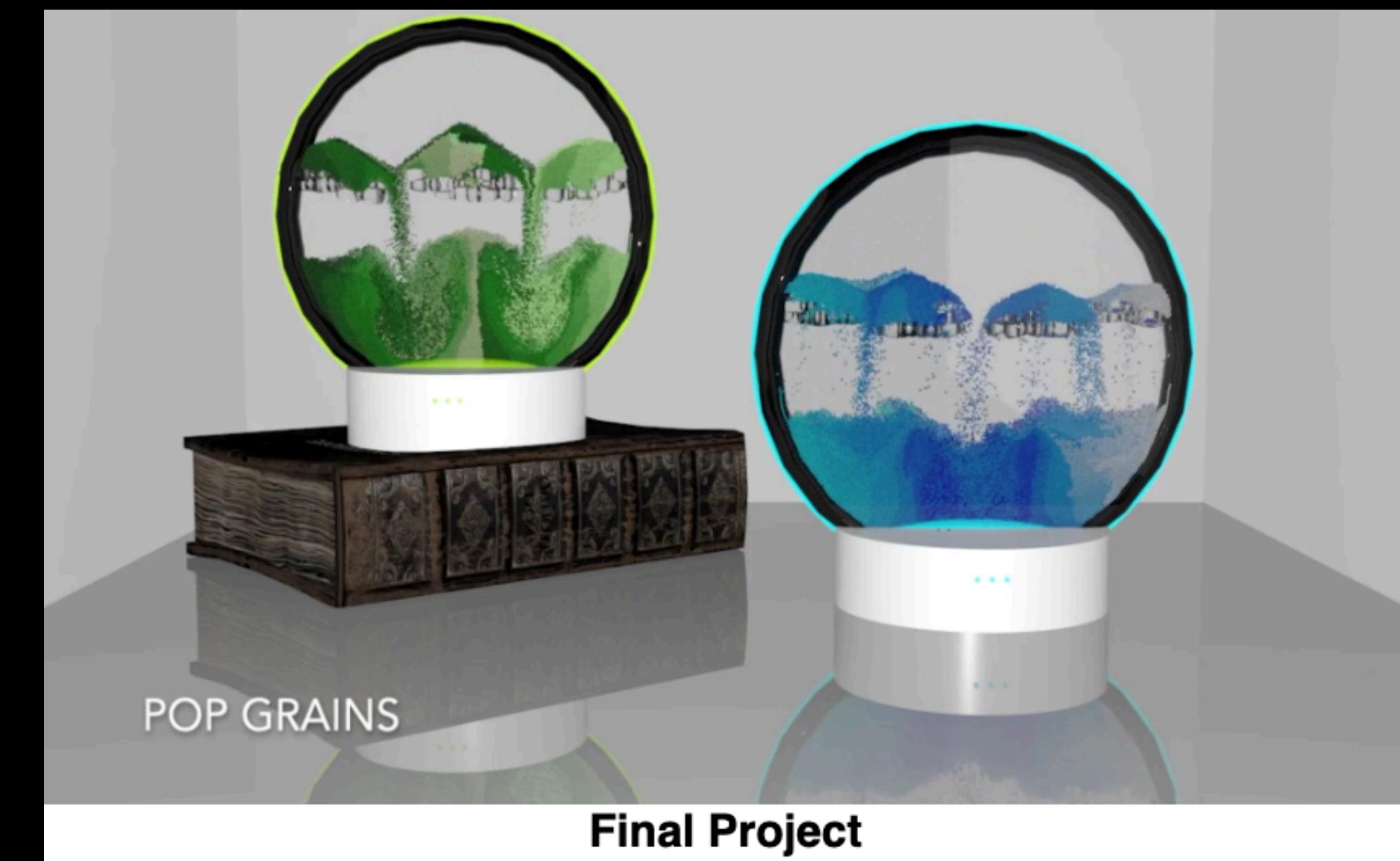
HW #2: Procedural Modeling



HW #3: Dynamics



HW #4: Character & Audio FX



POP GRAINS

Final Project