Lecture 10

Fluid Animation

A brief introduction

FUNDAMENTALS OF COMPUTER GRAPHICS

Animation & Simulation

Stanford CS248B, Fall 2022

PROFS. KAREN LIU & DOUG JAMES

Fluid Animation

Reference:

Robert Bridson and Matthias Muller-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]

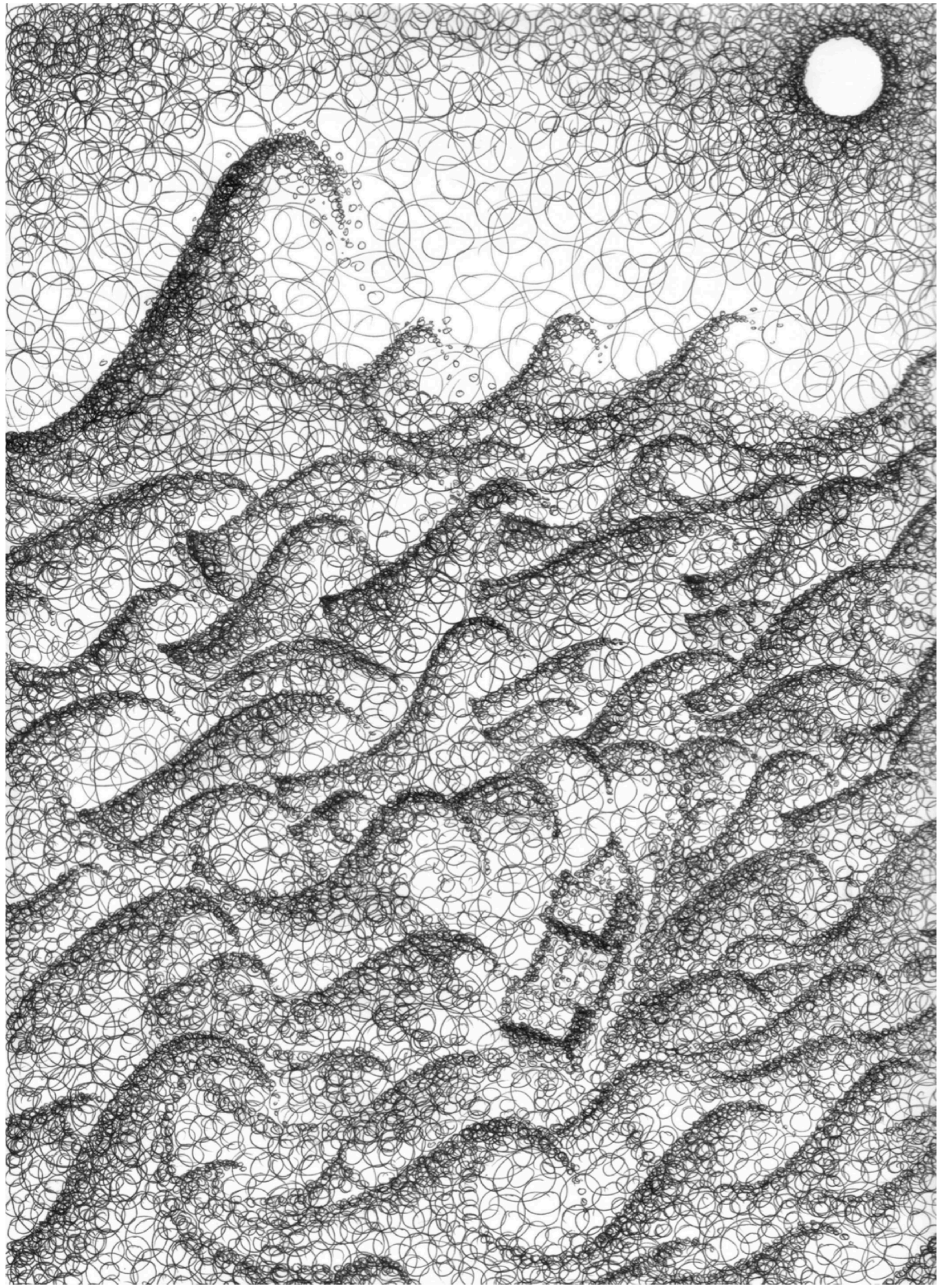
Navier-Stokes Equatiions

Momentum Equation ("a=f/m")

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

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

Incompressibility Constraint



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

Particle-based Fluids

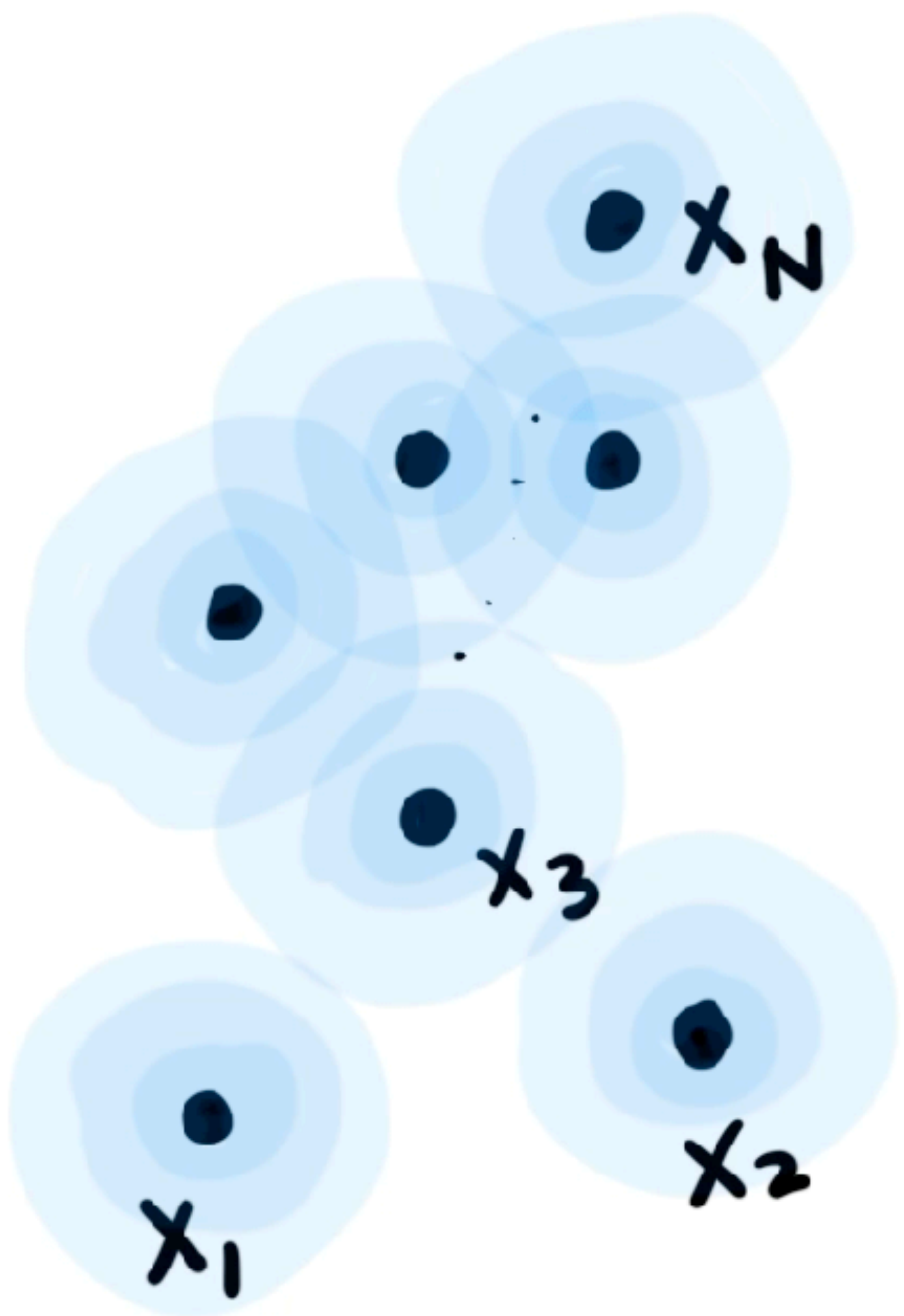
Lagrangian viewpoint: Data lives on particles

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

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



Kernel $W(r)$ models density of unit mass blob, radius d.

$\Longrightarrow$ $m_j W(\|x - x_j\|)$ is density of blob with mass m_j at position x_j.

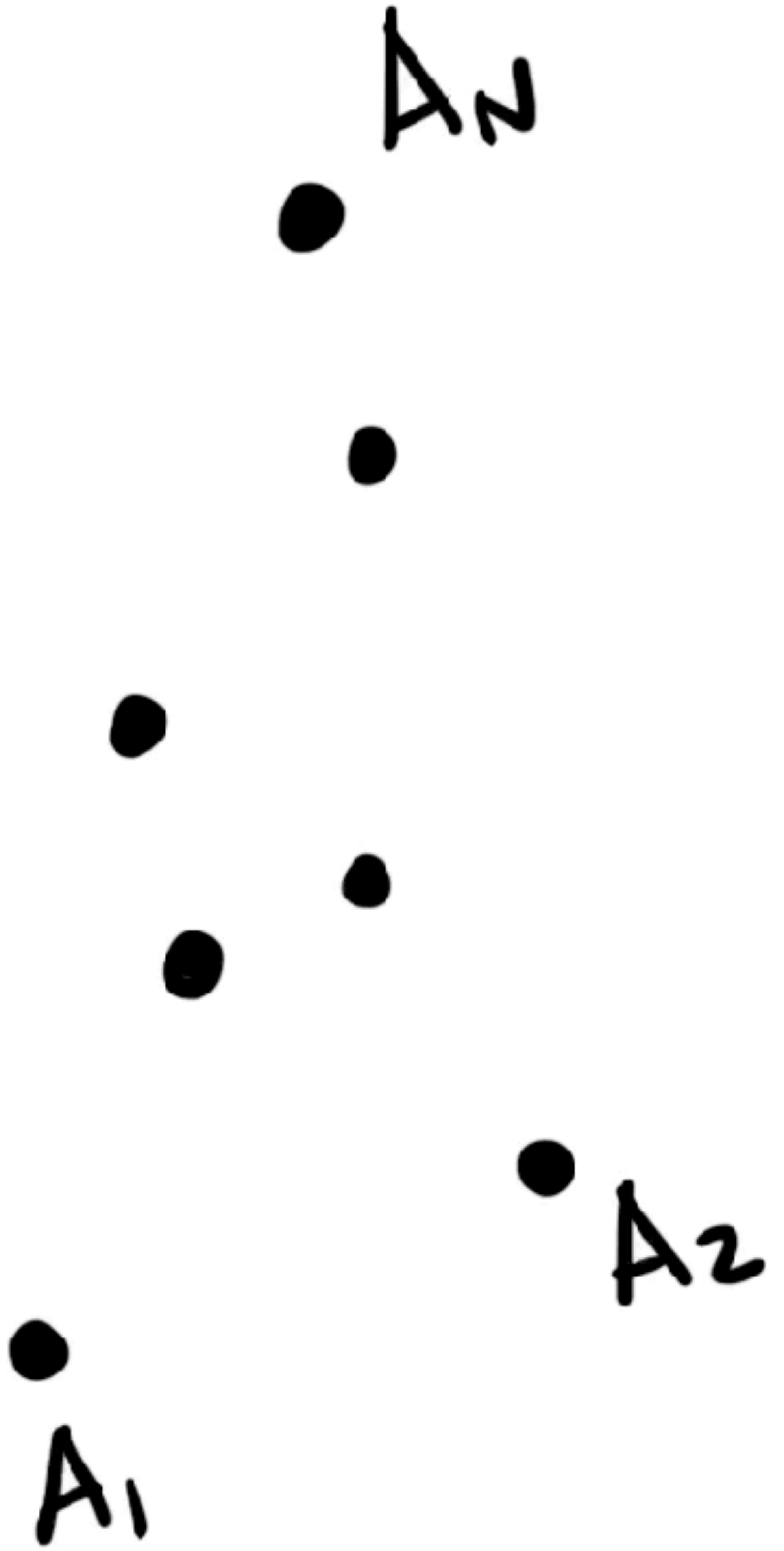
Density field of N blobs $= \rho(x) = \sum_{j=1}^{N} m_j W(\|x - x_j\|)$

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

Total Mass $= \int \rho(x)\, d^3x = \int \left(\sum_j m_j W(\|x - x_j\|) \right) d^3x = \sum_j m_j \underbrace{\boxed{\int W(\|x - x_j\|)\, d^3x}}_{1}$

$= \sum_j m_j$ ☺

SPH concepts: Any field, $A(\mathbf{x})$, from particle values, A_j



Generalize density interpolation formula:

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

"VOLUME" OF PARTICLE j

So, for <u>any</u> attributes A_j on particles,

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

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

SPH force density, $\mathbf{f}_i$ (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:** $-\nabla p(\mathbf{x})$

SPH Trick

pressure at x_i

$$f_i^{\text{pressure}} = -\nabla p(x_i) = -\sum_j \frac{m_j}{\rho_j} p_j \nabla W(\|x_i - x_j\|)$$

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

Symmetrized pressure forces:

$$f_i^{\text{pressure}} = -\sum_j \frac{m_j}{\rho_j} \frac{p_i + p_j}{2} \nabla W(\|x_i - x_j\|)$$

What is $p_i = p(x_i)$?

SPH pressure models

- Pressure is used to enforce incompressibility, $\rho(\mathbf{x}) \approx \rho_0 =$ liquid density
- Popular: Penalty-based pressure models:
  - Gas state equations:
    - Linear model (ideal gas): $p(\mathbf{x}) = k\left(\rho(\mathbf{x}) - \rho_0\right)$
    - 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

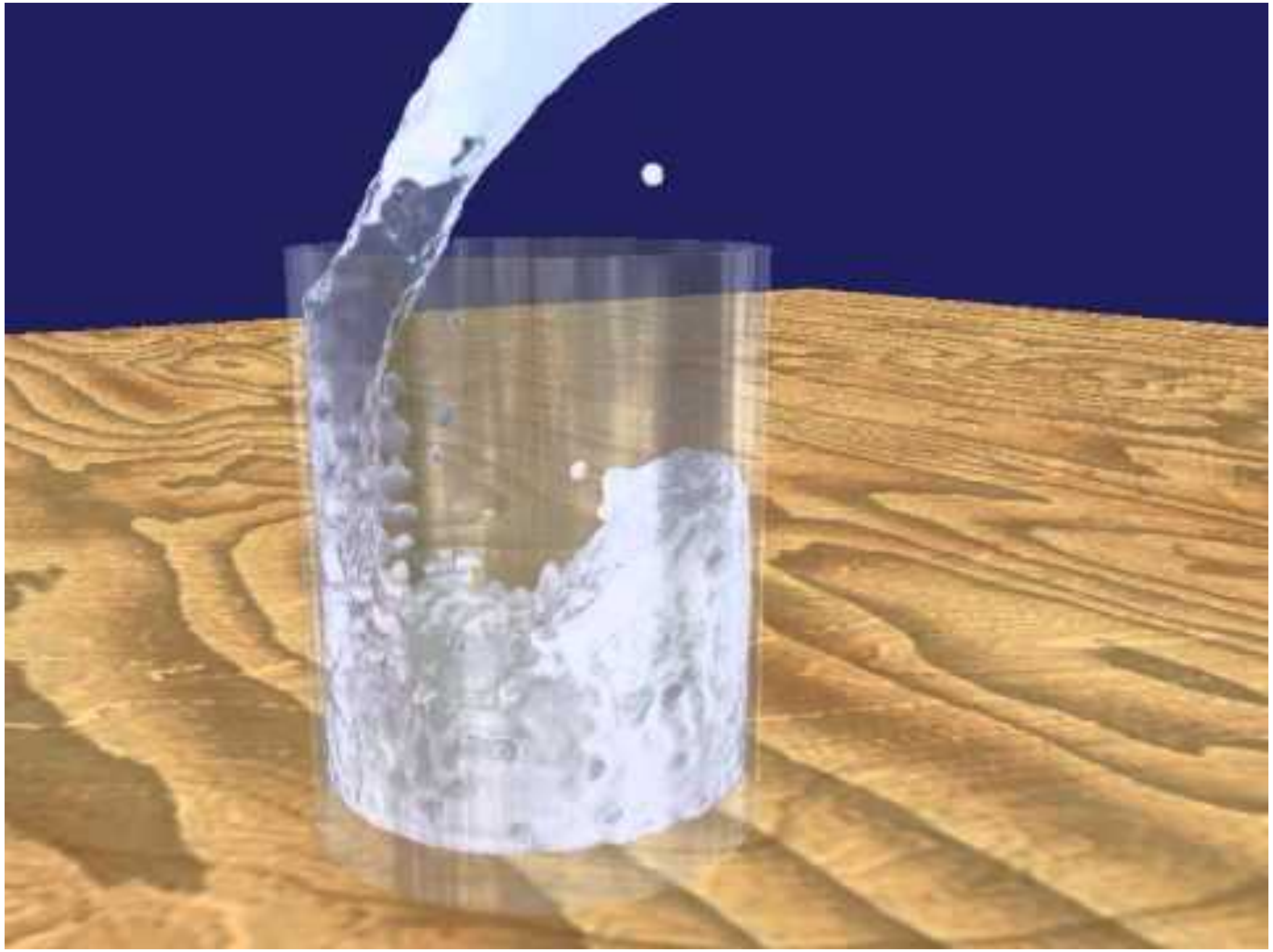
Typical SPH Implementation

Algorithm 1 SPH / WCSPH

while animating **do**
  for all i **do**
    find neighborhoods $N_i(t)$
  for all i **do**
    compute density $\rho_i(t)$
    compute pressure $p_i(t)$
  for all i **do**
    compute forces $\mathbf{F}^{p,v,g,ext}(t)$
  for all i **do**
    compute new velocity $\mathbf{v}_i(t+1)$
    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

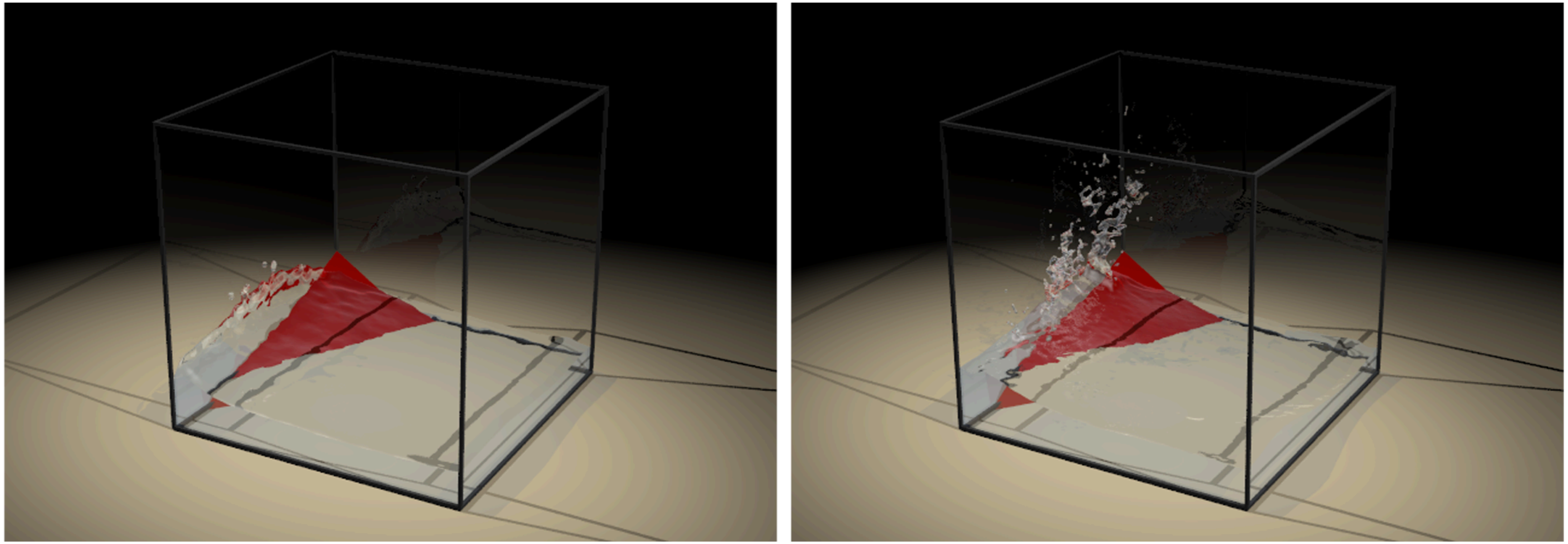


Figure 4: *Comparison of WCSPH (left, 17k particles) and PCISPH (right, 100k particles) with equal computation times.*

[Solenthaler and Pajarola 2009]

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

Implicit SPH: Predictive-Corrective Incompressible SPH (PCISPH)

Algorithm 2 PCISPH

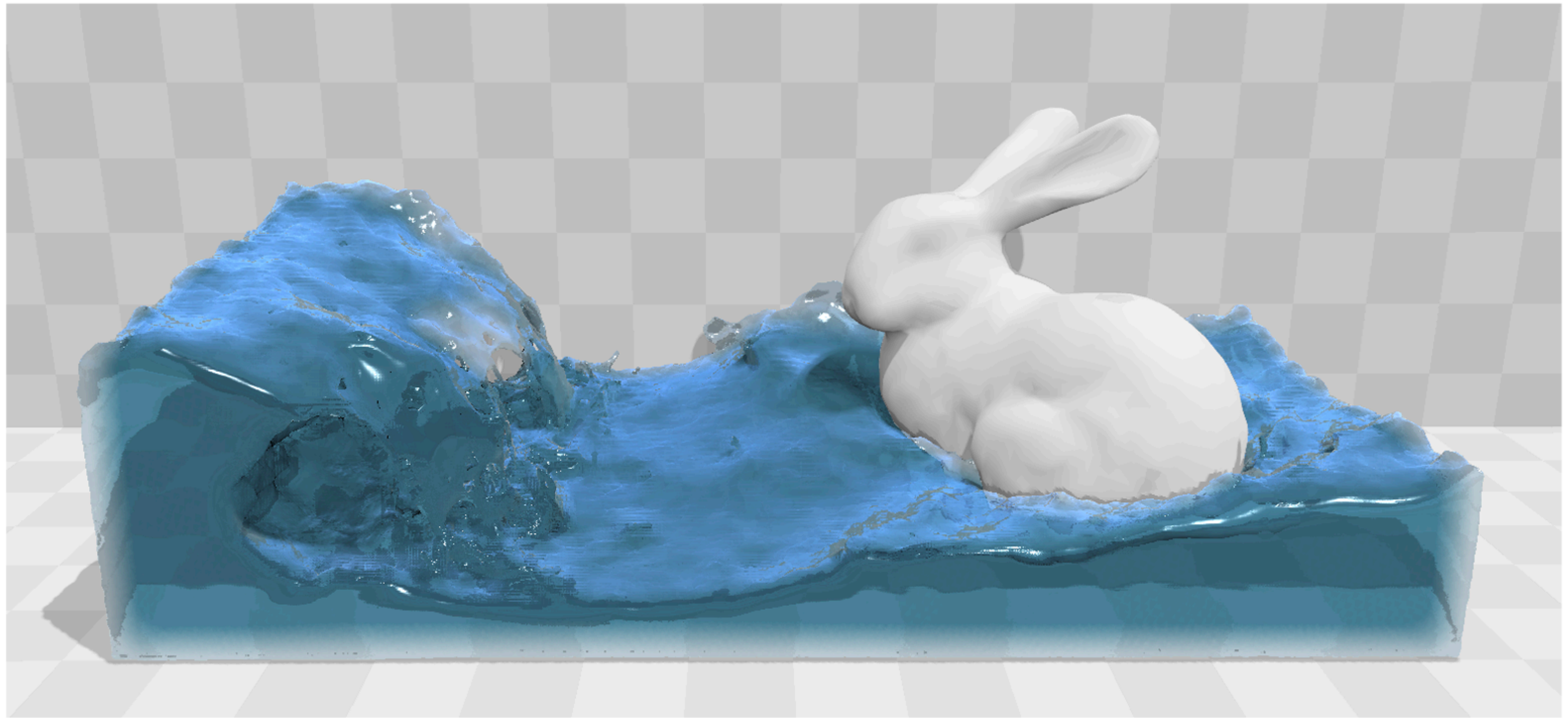
while animating **do**
    for all i **do**
        find neighborhoods $N_i(t)$
    for all i **do**
        compute forces $\mathbf{F}^{v,g,ext}(t)$
        initialize pressure $p(t) = 0.0$
        initialize pressure force $\mathbf{F}^p(t) = 0.0$
    while $(\rho^*_{err}(t+1) > \eta) \,||\, (iter < minIterations)$ **do**
        for all i **do**
            predict velocity $\mathbf{v}^*_i(t+1)$
            predict position $\mathbf{x}^*_i(t+1)$
        for all i **do**
            predict density $\rho^*_i(t+1)$
            predict density variation $\rho^*_{err}(t+1)$
            update pressure $p_i(t) += f(\rho^*_{err}(t+1))$
        for all i **do**
            compute pressure force $\mathbf{F}^p(t)$
    for all i **do**
        compute new velocity $\mathbf{v}_i(t+1)$
        compute new position $\mathbf{x}_i(t+1)$

[Solenthaler and Pajarola 2009]

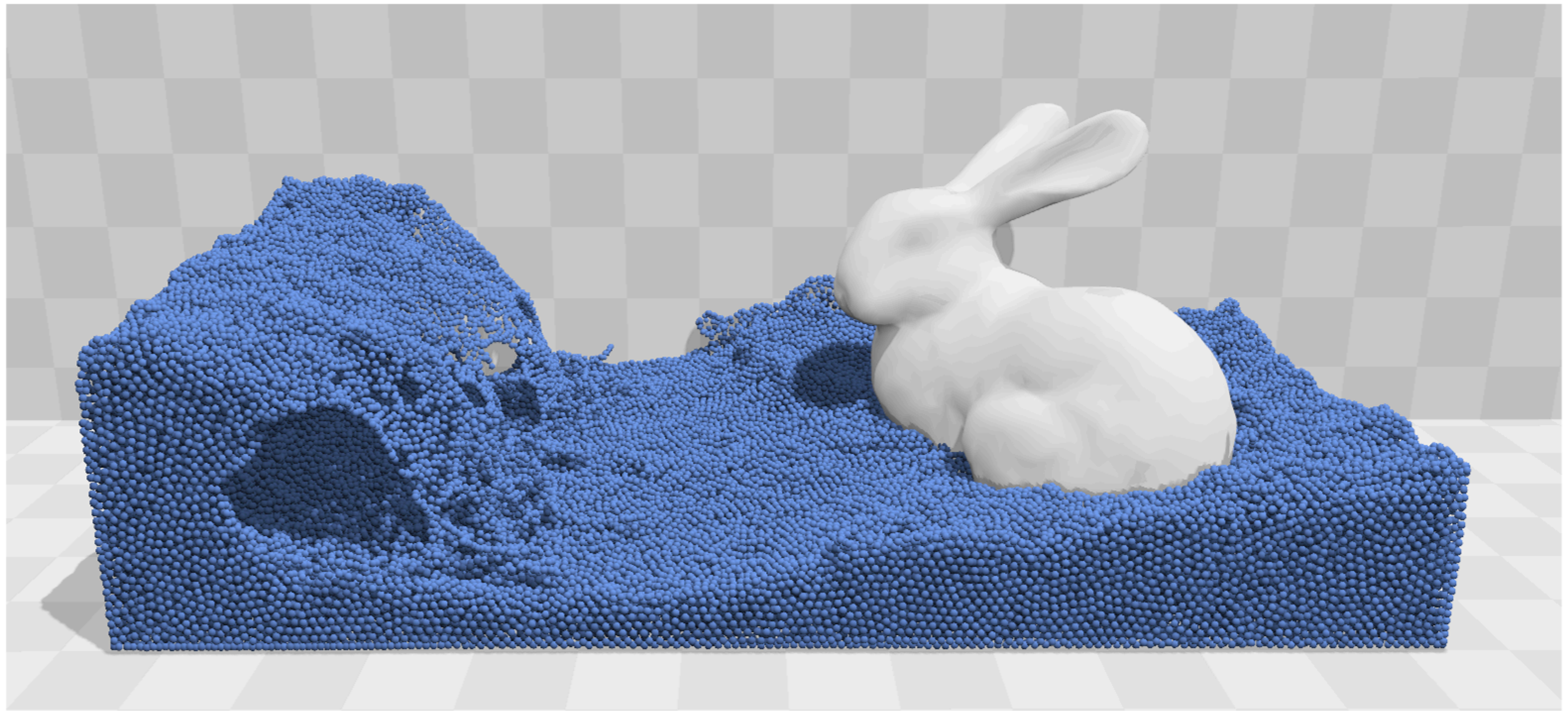
Implicit SPH: Predictive-Corrective Incompressible SPH (PCISPH)

[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

Example: Position Based Fluids [Mueller and Macklin 2013]

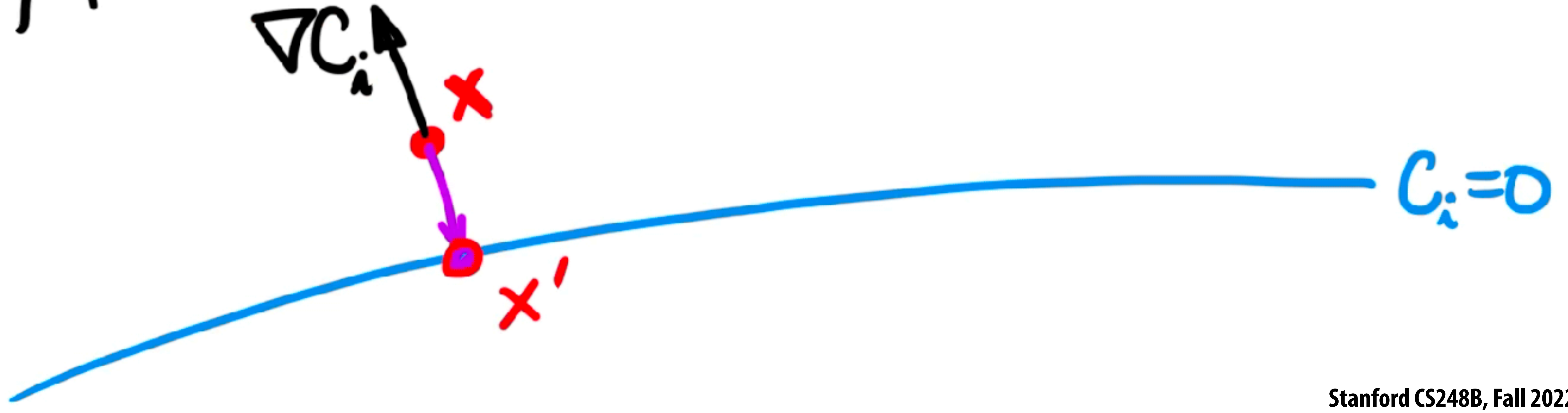
Want $\rho_i \cong \rho_0$ at each particle, i, where

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

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):

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

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

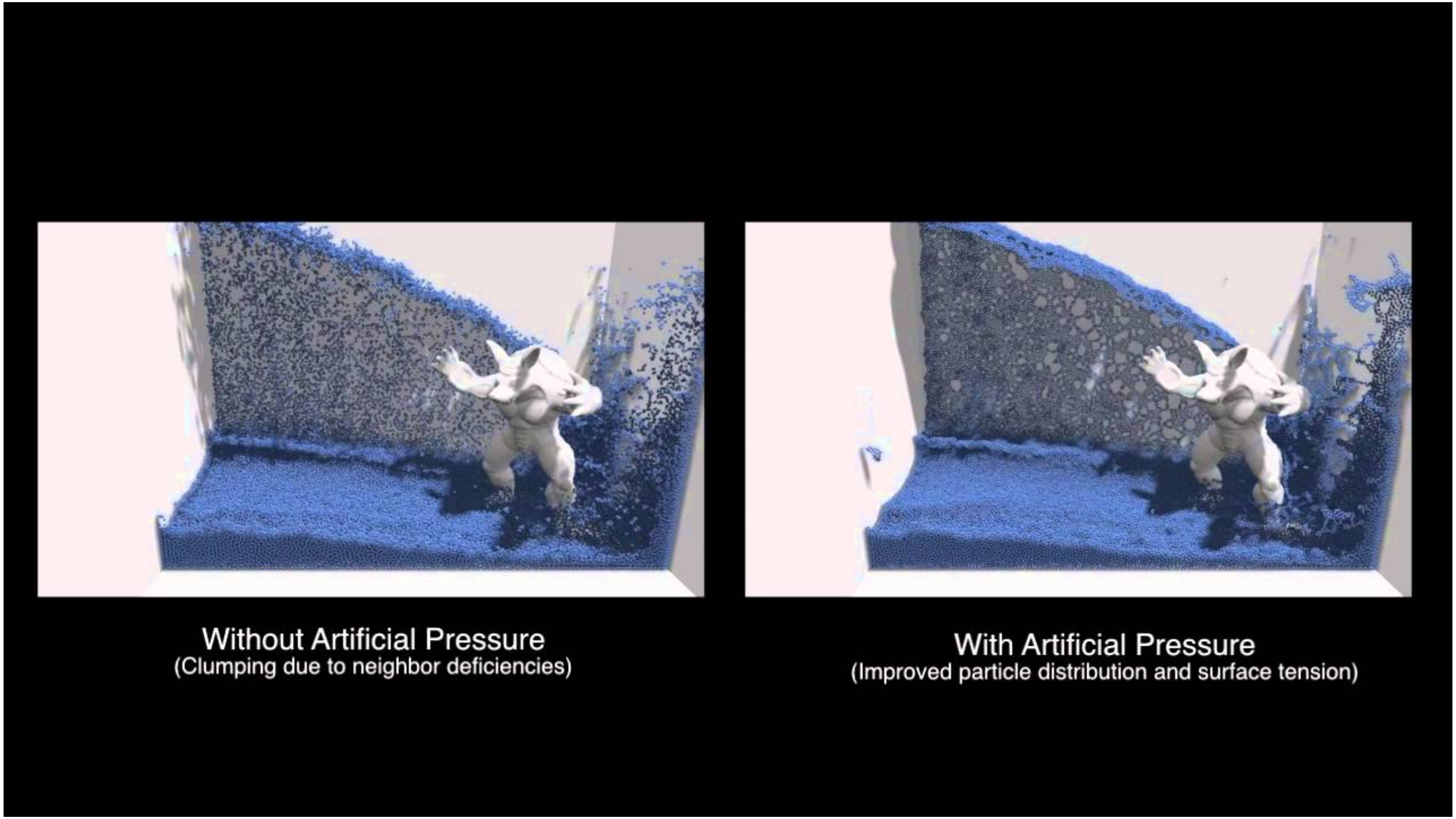
let $\Delta x = J_i^T \lambda_i$

$$= C_i + J_i J_i^T \lambda_i \longrightarrow \lambda_i = \frac{-C_i}{J_i J_i^T + \varepsilon}$$

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

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

Example: Position Based Fluids [Mueller and Macklin 2013]



https://www.youtube.com/watch?v=mgYztcj0vRQ&ab_channel=MilesMacklin

Particle-based rigid-fluid coupling

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

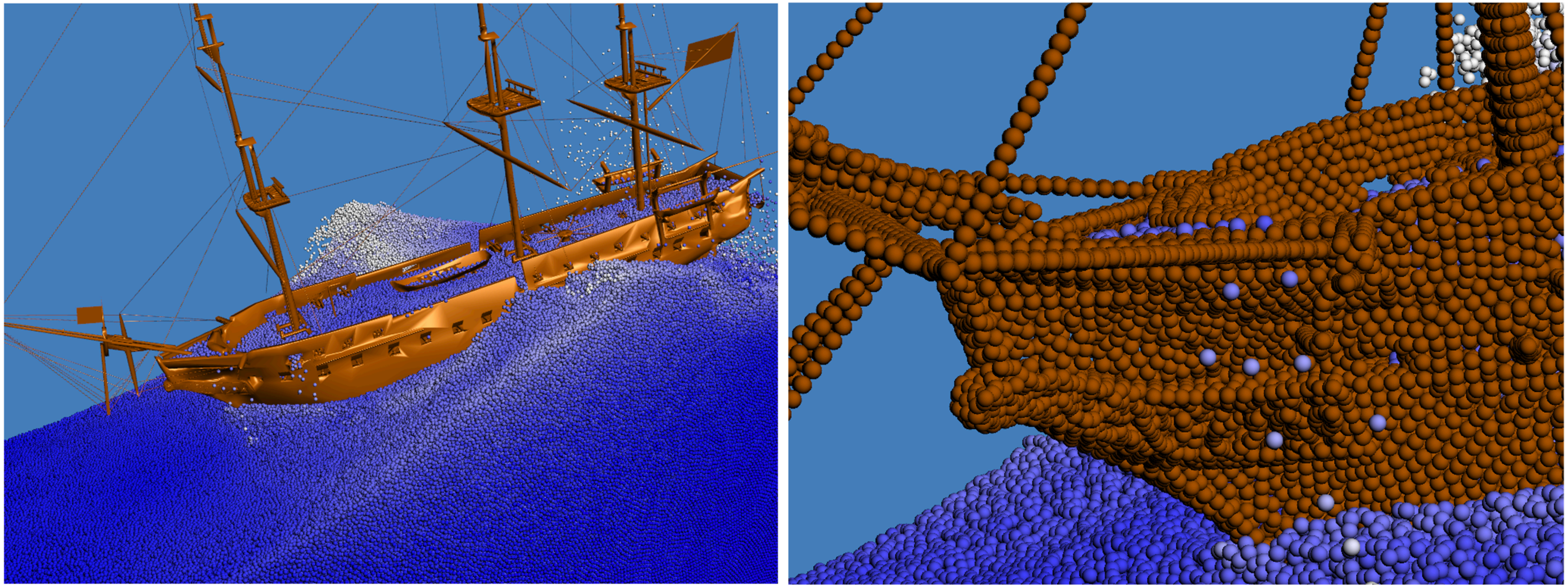


Figure 3: *A frigate is sailing on wavy sea. The right image shows the irregular sampling of boundary particles.*

Particle-based rigid-fluid coupling

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

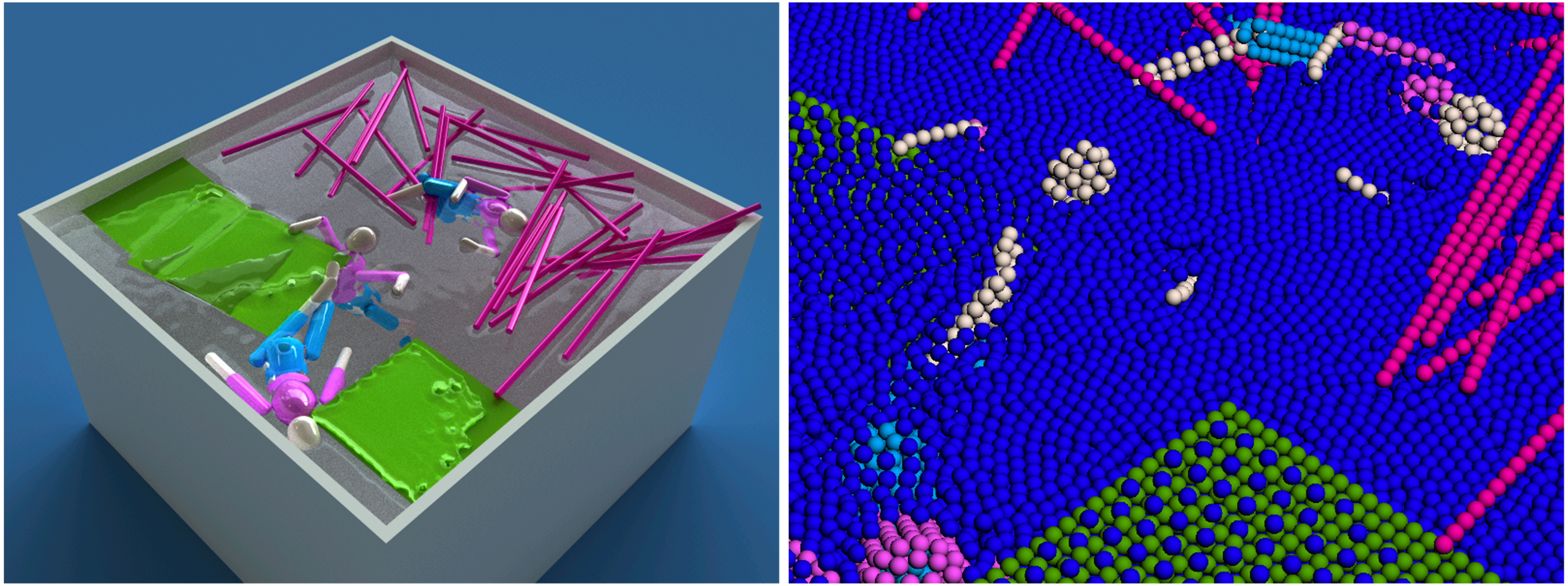
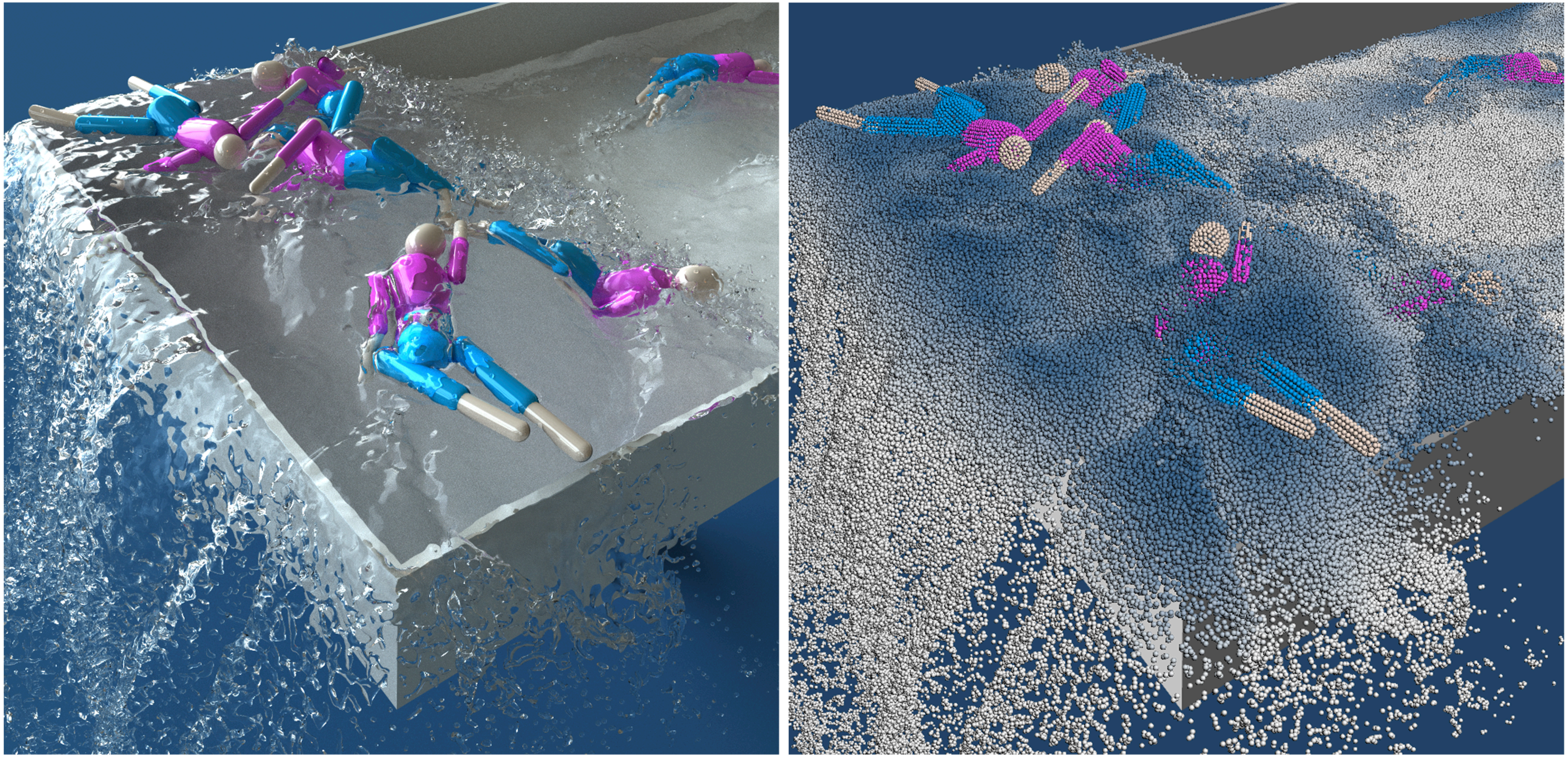


Figure 8: *Handling of lower dimensional objects. The right image shows the underlying particles. Note that planes and rods are modeled with single particle layers.*

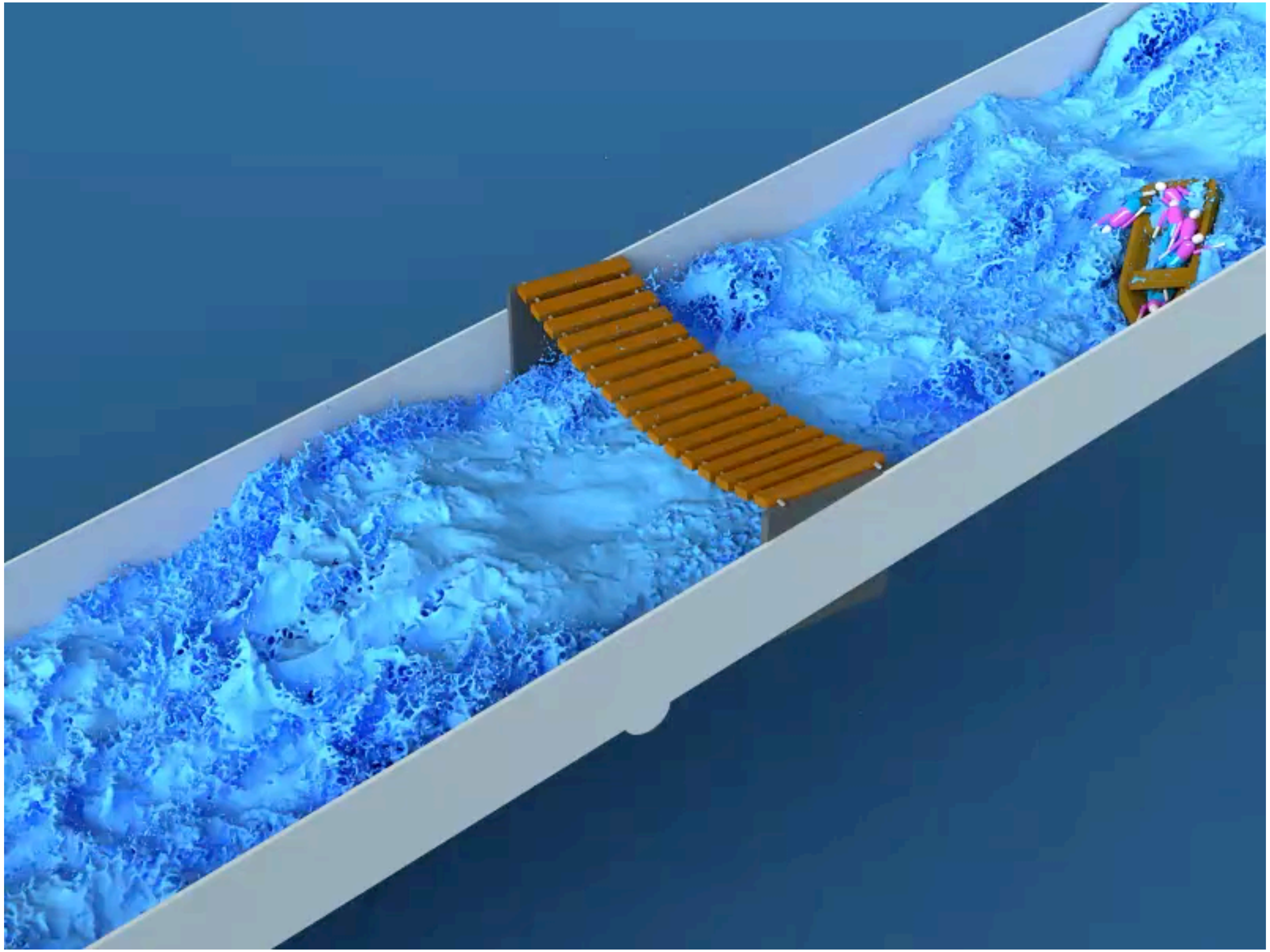
Particle-based rigid-fluid coupling

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

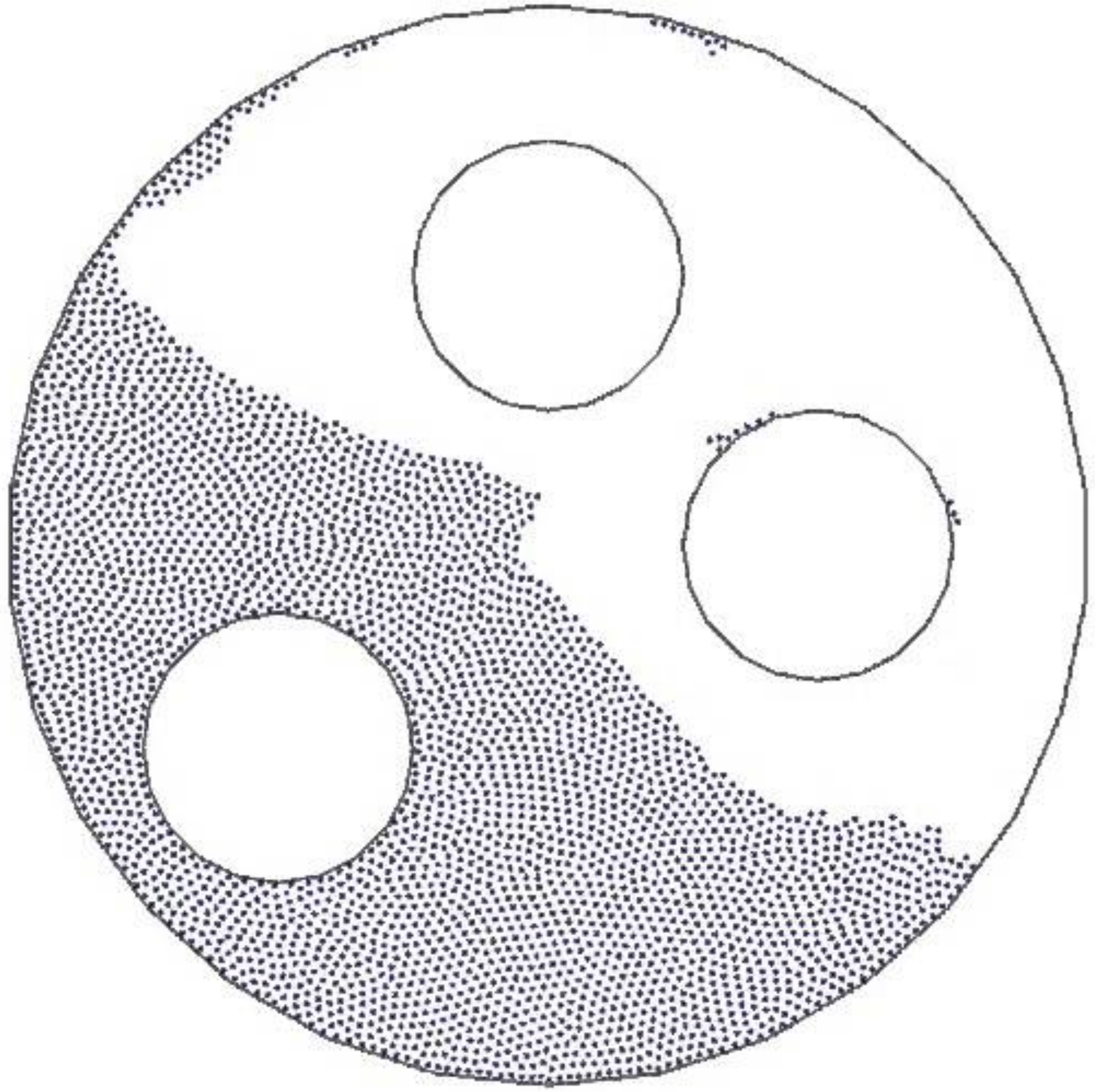


Particle-based rigid-fluid coupling

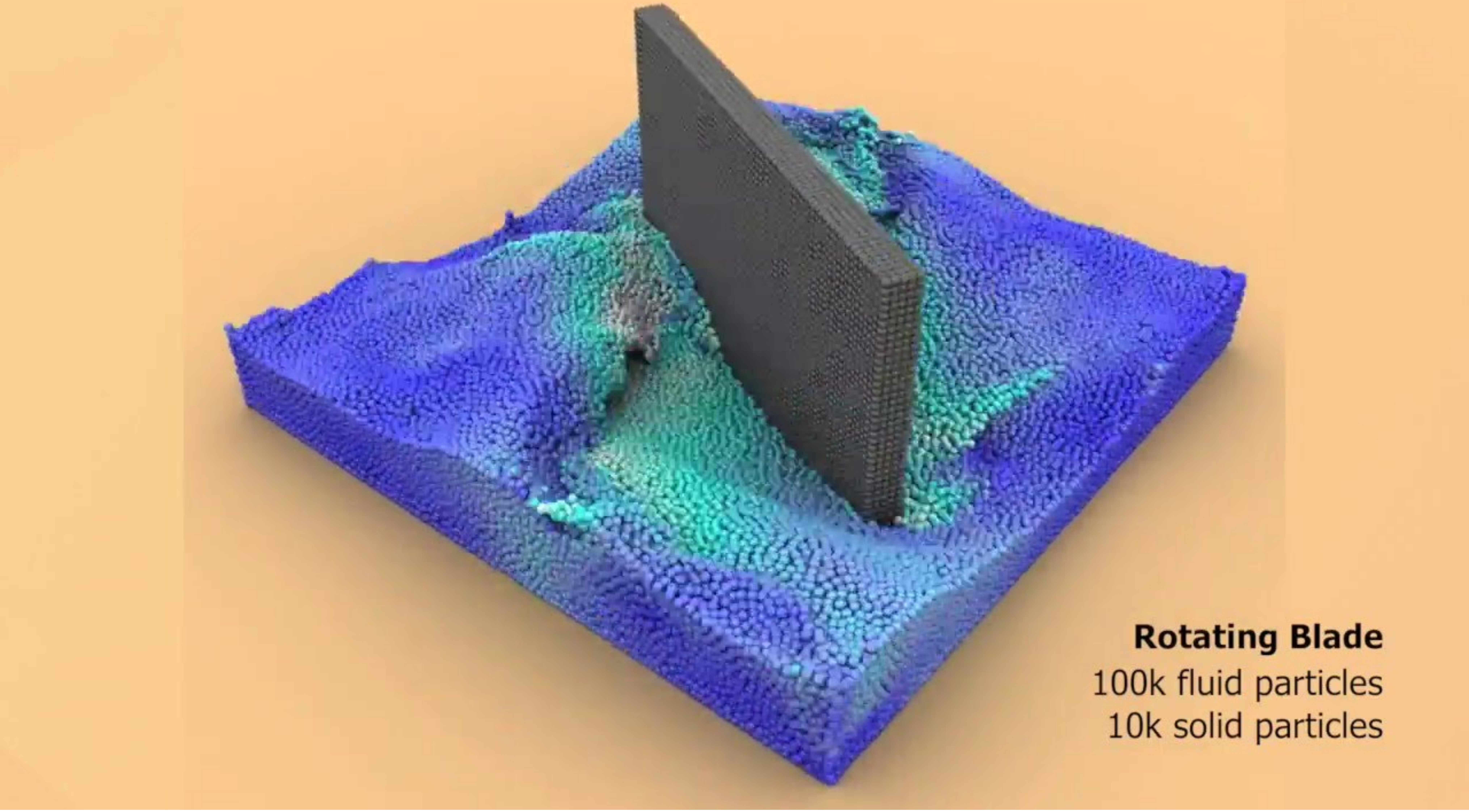
From "Versatile Rigid-Fluid Coupling for Incompressible SPH" [Akinci et al. 2012]



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



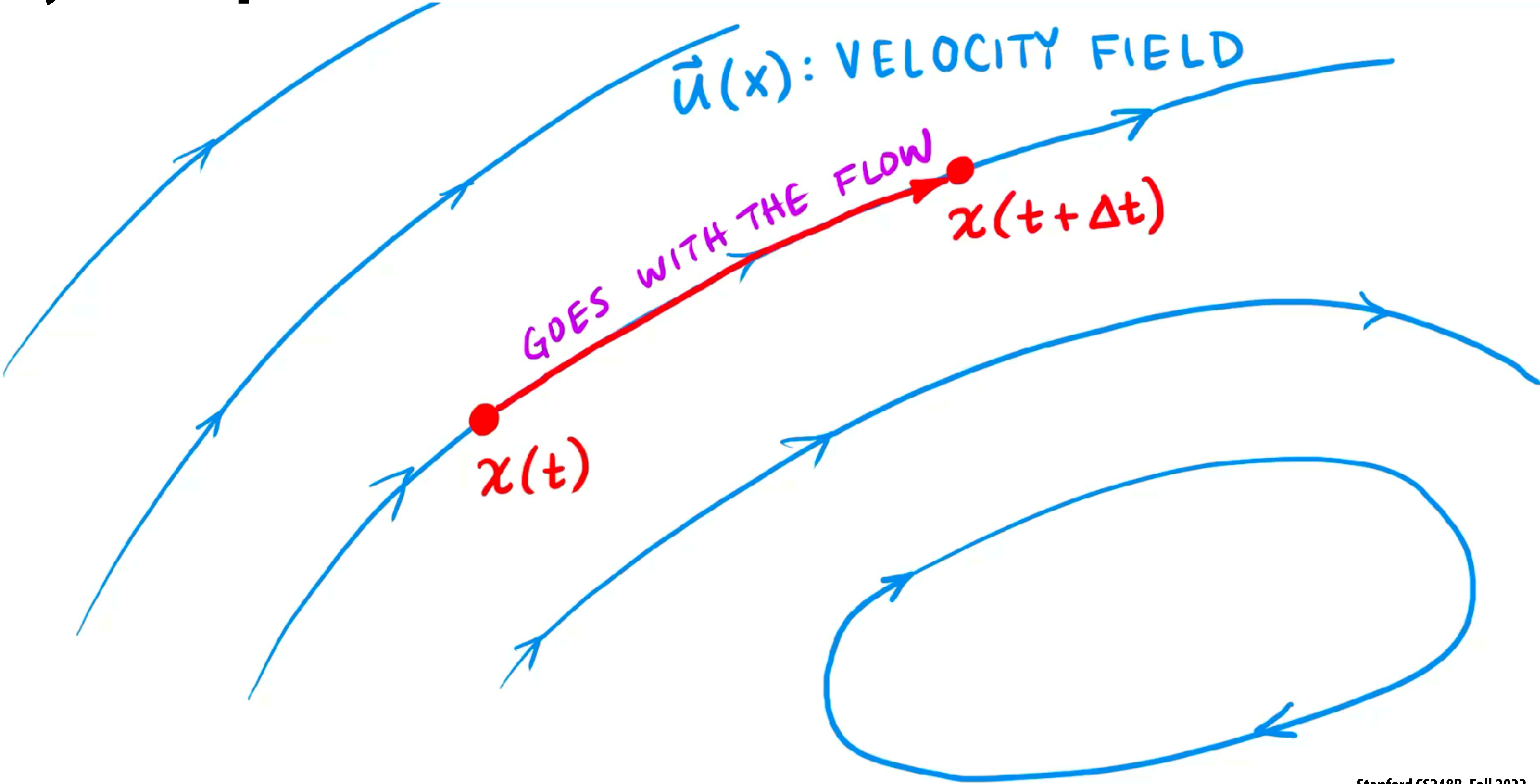
Example: Power Particles [de Goes et al. 2015]



Grid-based Fluids

Eulerian viewpoint: Velocity & fields live at grid points.

Key concept: Advection



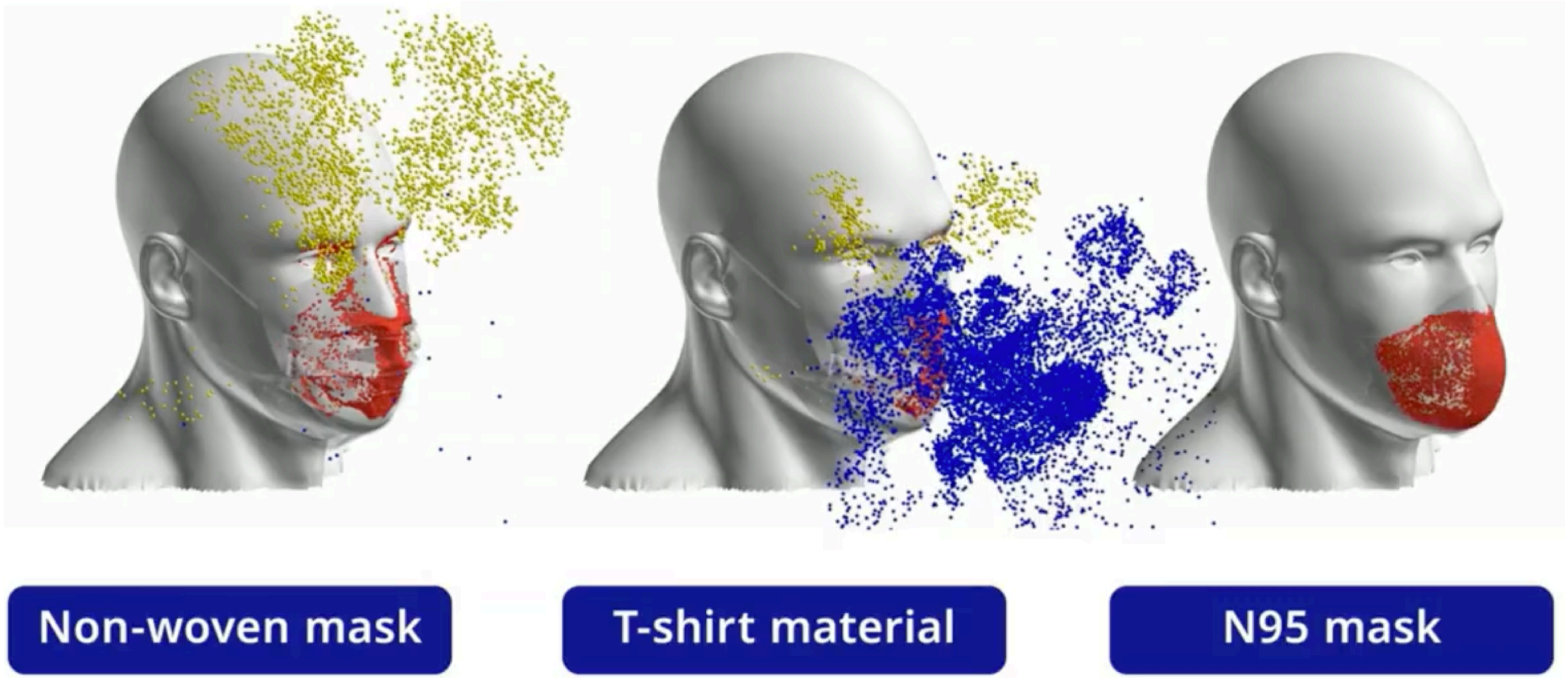
https://jangafx.com/software/embergen/download/free-vdb-animations/

Example: Bubble advection w/ buoyancy and drag forces

[Kim et al. 2010]

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

Example: Aerosol and droplet advection (RIKEN, Japan)



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

Example: Advecting smoke density through fluid velocity

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

Example: Advecting smoke density through fluid velocity

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

Key concept: The Material Derivative

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

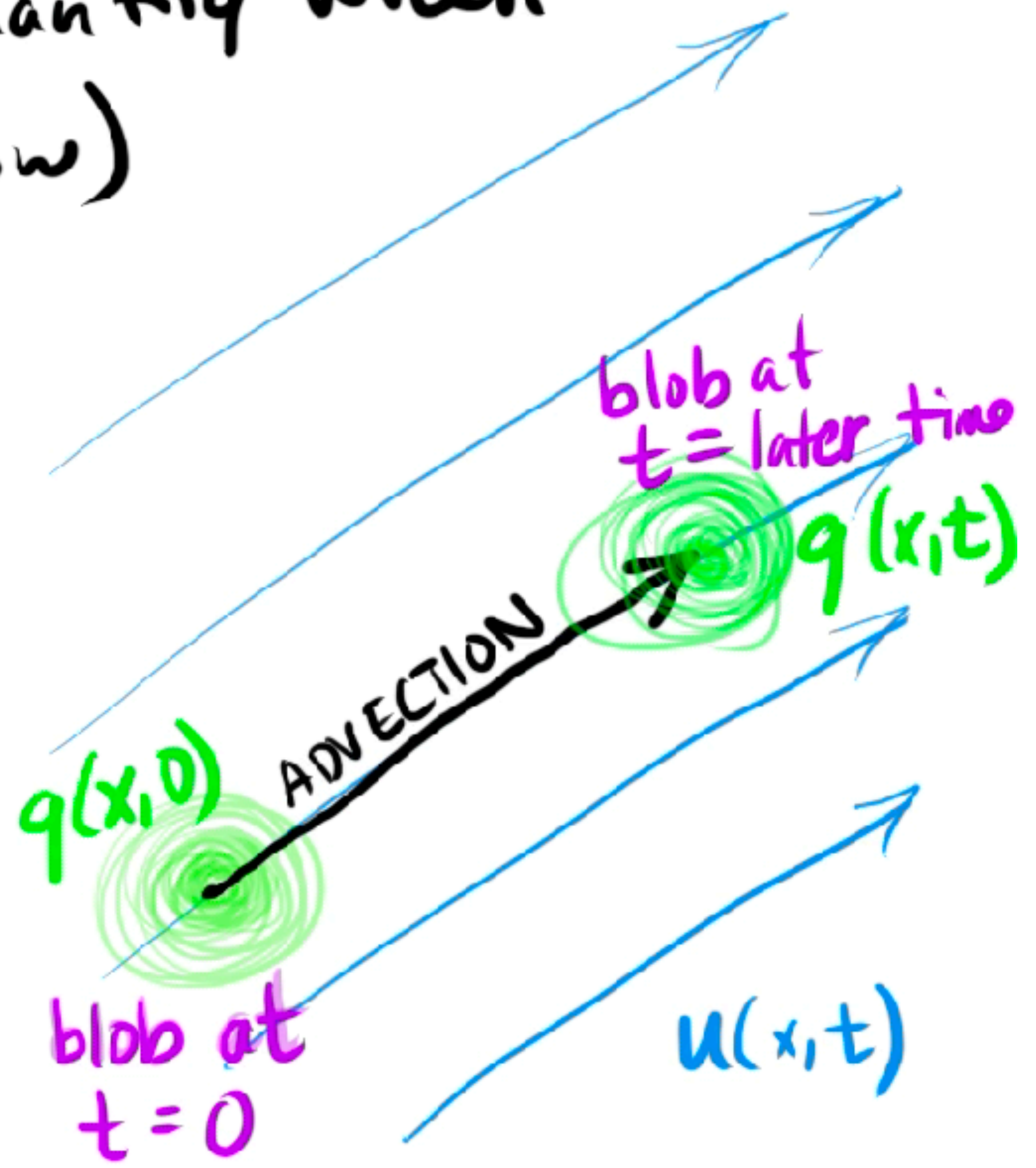
QUANTITIES DON'T CHANGE ALONG PATH, SO

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

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

$$\overset{\text{chain rule}}{=} \frac{\partial q}{\partial t} + \nabla_x q \cdot \underbrace{\frac{dx}{dt}}_{\text{fluid velocity, } u(x,t)}$$

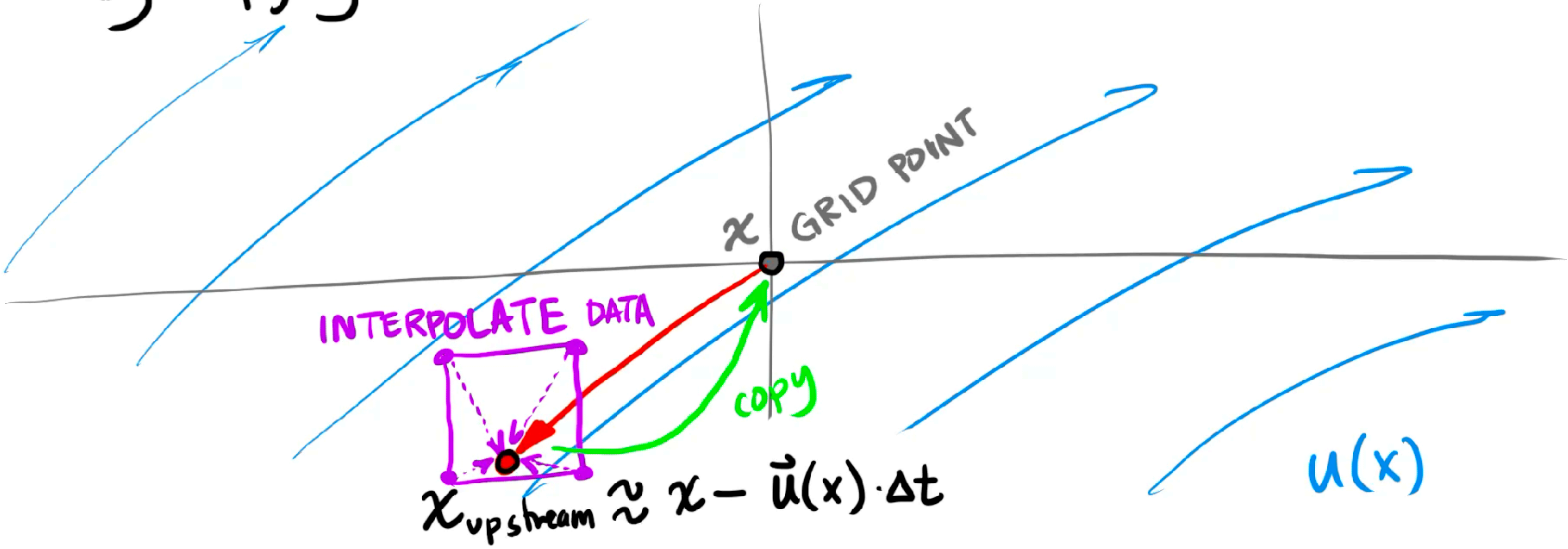
$$= \frac{\partial q}{\partial t} + \underbrace{(u \cdot \nabla)}_{\text{directional derivative}} q$$



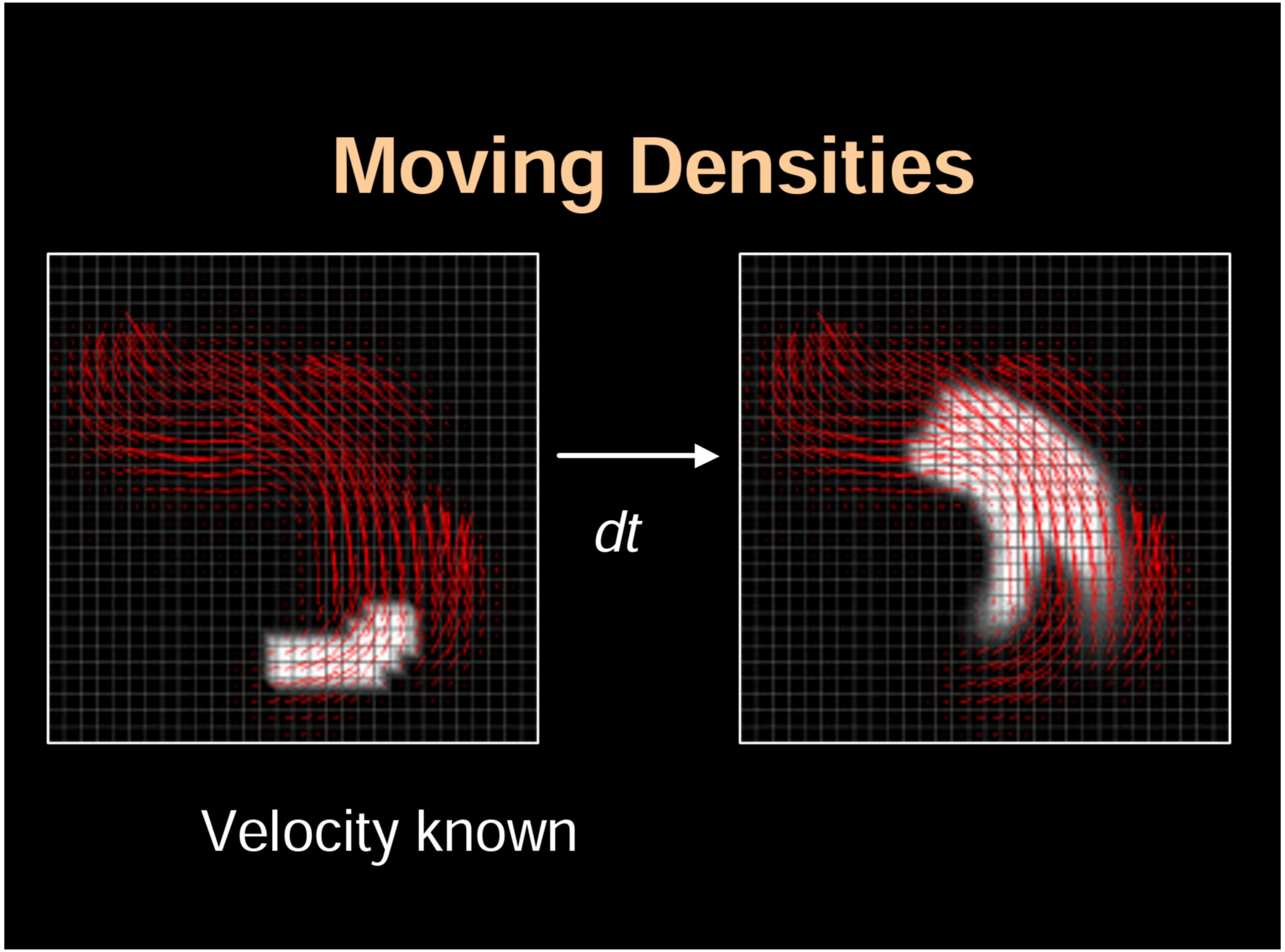
Important method: Semi-Lagrangian advection

Quantities stored at gridpoints.

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

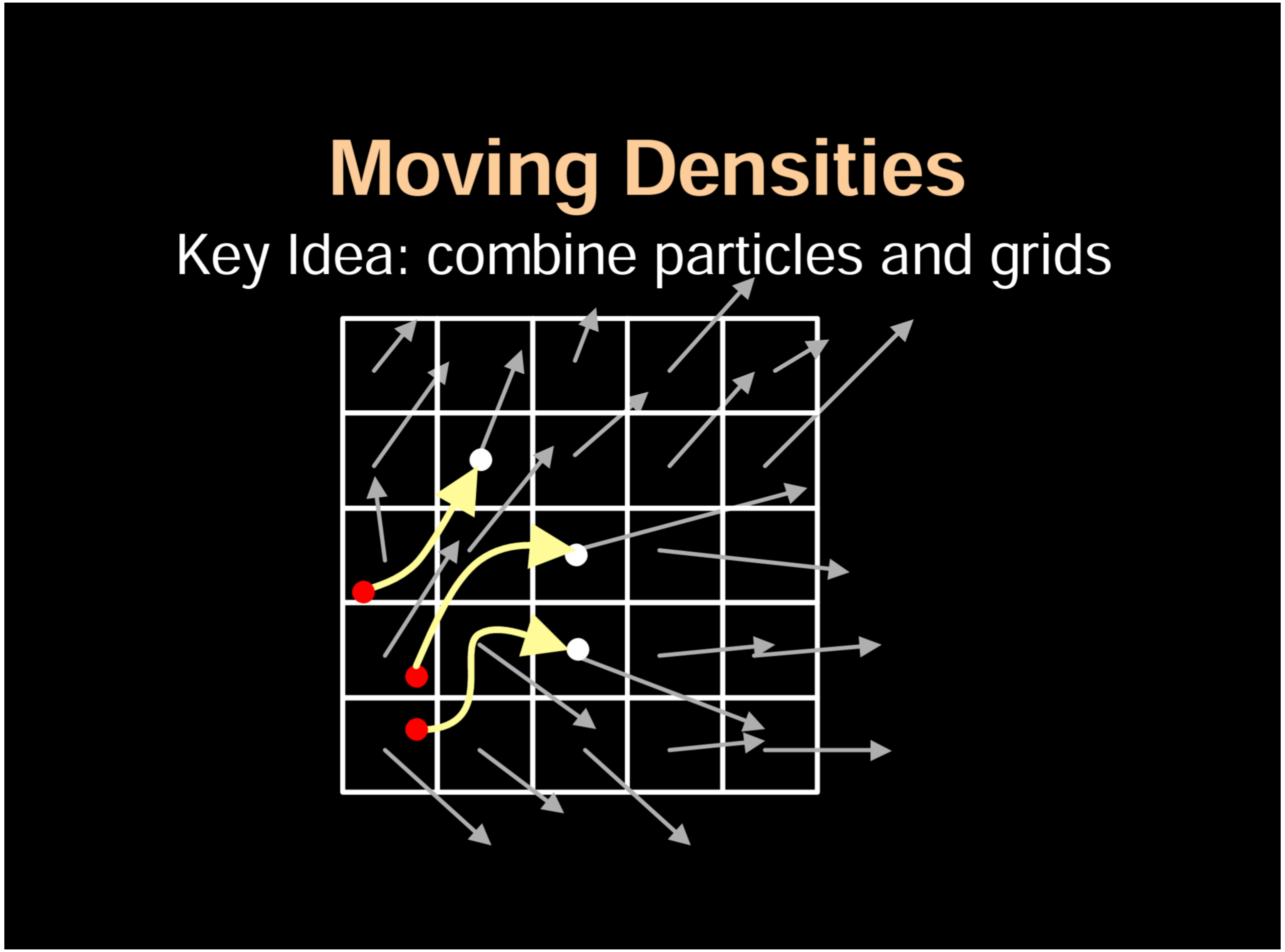


Important method: Semi-Lagrangian advection



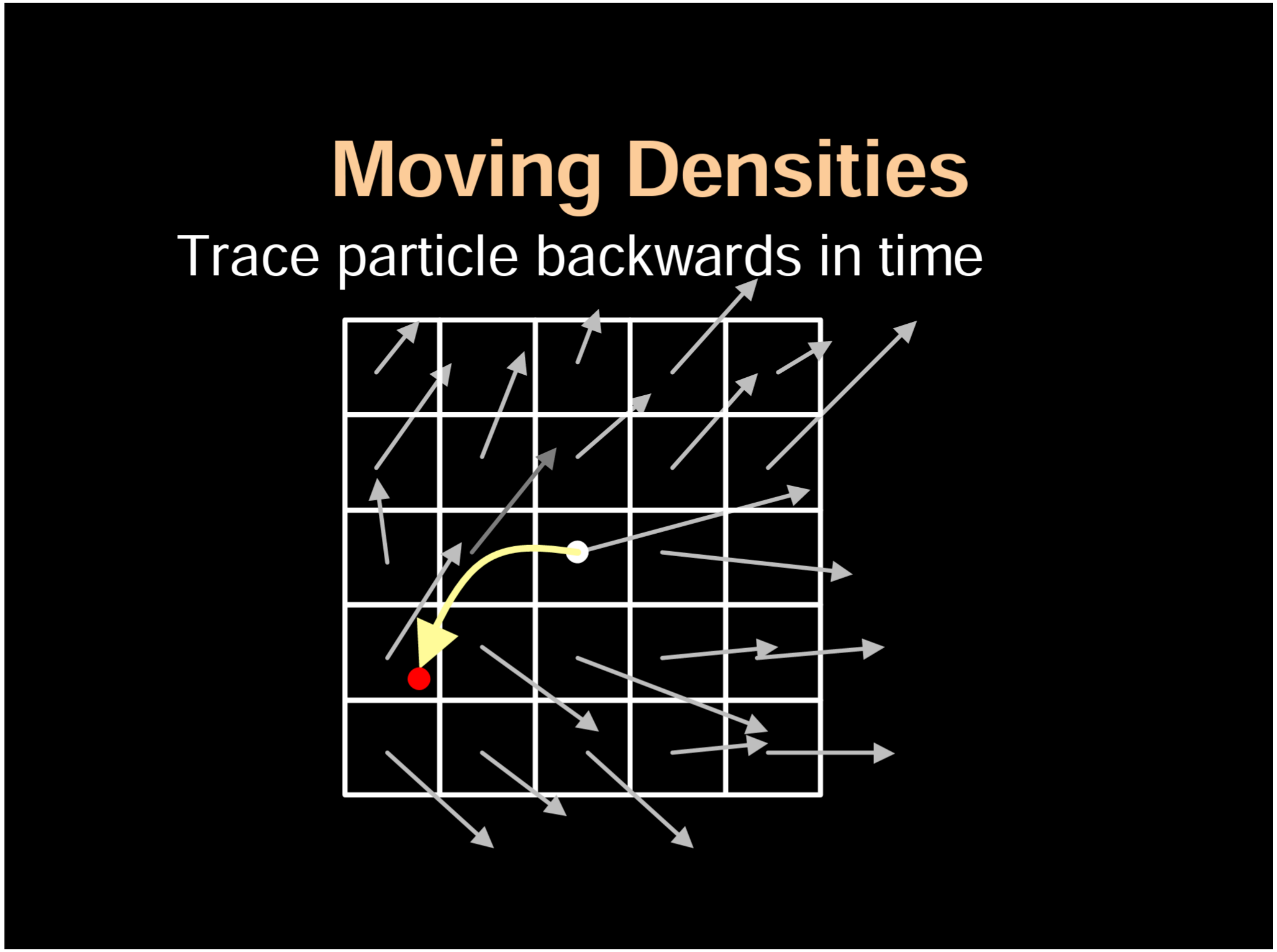
From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection



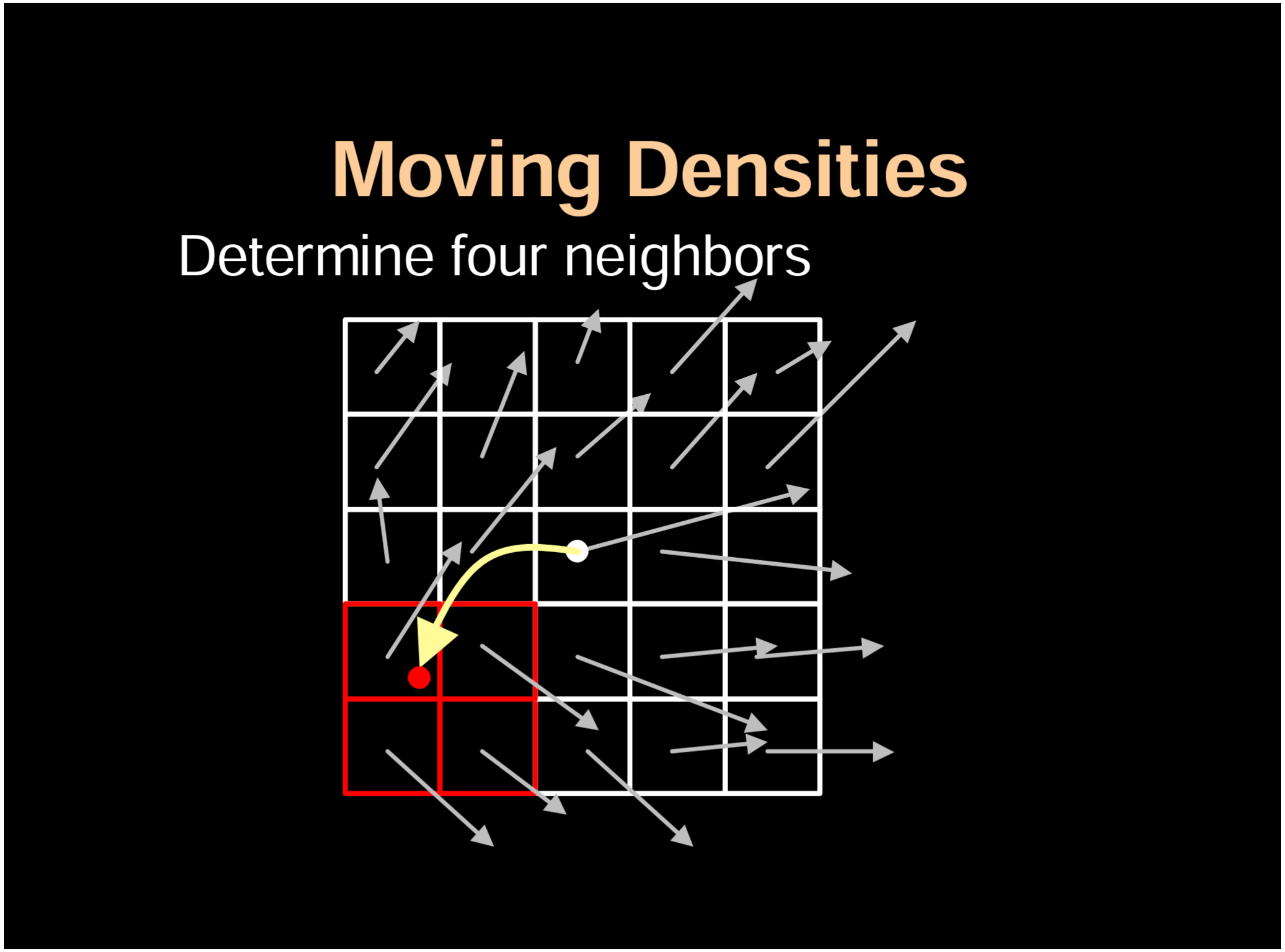
From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection



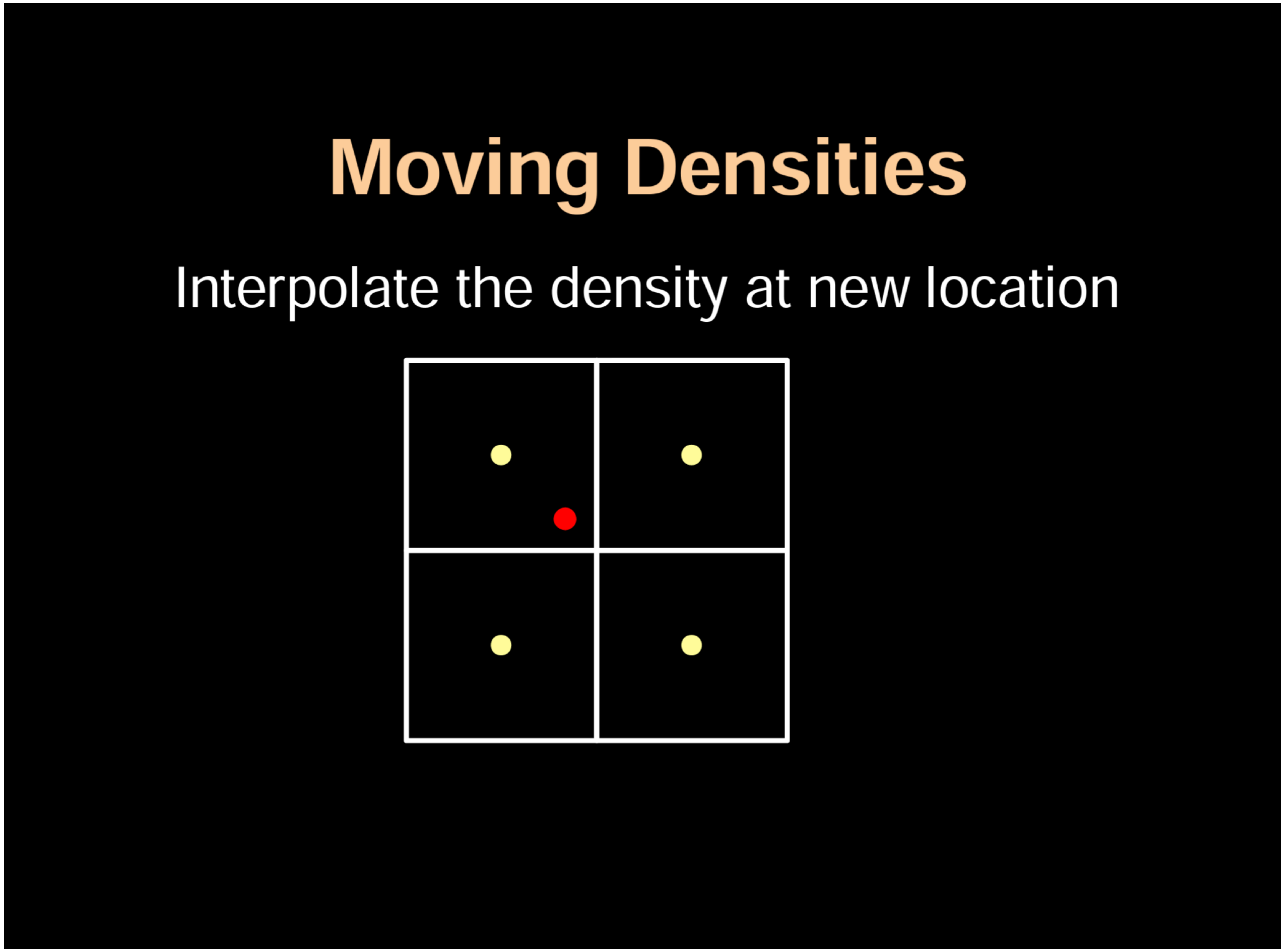
From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection



From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection



From "Stable Fluids" [Stam 1999]

Important method: Semi-Lagrangian advection

Moving Densities

This scheme is unconditionally stable:

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

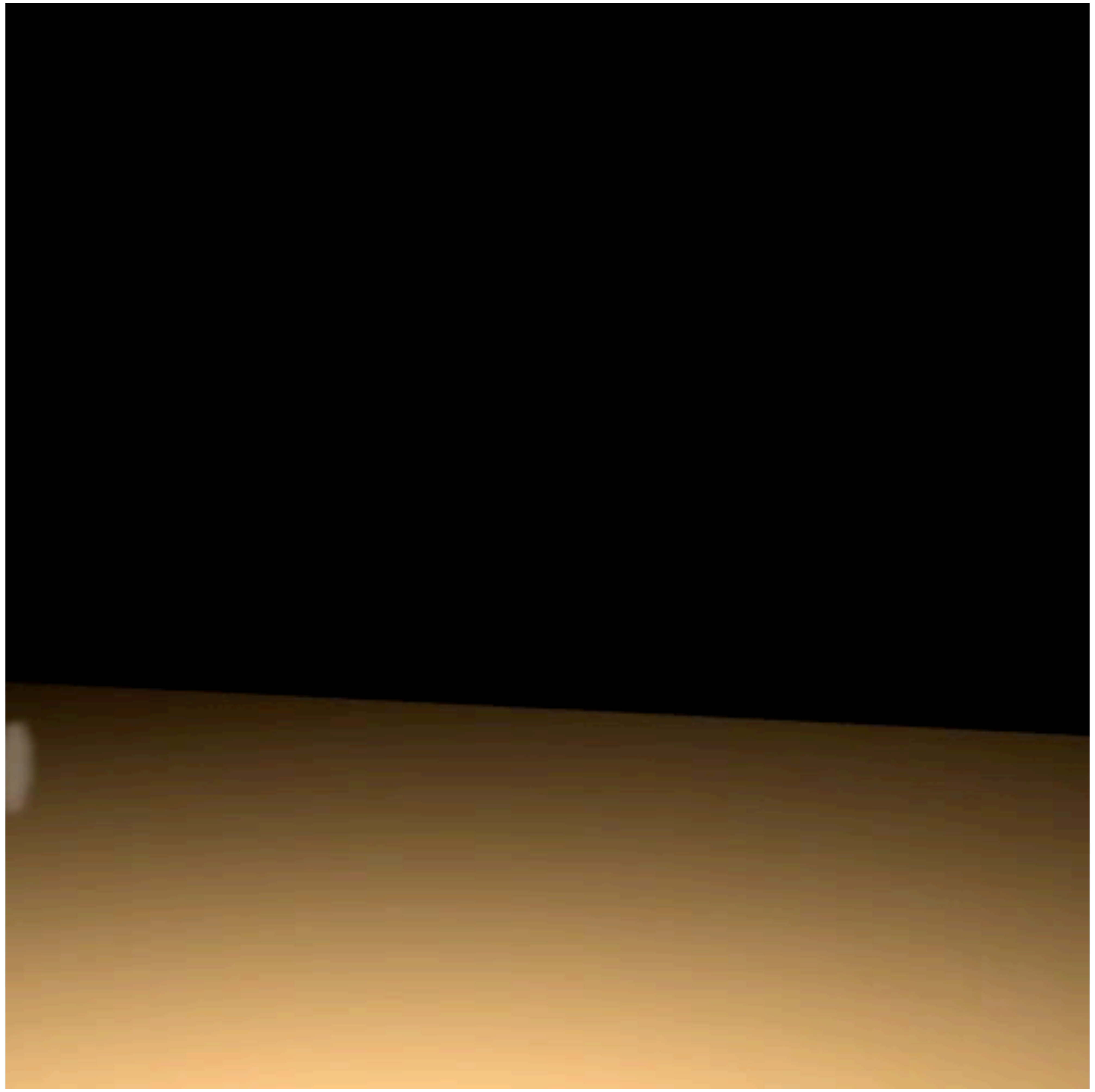
$$\rho_0, \rho_1 \leq \rho_{max}$$

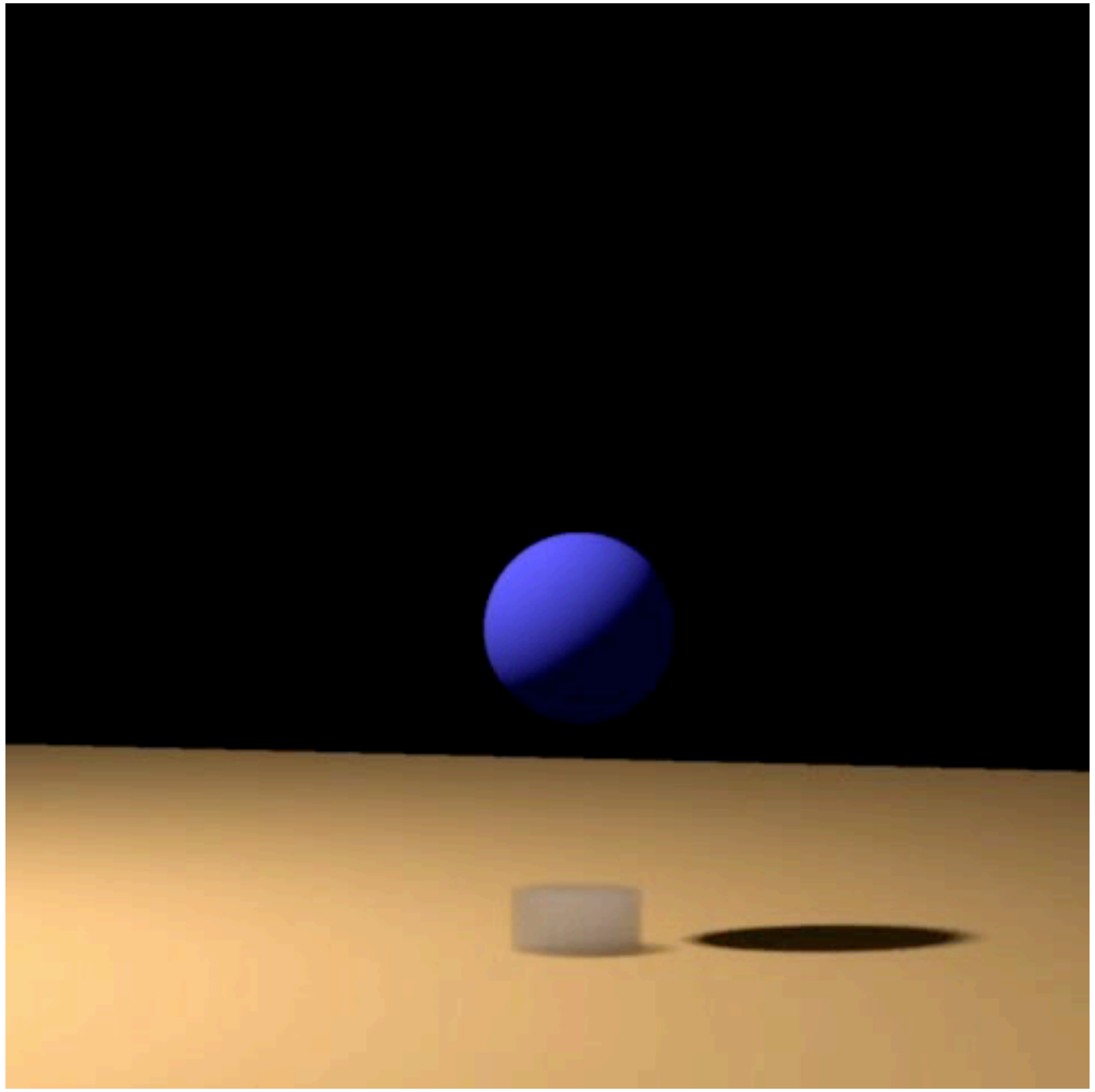
$$\rho_{int} \leq (1 - s + s)\rho_{max} \leq \rho_{max}$$

$\longrightarrow$ density is always bounded

From "Stable Fluids" [Stam 1999]

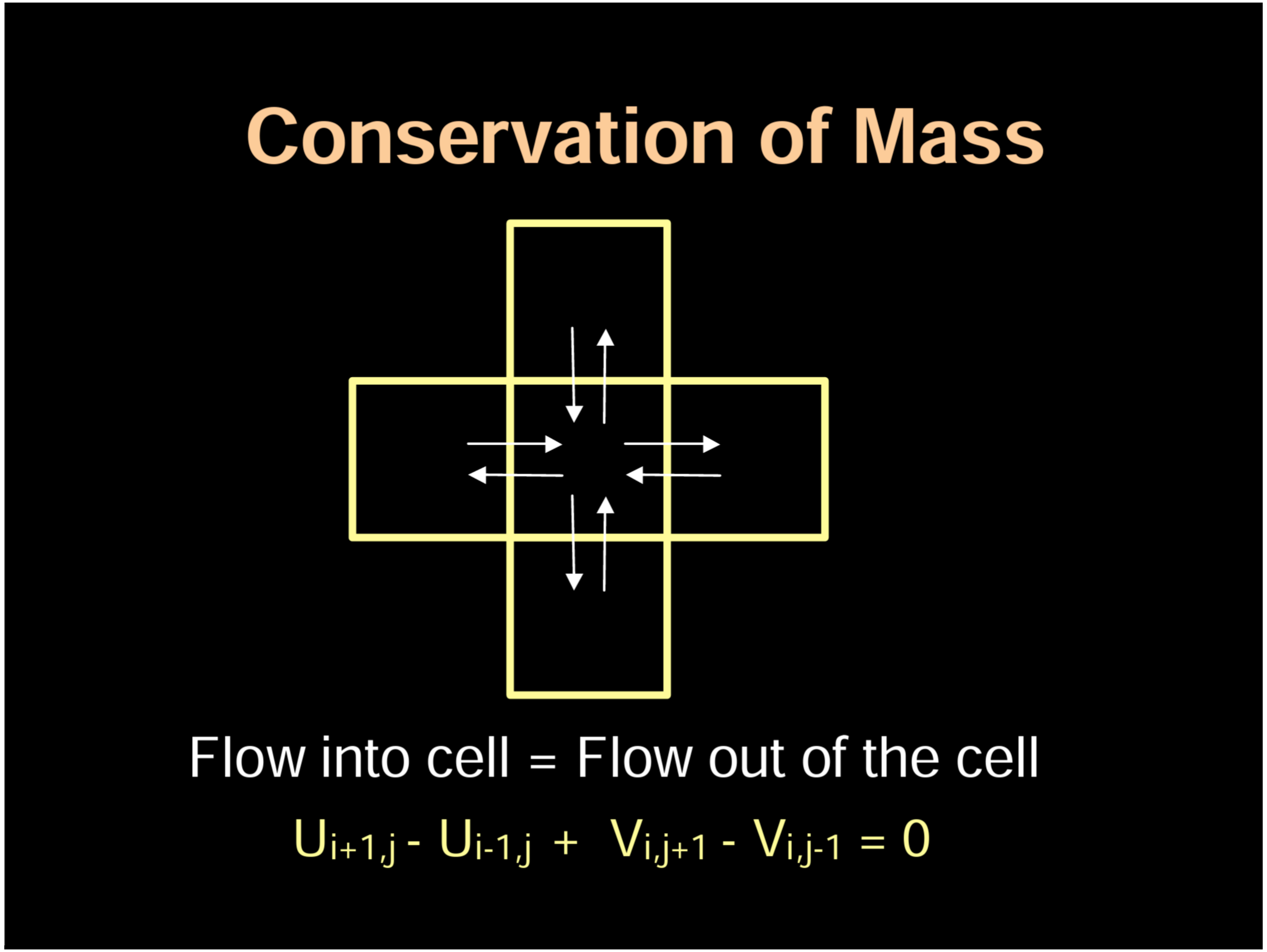
Semi-Lagrangian advection [Fedkiw et al. 2001]





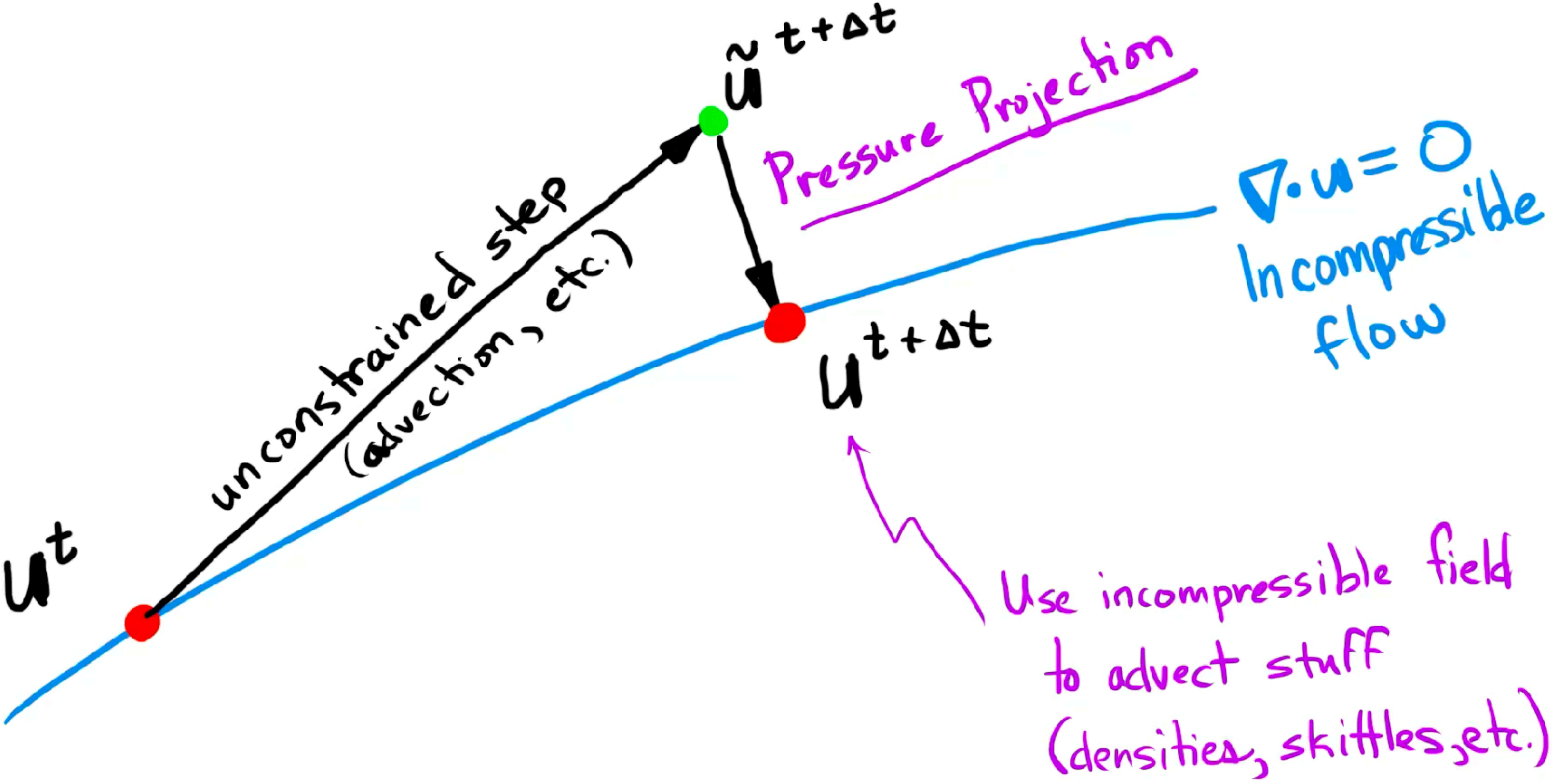
Enforcing Incompressibility via Projection

Key idea: Helmholtz-Hodge Decomposition

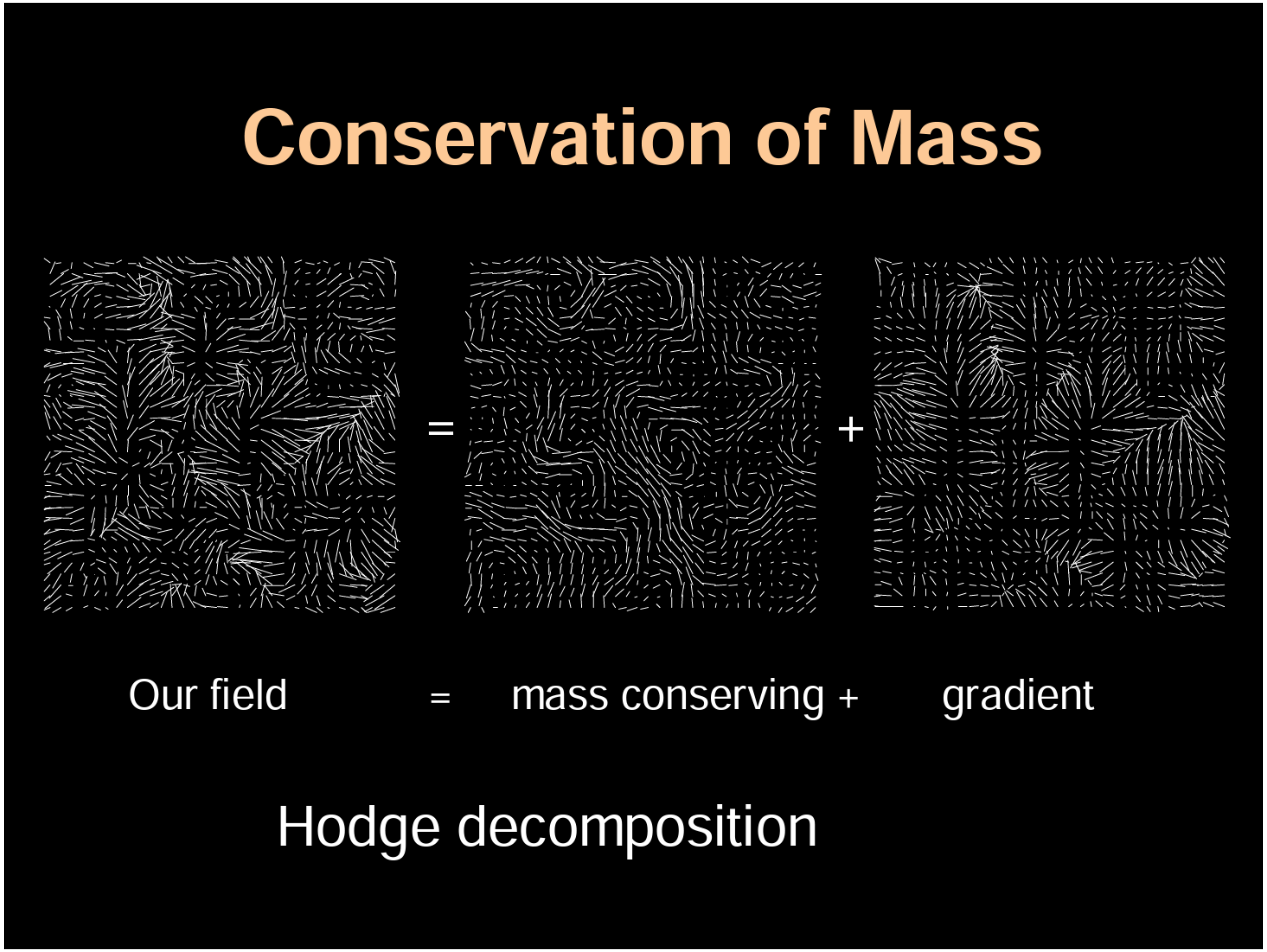


From "Stable Fluids" [Stam 1999]

Geometric interpretation: Pressure projection

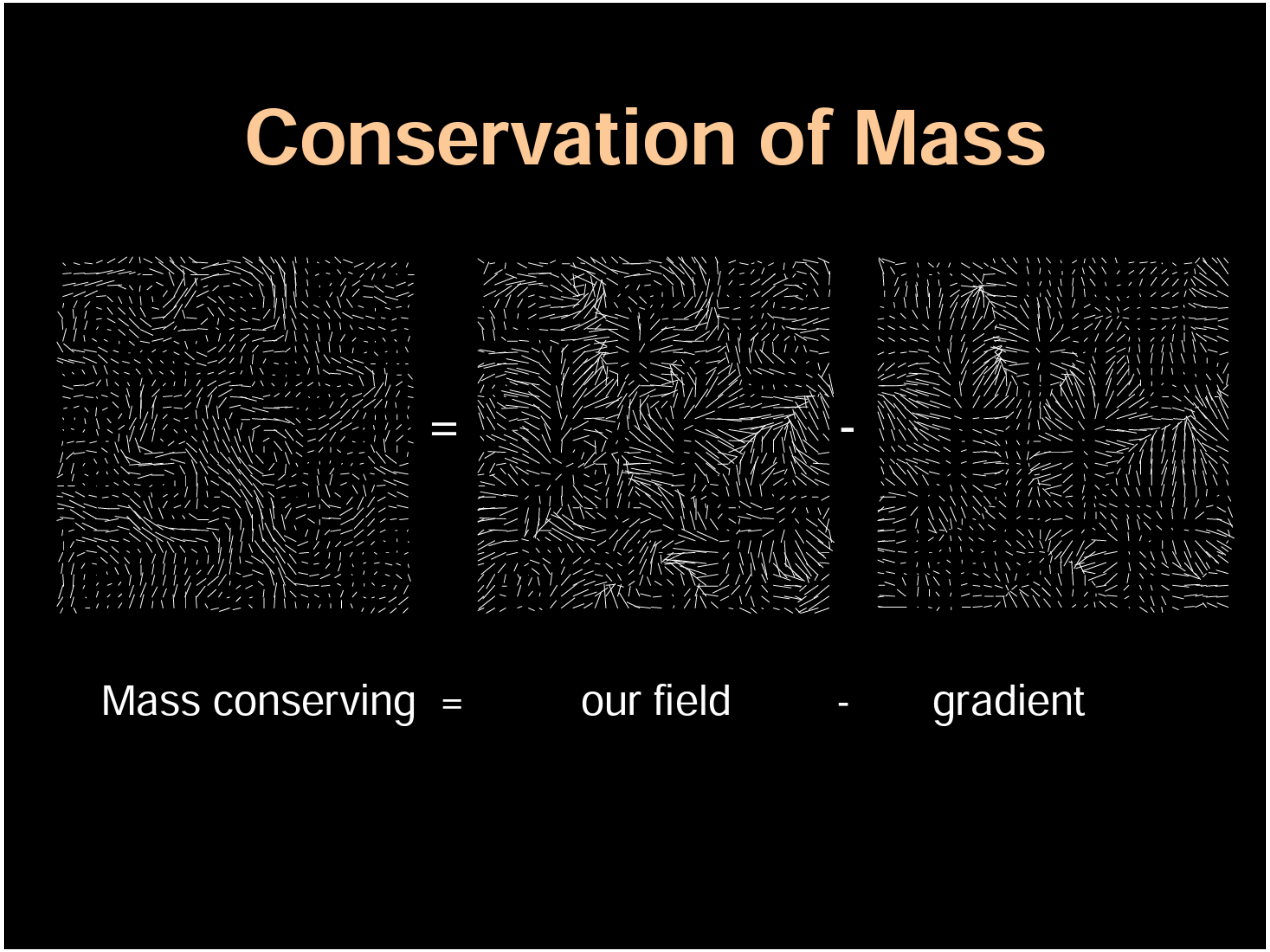


Key idea: Helmholtz-Hodge Decomposition (of $\tilde{\mathbf{u}}^{t+\Delta t}$)



From "Stable Fluids" [Stam 1999]

Key idea: Helmholtz-Hodge Decomposition



From "Stable Fluids" [Stam 1999]

Math of pressure projection

$$\tilde{u} = u + \nabla\phi \qquad (\text{where } \nabla\cdot u = 0)$$

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

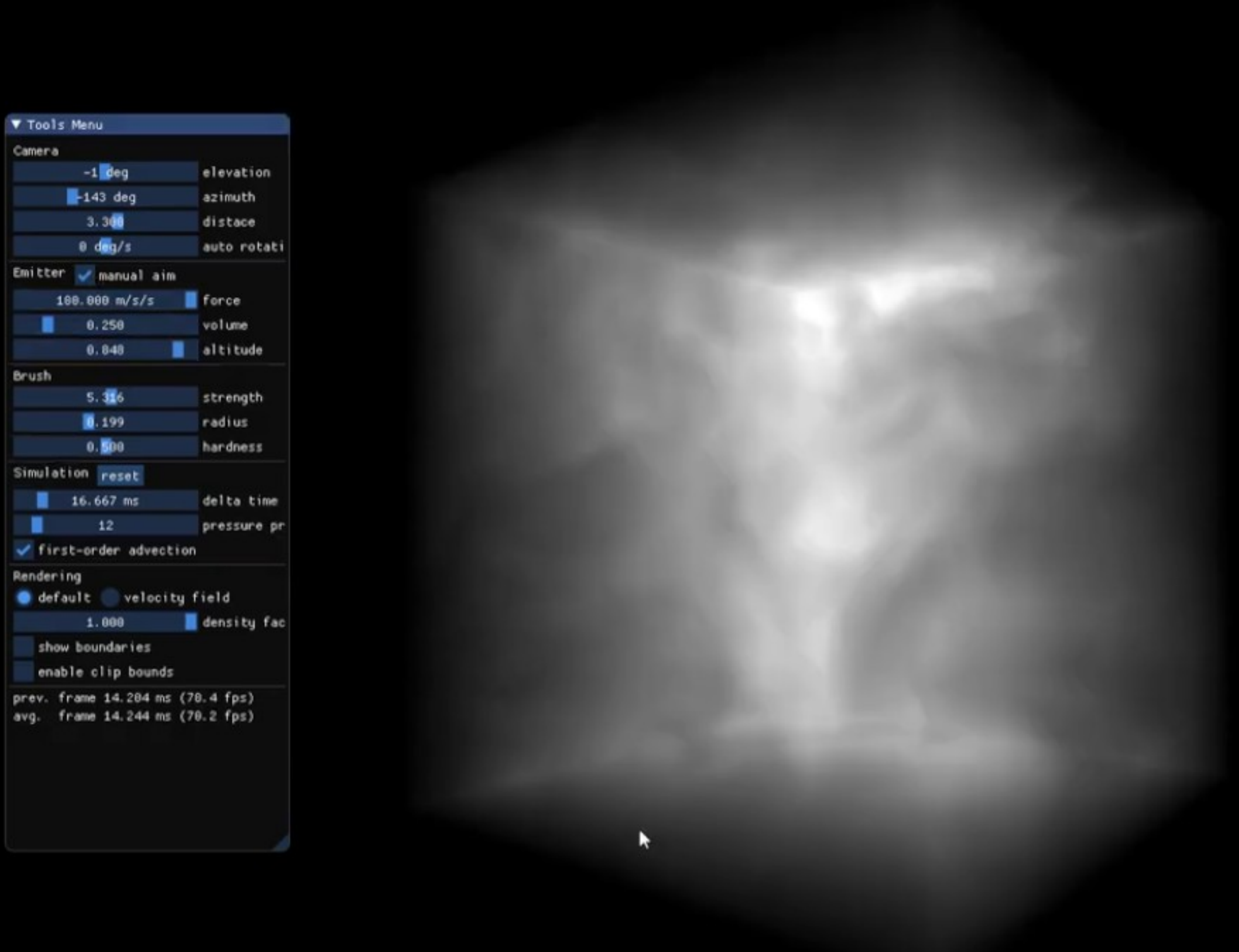
$$\nabla\cdot\nabla = (\partial_x\ \partial_y\ \partial_z)\begin{pmatrix}\partial_x\\ \partial_y\\ \partial_z\end{pmatrix} = \partial_x^2+\partial_y^2+\partial_z^2 = \nabla^2$$

Solve $\nabla^2\phi = (\nabla\cdot\tilde{u})$ for ϕ

- Linear system "$A\phi=b$"

ϕ is a pressure-like variable (Lagrange multiplier for the constraint)

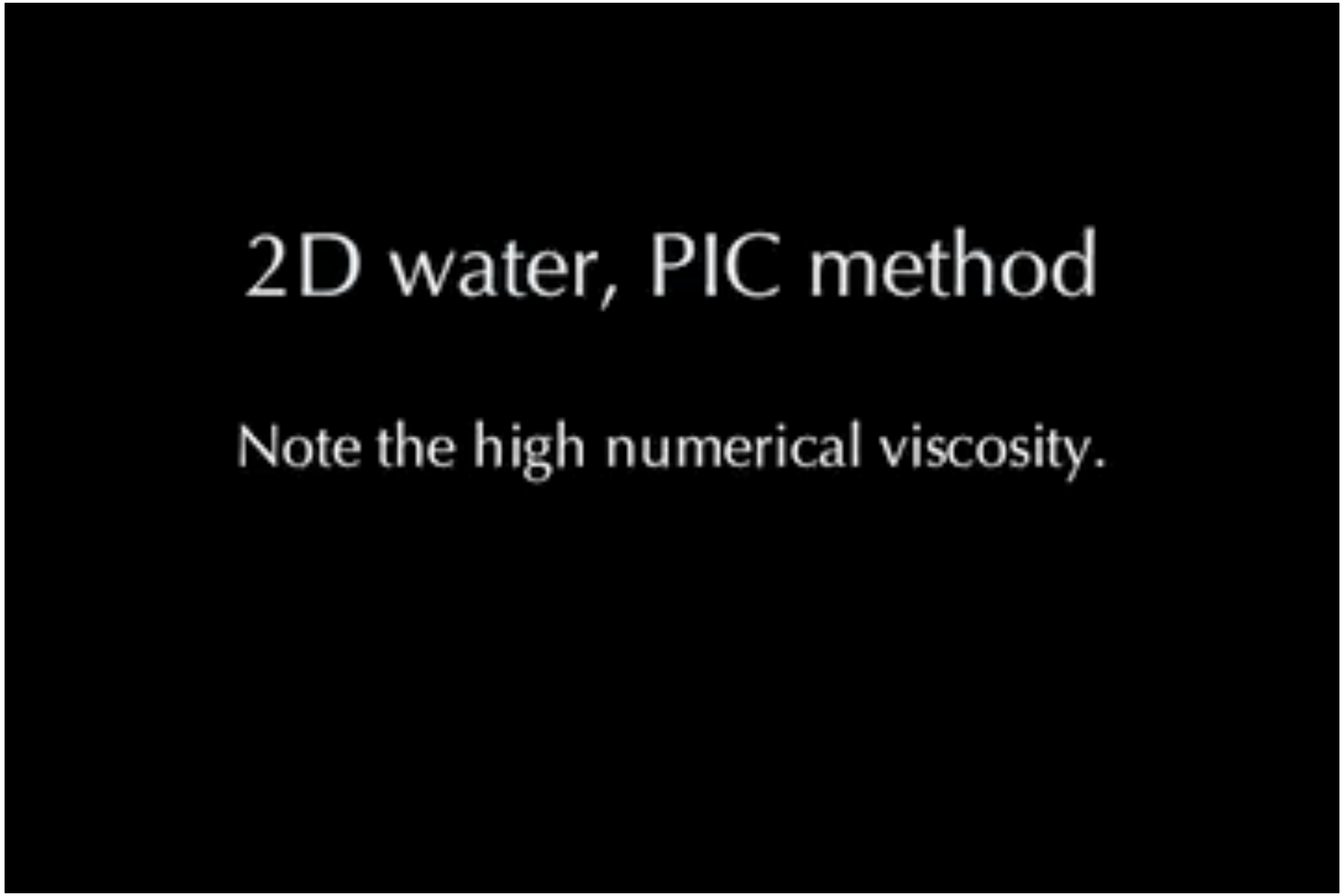
$$u = \tilde{u} - \nabla\phi$$



https://www.youtube.com/watch?v=twfqYyvT50s&ab_channel=MarcGluyas

Hybrid schemes

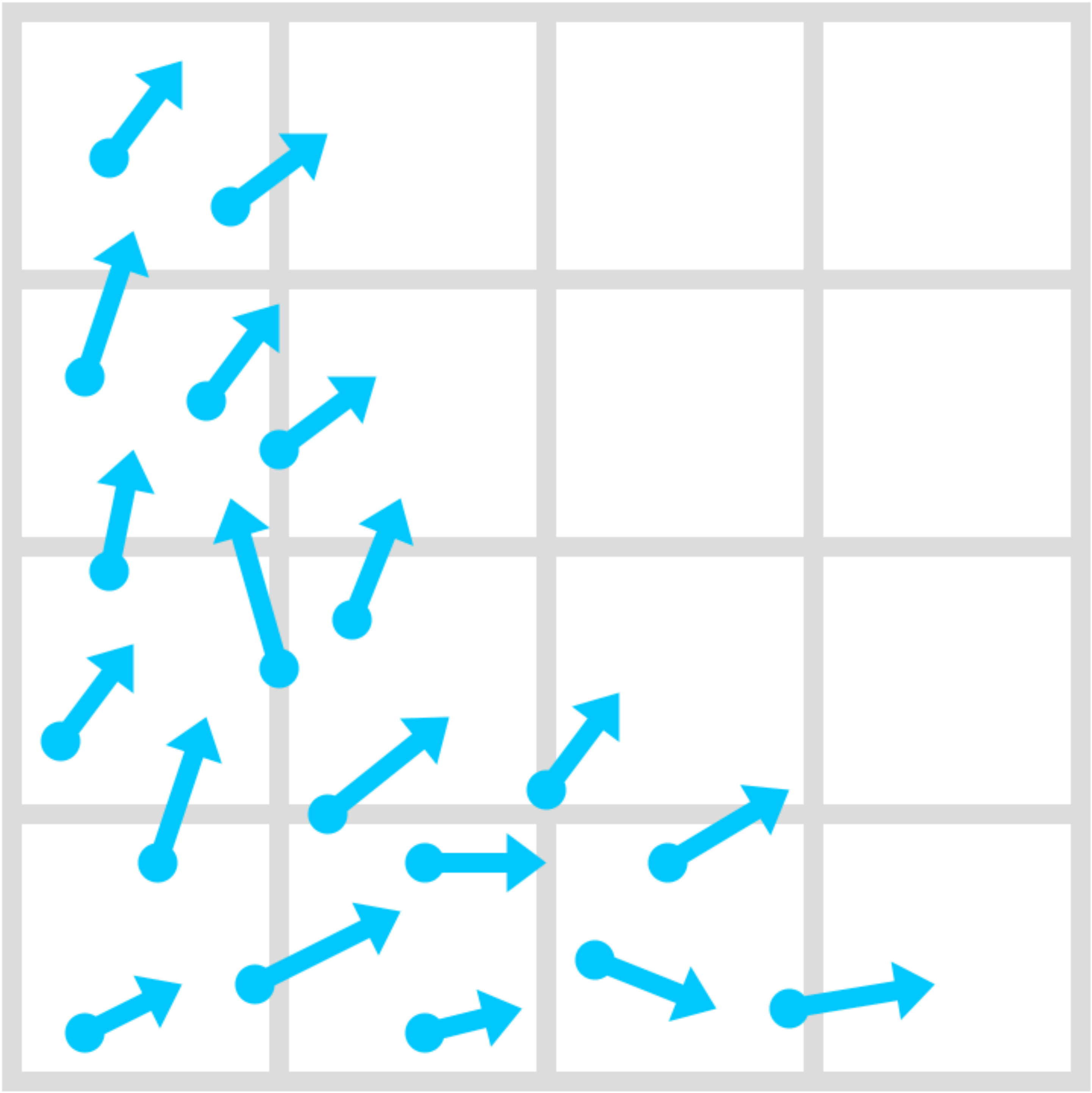
Particle-In-Cell (PIC)



PIC

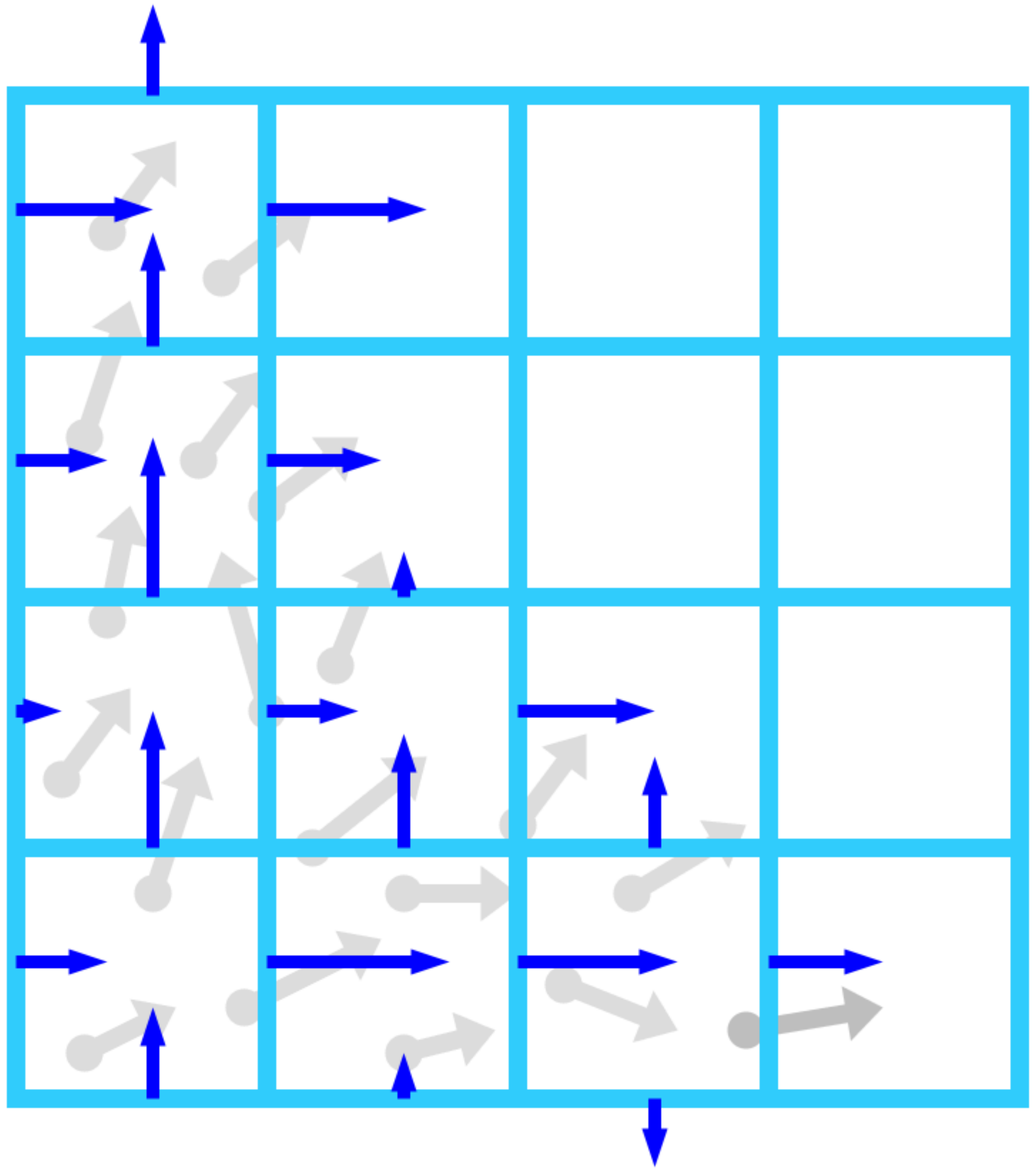
SIGGRAPH2006

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



PIC

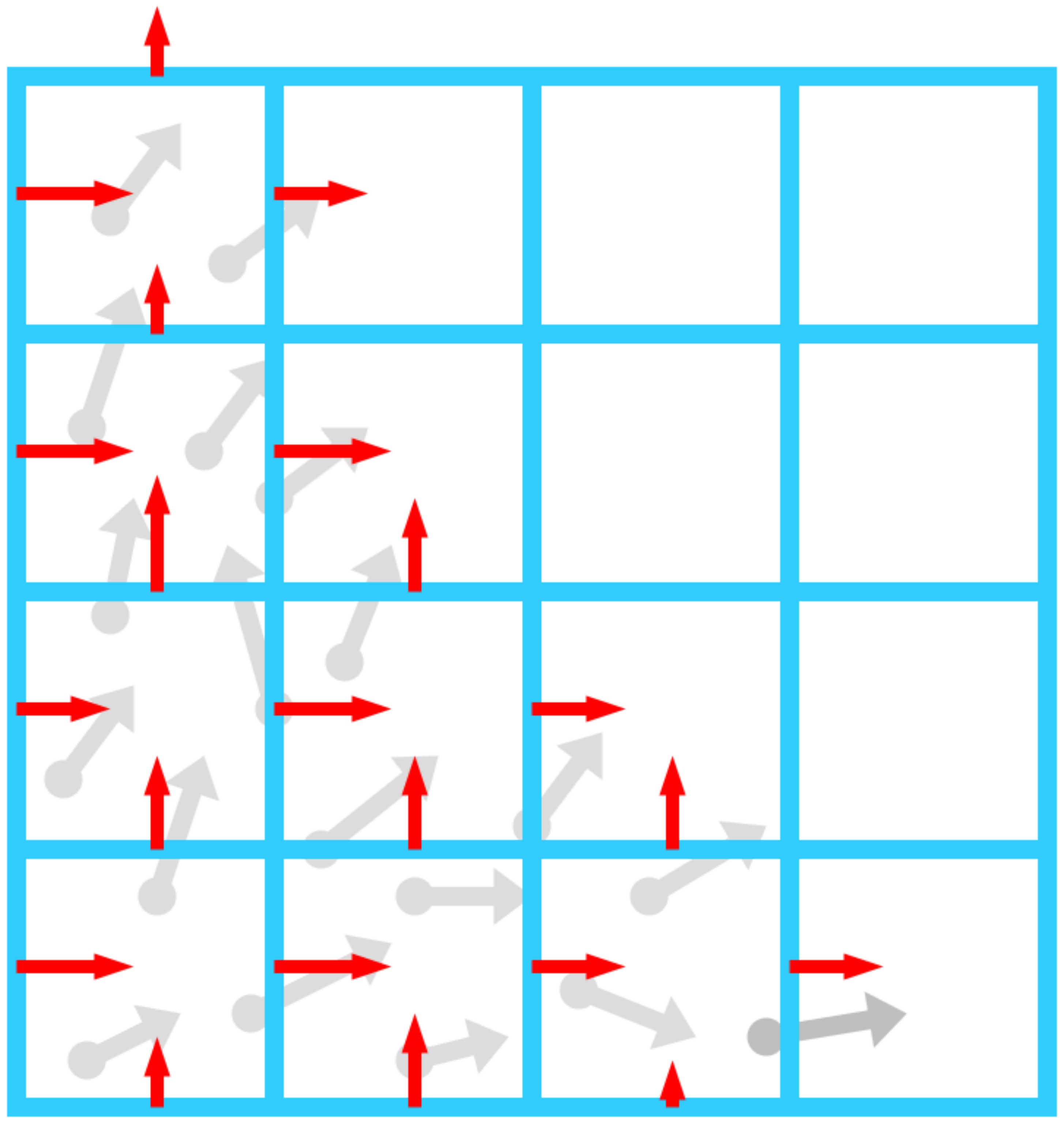
- Start with particles
- Transfer to grid
- Resolve forces on grid
  - Gravity, boundaries, pressure, etc.
- Transfer velocity back to particles
- Advect: move particles



PIC

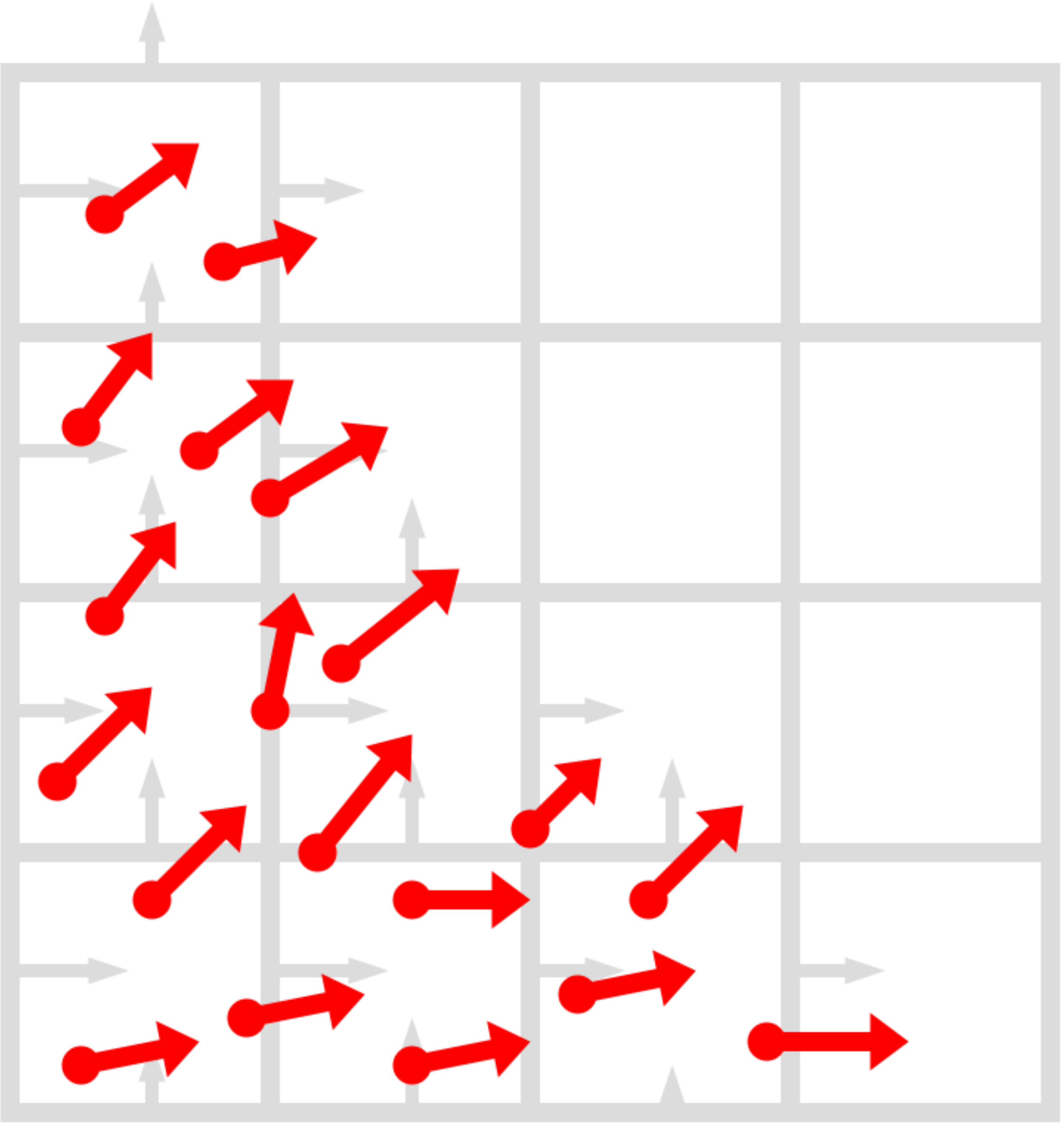
SIGGRAPH2006

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



PIC

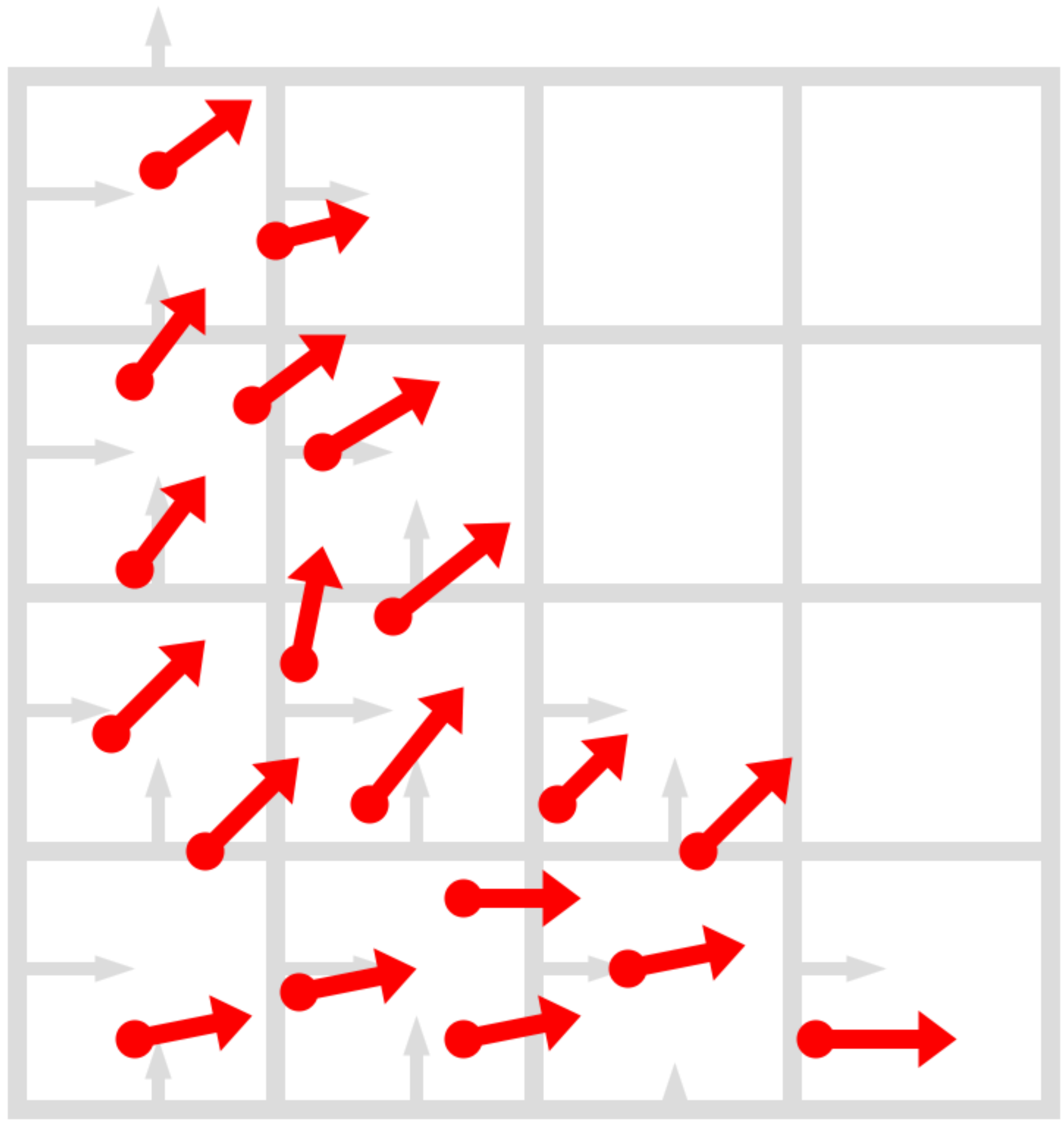
- Start with particles
- Transfer to grid
- Resolve forces on grid
  - Gravity, boundaries, pressure, etc.
- Transfer velocity back to particles
- Advect: move particles



PIC

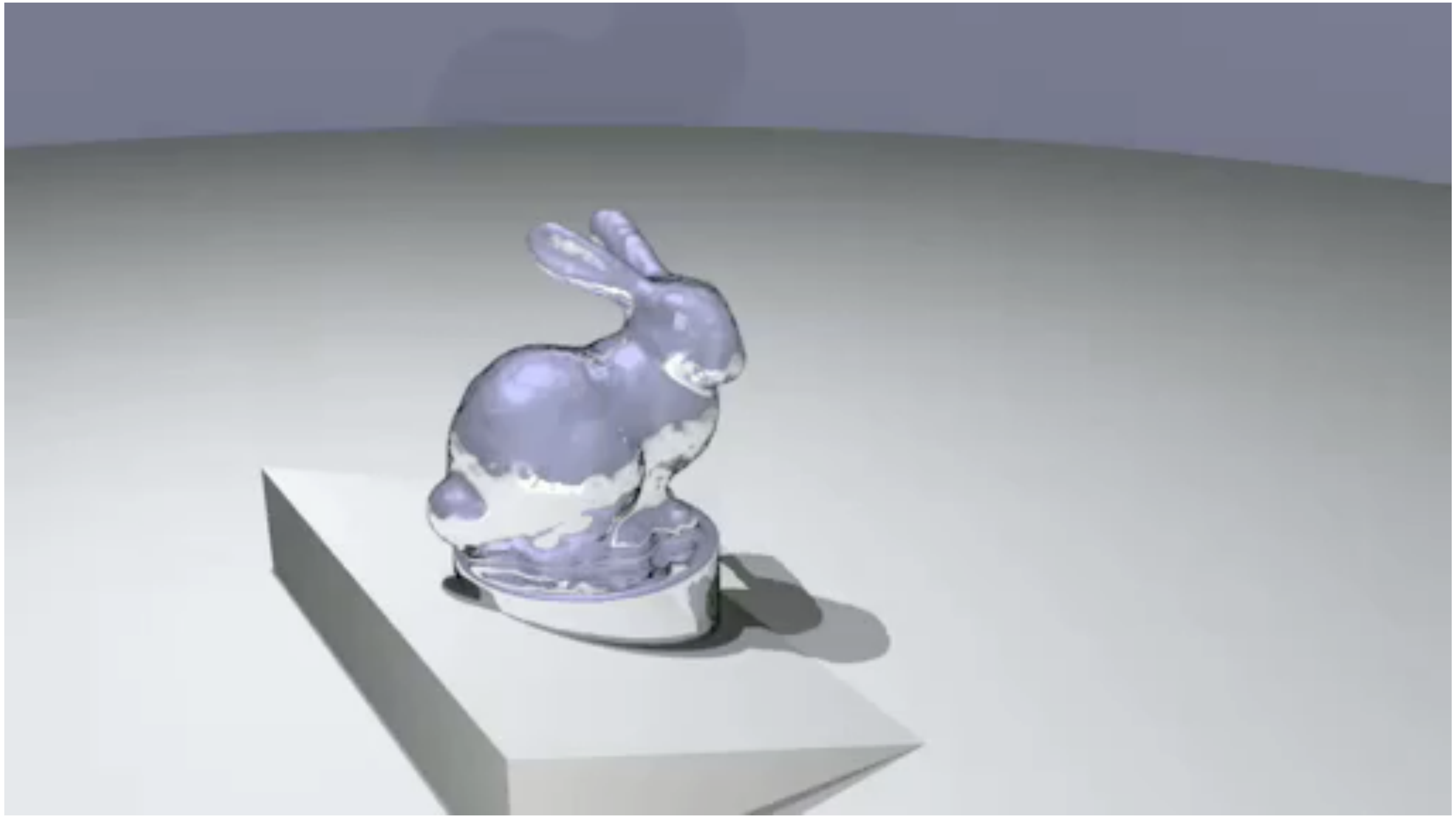
SIGGRAPH2006

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



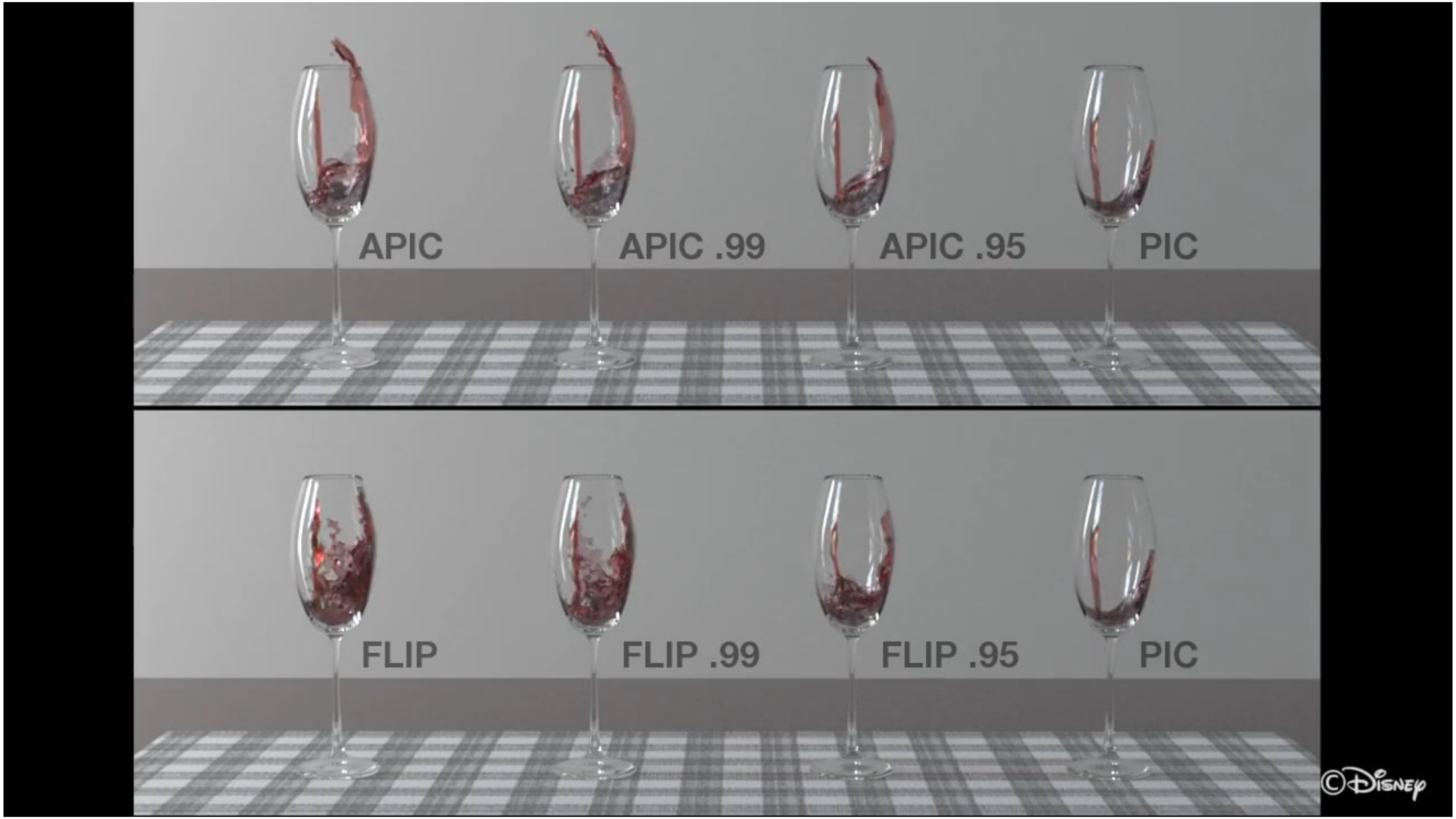
FLuid-Implicit-Particle (FLIP): Less dissipation than PIC

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)



Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. ACM Trans. Graph. 34, 4, Article 51 (July 2015), 10 pages. DOI=dx.doi.org/10.1145/2766996

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

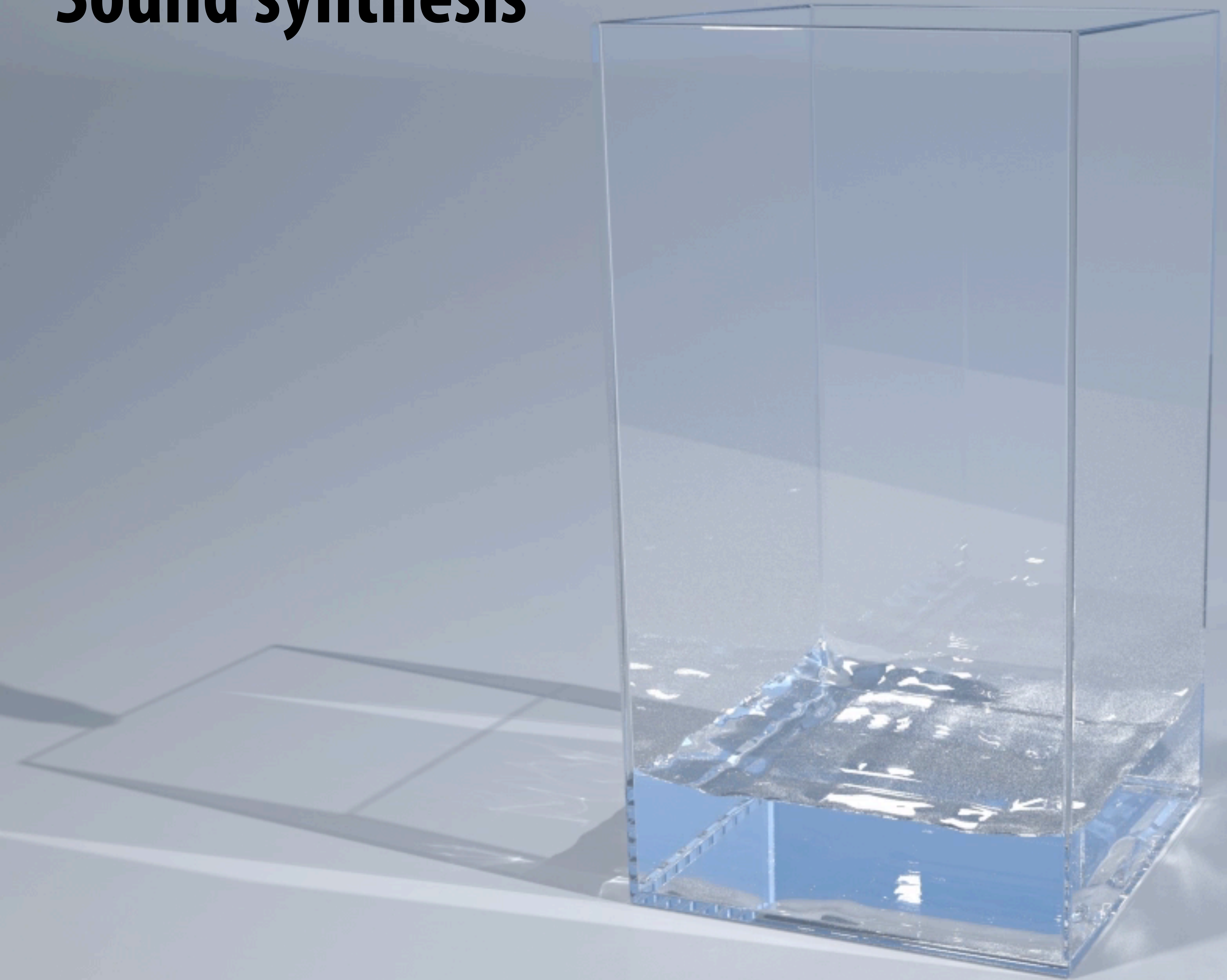
YIHUI MA, Shanghaitech University

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]

Just scratching the surface

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