Position Based Fluids

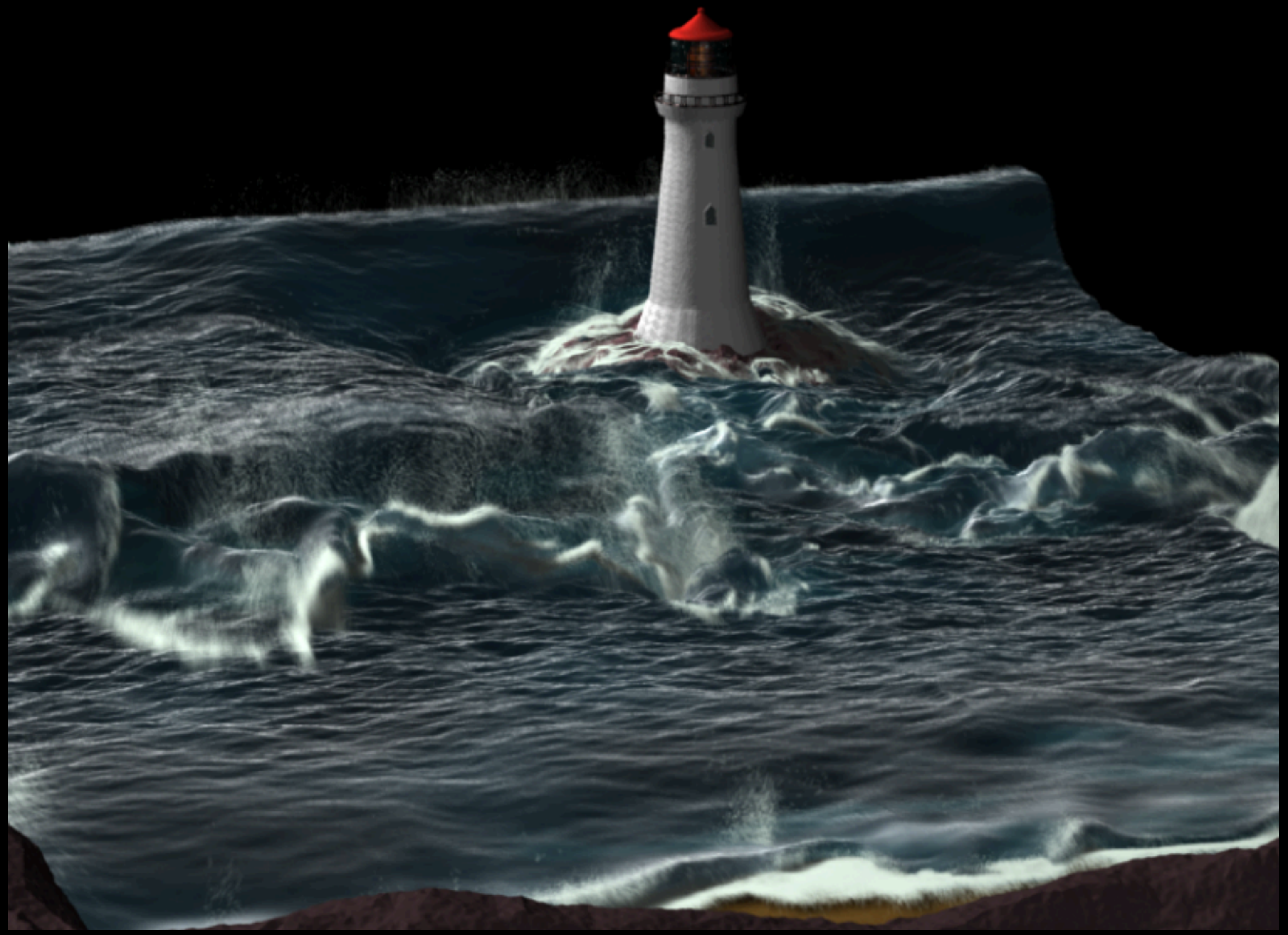
Miles Macklin Matthias Müller

NVIDIA

SIGGRAPH2013

Motivation

- Want great particle based fluid simulation for games
- Must be fast and stable
- Accuracy not so important



[Losasso et al. 2007]

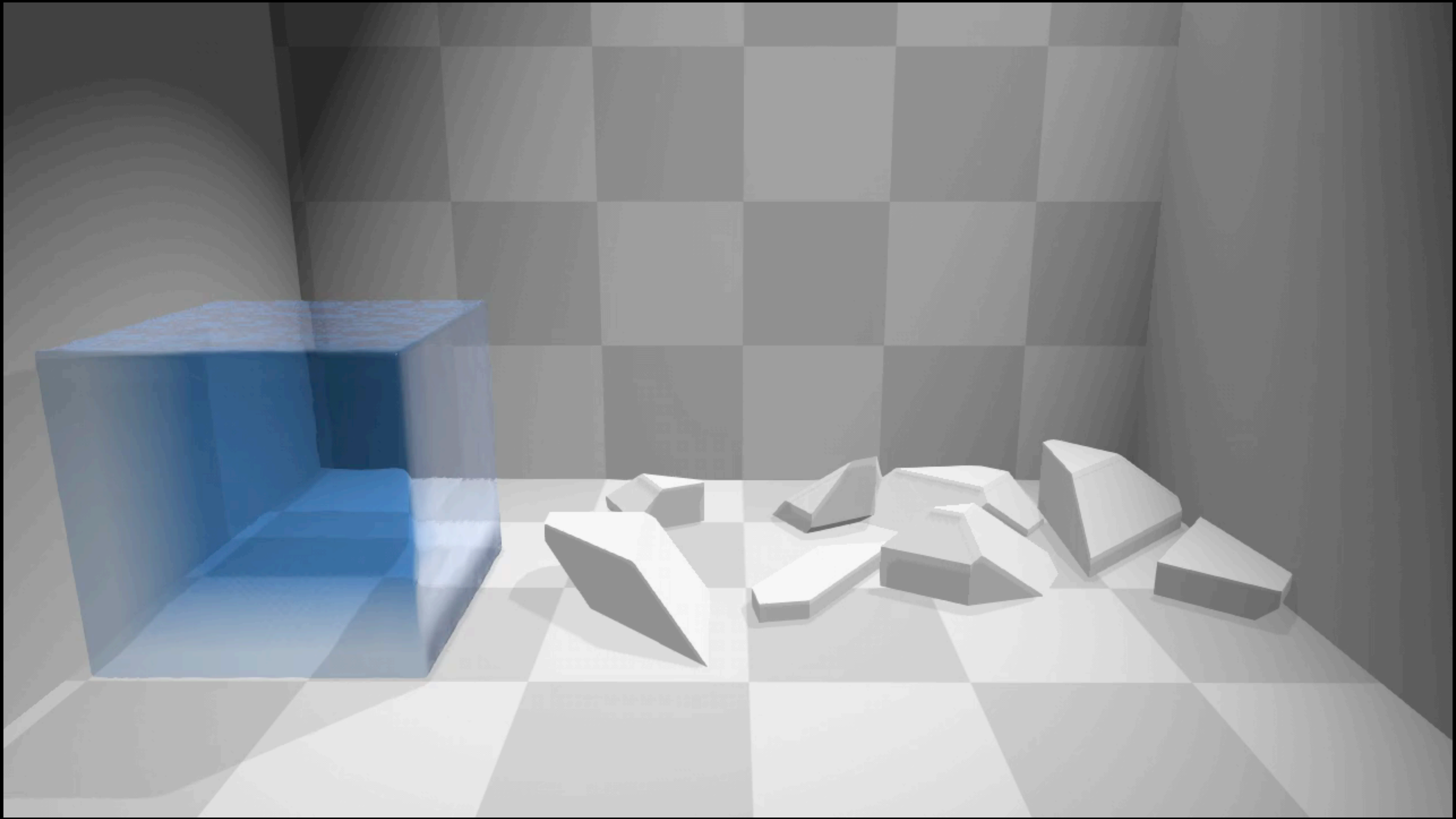
Traditional Particle Fluids

- Smoothed Particle Hydrodynamics (SPH)
- Difficult to tune stiffness (incompressibility)
- Require small time steps (< 0.0016s common)
- Require large smoothing radius

Our Approach

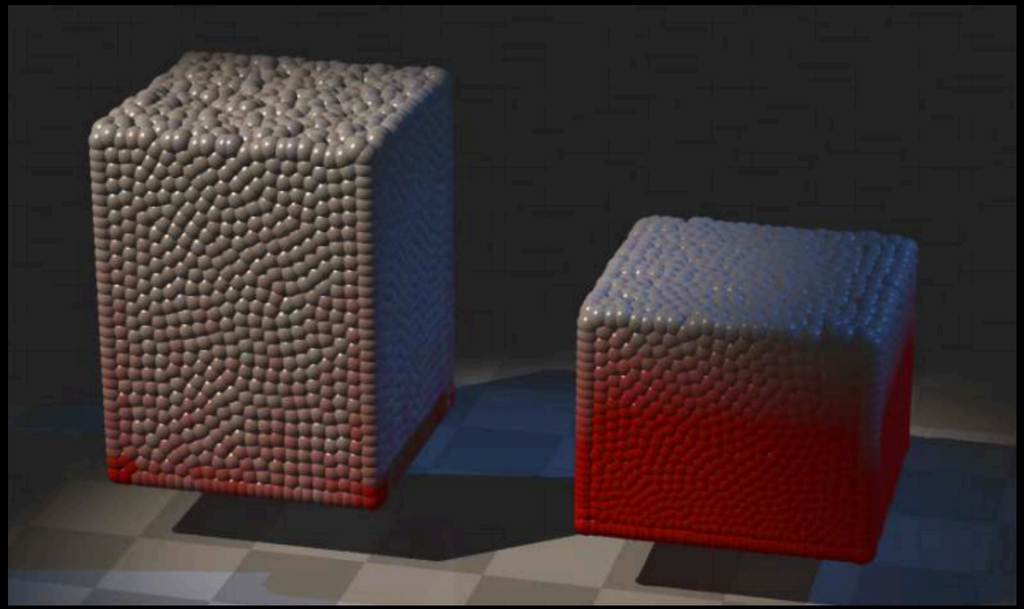
- Combines SPH with Position Based Dynamics
- Allows larger time-steps
- Smaller smoothing radius
- Practical for real-time applications

Examples

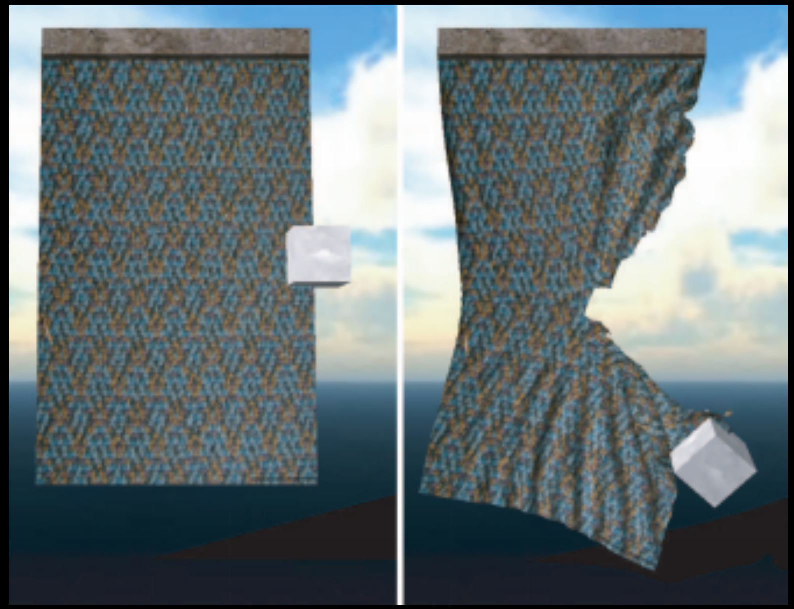


Related Work

PCISPH
[Solenthaler & Parajola 2009]



Constraint Fluids
[Bodin et al. 2012]



Position Based Dynamics
[Müller et al. 2007]

3 Ingredients

- Density Constraint
- Artificial Pressure
- Implementation

Incompressibility

- Most liquids are incompressible
- Want constant fluid density
- Key idea:

Enforce incompressibility using position constraints

Density Constraint

$$C_i(\mathbf{p}_1, ..., \mathbf{p}_n) = \frac{\rho_i}{\rho_0} - 1 = 0$$

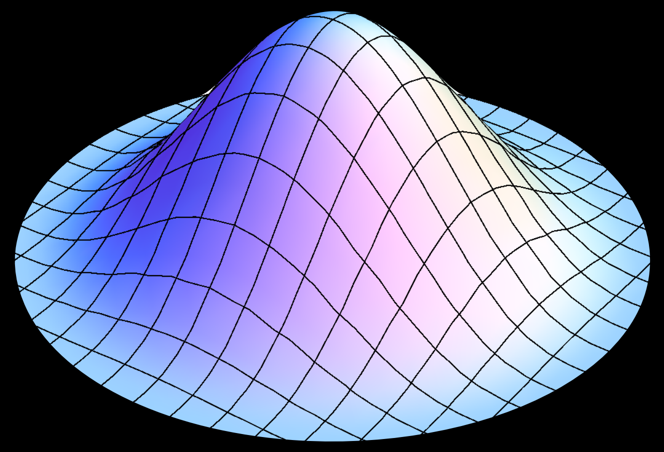
- Positions of particle *i* and its neighbors
- Density at particle *i*
- Rest density (1000kg/m^3)

SPH Density Estimator

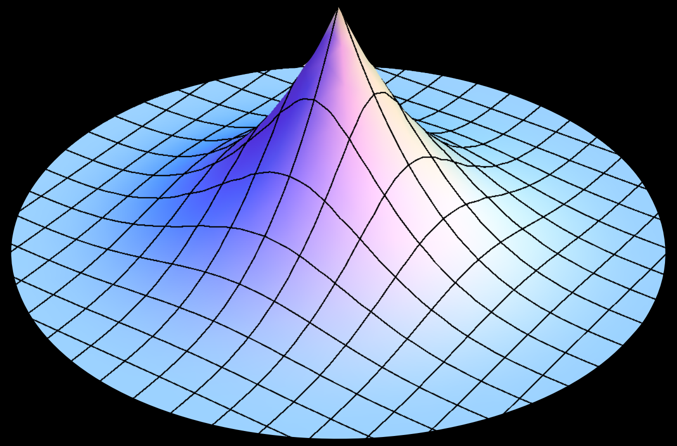
$$\rho_i = \sum_j m_j W(\mathbf{p}_i - \mathbf{p}_j, h)$$

- Density at particle *i*
- Sum over neighbors of *i*
- Smoothing kernel, taking position delta and smoothing radius

SPH Kernel Functions



$$W_{poly6}(\mathbf{r}, h) = \frac{315}{64\pi h^9}(h^2 - |\mathbf{r}|^2)^3$$

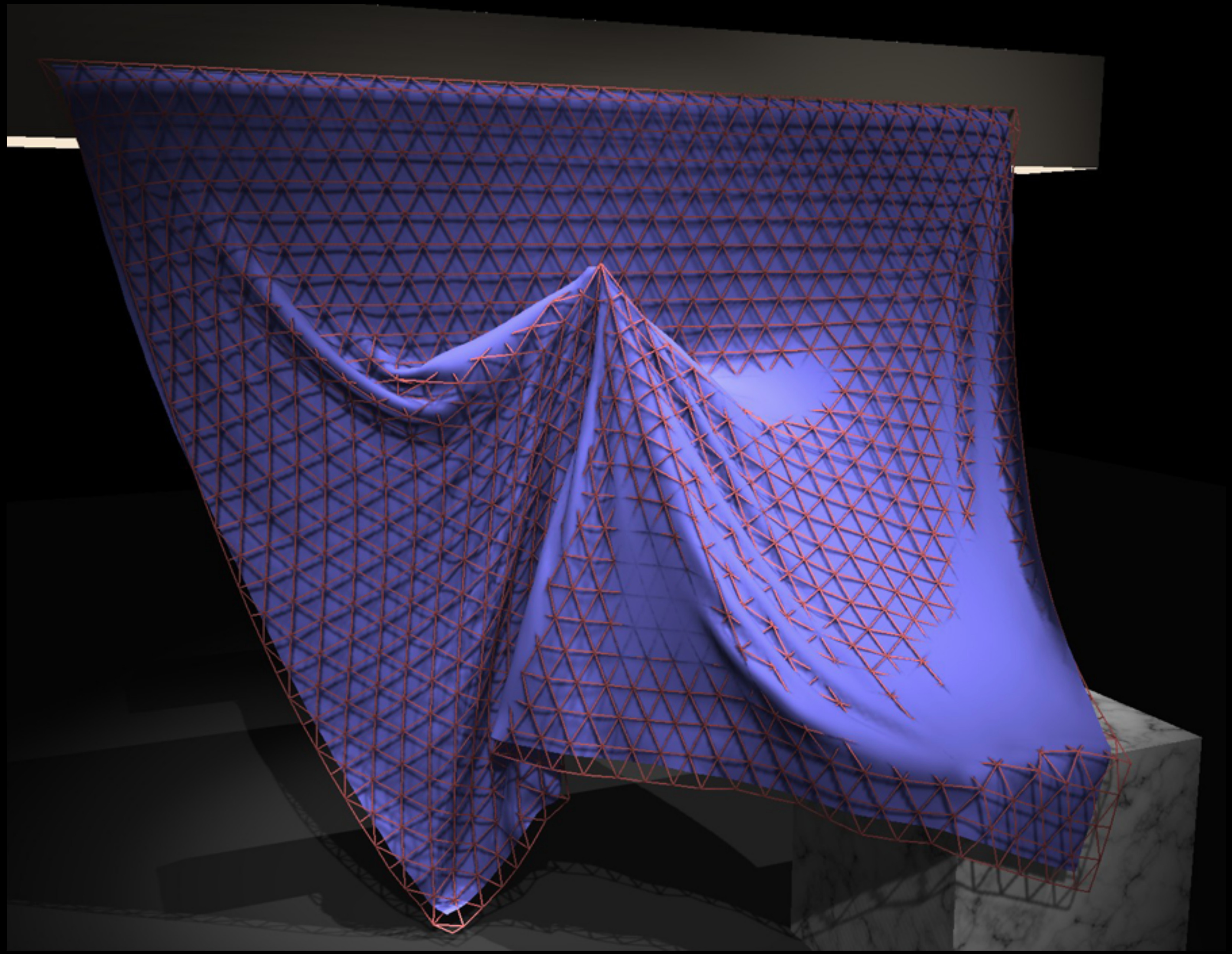


$$\nabla W_{spiky}(\mathbf{r}, h) = \frac{45}{\pi h^6}(h - |\mathbf{r}|)^2 \frac{\mathbf{r}}{|\mathbf{r}|}$$

How to solve the constraints?

Position Based Dynamics (PBD)

- Non-linear constraint solver
- Used extensively in games/film
- Unconditionally stable



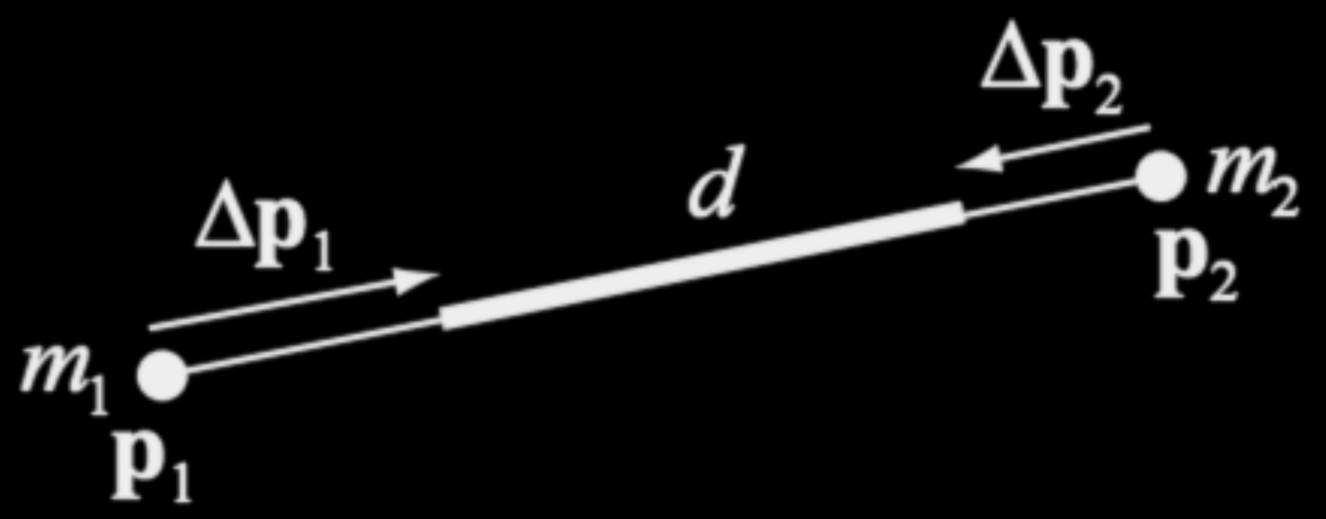
Constraint Types

- Distance constraint (mass spring)
- Non-penetration (collision)
- Bending constraint (triangle mesh)
- Density constraint (fluid)

[Bender et al. 2012]

How PBD solves constraints

- For each constraint
  - Project particles along constraint gradient
- Newton's method applied to each constraint individually
- Iterate until ~converged



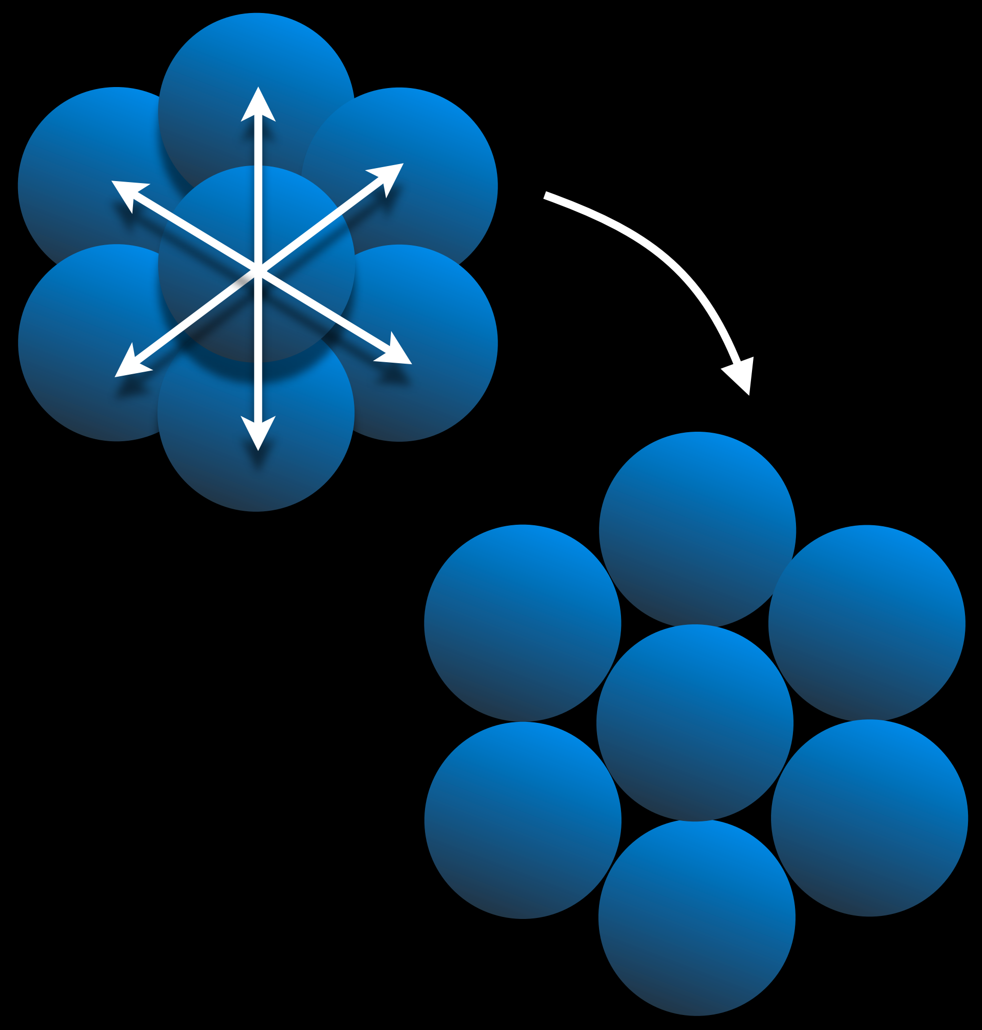
Solving the Density Constraint

- Position delta:

$$\Delta \mathbf{p}_i = \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j) \nabla W(\mathbf{p}_i - \mathbf{p}_j, h)$$

- Constraint force:

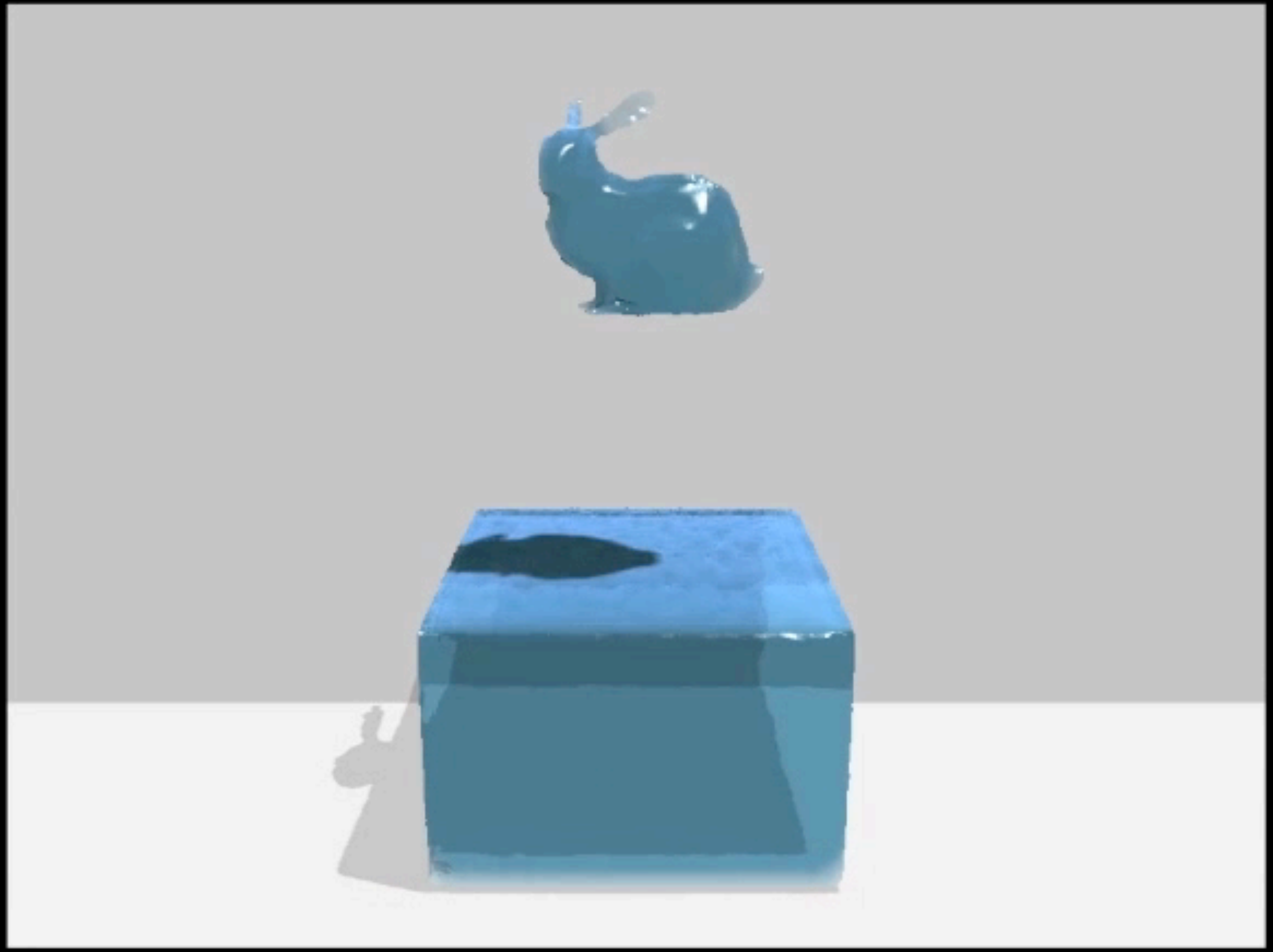
$$\lambda_i = -\frac{C_i(\mathbf{p}_1, \ldots, \mathbf{p}_n)}{\sum_k |\nabla_{\mathbf{p}_k} C_i|^2}$$



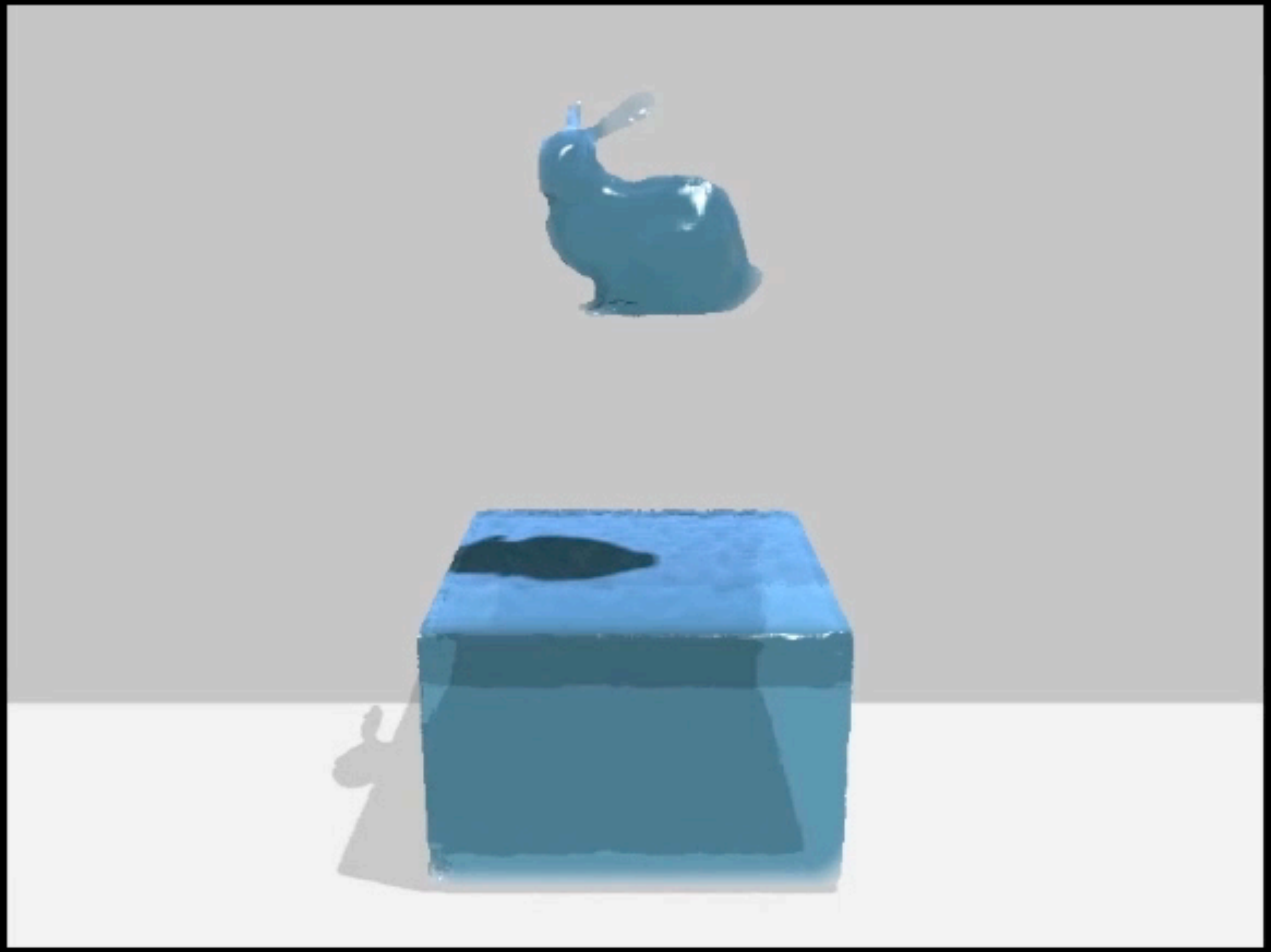
Advantages

- Unlike PCISPH does not accumulate pressure each iteration
- Avoids problems with isolated/trapped particles
- Allows larger time steps
- Integrates easily with other position based methods

Comparison with PCISPH

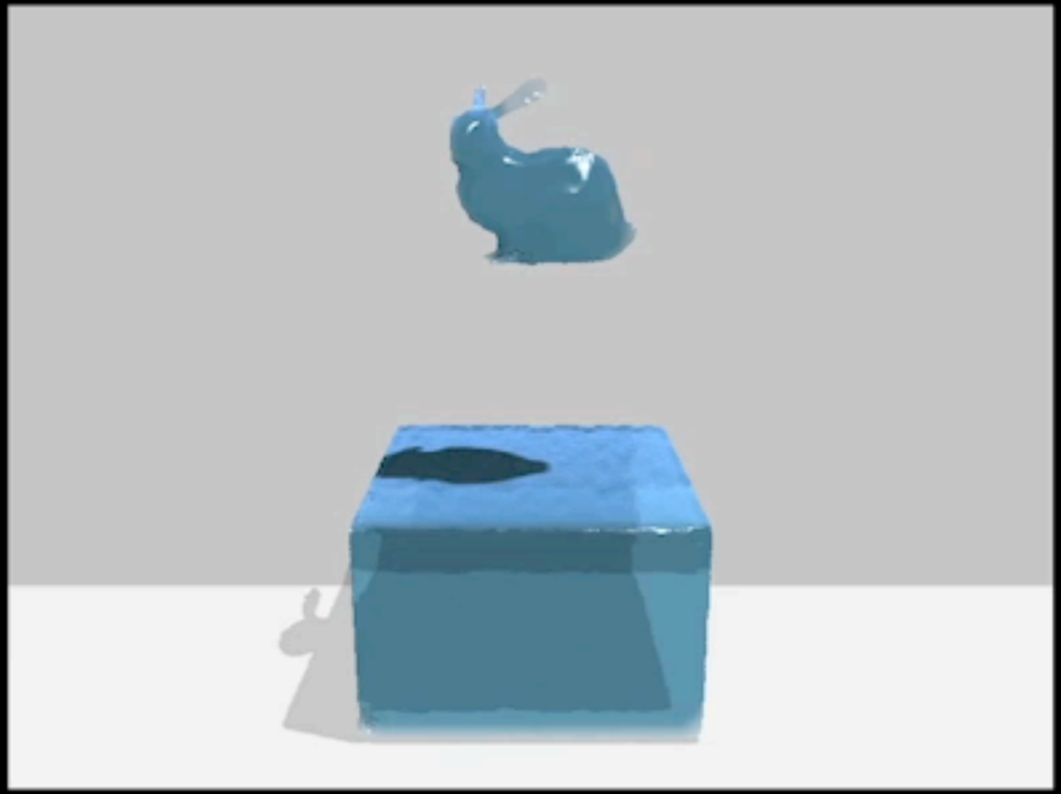


PCISPH
(10 steps, 4 iterations / frame)

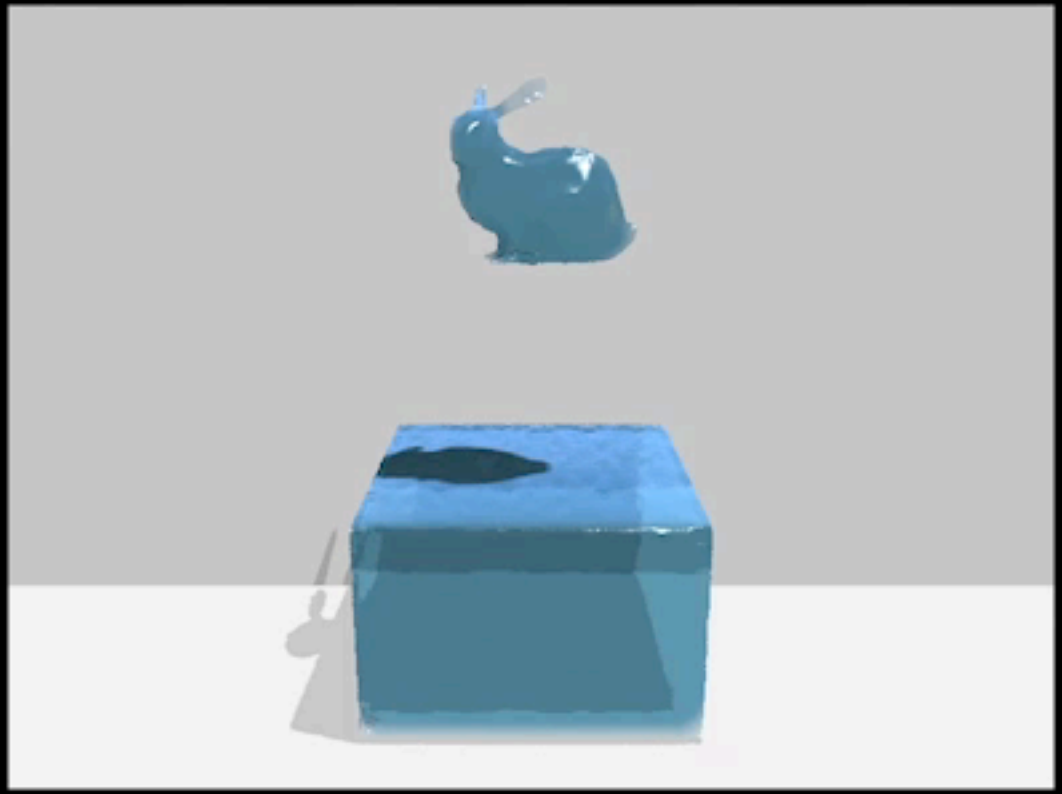


Our Method
(4 steps, 10 iterations / frame)

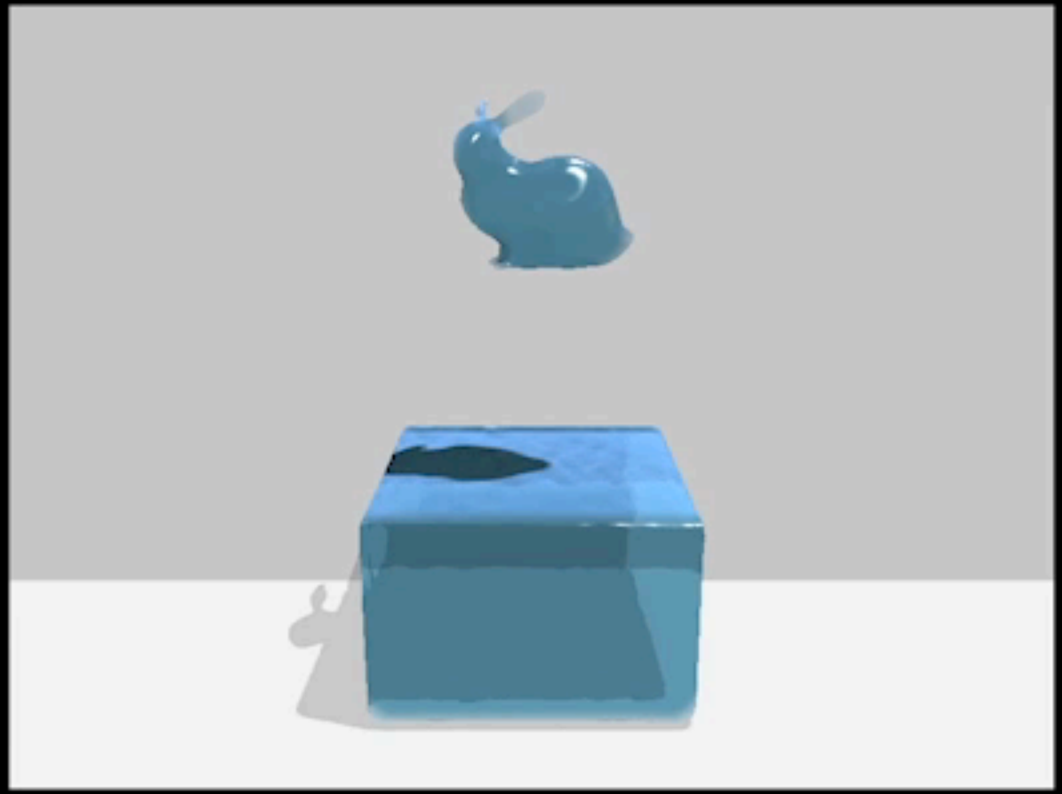
Graceful Degradation



1 step / 40 iterations



1 step / 10 iterations



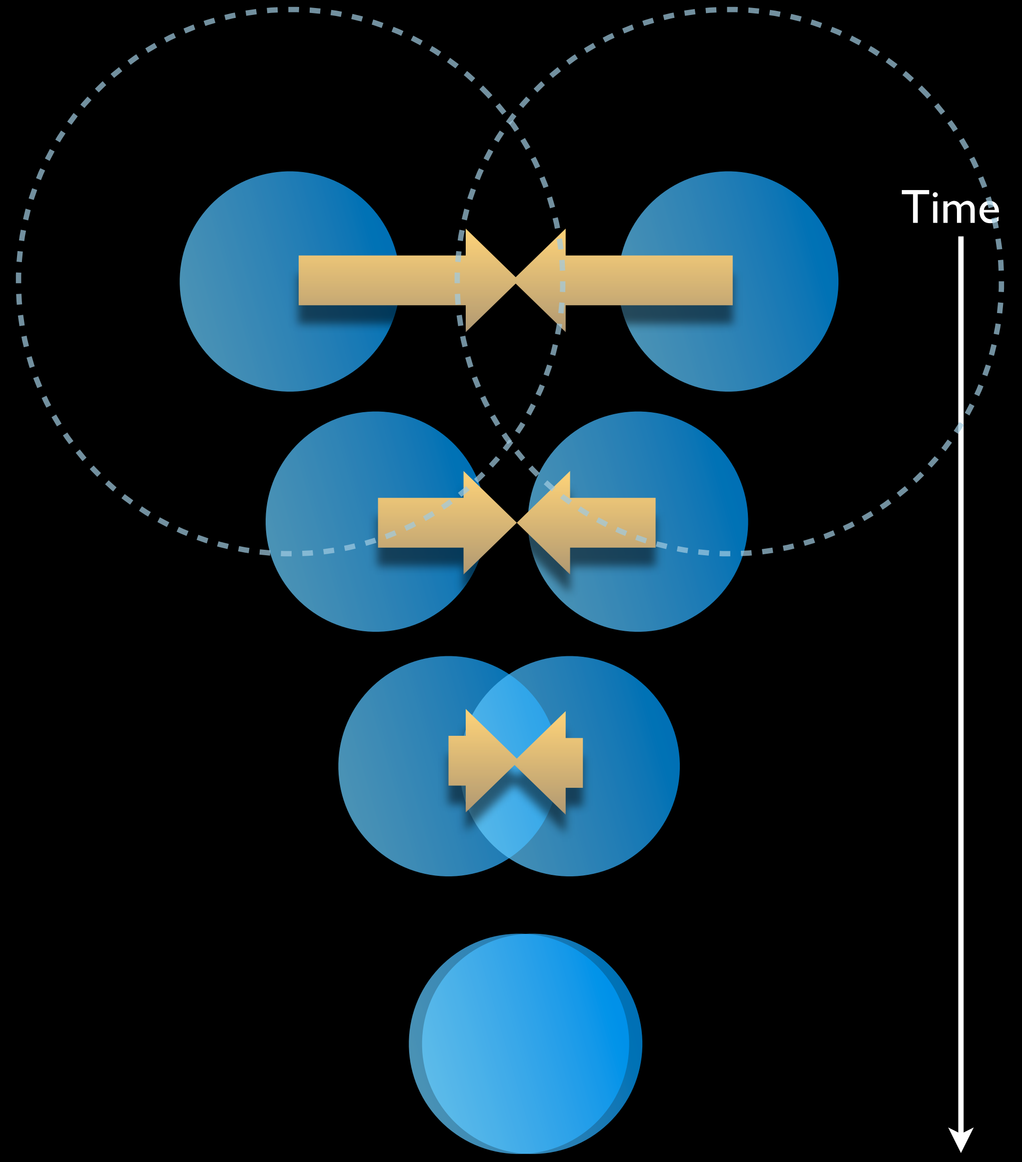
1 step / 1 iteration

3 Ingredients

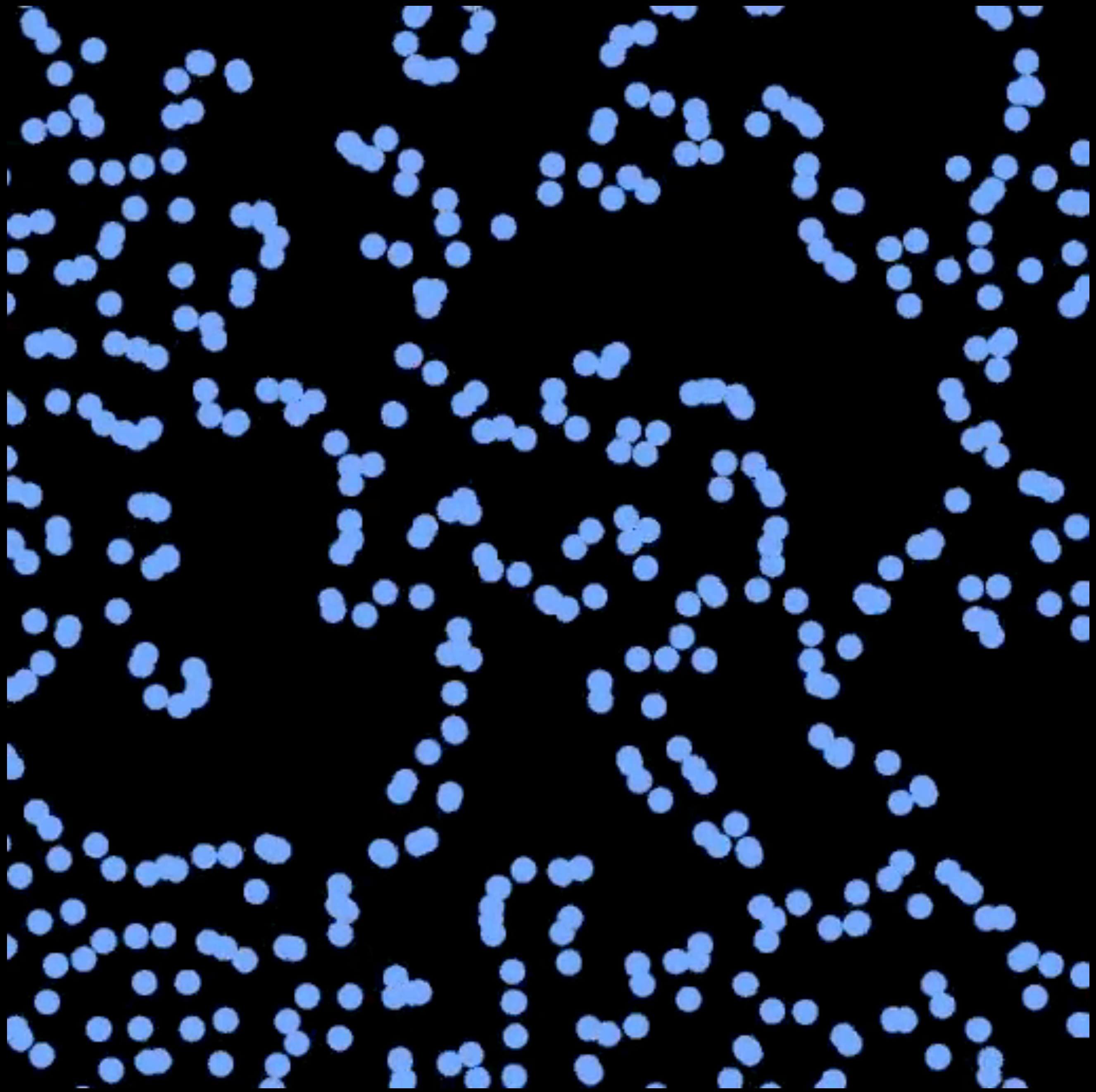
- Density Constraint
- Artificial Pressure
- Implementation

Tensile Instability

- SPH requires a minimum number of neighbors
- Particles "pull" each other together to try and reach rest density
- Form into clumps

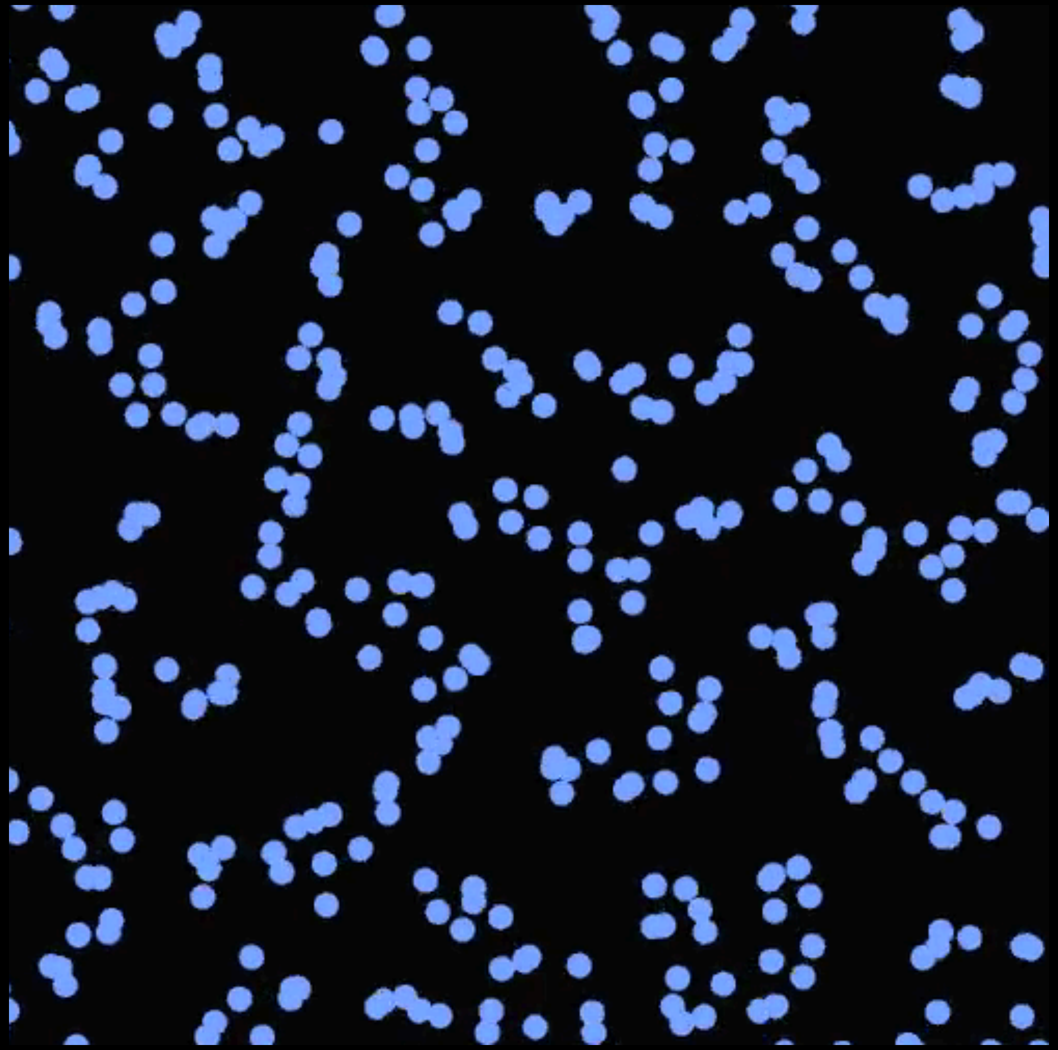


Without Artificial Pressure



Possible Solutions to Clumping

- Clamp constraint function >= 0
- Use more particles
- Use smaller time-steps
- Use larger smoothing radius



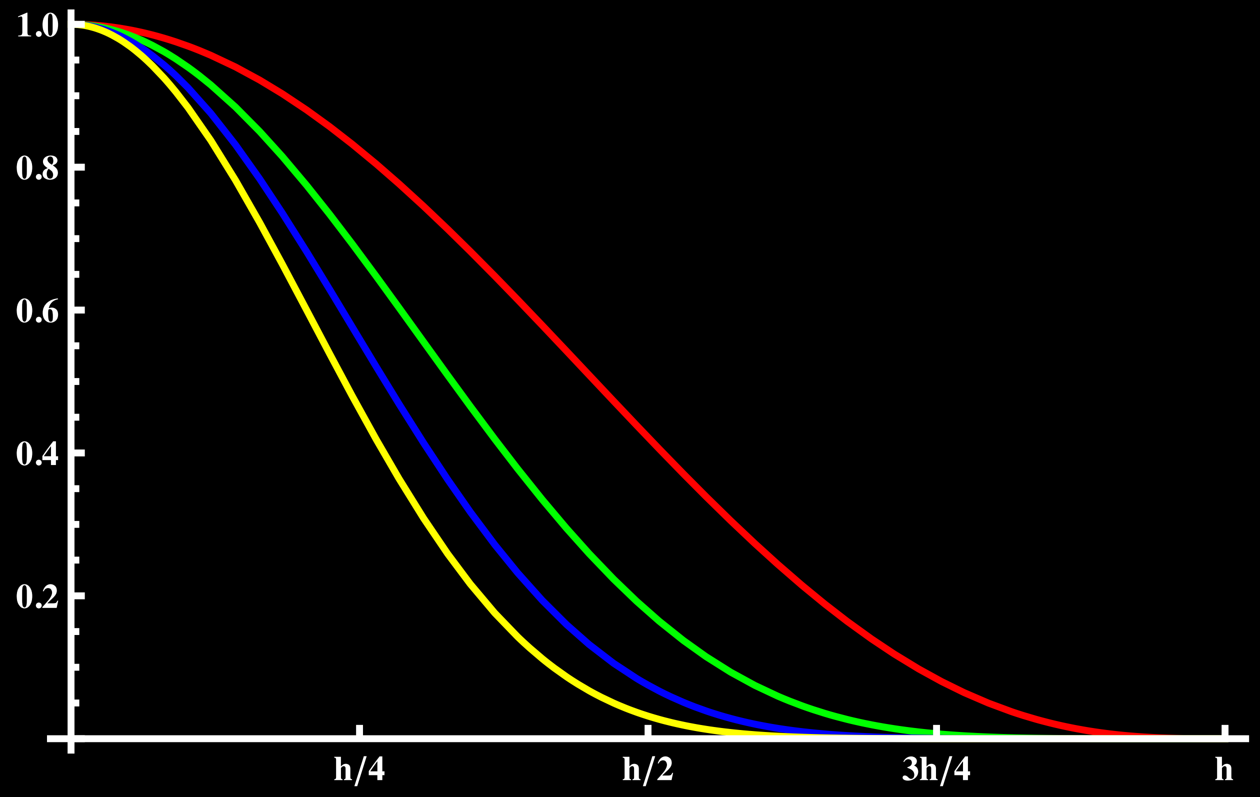
Artificial Pressure

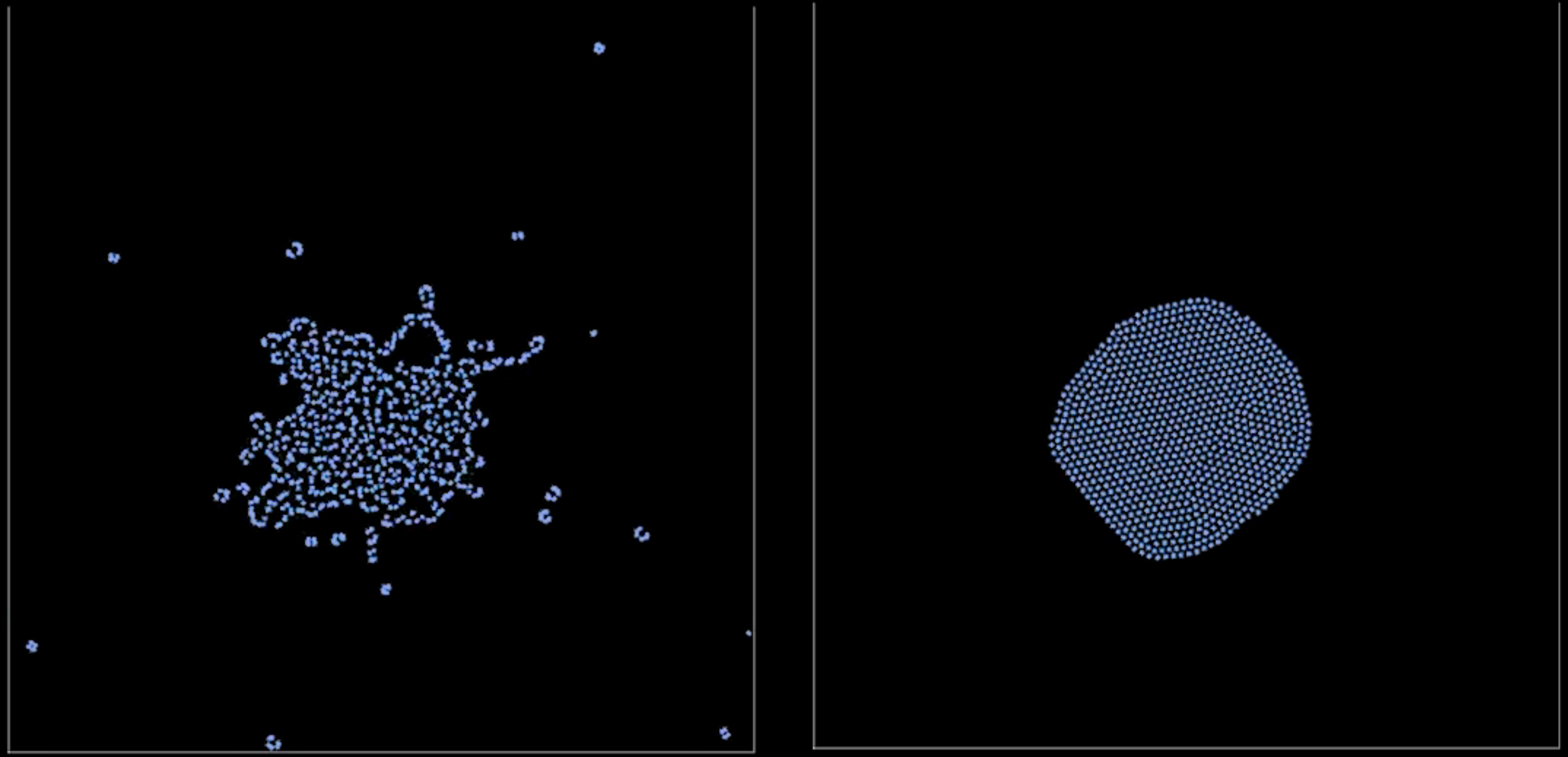
- Inspired by [Monaghan 2000]
- Add a small repulsive term to constraint
- Typical values: $k = 0.001$

$$n = 4$$

$$\Delta \mathbf{q} = \mathbf{0}$$

$$\lambda_{corr} = -k \left(\frac{W(\mathbf{p}_i - \mathbf{p}_j, h)}{W(\Delta \mathbf{q}, h)} \right)^n$$

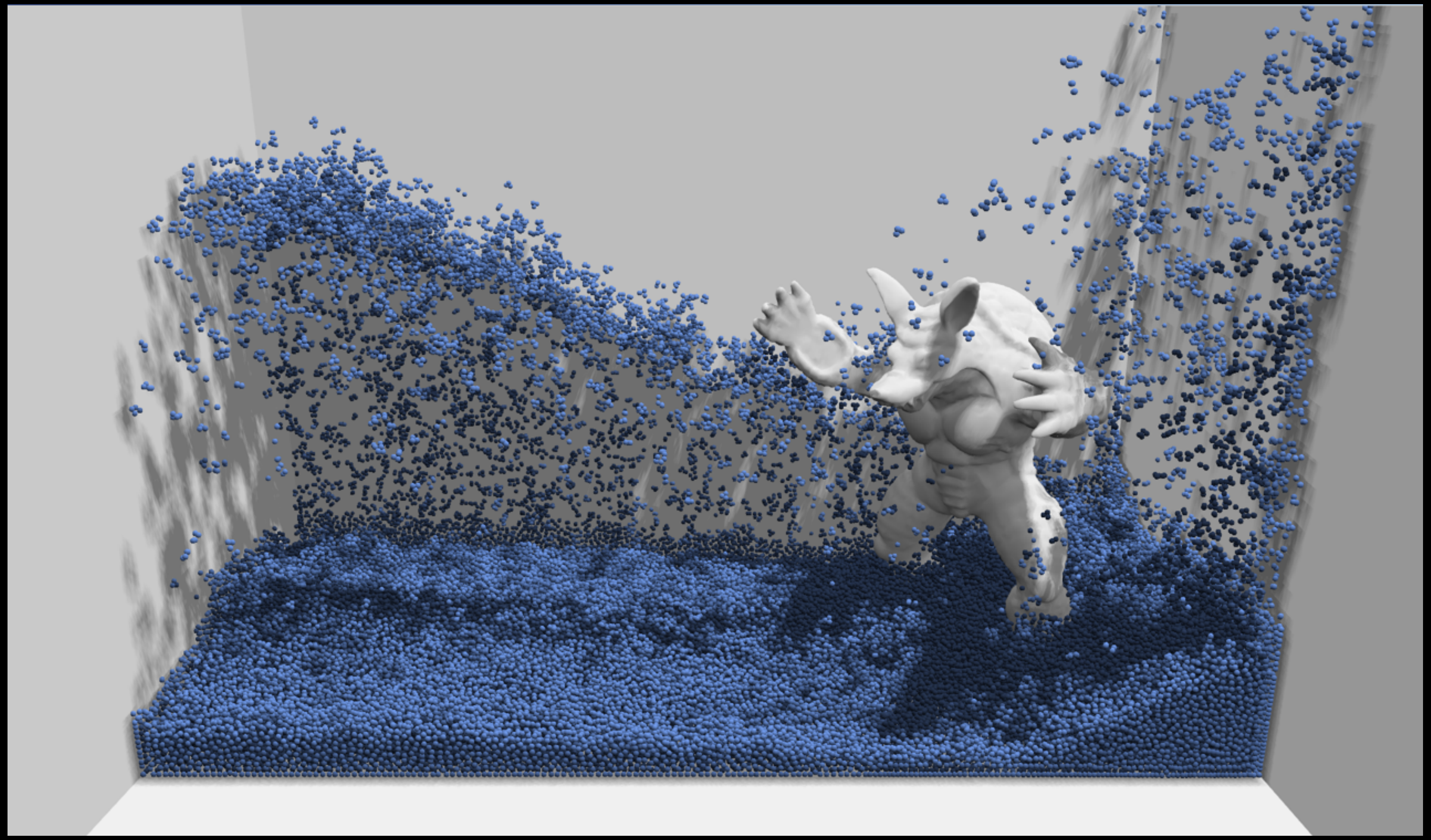




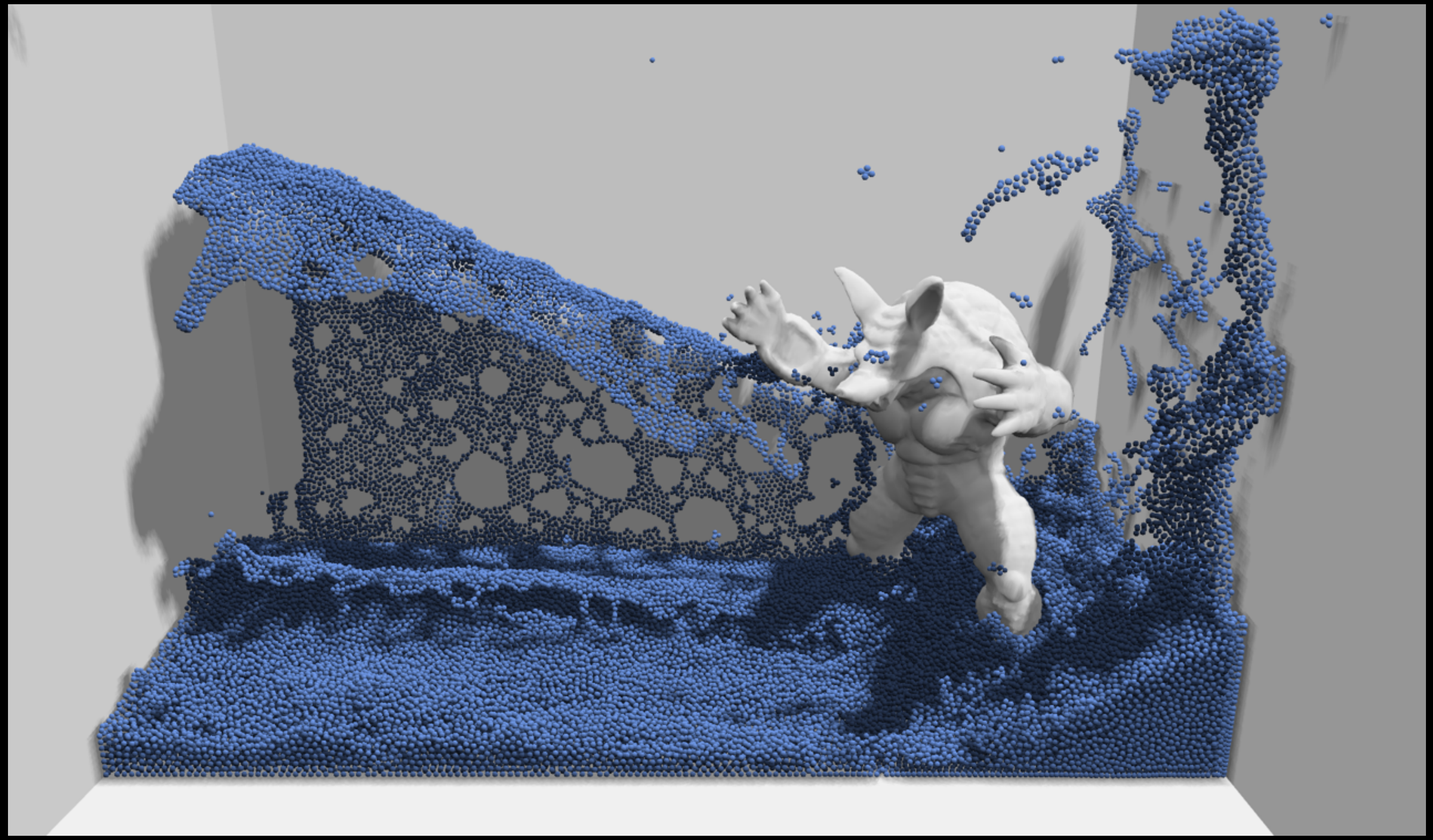
Without Artificial Pressure

With Artificial Pressure

Example in 3D



Without Artificial Pressure



With Artificial Pressure

Artificial Pressure Advantages

- Uniform particle distribution
- Particle cohesion (approximate surface tension)
- Smaller smoothing radius

3 Ingredients

- Density Constraint
- Artificial Pressure
- Implementation

Implementation

- 10 CUDA kernels
- GPU Hash Grid [Green 2008]
- Parallel Jacobi iteration
- 128k particles, 6 iterations, 10ms/frame on GTX680

Solver Loop

1. Predict New Positions

$$\mathbf{v}^* = \mathbf{v}^{\mathbf{t}} + \Delta t \mathbf{f}$$

$$\mathbf{p}^* = \mathbf{p}^{\mathbf{t}} + \Delta t \mathbf{v}^*$$

2. Find Particle Neighbors
3. For *n* iterations
   a. Calculate Density
   b. Calculate Position Delta (output delta to temp memory)
   c. Update $\mathbf{p}^*$ (apply delta)
   d. Collision Handling (project to surface)
4. Update Velocity

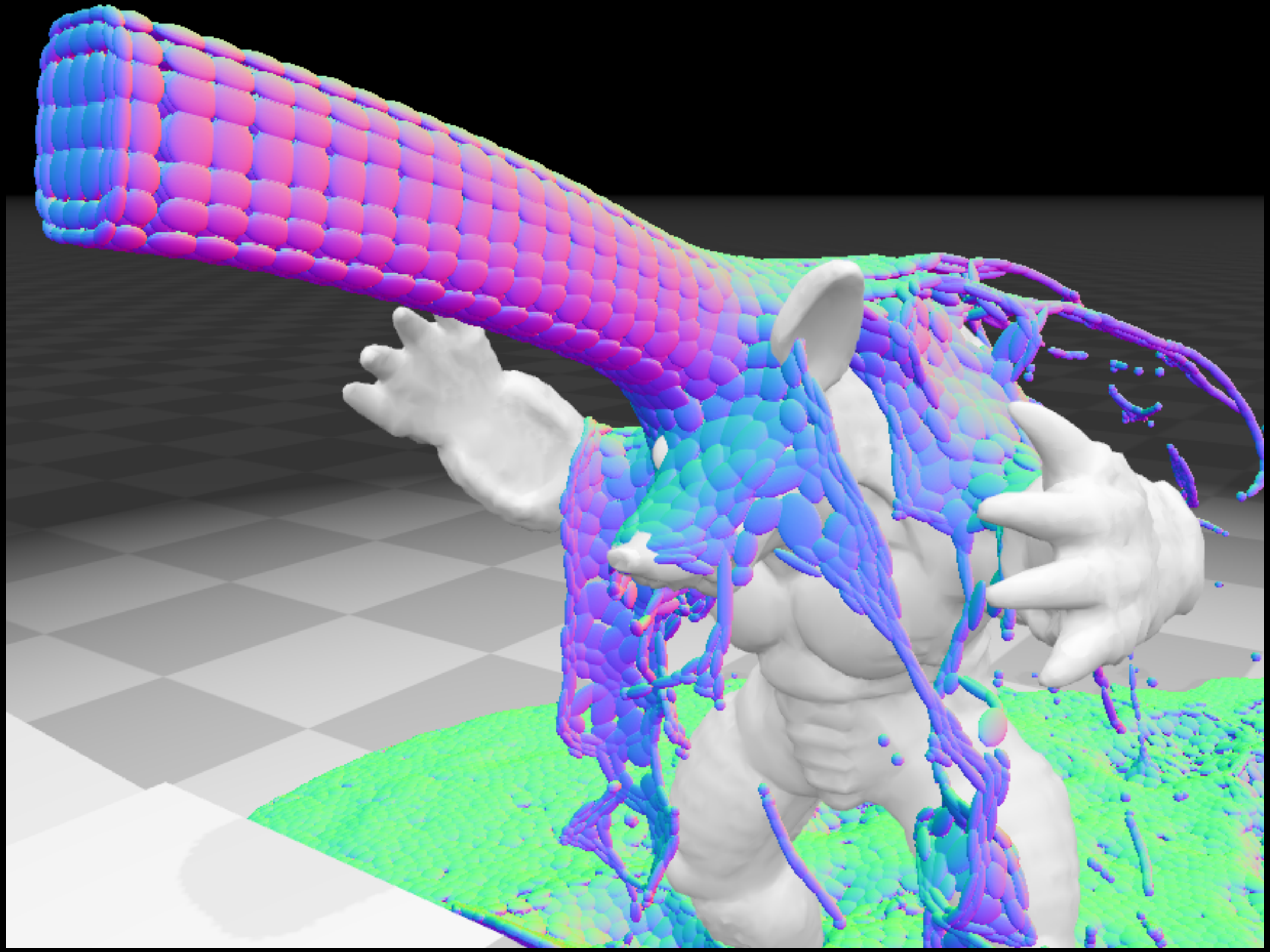
$$\mathbf{v}^{\mathbf{t+1}} = \frac{\mathbf{p}^* - \mathbf{p}^{\mathbf{t}}}{\Delta t}$$

$$\mathbf{p}^{\mathbf{t+1}} = \mathbf{p}^*$$

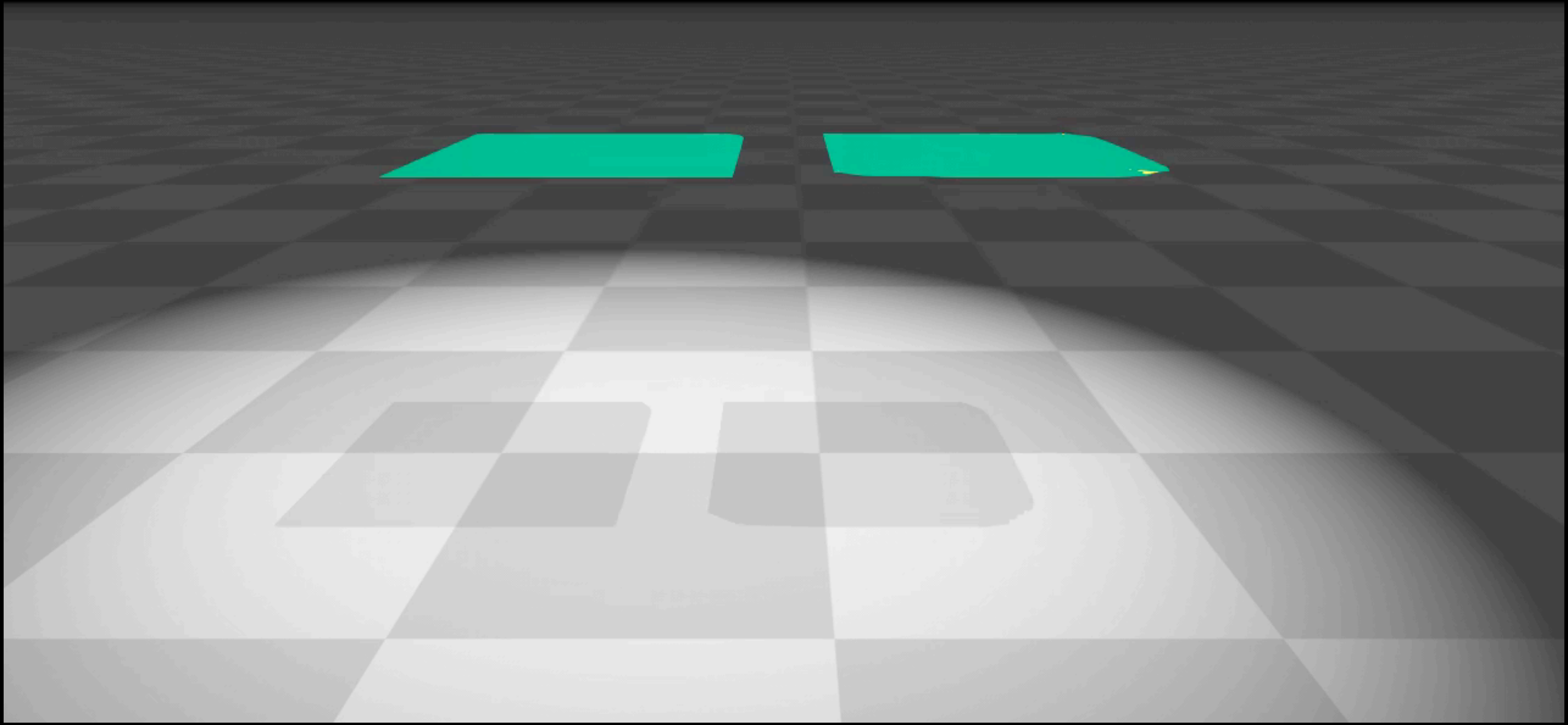
5. Apply XSPH Viscosity + Vorticity Confinement

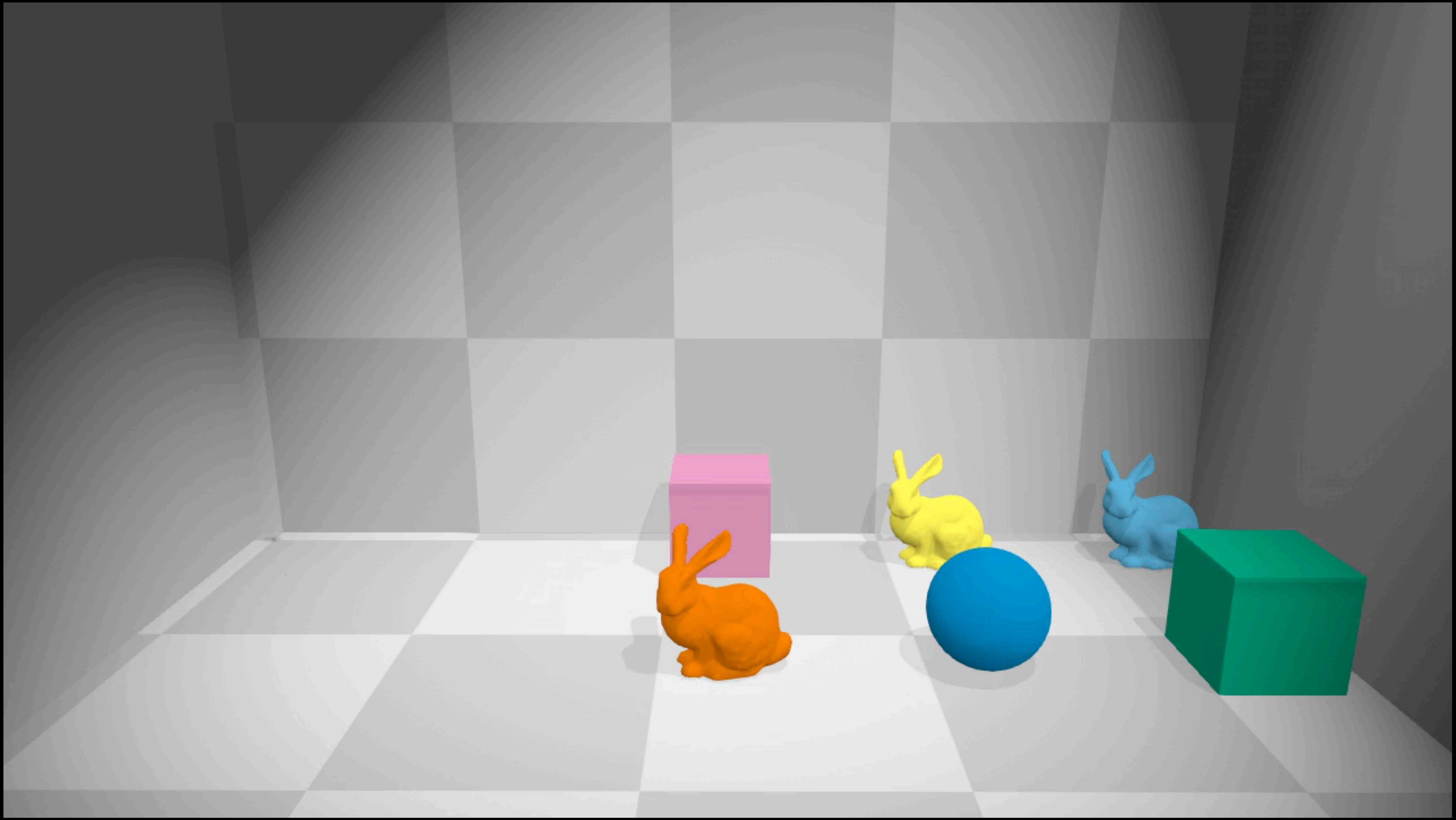
Rendering

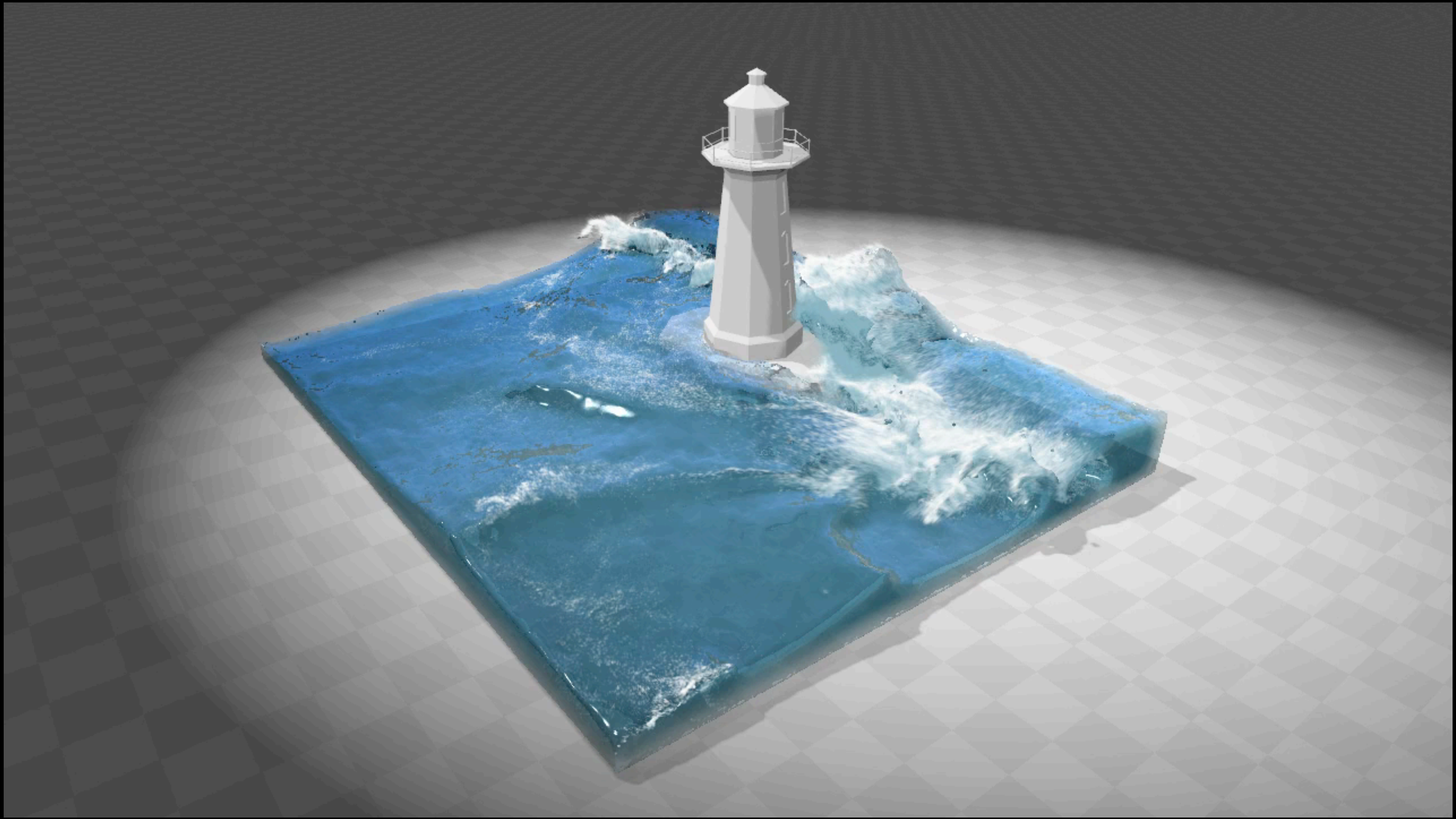
- Anisotropic Particles [Yu & Turk 2013]
- Ray-Traced Quadrics [Sigg 2006]
- Screen Space Surface Reconstruction [van der Laan et al. 2009]



Recent Results







Thank You

Acknowledgments

- Thanks to the PhysX team and paper reviewers

- Contact details:
- mmacklin@nvidia.com
- @milesmacklin

References

- [Solenthaler & Parajola 2009] PCISPH https://sph-sjtu-f06.googlecode.com/files/a40-solenthaler.pdf
- [Bodin et al. 2012] Constraint Fluids http://www.physics.umu.se/digitalAssets/60/60425_constraint-fluids-ieee.pdf
- [Green 2008] CUDA SDK Particles Sample http://docs.nvidia.com/cuda/samples/5_Simulations/particles/doc/particles.pdf
- GPU-Based Ray Casting of Quadratic Surfaces [Sigg 2006] www.cs.ucl.ac.uk/staff/t.weyrich/projects/quadrics/pbg06.pdf
- [Yu & Turk 2010] Reconstructing Surfaces of Particle-Based Fluids Using Anisotropic Kernels http://www.cc.gatech.edu/~turk/my_papers/sph_surfaces.pdf
- [Müller et al. 2009] Position Based Dynamics http://www.matthiasmueller.info/publications/posBasedDyn.pdf
- [Monaghan 2000] SPH without a Tensile Instability
- [Bender et al. 2012] Position-based Methods for the Simulation of Solid Objects in Computer Graphics http://www.matthiasmueller.info/publications/STAR2013.pdf