Instructor: Minchen Li

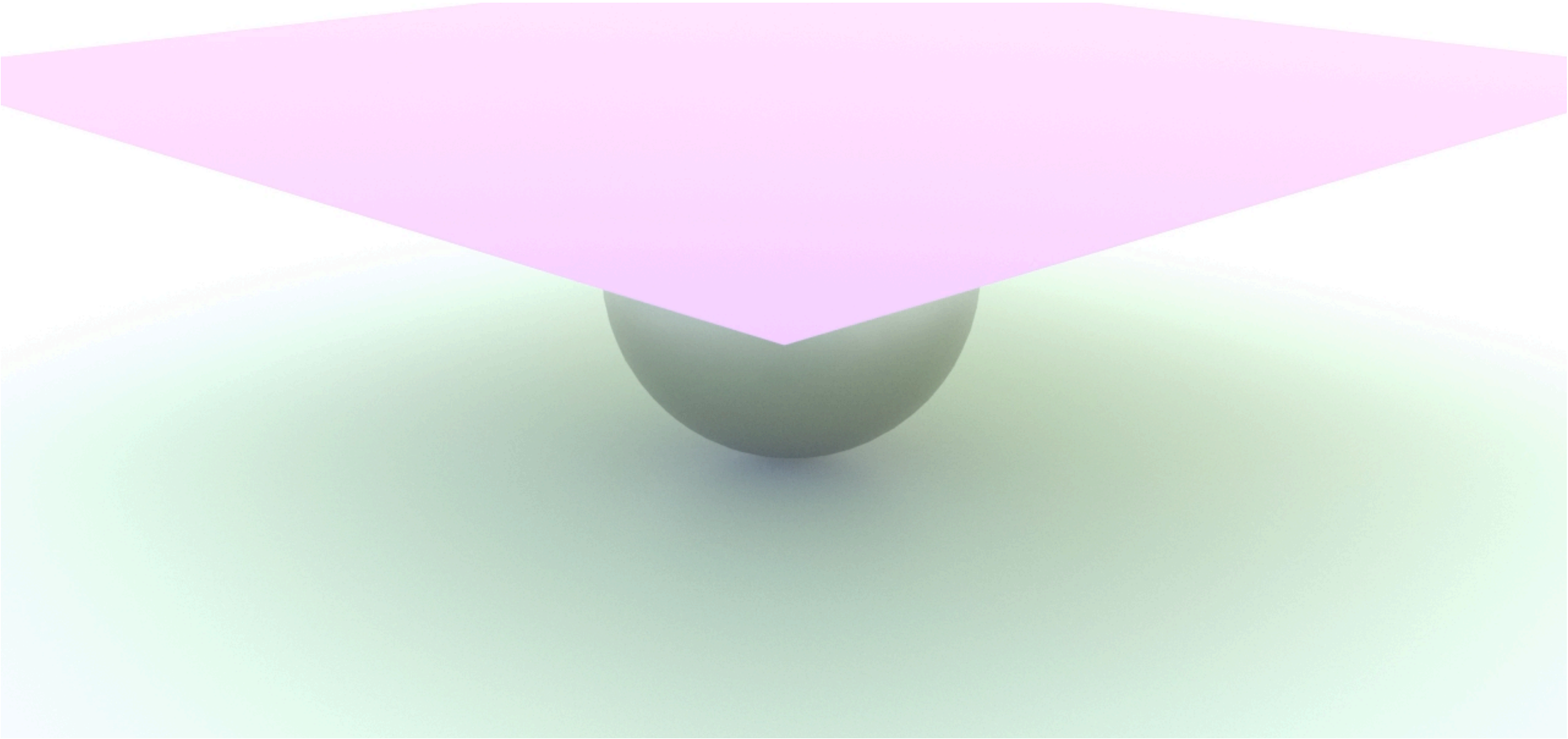
$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + \nu \nabla \cdot \nabla \vec{u},$$
$$\nabla \cdot \vec{u} = 0.$$

Lec 15: Fluid Simulation Fundamentals, SPH

15-769: Physically-based Animation of Solids and Fluids (F23)

Recap: Codimensional Solids — Thin Shells

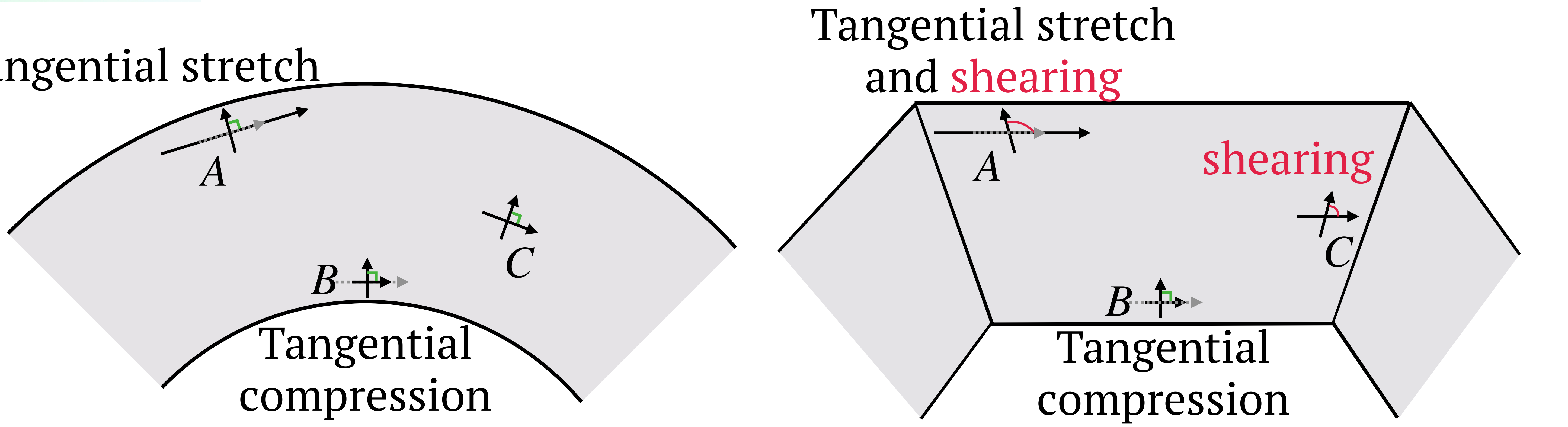
Simulating Using Surface Meshes



Avoid Ill-conditioning:

Avoid shear locking issue (linear shape functions):

Higher-order shape functions are expensive



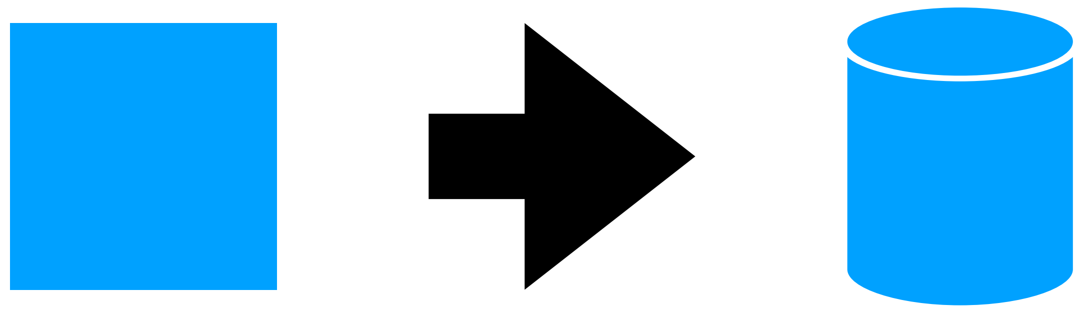
(a) Ideal Setting

(b) Coarse Tessellation with Linear Basis

Recap: Codimensional Solids — Thin Shells

Bending Energy

With only tangent space elasticity, no force under isometric deformation:



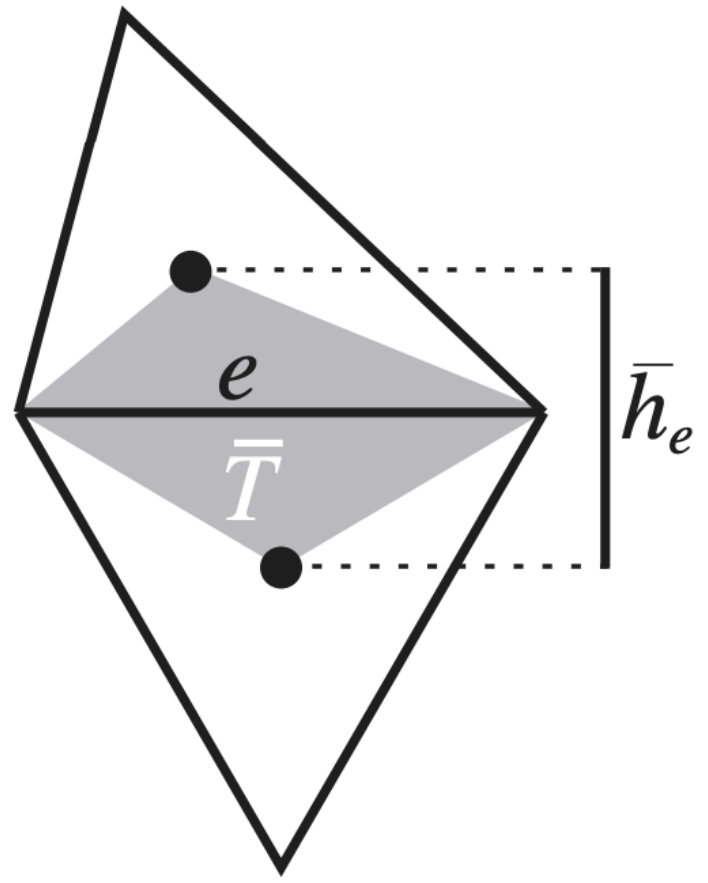
After discretization:

$$\Psi_{\text{bend}}(x) = \sum_i k \frac{3||\bar{e}_i||^2}{\bar{A}_i}(\theta_i - \bar{\theta}_i)^2$$

Model the strain energy for bending directly as a penalty of mean curvature changes

$$\int_{\bar{\Omega}} (H \circ \varphi - \bar{H})^2 d\bar{A}$$

$$k = \frac{E\xi^3}{24(1-\nu^2)}$$



Garg et al. [2007]:
For isometric deformation,
A bending energy can be formulated as
a cubic polynomial of x

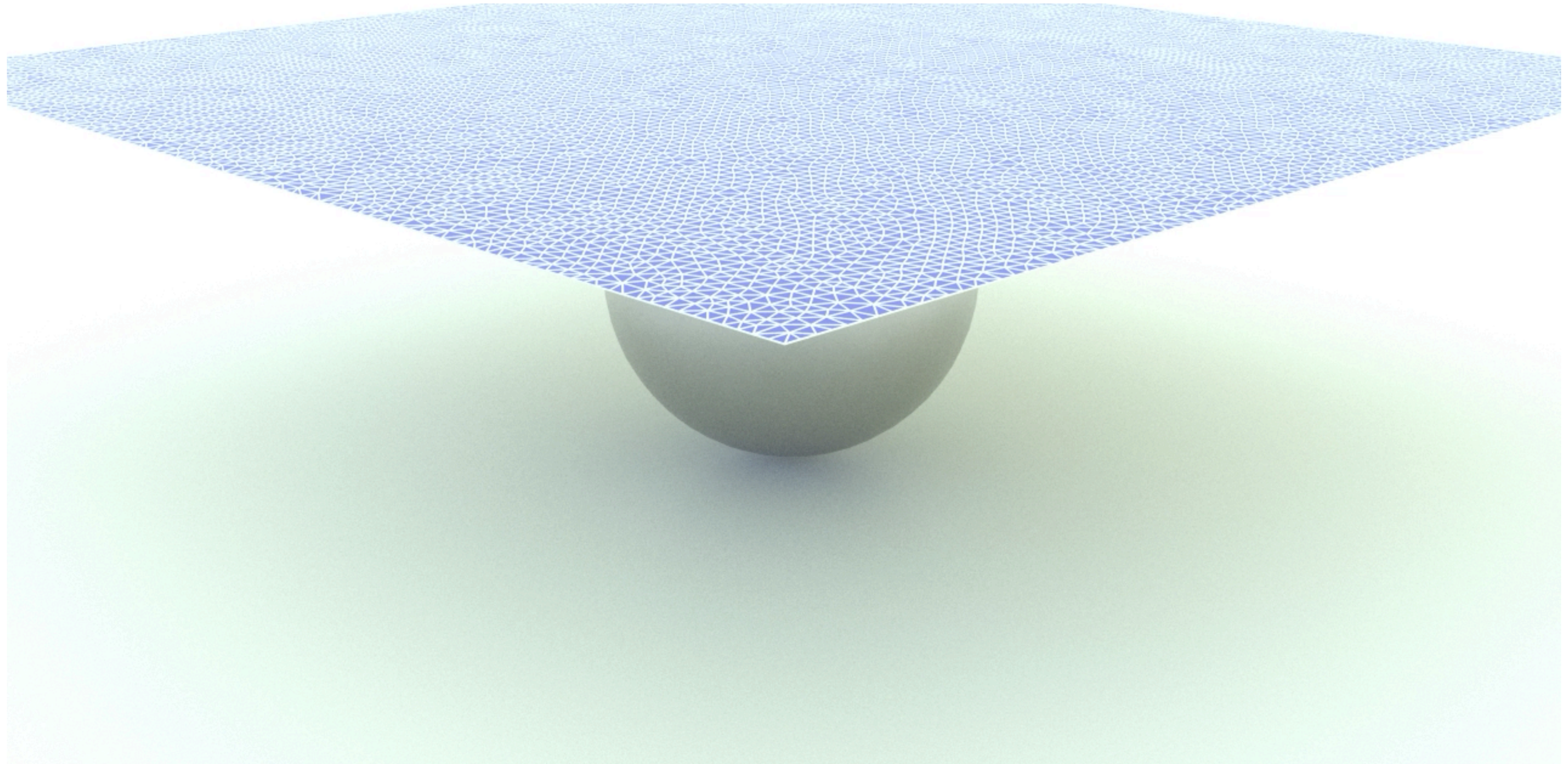
Bergou et al. [2006]:
For isometric deformation of plates (flat rest shapes),
A bending energy can be formulated as
a quadratic polynomial of x

Recap: Codimensional Solids — Thin Shells

Membrane Locking

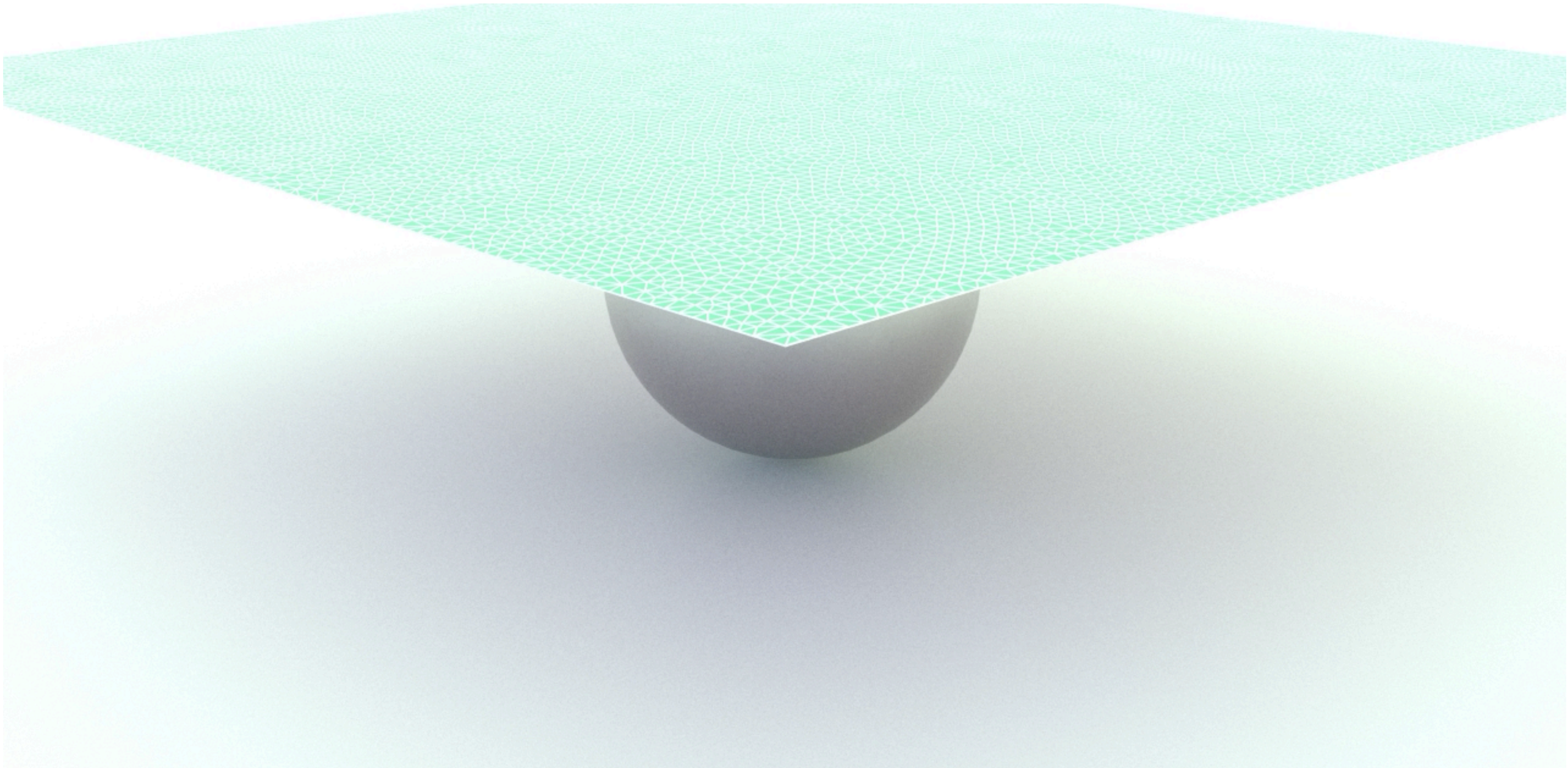
Cloth are nearly unstretchable — stiff stretch resistance, E = ~10^7 Pa

With low-res triangulation, there can be geometric lockings:



Stiff membrane creates extra bending penalty
(real material parameter)

Solution:
Softer material parameters + **Strain limiting**



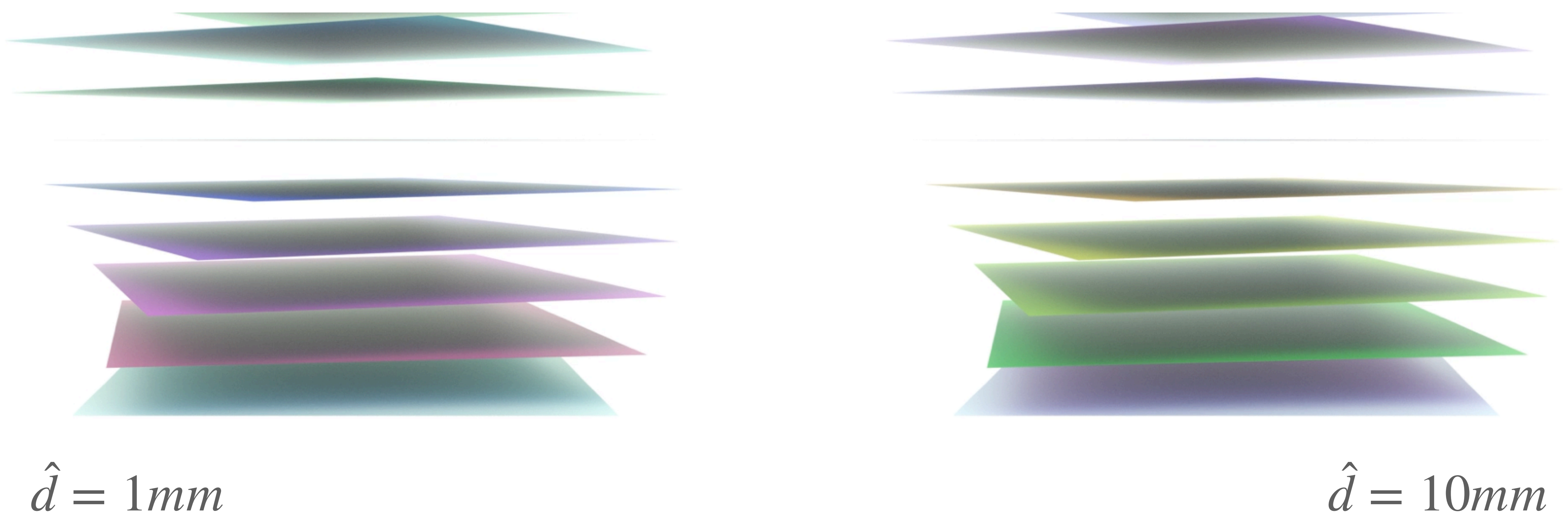
0.01x membrane stiffness + 10% strain limit

Recap: Codimensional Solids — Thin Shells

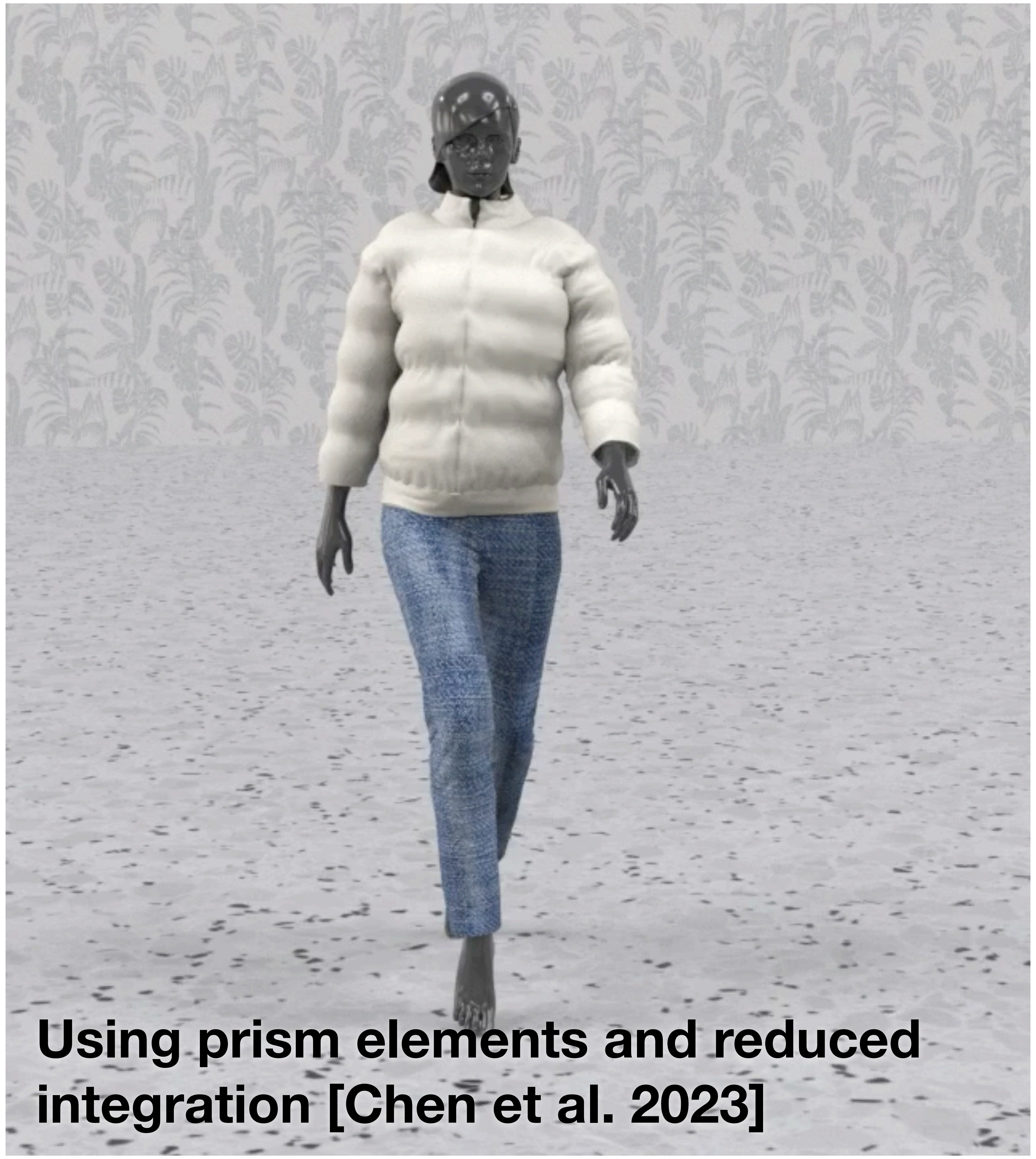
Thickness Modeling

Using IPC:

Elastic Thickness

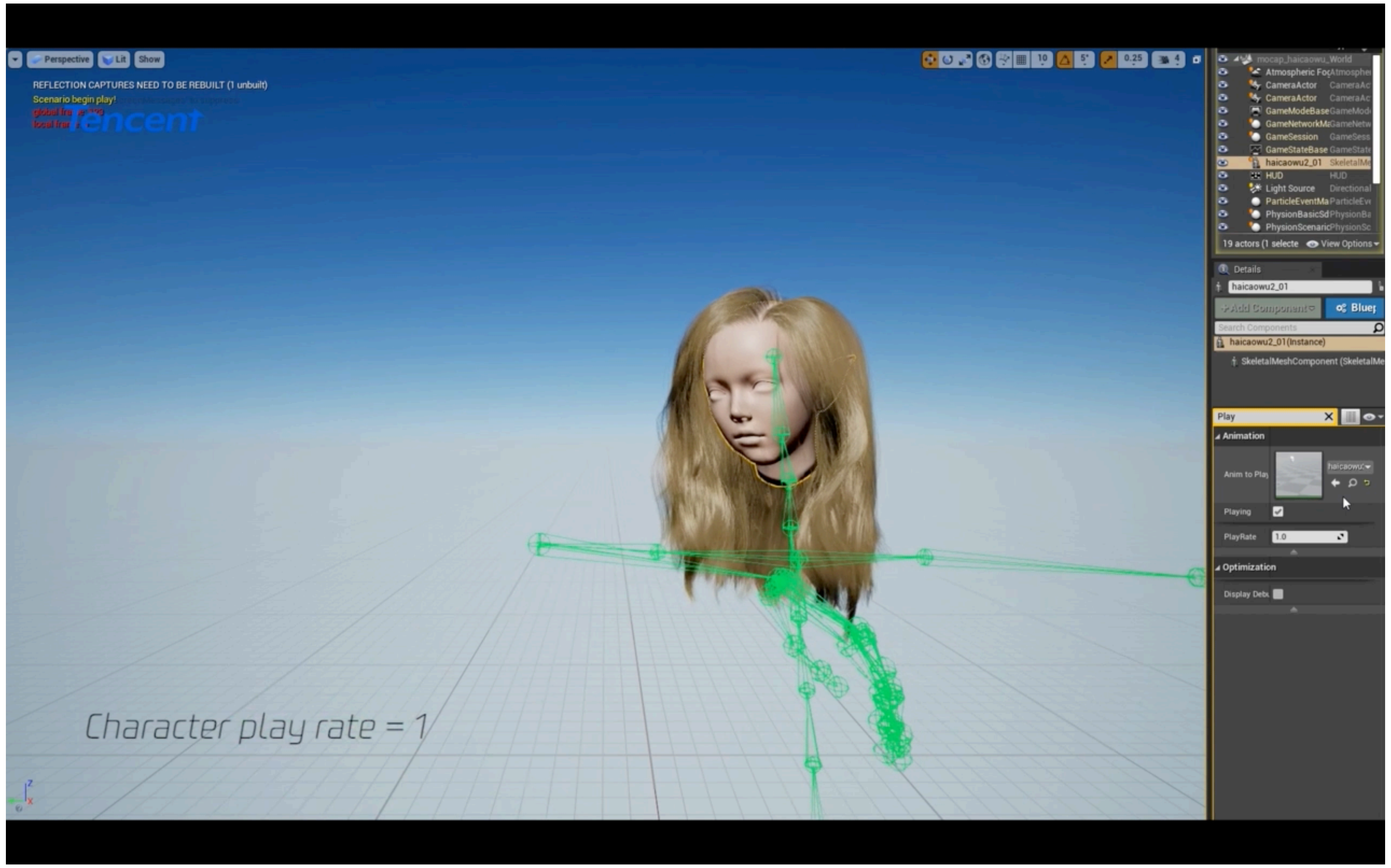


Inelastic Thickness (Contact Gap)

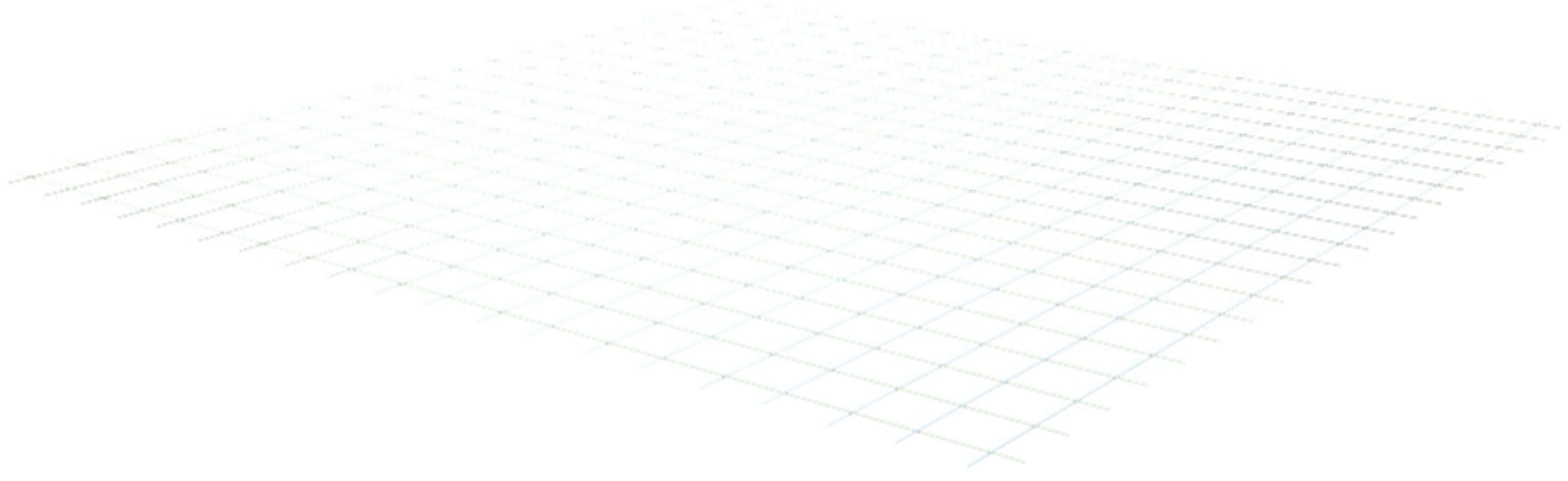


Using prism elements and reduced integration [Chen et al. 2023]

Recap: Codimensional Solids — Rods and Particles



Hair simulation [Huang et al. 2023] based on Discrete Elastic Rod and MPM



Coupling codimension-0,1,2,3 solids using IPC

Today: Fluid Simulation

Fluid as a Special Kind of Solid

- Fluid: as a special kind of solid whose strain energy only penalizes volume change
    - i.e. no resistance to volume-preserving shearing, nor rotation
    - Dissipative effects can be modeled via viscosity

$$x^{n+1} = arg\min_x \frac{1}{2}\|x - \tilde{x}^n\| + h^2 \sum P(x)$$

e.g. $P_{fluid}(x) = \sum_e V_e^0 \frac{\kappa}{2}(\det(\mathbf{F}_e(x)) - 1)^2$

Frequent and large topology changes -> mesh quality gets really bad!

Frequent remeshing is not practical!

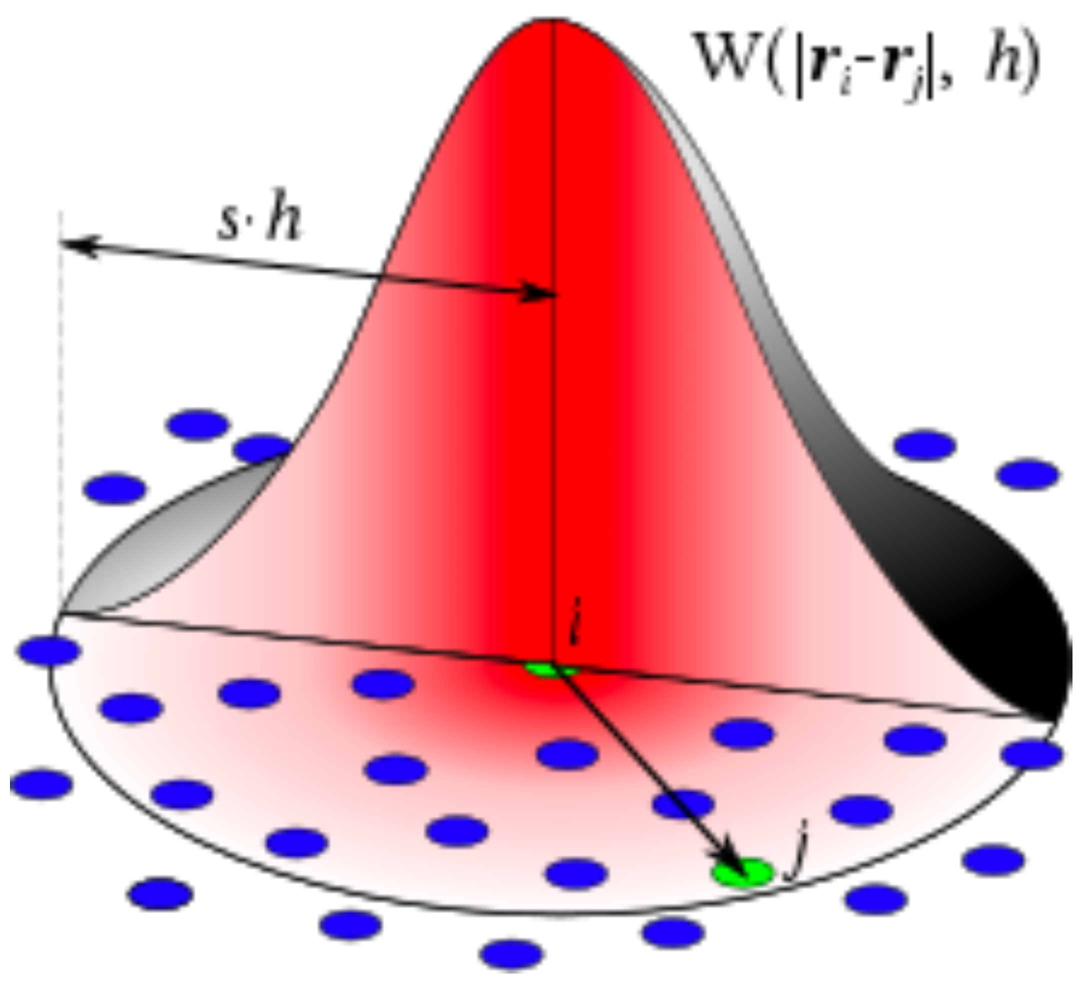
Simulating Fluids in Eulerian View using Particles

Use particles to track/represent fluid regions

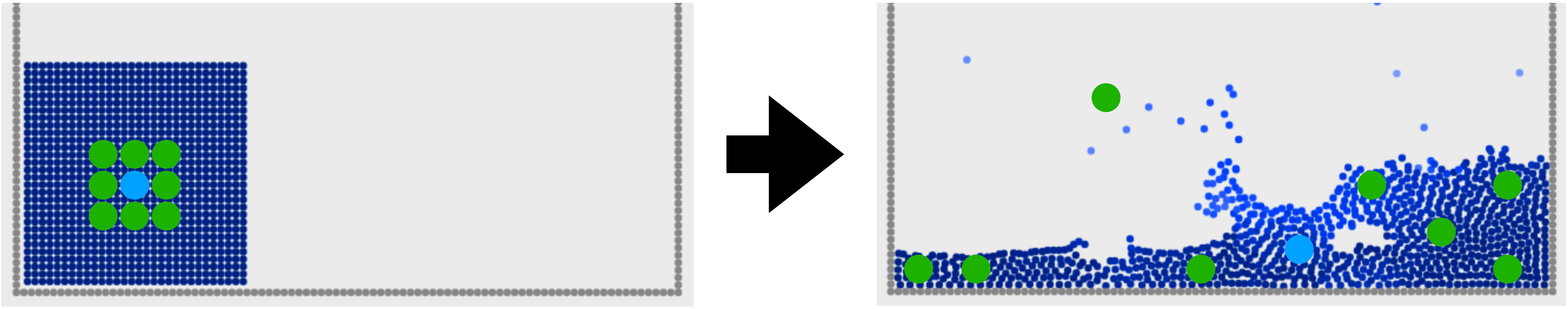
(The particles are macroscopic markers, not molecules!)

Use shape functions directly defined in space (not on meshes)

Topology change gets easy!



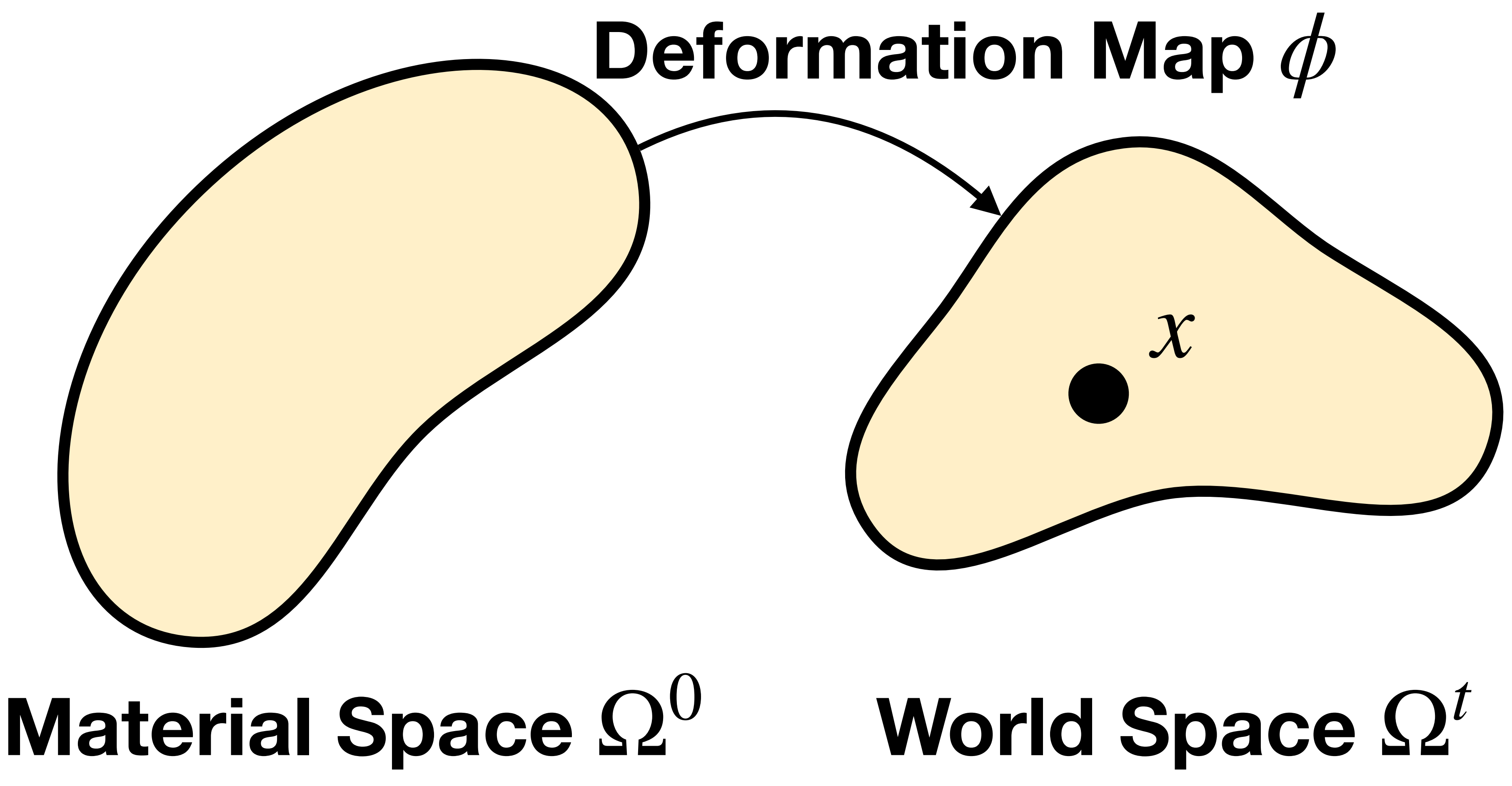
Material-space shape functions can barely work:



Use world-space shape functions! (Eulerian view)*

*Using world-space shape functions in Eulerian simulation, the time integration is subject to CFL conditions.

Lagrangian v.s Eulerian View



$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \phi(\mathbf{X}, t)$$

$$\mathbf{X} = \phi^{-1}(\mathbf{x}, t)$$

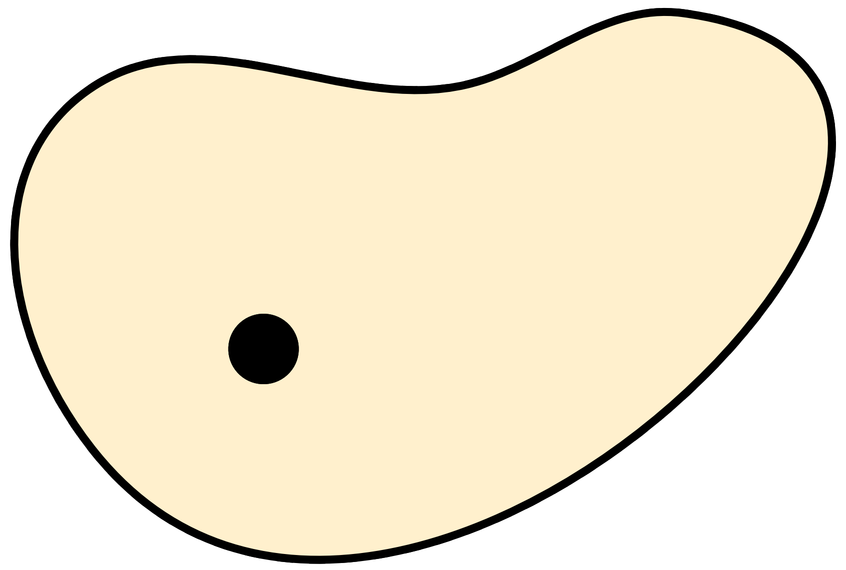
$$\mathbf{Q}(\mathbf{X}, t) = \mathbf{Q}(\phi^{-1}(\mathbf{x}, t), t) \equiv \mathbf{q}(\mathbf{x}, t)$$

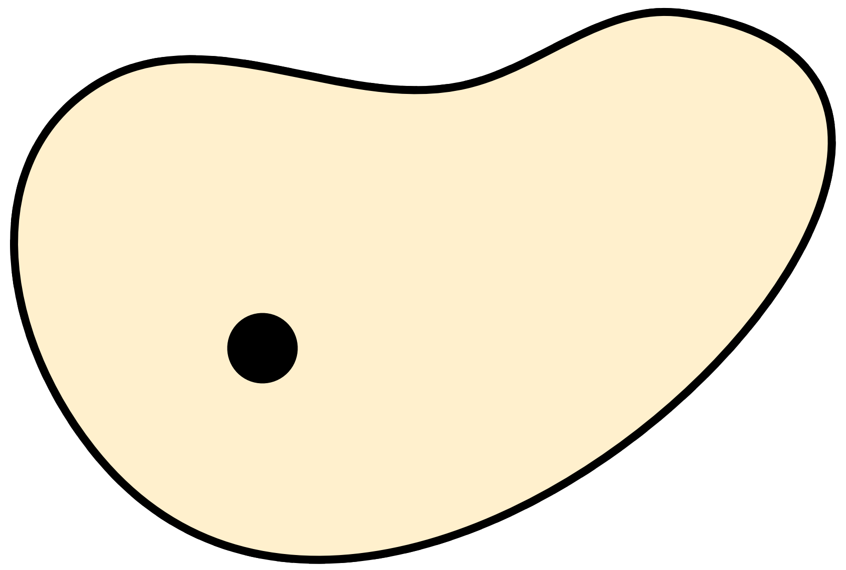
— Push forward

Pull back:

$$\mathbf{q}(\mathbf{x}, t) = \mathbf{q}(\phi(\mathbf{X}, t), t) \equiv \mathbf{Q}(\mathbf{X}, t)$$

Lagrangian view:
Quantity measured
at a point on the solid





Eulerian view:
Quantity measured
at a point in space

Lagrangian v.s Eulerian View

The Material Derivative of Eulerian Quantities

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \phi(\mathbf{X}, t) \qquad \mathbf{X} = \phi^{-1}(\mathbf{x}, t)$$

Push forward: $\mathbf{Q}(\mathbf{X}, t) = \mathbf{Q}(\phi^{-1}(\mathbf{x}, t), t) \equiv \mathbf{q}(\mathbf{x}, t)$ **Pull back:** $\mathbf{q}(\mathbf{x}, t) = \mathbf{q}(\phi(\mathbf{X}, t), t) \equiv \mathbf{Q}(\mathbf{X}, t)$

$$\mathbf{V}(\mathbf{X}, t) = \frac{\partial \phi}{\partial t}(\mathbf{X}, t)$$

$$\mathbf{A}(\mathbf{X}, t) = \frac{\partial^2 \phi}{\partial t^2}(\mathbf{X}, t) = \frac{\partial \mathbf{V}}{\partial t}(\mathbf{X}, t).$$

$$\boldsymbol{v}(\boldsymbol{x}, t) = \mathbf{V}(\phi^{-1}(\boldsymbol{x}, t), t),$$

$$\boldsymbol{a}(\boldsymbol{x}, t) = \mathbf{A}(\phi^{-1}(\boldsymbol{x}, t), t).$$

$$\mathbf{V}(\mathbf{X}, t) = \boldsymbol{v}(\phi(\mathbf{X}, t), t),$$

$$\mathbf{A}(\mathbf{X}, t) = \boldsymbol{a}(\phi(\mathbf{X}, t), t).$$

$$\mathbf{A}(\mathbf{X}, t) = \frac{\partial}{\partial t}\mathbf{V}(\mathbf{X}, t) = \frac{\partial \boldsymbol{v}}{\partial t}(\phi(\mathbf{X}, t), t) + \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}(\phi(\mathbf{X}, t), t)\frac{\partial \phi}{\partial t}(\mathbf{X}, t).$$

$$A_i(\mathbf{X}, t) = \frac{\partial}{\partial t}V_i(\mathbf{X}, t) = \frac{\partial v_i}{\partial t}(\phi(\mathbf{X}, t), t) + \frac{\partial v_i}{\partial x_j}(\phi(\mathbf{X}, t), t)\frac{\partial \phi_j}{\partial t}(\mathbf{X}, t).$$

$$a_i(\boldsymbol{x}, t) = A_i(\phi^{-1}(\boldsymbol{x}, t), t) = \frac{\partial v_i}{\partial t}(\boldsymbol{x}, t) + \frac{\partial v_i}{\partial x_j}(\boldsymbol{x}, t)v_j(\boldsymbol{x}, t)$$

$$a_i(\boldsymbol{x}, t) \neq \frac{\partial v_i}{\partial t}(\boldsymbol{x}, t).$$

$$\mathbf{a}(\mathbf{x}, t) = \frac{D\mathbf{v}(\mathbf{x}, t)}{Dt}$$

(Material Derivative)

Conservation of Momentum

Lagrangian View: $$R(\mathbf{X},0)\frac{\partial \mathbf{V}}{\partial t}(\mathbf{X},t) = \nabla^{\mathbf{X}} \cdot \mathbf{P}(\mathbf{X},t) + R(\mathbf{X},0)\boldsymbol{g}$$

Newton's 2nd Law on B_ϵ^0:

$$\int_{B_\epsilon^0} R(\mathbf{X},0)\frac{\partial \mathbf{V}}{\partial t}(\mathbf{X},t)d\mathbf{X} = \int_{\partial B_\epsilon^0} \mathbf{P}(\mathbf{X},t)\mathbf{N}(\mathbf{X})ds(\mathbf{X}) + \int_{B_\epsilon^0} R(\mathbf{X},0)\mathbf{A}^{\text{ext}}(\mathbf{X},t)d\mathbf{X}, \quad \forall\ B_\epsilon^0 \subset \Omega^0 \text{ and } t \geq 0.$$

Applying Divergence Theorem:

$$\int_{B_\epsilon^0} R(\mathbf{X},0)\frac{\partial \mathbf{V}}{\partial t}(\mathbf{X},t)d\mathbf{X} = \int_{B_\epsilon^0} \nabla^{\mathbf{X}} \cdot \mathbf{P}(\mathbf{X},t)d\mathbf{X} + \int_{B_\epsilon^0} R(\mathbf{X},0)\mathbf{A}^{\text{ext}}(\mathbf{X},t)d\mathbf{X}, \quad \forall\ B_\epsilon^0 \subset \Omega^0 \text{ and } t \geq 0$$

Push forward and extract the integrand

Eulerian View: $$\rho(\mathbf{x},t)\frac{D\mathbf{v}}{Dt}(\mathbf{x},t) = \nabla^{\mathbf{x}} \cdot \sigma(\mathbf{x},t) + \rho(\mathbf{x},t)\mathbf{g}$$

Cauchy stress: $\sigma = \frac{1}{J}\mathbf{P}\mathbf{F}^T$ $\qquad Q_{i,j} = \frac{\partial Q_i}{\partial X_j} = \frac{\partial q_i}{\partial x_k}\frac{\partial x_k}{\partial X_j} = q_{i,k}F_{kj}.$ $\qquad d\mathbf{X} = \frac{1}{J}d\mathbf{x}$

Inviscid Navier-Stoke's Equation

How is Cauchy stress modeled?

Consider a fluid constitutive model, e.g. $\Psi_{fluid}(\mathbf{F}) = \frac{\kappa}{2}(\det(\mathbf{F}) - 1)^2$

$$\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}} = \frac{\partial \Psi}{\partial J}\frac{\partial J}{\mathbf{F}} = \kappa(J-1)J\mathbf{F}^{-T}$$

$$\sigma = \frac{1}{J}\mathbf{P}\mathbf{F}^T = \kappa(J-1)\mathbf{I} = -pI$$

$p = -\frac{\partial \Psi}{\partial J}$ **is called Pressure**

Momentum Conservation (Eulerian View): $\rho(\mathbf{x}, t)\frac{D\mathbf{v}}{Dt}(\mathbf{x}, t) = \nabla^{\mathbf{x}} \cdot \sigma(\mathbf{x}, t) + \rho(\mathbf{x}, t)\mathbf{g}$

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla_{\mathbf{x}} p + \rho\mathbf{g}$$

— Euler Equation

— Navier Stoke's Equation (Inviscid) $\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho}\nabla p = \vec{g}$

Incompressibility

Consider a fluid constitutive model, e.g. $\Psi_{fluid}(\mathbf{F}) = \frac{\kappa}{2}(\det(\mathbf{F}) - 1)^2$

κ is called bulk modulus, similar to Young's modulus for solids

How large should κ be?

Water	2.2 GPa

(en.wikipedia.org/wiki/Bulk_modulus)

Very stiff!

What if we model volume-preserving fluids using equality constraints?

Applying divergence theorem:

$$\frac{d}{dt}V(B_\epsilon^t) = \int_{\partial B_\epsilon^t} \mathbf{v} \cdot \mathbf{n} d\mathbf{x} = 0 \quad \forall B_\epsilon^t \in \Omega^t \Leftrightarrow \int_{B_\epsilon^t} \nabla \cdot \mathbf{v} d\mathbf{x} = 0 \quad \forall B_\epsilon^t \in \Omega^t \Leftrightarrow \nabla \cdot \mathbf{v} = 0$$

Incompressible Navier-Stoke's Equation (Inviscid):

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

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

Lagrange multiplier term

Solving the KKT is still hard,
But magic tricks can be applied!

Viscosity

Can be viewed as fluid friction — penalizing stretching and shearing motion

Strain rate tensor: $\mathbf{D} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$

Newtonian fluids: $\sigma_{viscosity} = 2\mu\mathbf{D} + \lambda \text{tr}(\mathbf{D})\mathbf{I}$

$$\rho(\mathbf{x}, t)\frac{D\mathbf{v}}{Dt}(\mathbf{x}, t) = \nabla^{\mathbf{x}} \cdot \sigma(\mathbf{x}, t) + \rho(\mathbf{x}, t)\mathbf{g}$$

Newtonian fluids: $\sigma = -p\mathbf{I} + \mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T) + \lambda \text{tr}(\mathbf{D})\mathbf{I}$

$\text{tr}(\mathbf{D}) = \nabla \cdot \mathbf{v} = 0$ for incompressible fluids

$$\nabla \cdot (\mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T)) = \mu(\nabla \cdot \nabla \mathbf{v} + \boxed{\nabla \cdot (\nabla \mathbf{v})^T})$$

$$\nabla \cdot \begin{bmatrix} \frac{\partial \mathbf{v}}{\partial x_1}^T \\ \frac{\partial \mathbf{v}}{\partial x_2}^T \\ \frac{\partial \mathbf{v}}{\partial x_3}^T \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} \sum_i \frac{\partial v_i}{\partial x_i} \\ \frac{\partial}{\partial x_2} \sum_i \frac{\partial v_i}{\partial x_i} \\ \frac{\partial}{\partial x_3} \sum_i \frac{\partial v_i}{\partial x_i} \end{bmatrix} = \nabla(\nabla \cdot \mathbf{v}) = 0$$

for incompressible fluids

Incompressible Navier-Stoke's Equation

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

Time Splitting

Consider a generic ODE:

$$\frac{dq}{dt} = f(q) + g(q).$$

Explicit time integration with splitting:

$$\begin{aligned} \tilde{q} &= q^n + \Delta t f(q^n), \\ q^{n+1} &= \tilde{q} + \Delta t g(\tilde{q}). \end{aligned}$$

$$\begin{aligned} q^{n+1} &= (q^n + \Delta t f(q^n)) + \Delta t g(q^n + \Delta t f(q^n)) \\ &= q^n + \Delta t f(q^n) + \Delta t\,(g(q^n) + O(\Delta t)) \\ &= q^n + \Delta t(f(q^n) + g(q^n)) + O(\Delta t^2) \\ &= \boxed{q^n + \frac{dq}{dt}\Delta t} + O(\Delta t^2). \end{aligned}$$

Explicit Euler

Incompressible Navier-Stoke's Equation

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

For each time step n:

$u^a \leftarrow$ **Solve** $\frac{\partial u}{\partial t} + u \cdot \nabla u = 0$ **(advection)**

$u^b \leftarrow$ **Solve** $\frac{\partial u}{\partial t} = g$ **(apply external force)**

$u^c \leftarrow$ **Solve** $\frac{\partial u}{\partial t} = \nu \nabla \cdot \nabla u$ **(diffusion)**

$u^{n+1} \leftarrow$ **Solve** $\nabla \cdot u = 0$ **(pressure projection)**

With constraint view, this step is stable!

The Smoothed Particle Hydrodynamics (SPH) Method

A Brief Introduction

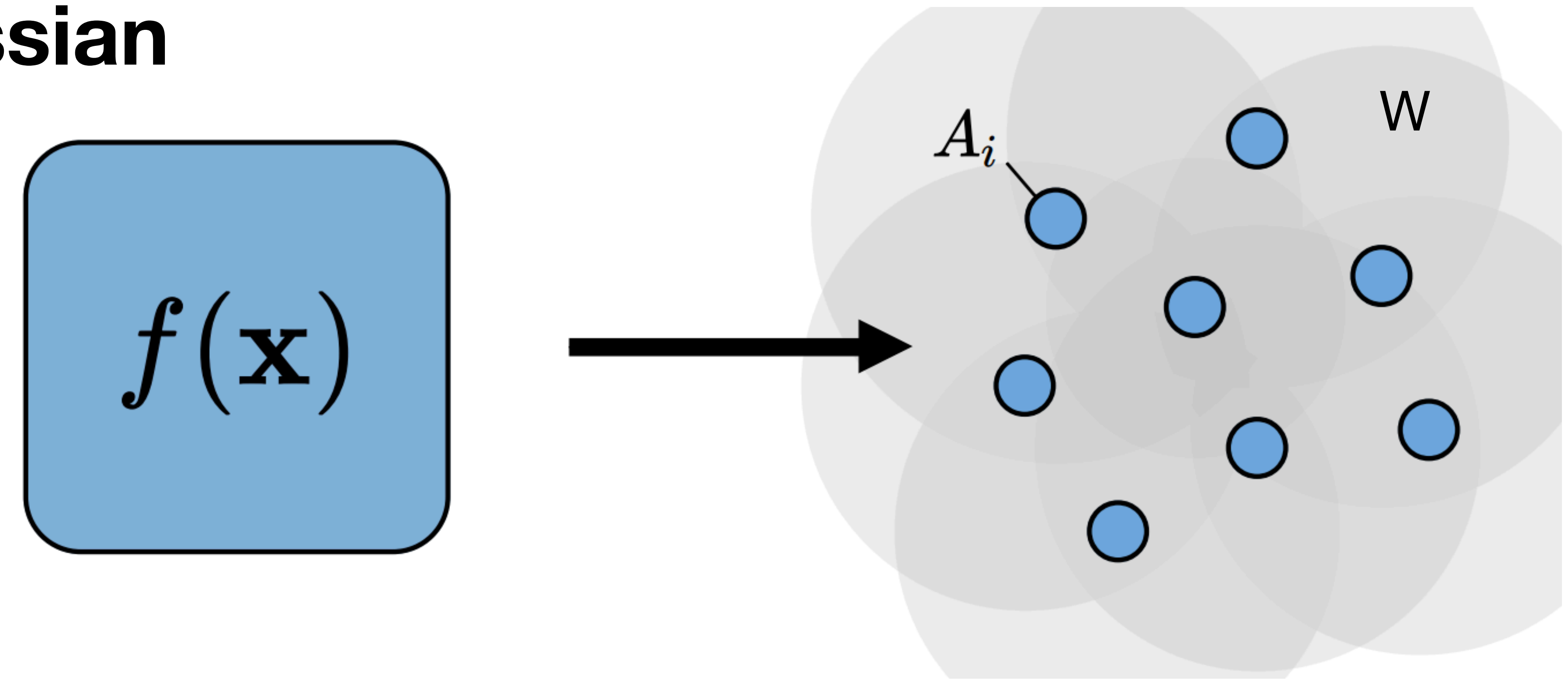
Given a field A and a smoothing kernel function W, e.g. Gaussian

A smoother version of A as an approximation of it is

$$A(\mathbf{x}) \approx (A * W)(\mathbf{x}) = \int A(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)\,dv'$$

Favored properties of W:

$\int_{\mathbb{R}^d} W(\mathbf{r}', h)\,dv' = 1$	(normalization condition)
$\lim_{h' \to 0} W(\mathbf{r}, h') = \delta(\mathbf{r})$	(Dirac-δ condition)
$W(\mathbf{r}, h) \geq 0$	(positivity condition)
$W(\mathbf{r}, h) = W(-\mathbf{r}, h)$	(symmetry condition)
$W(\mathbf{r}, h) = 0$ for $\lVert\mathbf{r}\rVert \geq \hbar$,	(compact support condition)



Discretization using particles:

$$(A * W)(\mathbf{x}_i) = \int \frac{A(\mathbf{x}')}{\rho(\mathbf{x}')} W(\mathbf{x}-\mathbf{x}', h) \underbrace{\rho(\mathbf{x}')\,dv'}_{dm'}$$

$$\approx \sum_{j \in \mathcal{F}} A_j \frac{m_j}{\rho_j} W(\mathbf{x}_i - \mathbf{x}_j, h) =: \langle A(\mathbf{x}_i) \rangle$$

The kernel needs to involve a large number of neighbors for accurate estimation!

A Brief Introduction to SPH

To solve the incompressible Navier-Stoke's Equation

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

Just need to approximate the differential operators,
And relate velocity to pressure via constitutive models

Direct discretization are not accurate and can lead to instability:

$$\nabla \mathbf{A}_i \approx \sum_j \frac{m_j}{\rho_j} \mathbf{A}_j \otimes \nabla W_{ij}$$
$$\nabla \cdot \mathbf{A}_i \approx \sum_j \frac{m_j}{\rho_j} \mathbf{A}_j \cdot \nabla W_{ij}$$

Difference and symmetric formula are often used:

$$\nabla A_i \approx \langle \nabla A_i \rangle - A_i \langle \nabla 1 \rangle = \sum_j \frac{m_j}{\rho_j} (A_j - A_i) \nabla_i W_{ij} .$$

$$\nabla A_i \approx \rho_i \left(\frac{A_i}{\rho_i^2} \langle \nabla \rho \rangle + \langle \nabla \left(\frac{A_i}{\rho_i} \right) \rangle \right) = \rho_i \sum_j m_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2} \right) \nabla_i W_{ij} .$$

A Brief Introduction to SPH

To solve the incompressible Navier-Stoke's Equation

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

Just need to approximate the differential operators,
And relate velocity to pressure via constitutive models

Relate velocity to pressure via state equation:

$$p = -\frac{\partial \Psi}{\partial J}$$ **e.g.** $p = -\kappa(J-1)$ **for** $\Psi = \frac{\kappa}{2}(J-1)^2$

— Weakly-Compressible SPH, or WCSPH

Handling pressure term by solving $\nabla \cdot u = 0$**:**

- **Implicit Imcompressible SPH (IISPH)**
- **Divergence-Free SPH (DFSPH)**

A Brief Introduction to SPH

To solve the incompressible Navier-Stoke's Equation

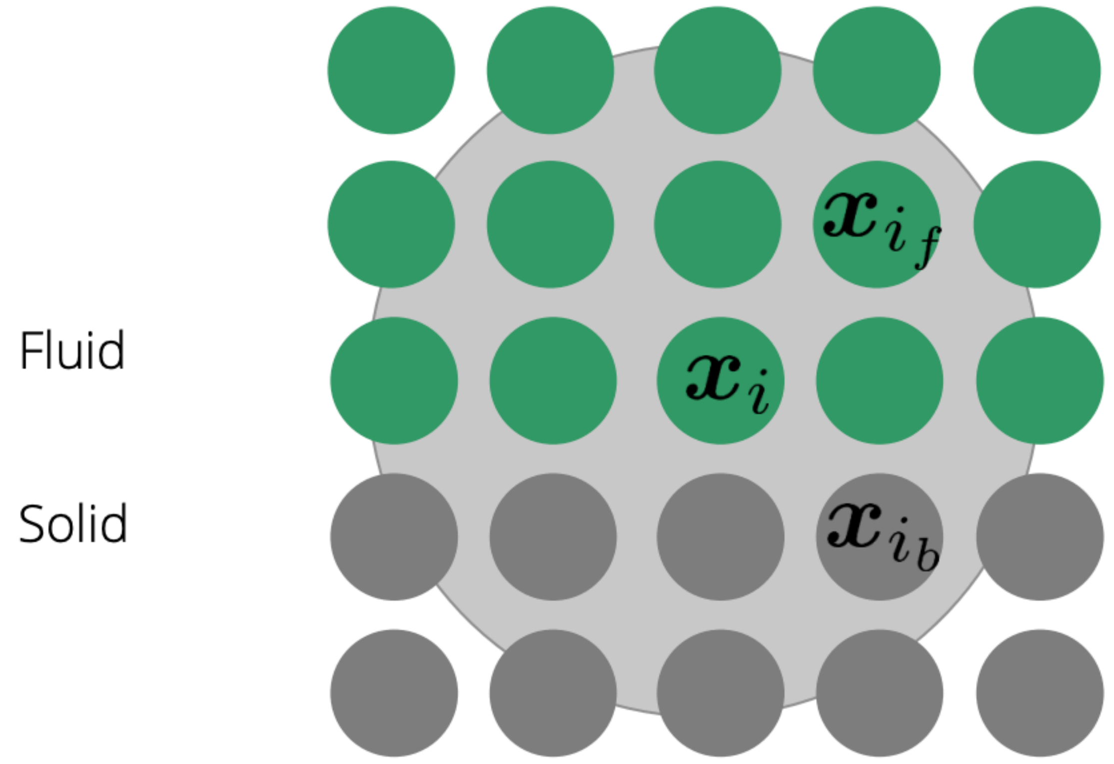
$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + \nu \nabla \cdot \nabla \vec{u},$$
$$\nabla \cdot \vec{u} = 0.$$

Just need to approximate the differential operators,
And relate velocity to pressure via constitutive models

CFL condition: $\Delta t \leq \lambda \frac{\tilde{h}}{\|\mathbf{v}^{\max}\|}$

— All particles are only allowed to move less than the particle diameter per time step for λ = 1

Use ghost particles to represent solids/air:



This also avoids density underestimation.

Demo!

interactivecomputergraphics.github.io/physics-simulation

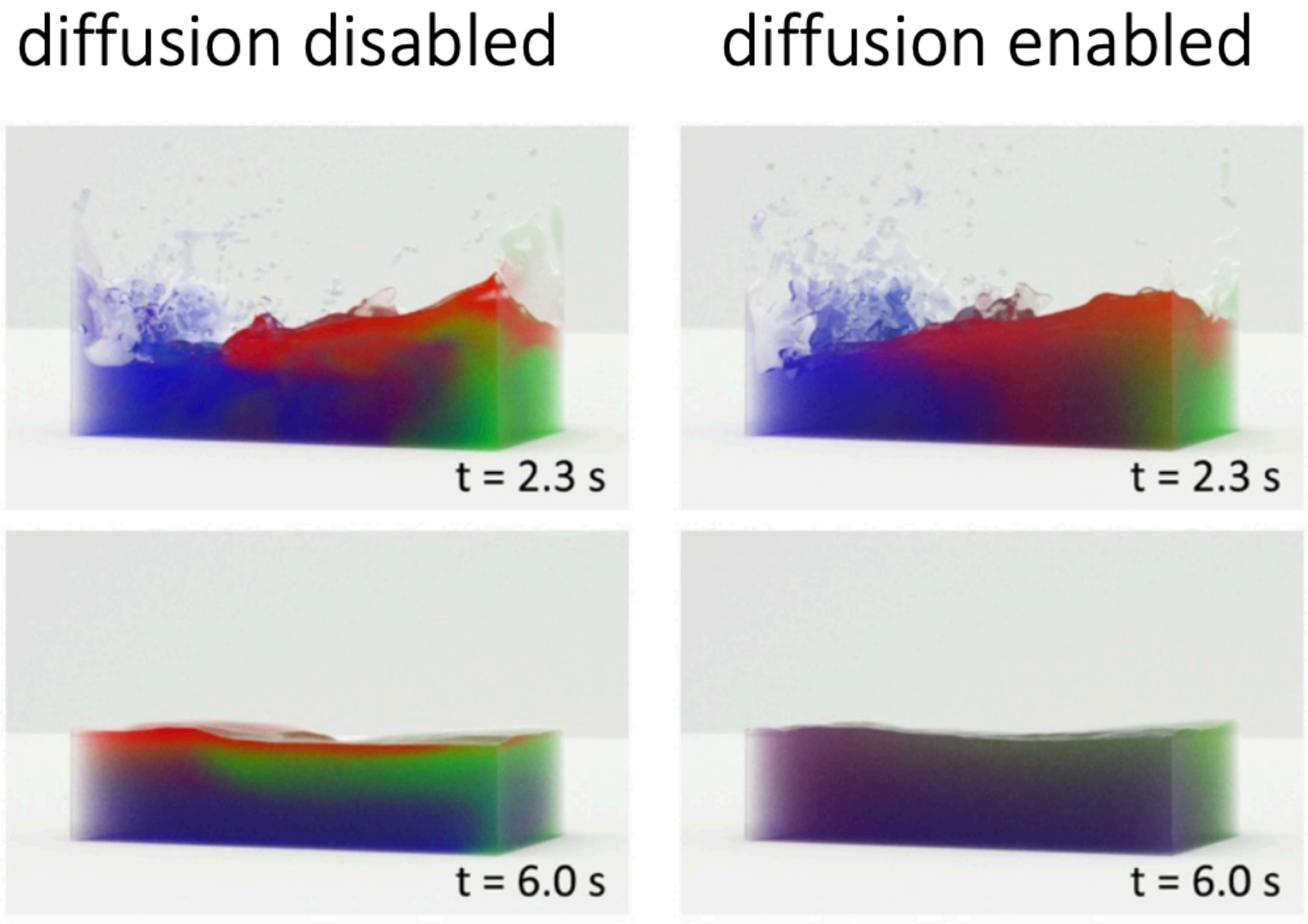
More on SPH

Optimization-based SPH [Xie et al. 2023]

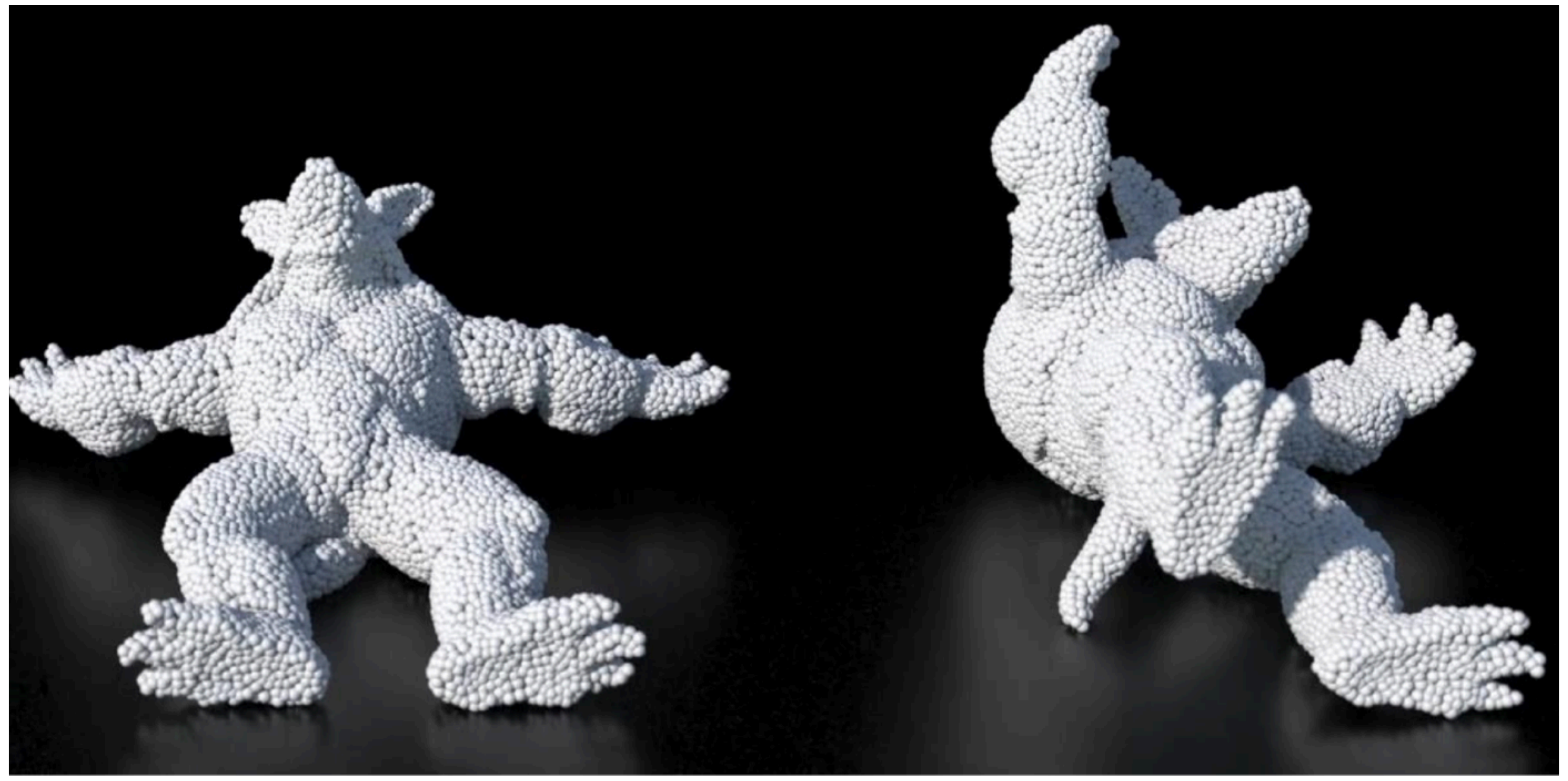
$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu_t \nabla \times \boldsymbol{\omega} + \mathbf{f}$$

$$\rho\Theta \frac{D\boldsymbol{\omega}}{Dt} = \mu_t(\nabla \times \mathbf{v} - 2\boldsymbol{\omega}) + \boldsymbol{\tau}$$

Micropolar SPH [Bender et al. 2017]
(particle with self-rotation)

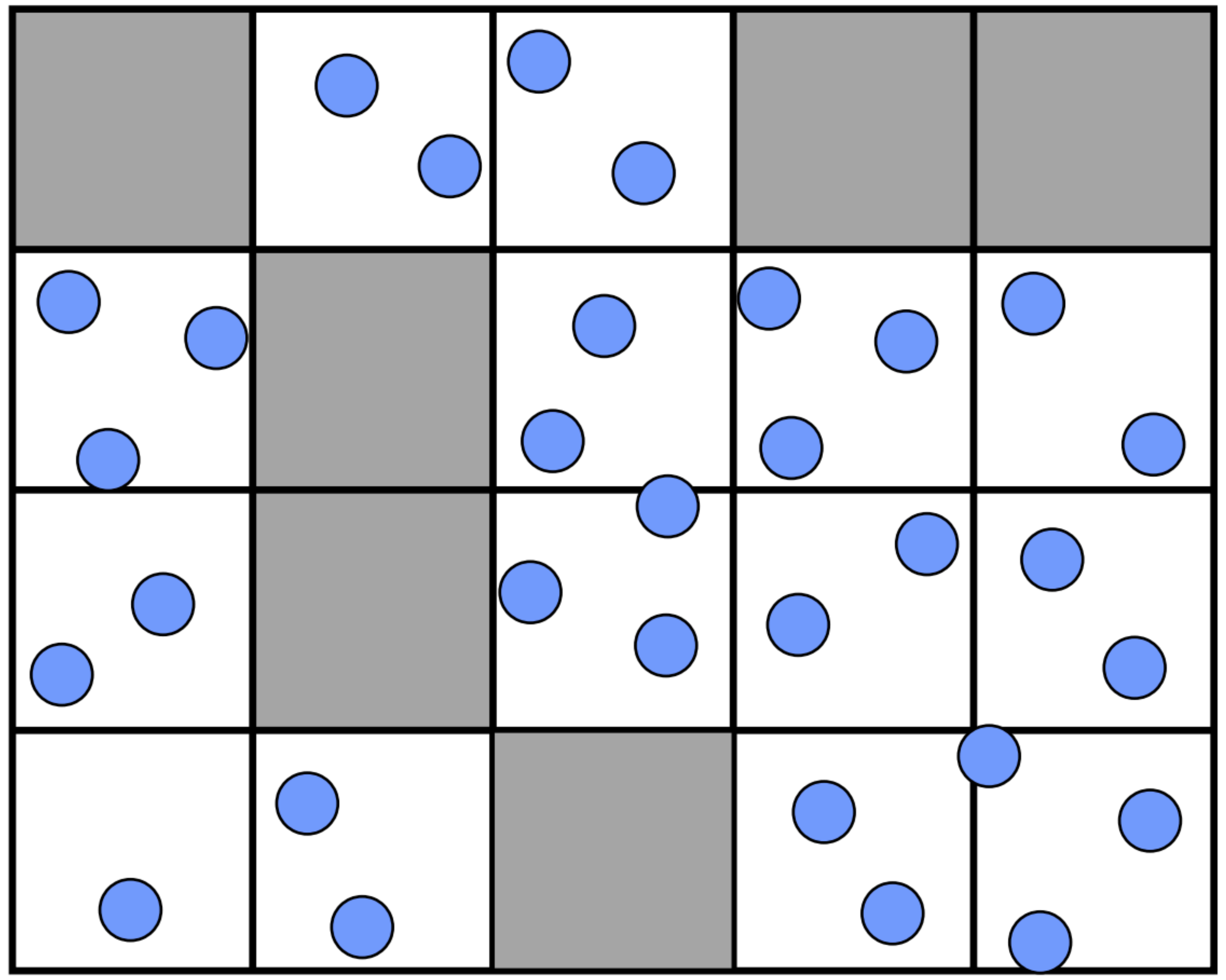


Multiphase Fluids [Ren et al. 2014]



SPH solids [Peer et al. 2018]

Next Lecture: Hybrid Lagrangian/Eulerian Methods



Next Week

- Nov 14: Paper Presentation
  - Chen et al. SIERE: A Hybrid Semi-Implicit Exponential Integrator for Efficiently Simulating Stiff Deformable Objects. ToG 2020 (Presenter: **Kevin You**)
  - Wolper et al. CD-MPM: Continuum Damage Material Point Methods for Dynamic Fracture Animation. SIGGRAPH 2018 (Presenter: **Shilin Ma**)
- Nov 16: Paper Presentation
  - Sharp et al. Data-Free Learning of Reduced-Order Kinematics. SIGGRAPH 2023 (Presenter: **Zoë Marschner**)
  - Sperl et al. Homogenized Yarn-Level Cloth. SIGGRAPH 2020 (Presenter: **Sarah Di**)

Image Sources

- http://multires.caltech.edu/pubs/ds.pdf
- https://www.youtube.com/watch?v=UDQaw4Ff3sg
- https://en.wikipedia.org/wiki/Smoothed-particle_hydrodynamics
- https://sph-tutorial.physics-simulation.org/