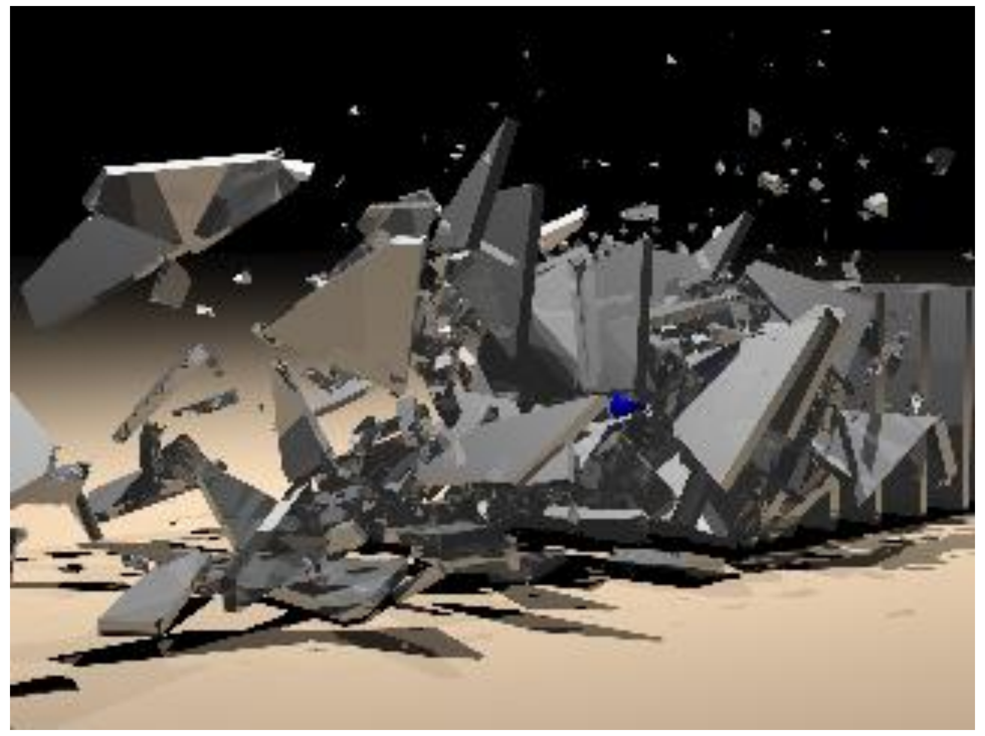
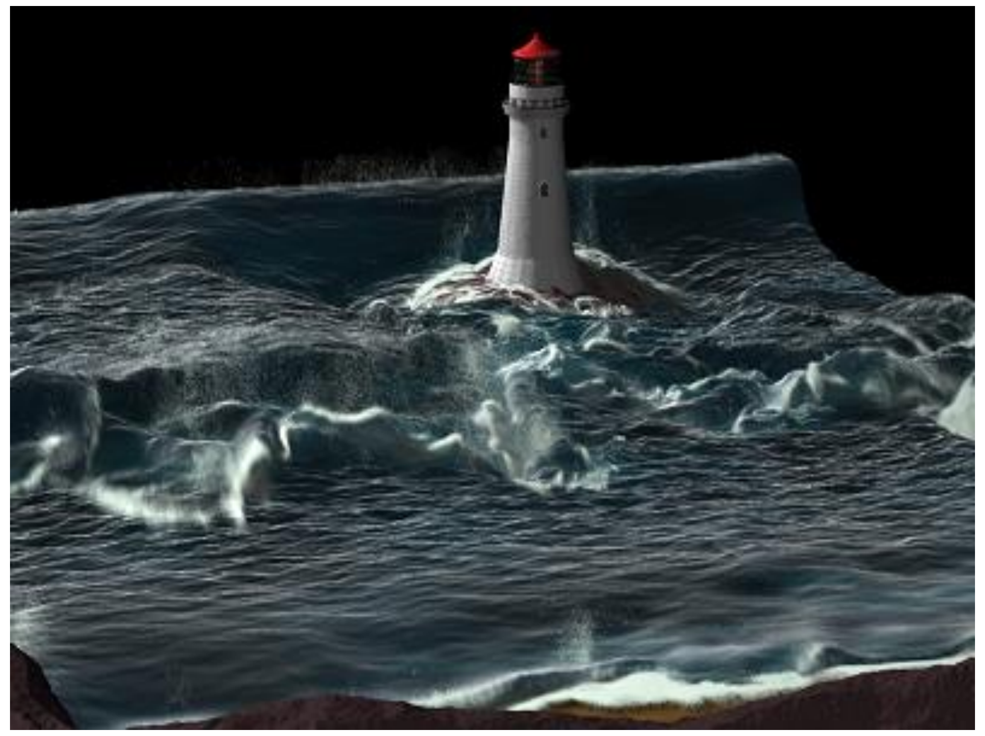
Physically-based Animation

Bo Zhu
School of Interactive Computing
Georgia Institute of Technology

Motivational Video: Pirates of the Caribbean

CG Tech behind the Maelstrom of Pirates of the Caribbean





Prof. Ronald Fedkiw's research group at Stanford developed a physics simulation engine **PhysBAM** that simulates the complex physics effects in many Hollywood movies --- including the maelstrom of Pirates of the Caribbean

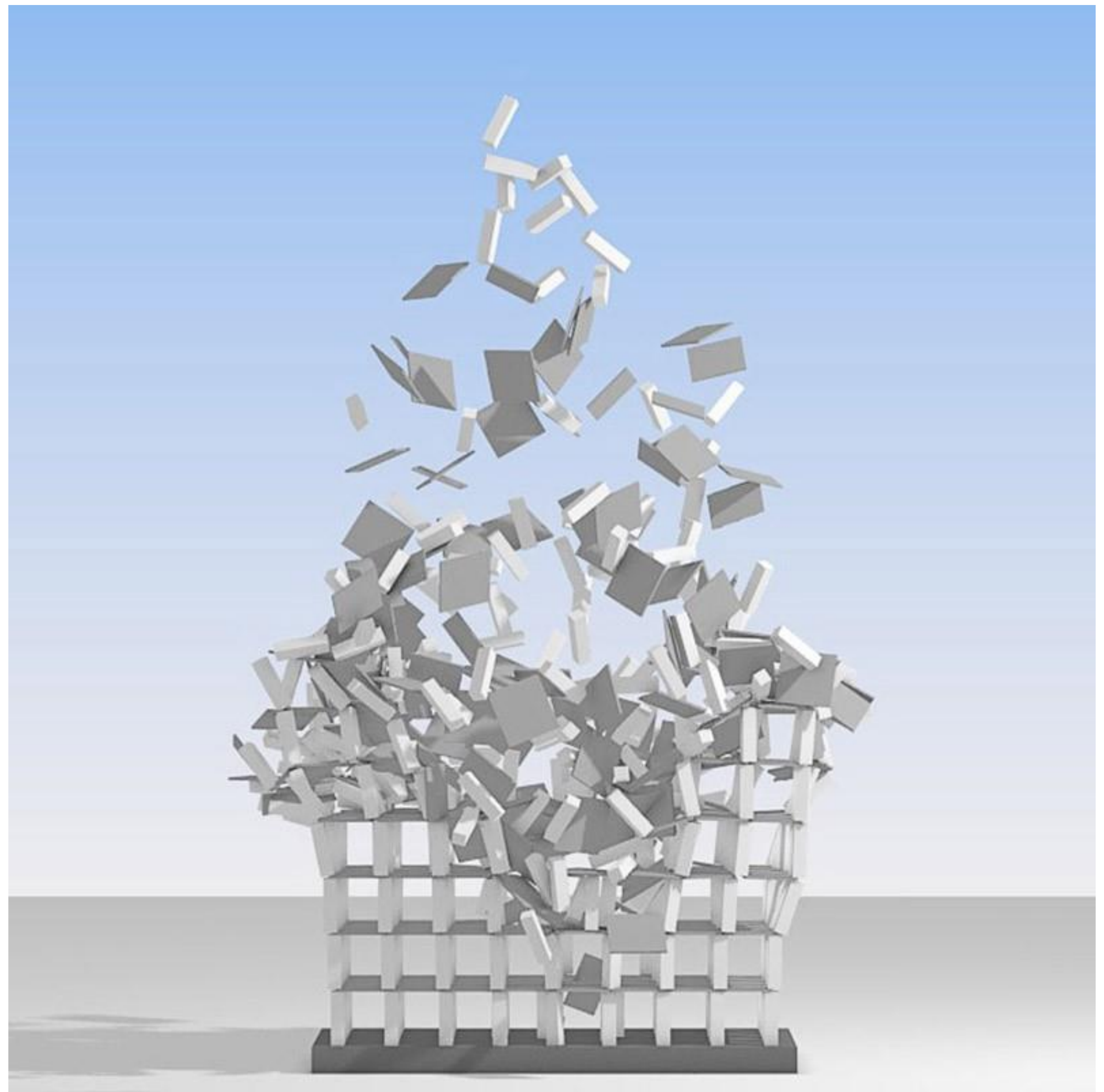
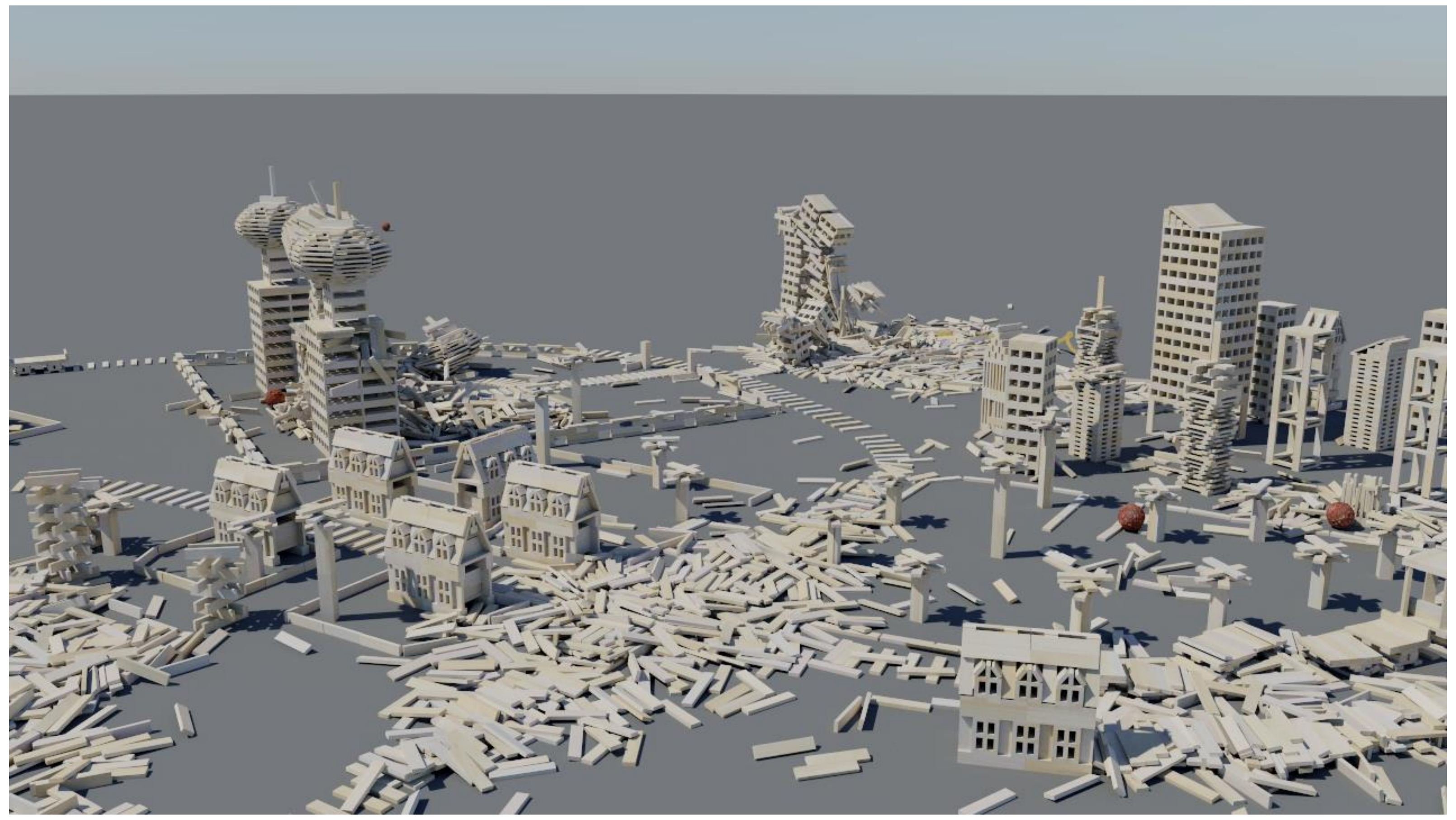
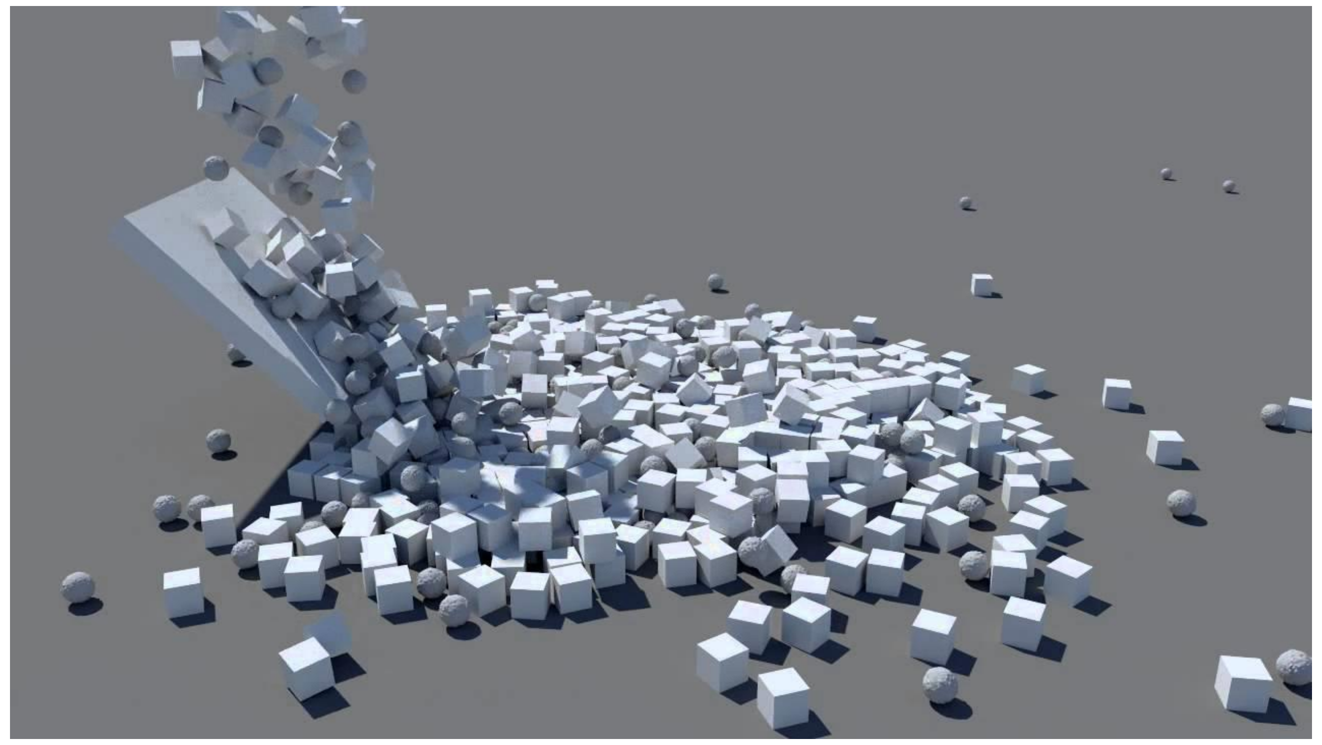
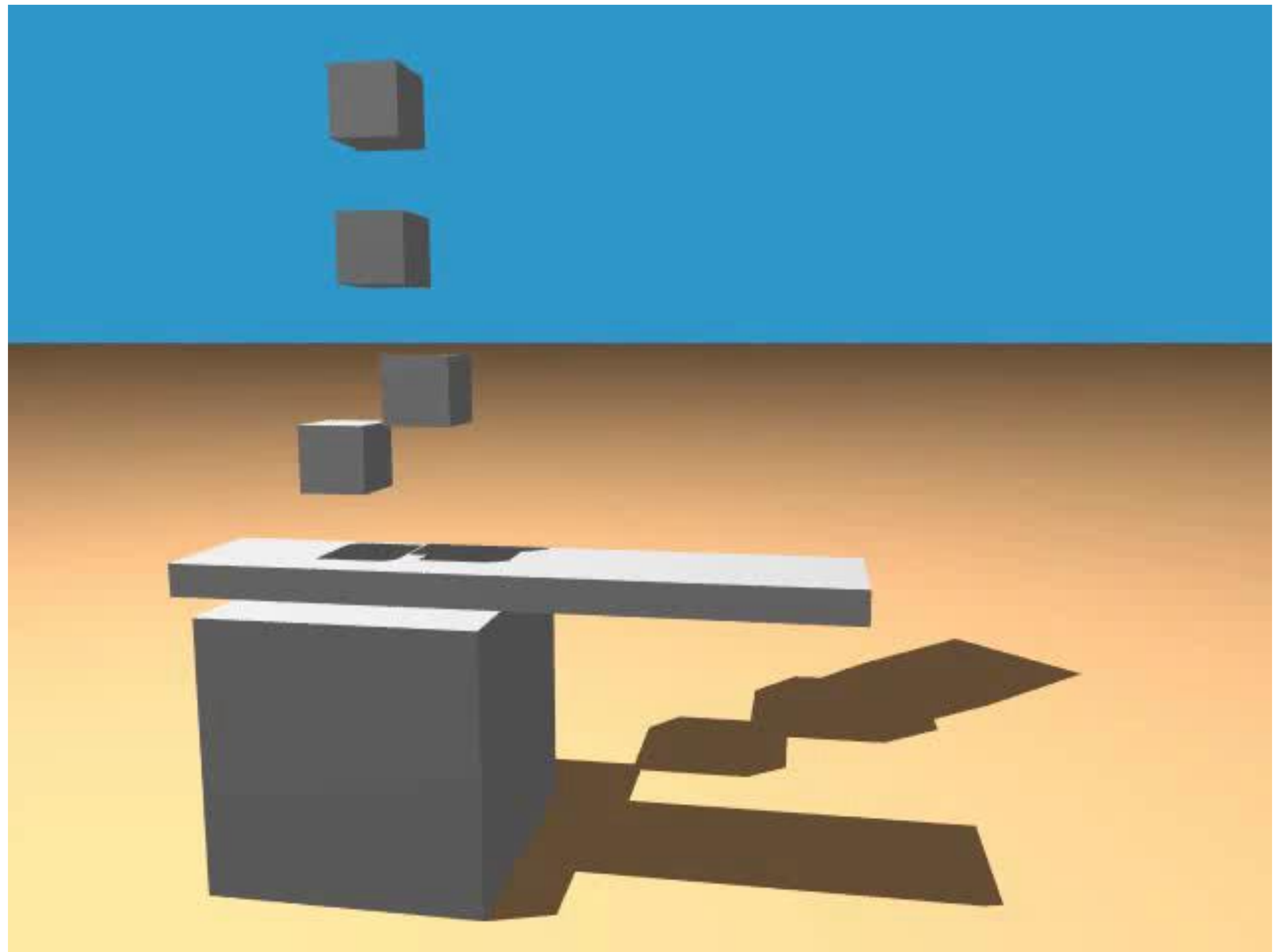
Physics in Everywhere in the World of Computer Graphics




Physical System Overview

Rigid Body

- Solid that **does not deform** its shape during simulation
- Motion can be described with translation and rotation
- Interactions among objects (collision and contact) need to be handled
- Typically many objects are simulated at the same time

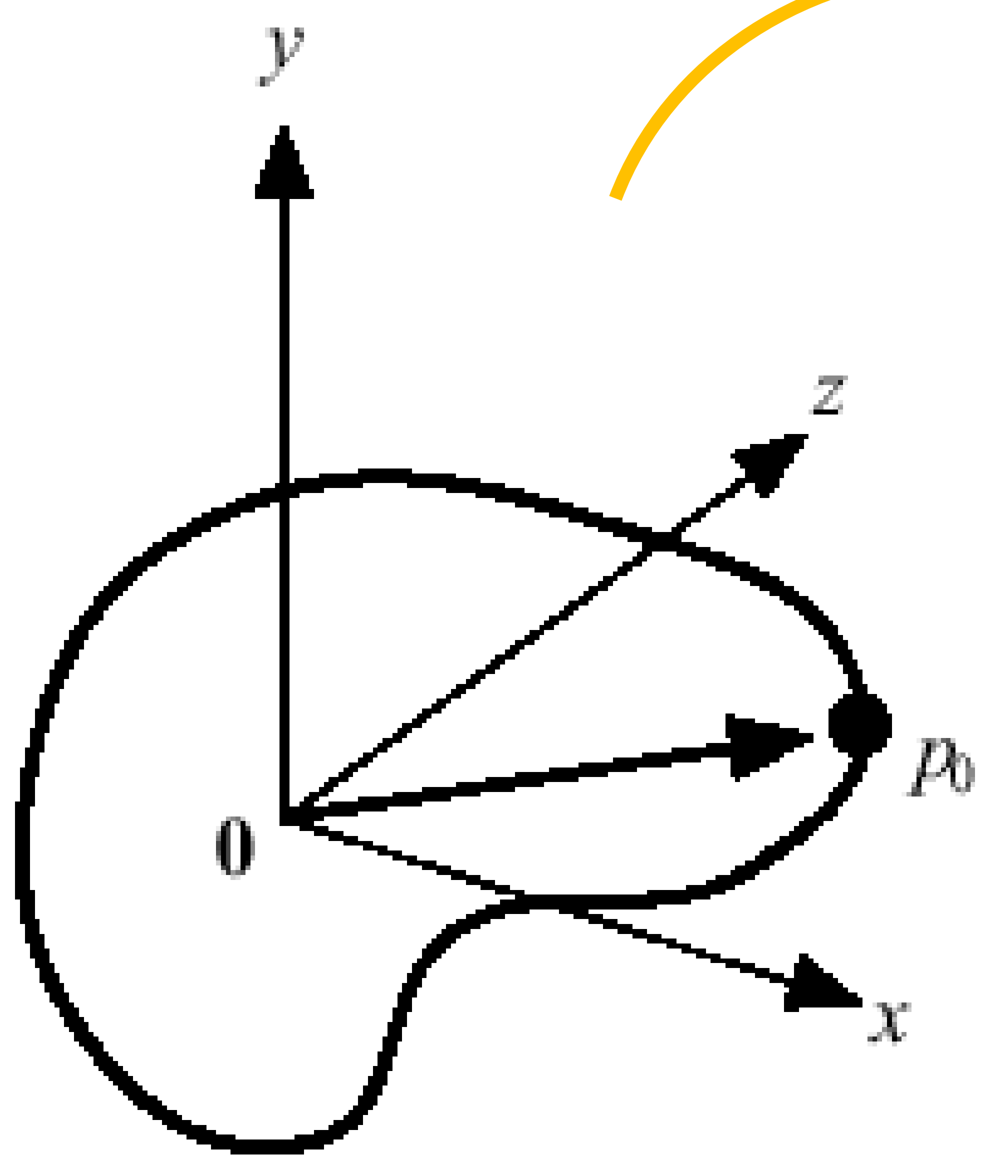
Mathematical Model of Rigid Bodies

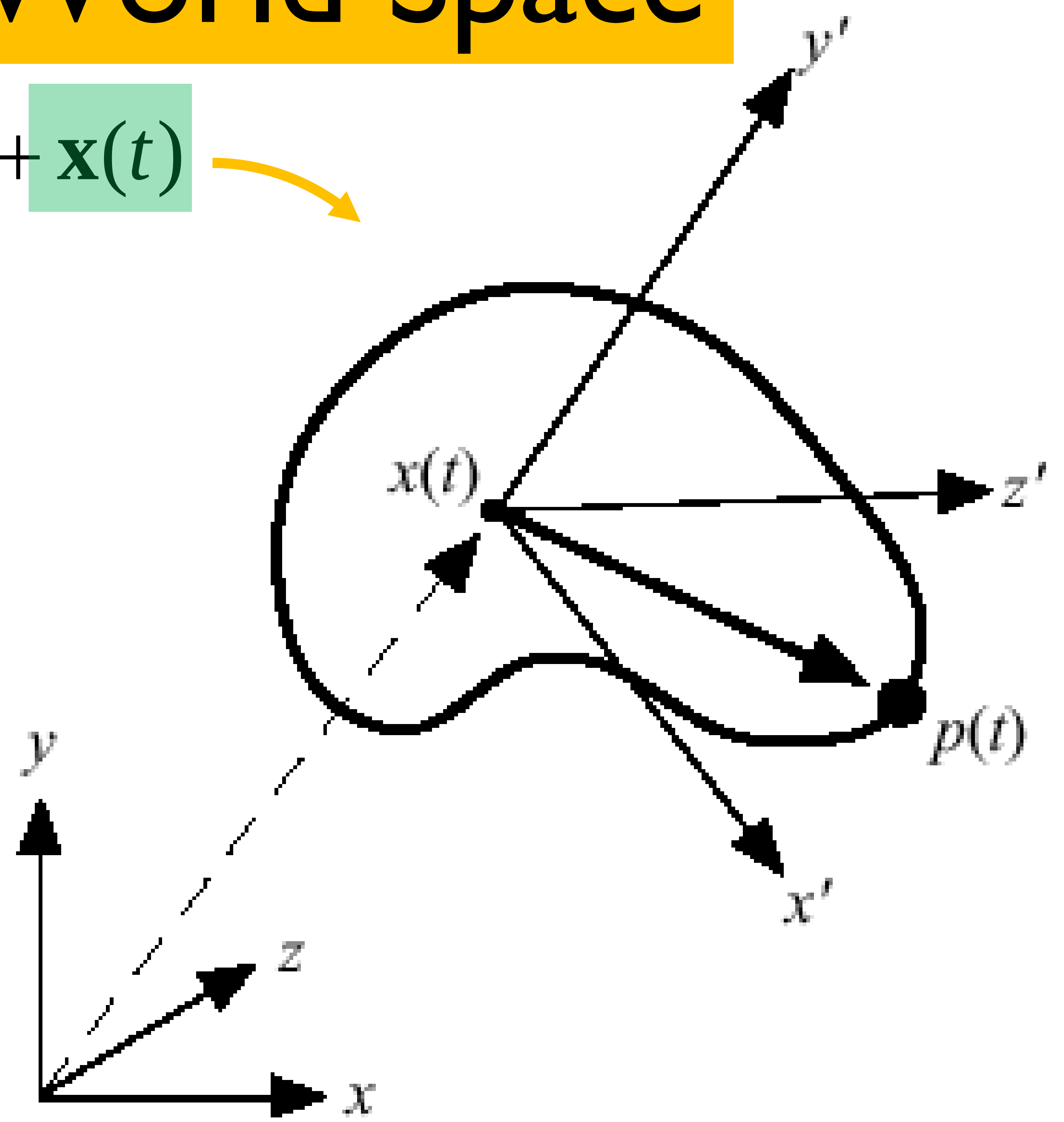
- A rigid body is defined by its **mass center (position)** and **orientation**: in each time, we calculate velocity and angular velocity to update its mass center and orientation

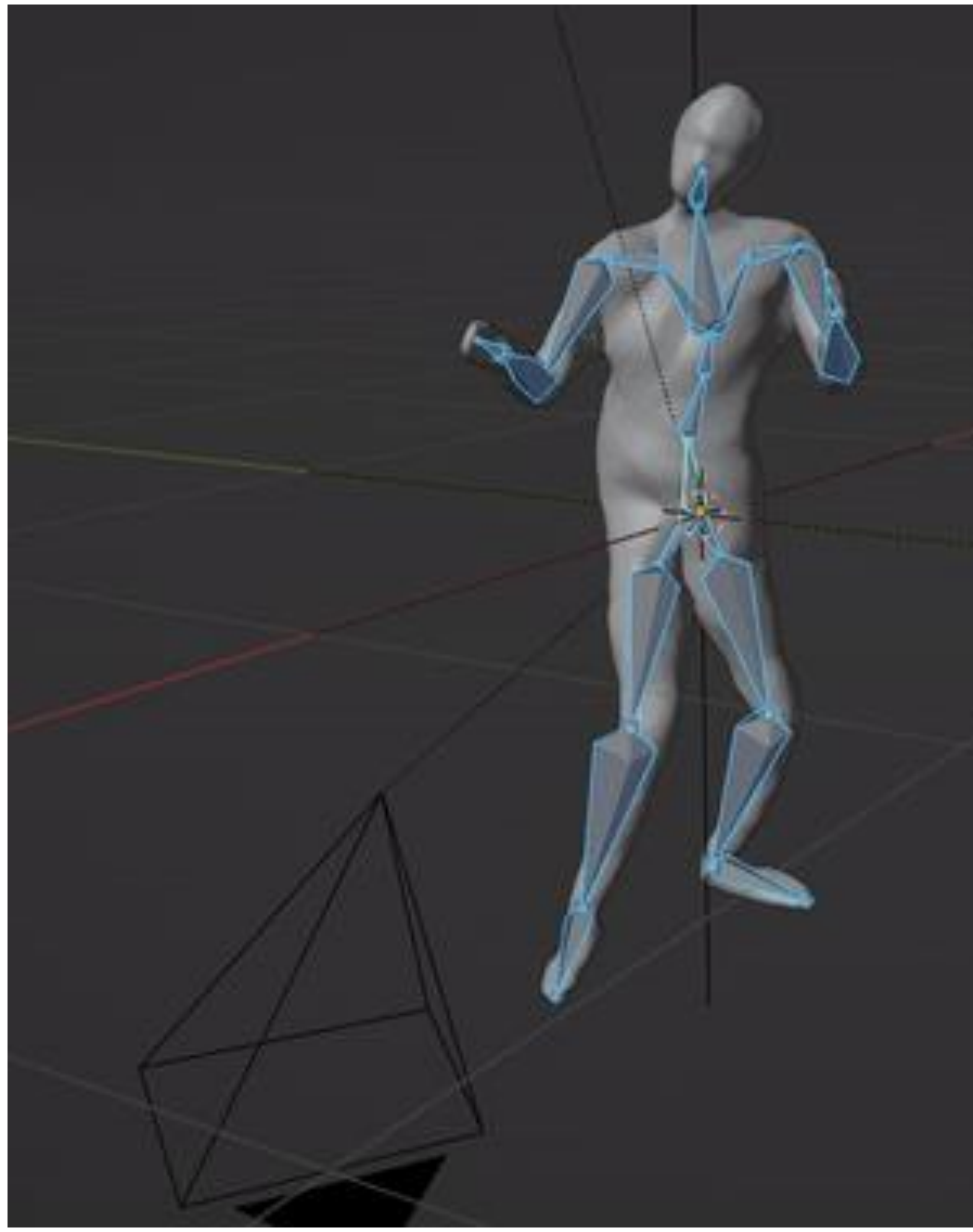
Body Space

$$\mathbf{p}(t) = \mathbf{R}(t)\,\mathbf{p}_0 + \mathbf{x}(t)$$

World Space

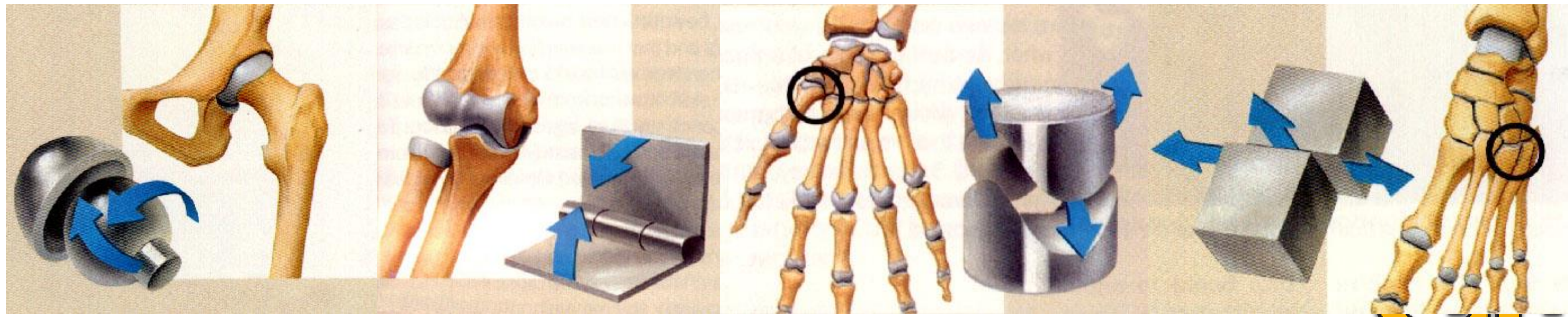






Articulated Bodies

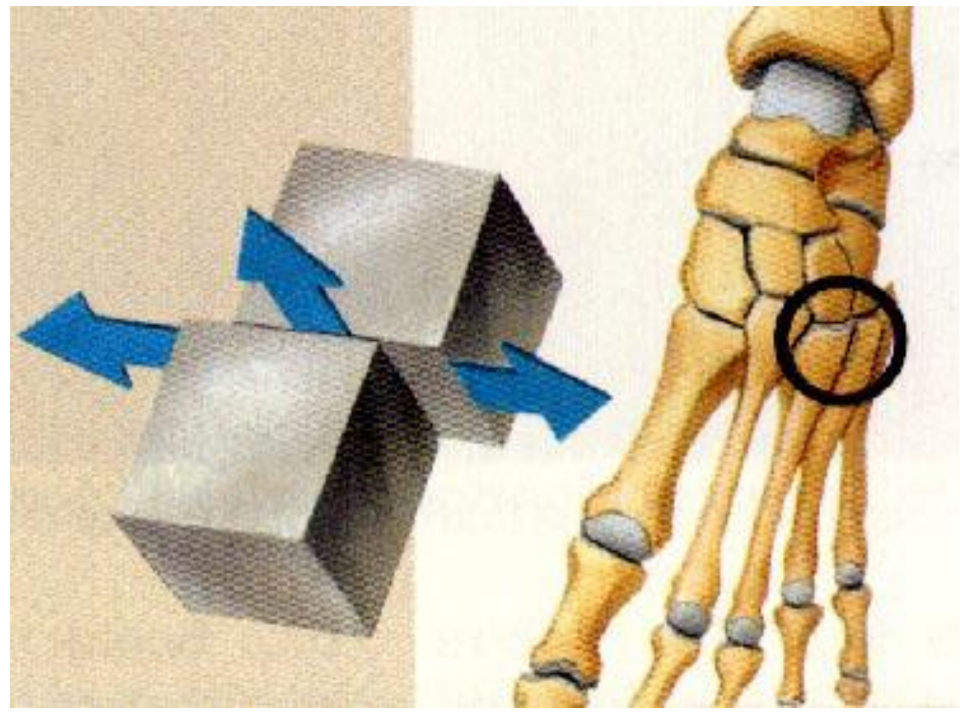
- Rigid-body pieces connected by joints
- Joints serve as constraints in simulation



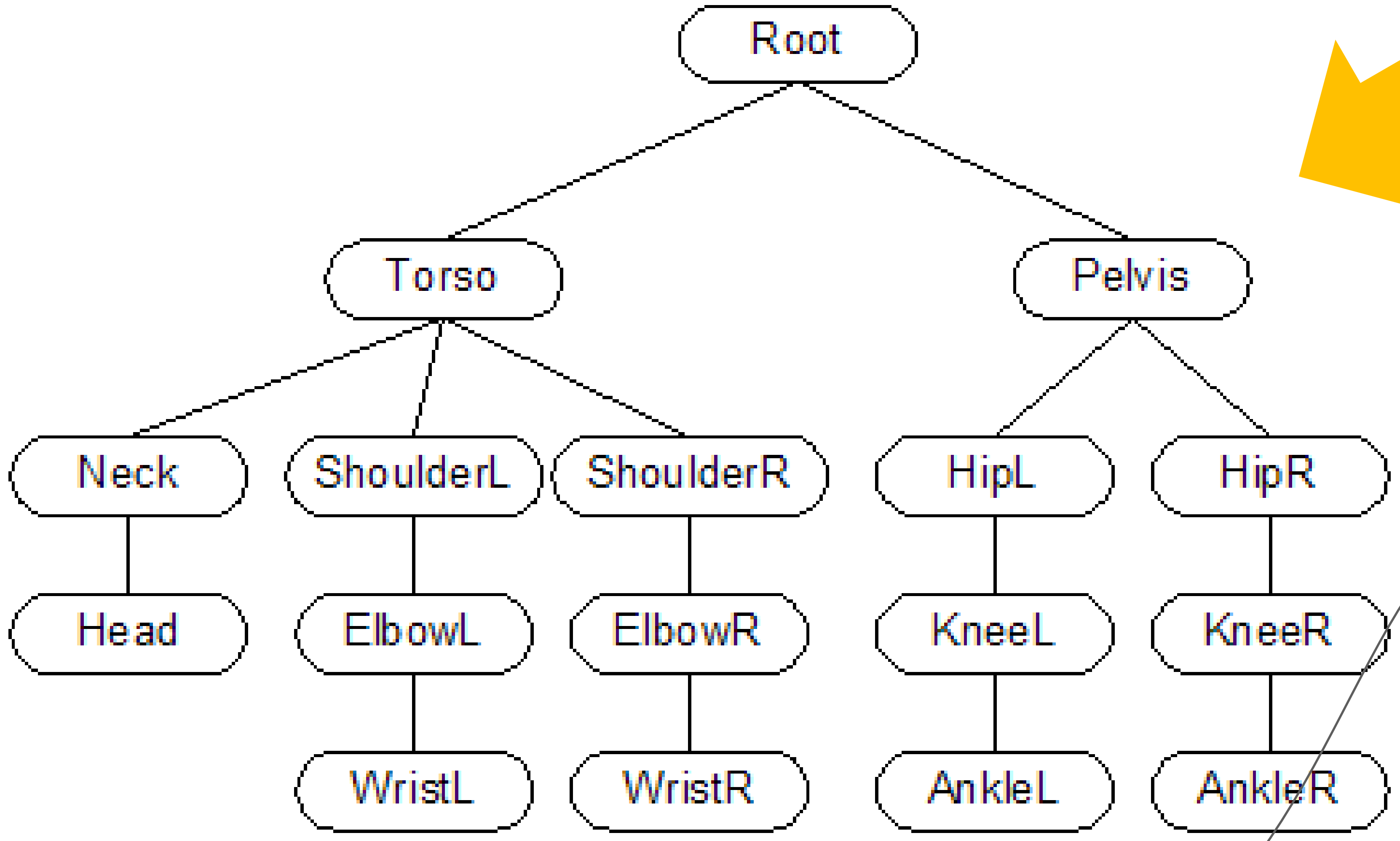
Mathematical Model of Articulated Bodies

- Articulated bodies are connected together by a number of joints
- Each joint has a specific type of constraint
- All joints are organized in a hierarchy

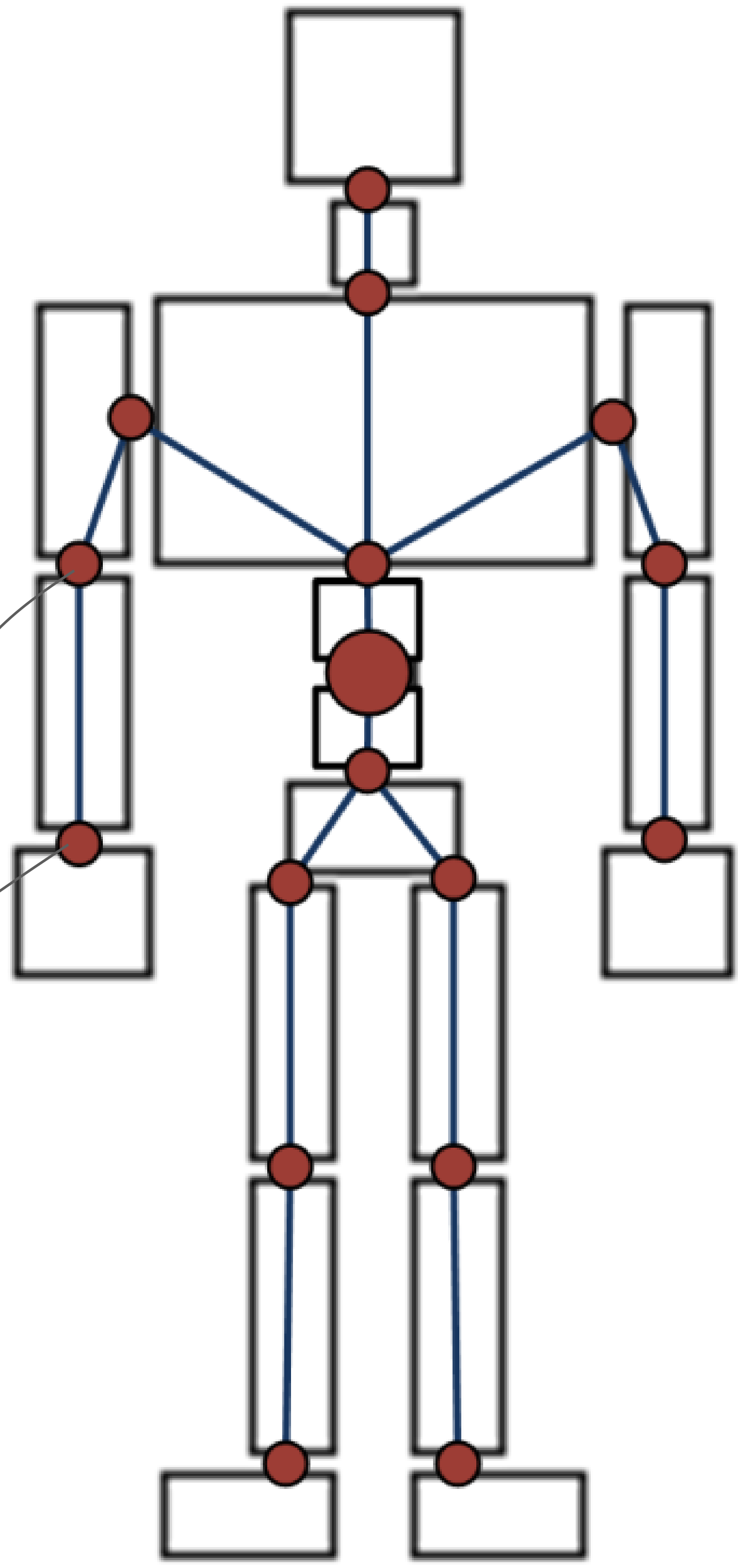
Hinge Joint



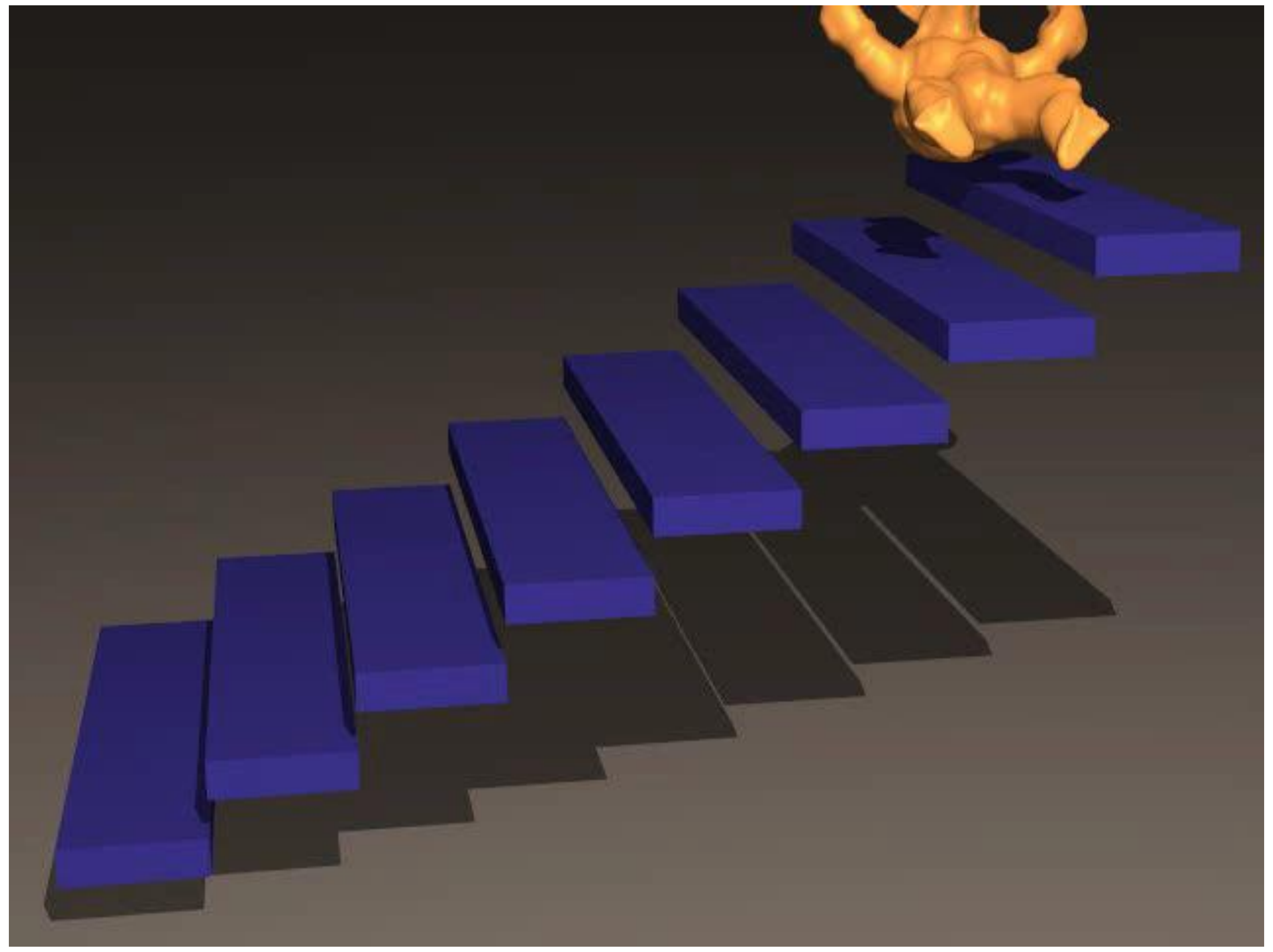
Sliding Joint

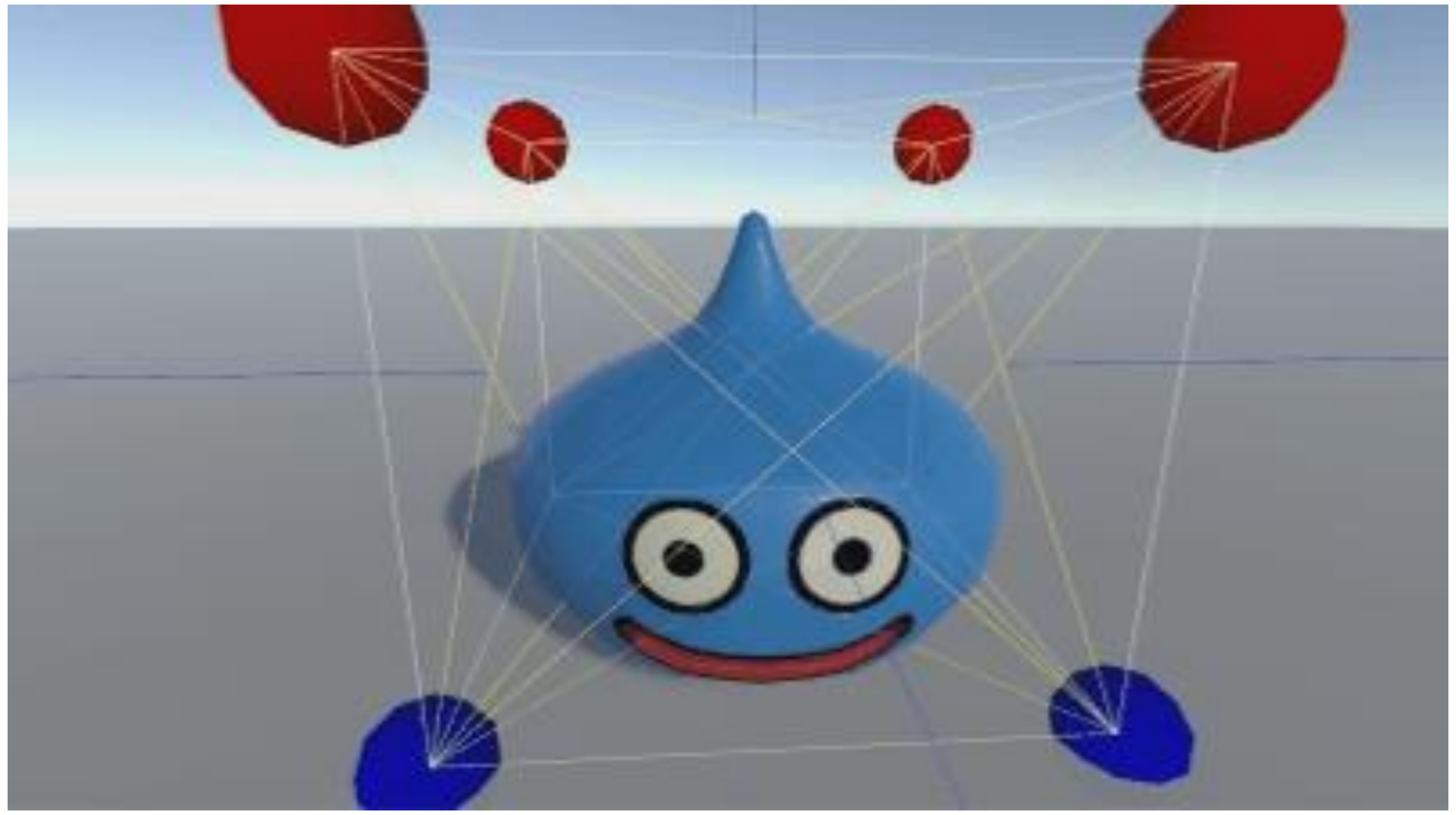


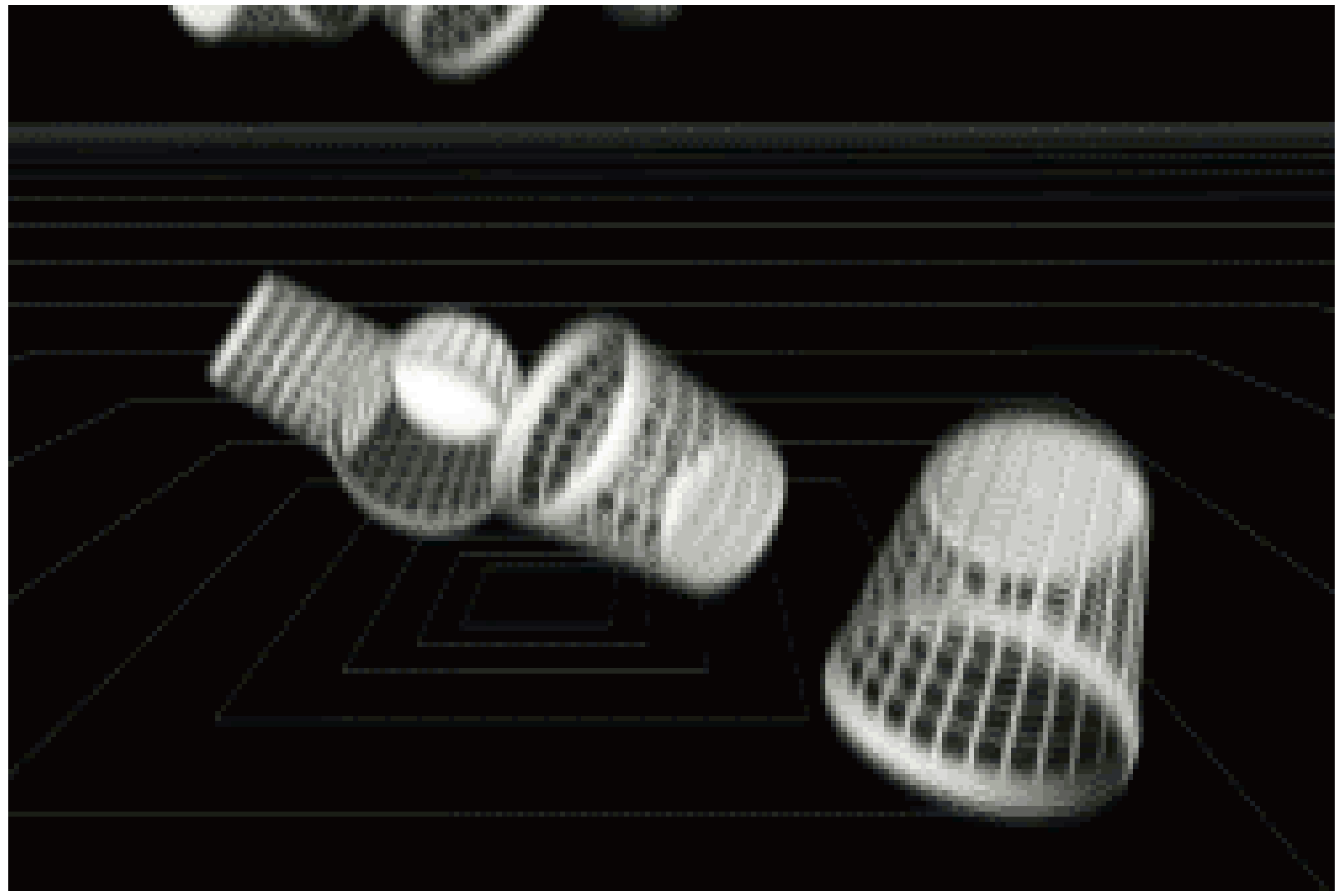
Joint Hierarchy



Articulated Bodies





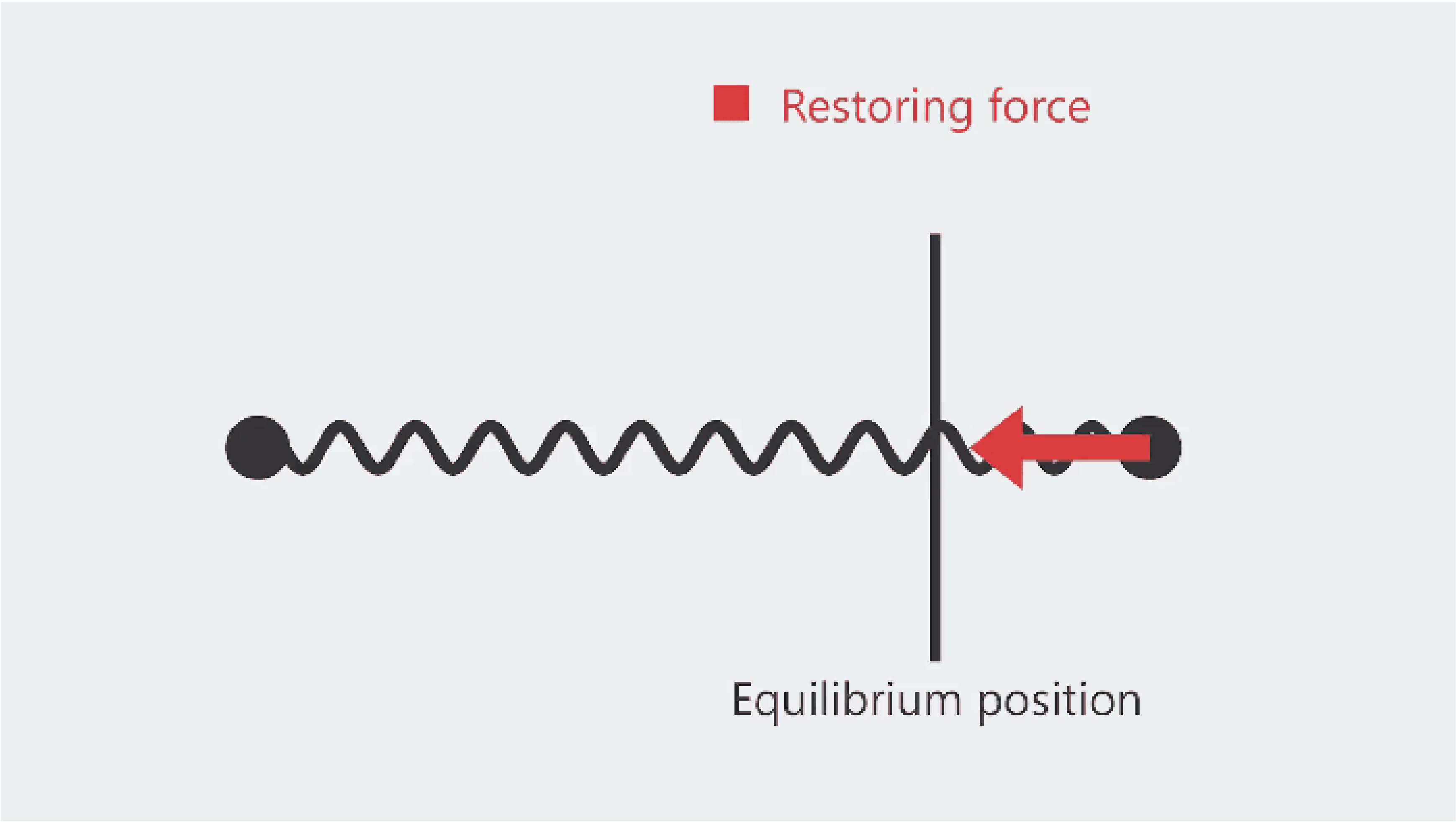


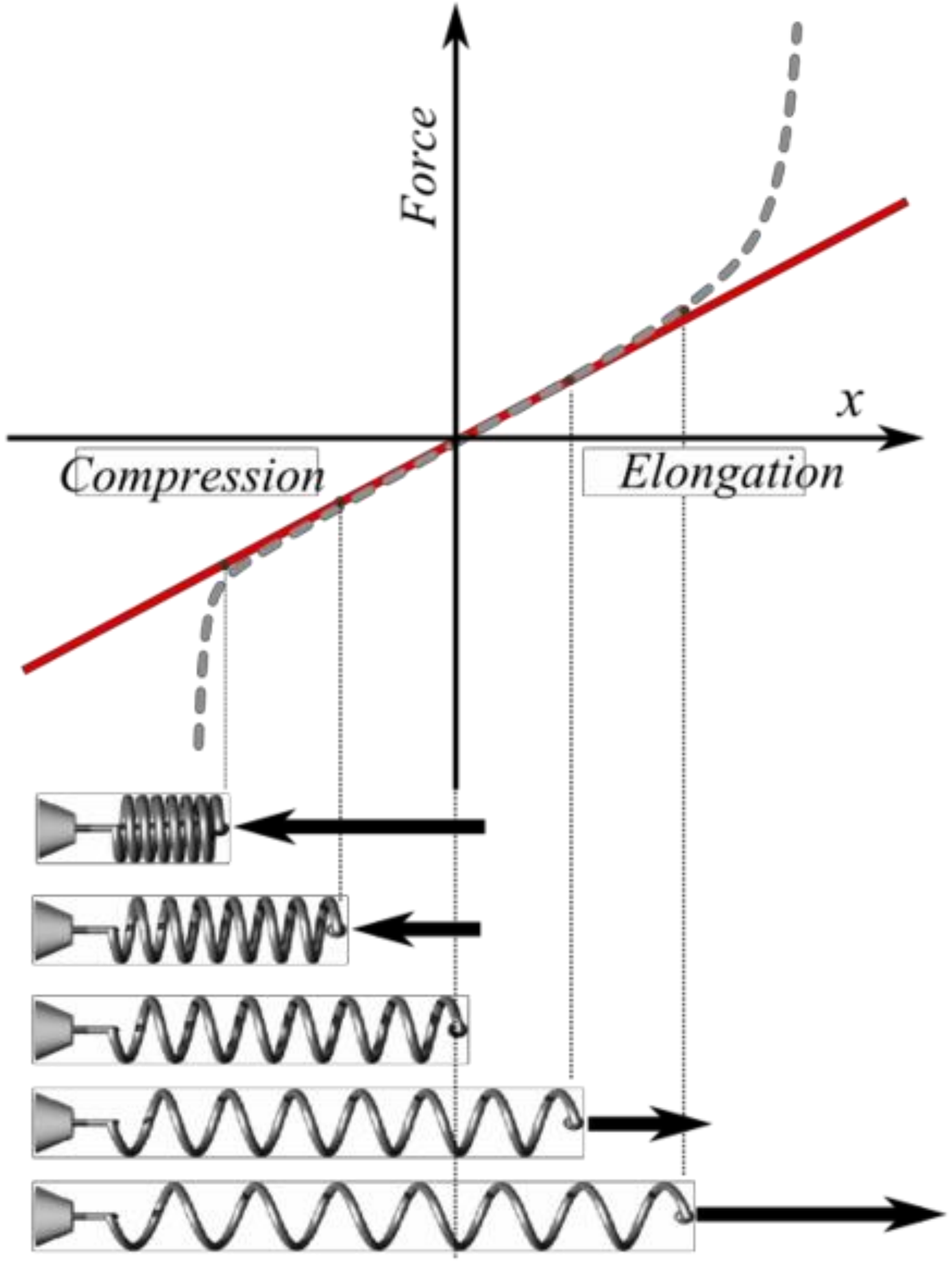
Soft Body

- Solid that deforms its shape while being acted on by external forces
- Motion needs to be described with many degrees of freedom Interactions among objects
- Different material properties can be simulated
- Many engineering methods such as finite elements (FEM) are used in graphical simulations

Mathematical Model of Soft Bodies

- A deformable object can be modeled as a set of particles connected by a set of springs (**Mass-Spring Model**)
- $f_{spring} = -kx$

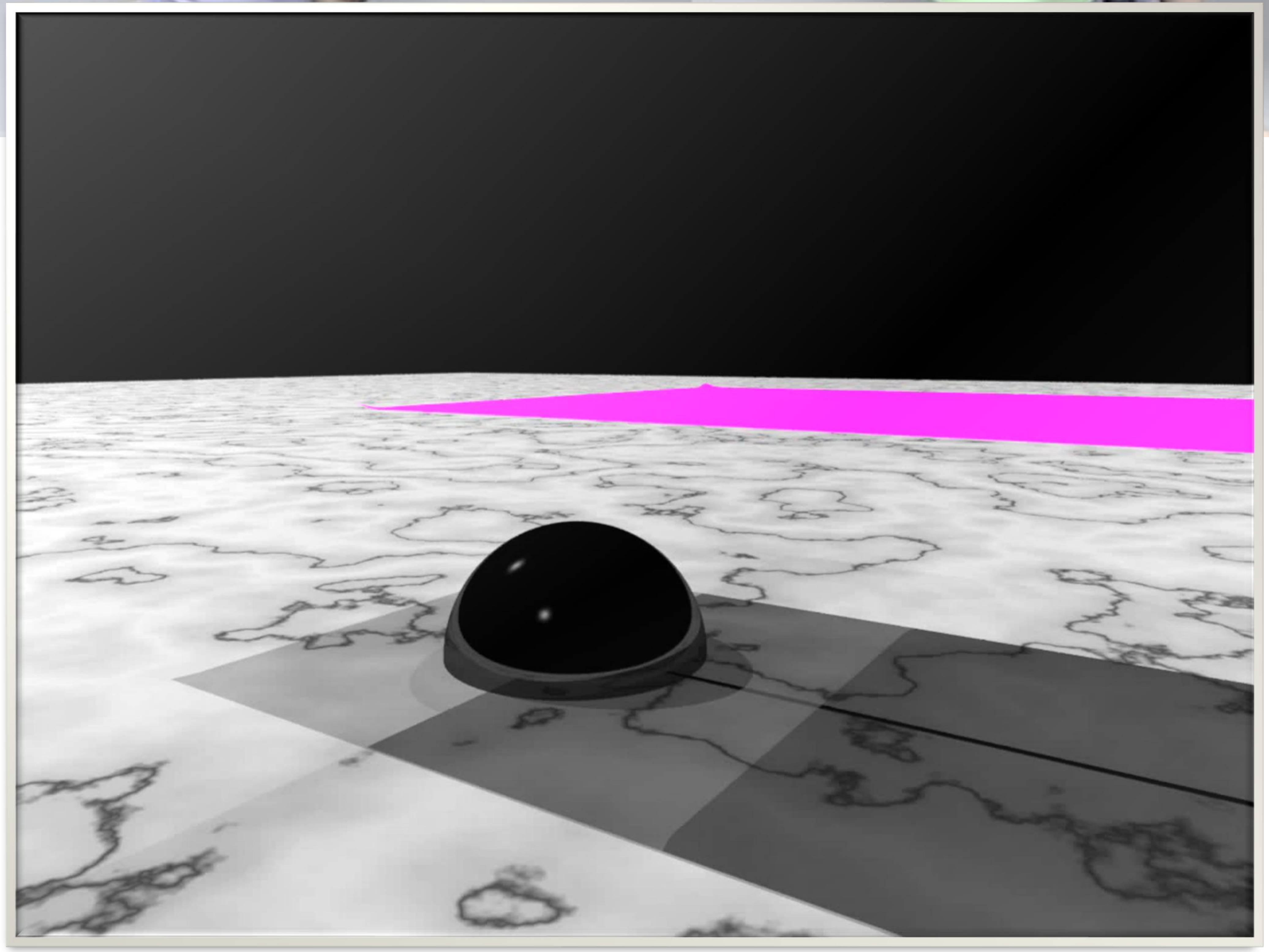


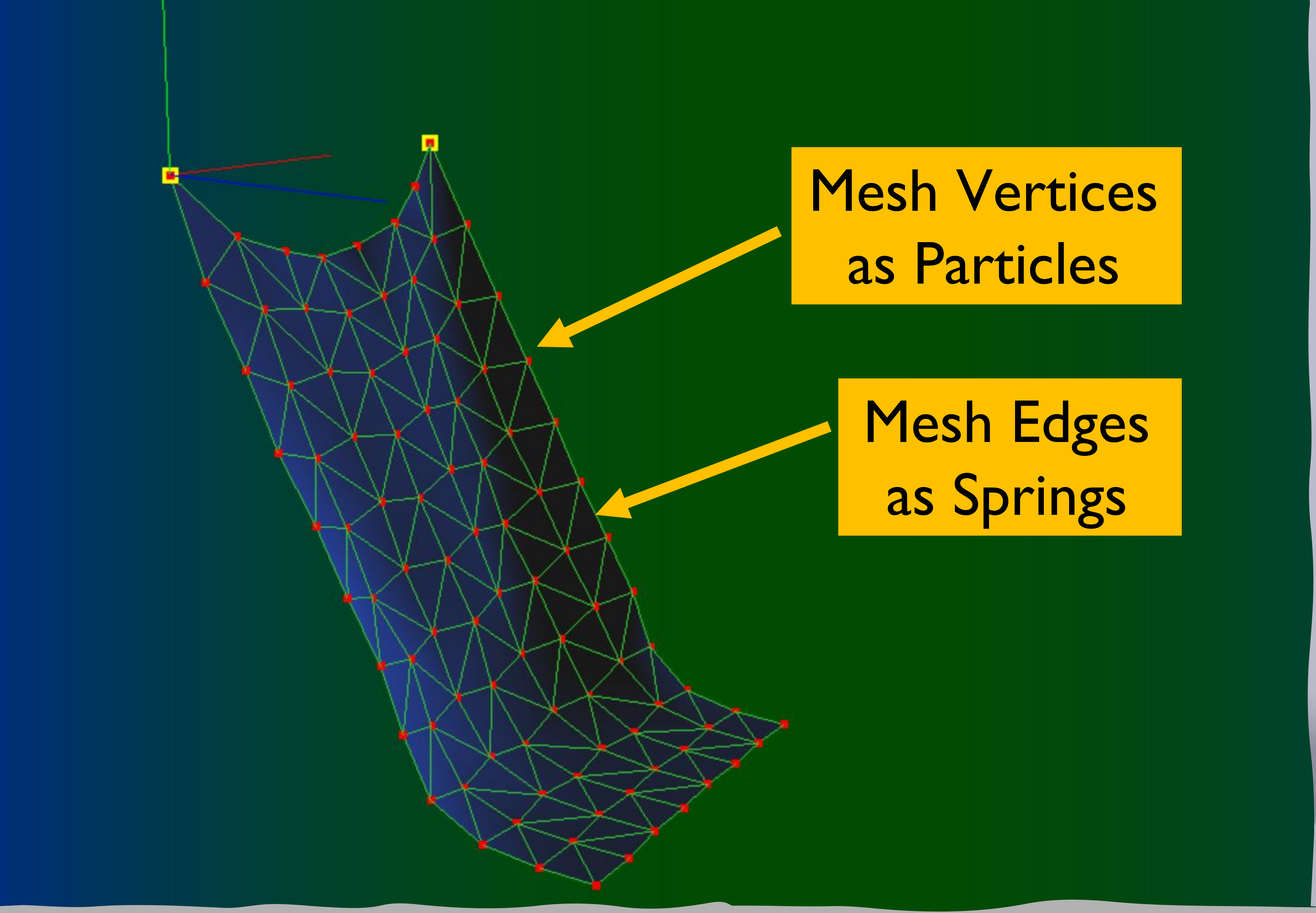



#THEGAMEAWARDS

Cloth

- Solid pieces that can bend but difficult to stretch
- Typically simulated as a particle systems connected by springs





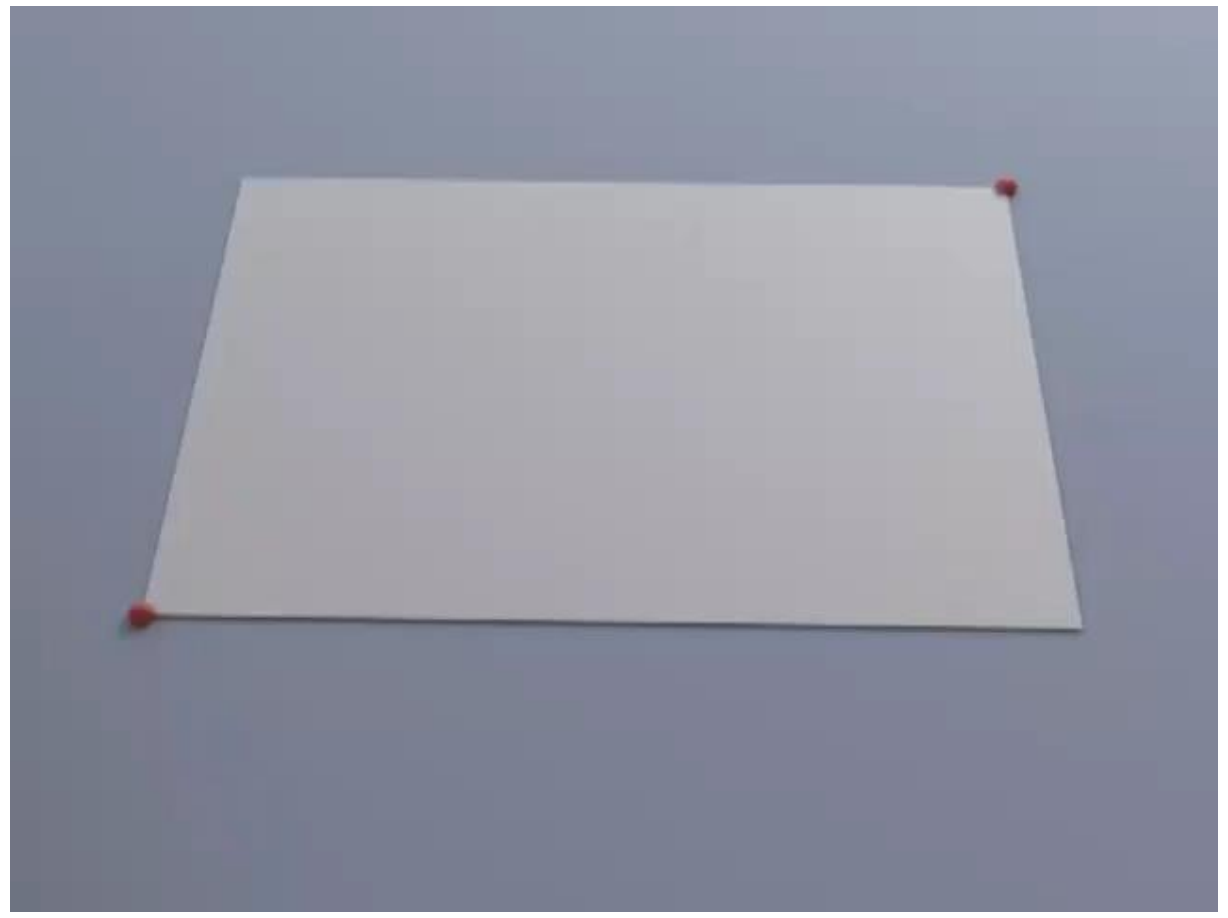
Mathematical Model of Cloth

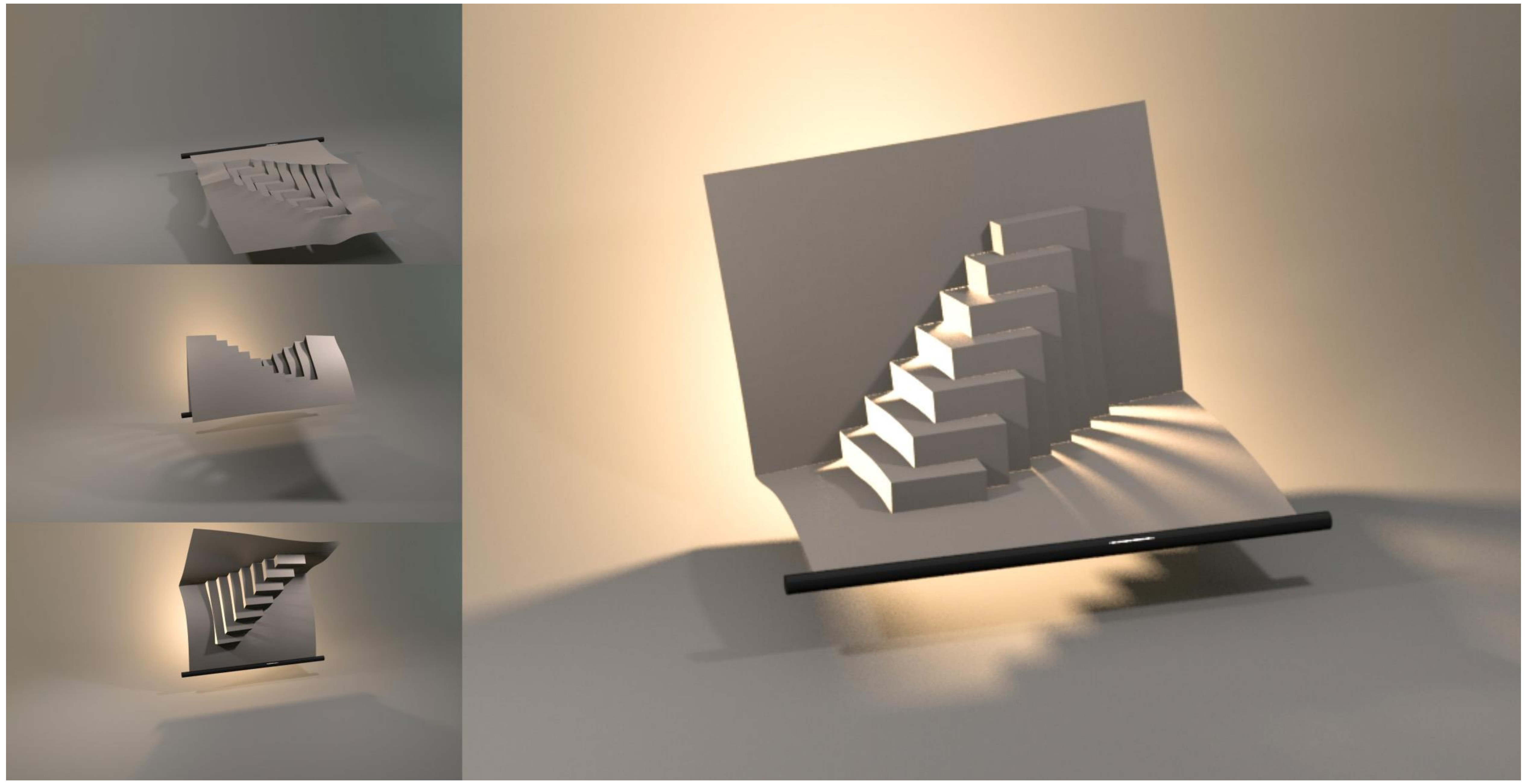
- A piece of cloth can be modeled as a **mass-spring model on a triangle mesh**
- The mesh **vertices** as particles and mesh **edges** as springs
- We usually have two types of springs to handle **stretching** and **bending** in separate

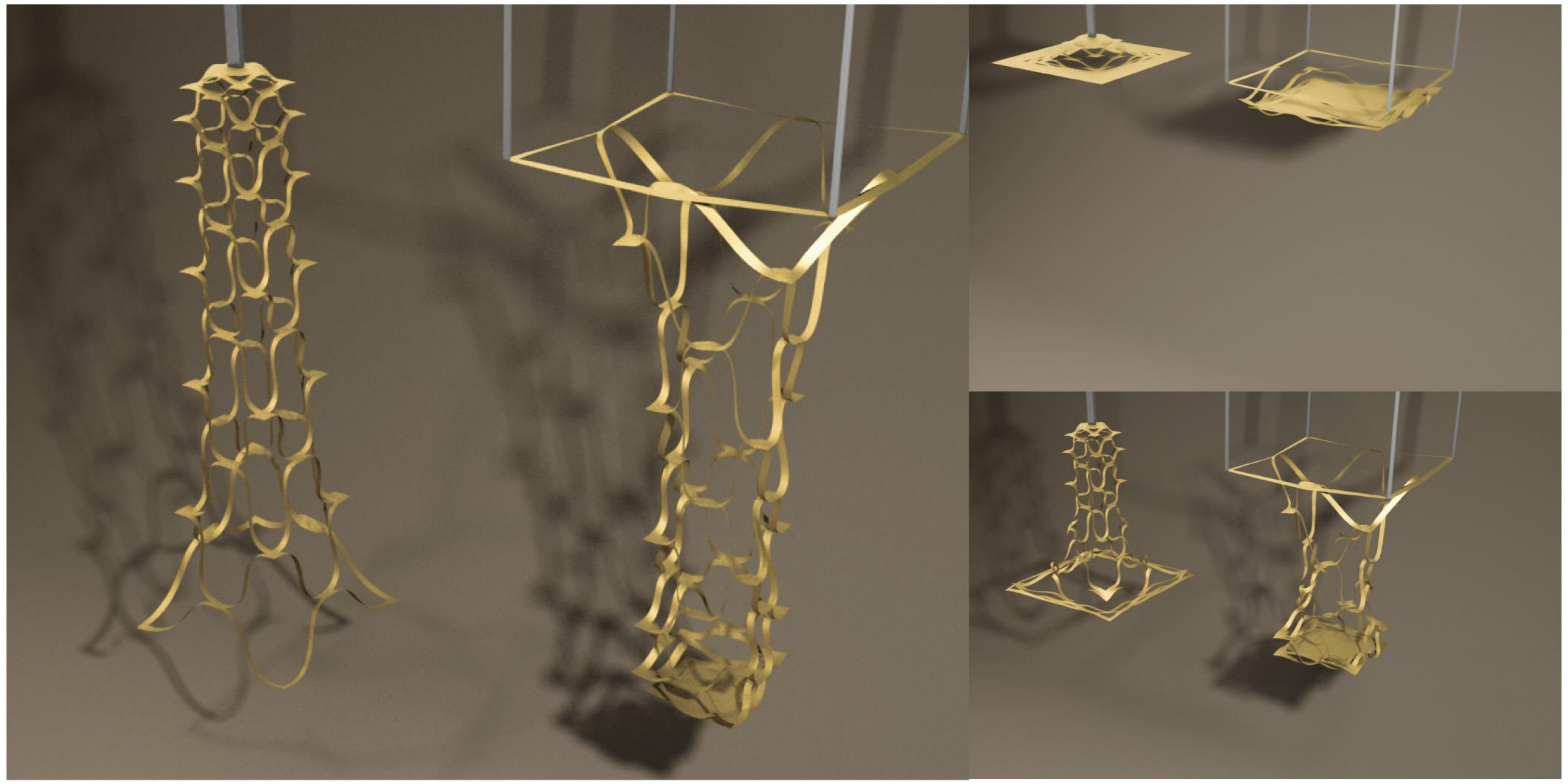

アウトライナー
アイテムラベル
CameraActor
CameraActor1
CameraActor2
CineCameraActor
CineCameraActor1
DefaultPawn
DirectionalLight
ExponentialHeightFog
Floor
GameModeBase
GameSession
GameStateBase
HUD
47 アクタ
詳細
シーケンサー

Paper

- Paper does not stretch (like cloth) but easy to bend and fold ceases (unlike cloth)

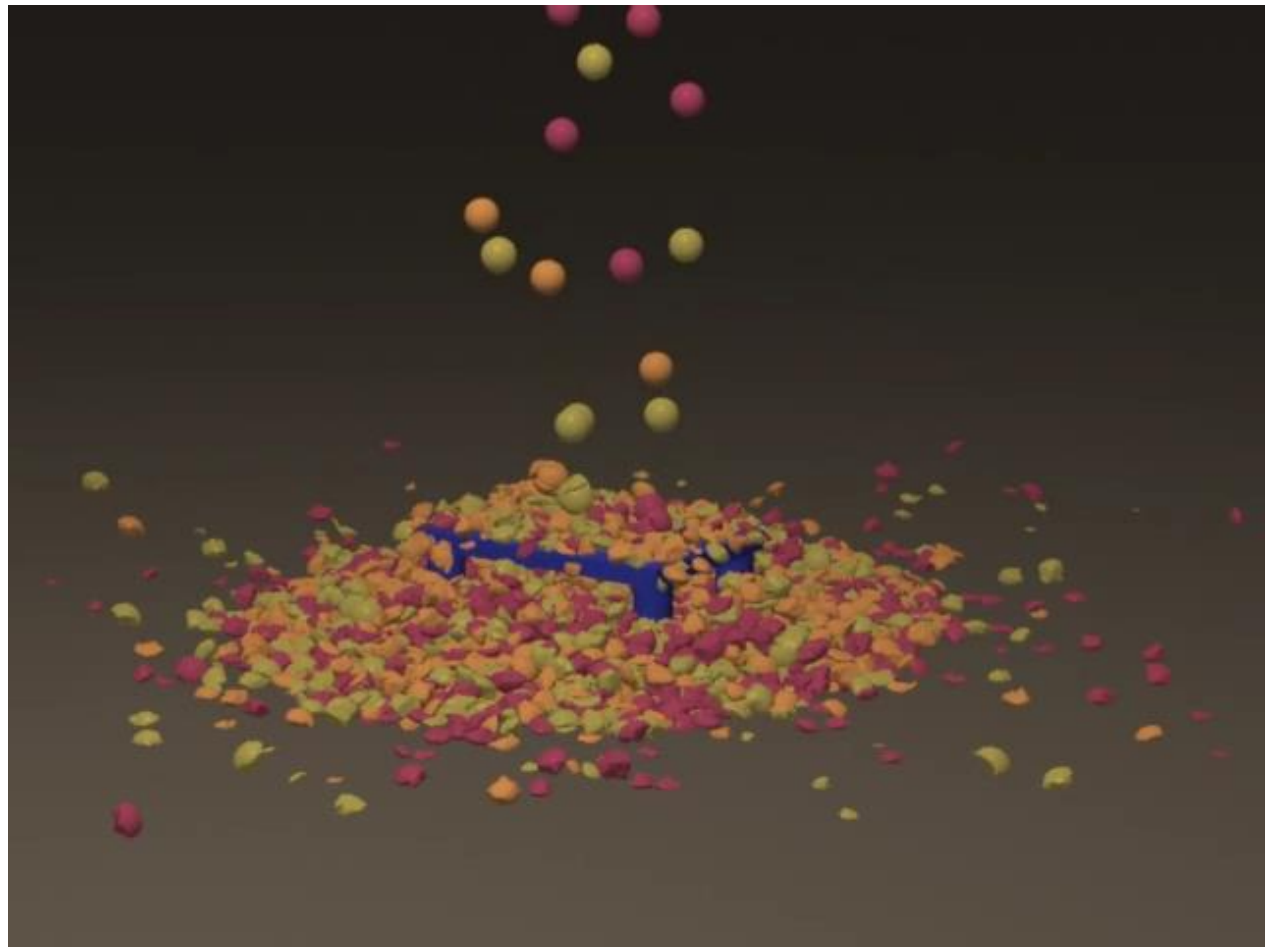
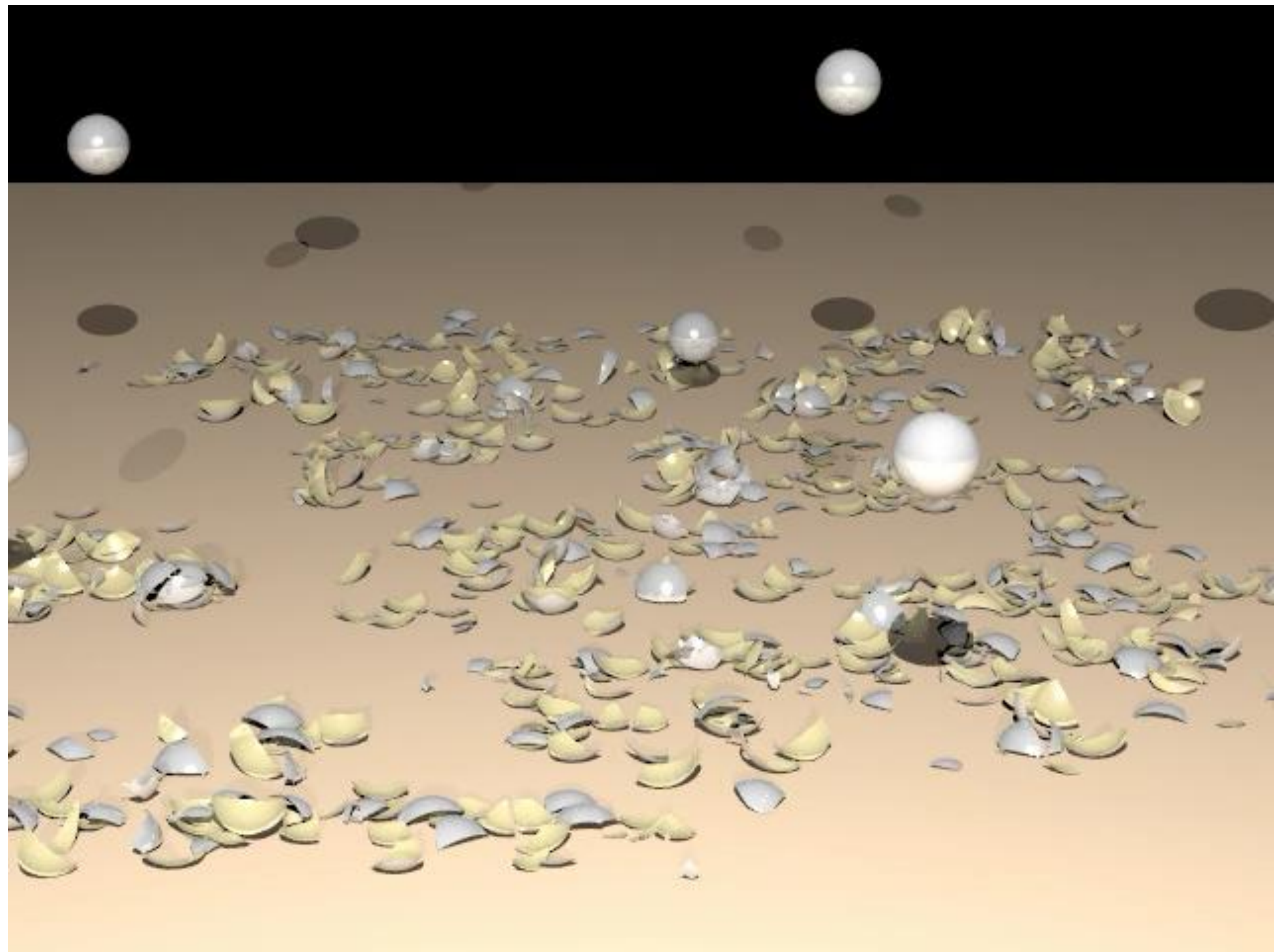
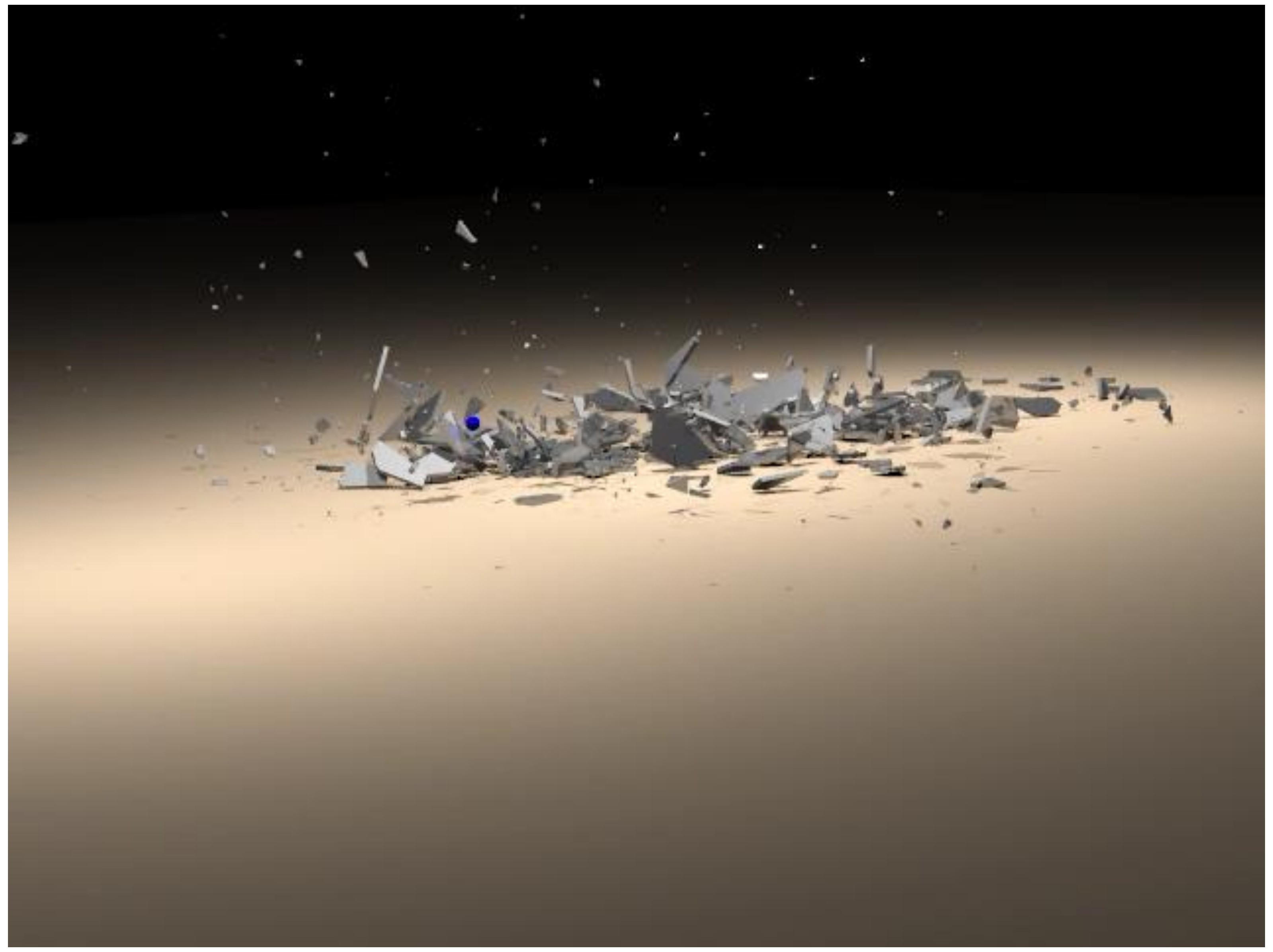






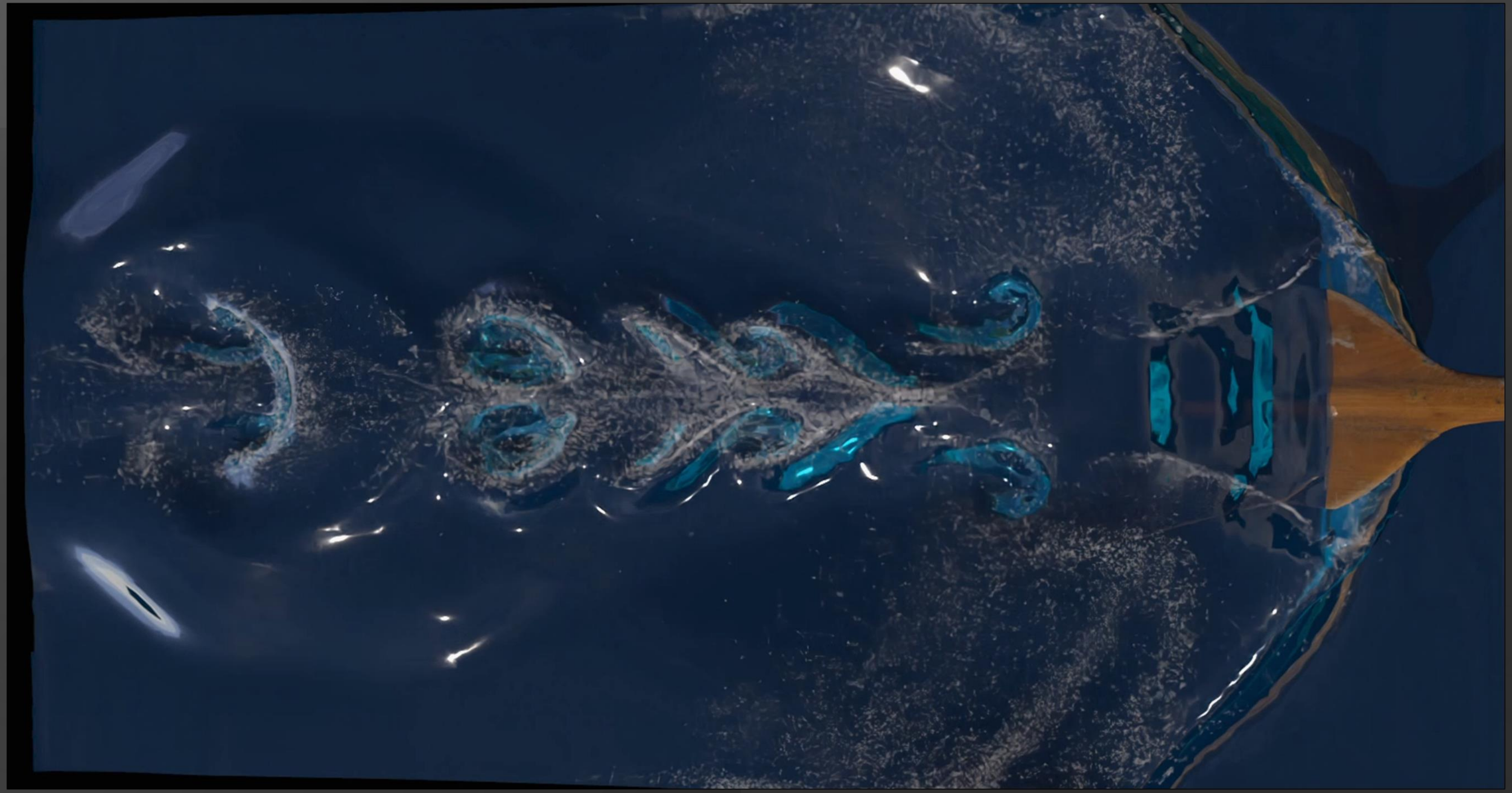
Fracture

Objects breaks into multiple pieces due to collision and impulse.

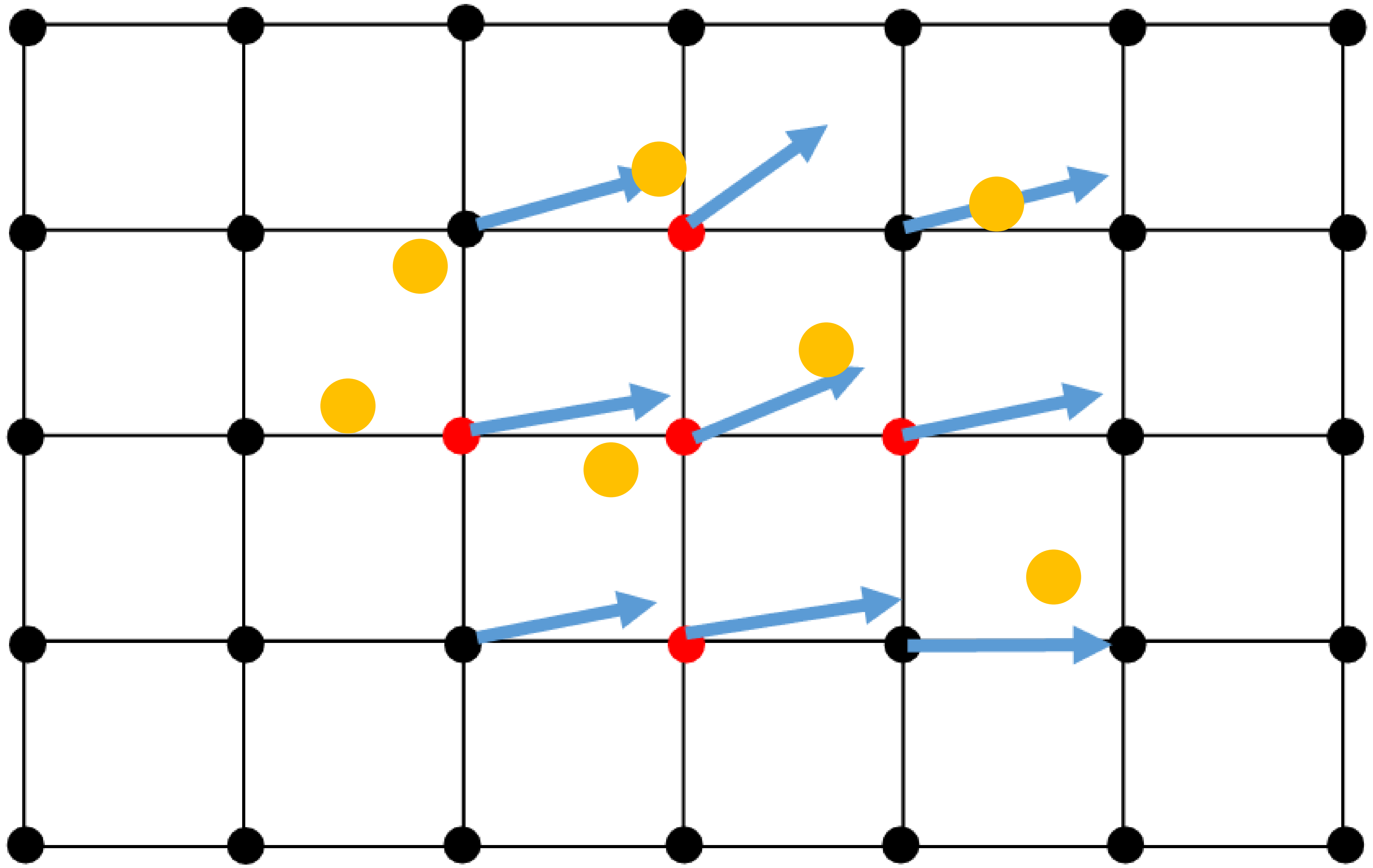

OBJECTIVES COMPLETE
PRESS [SPACE] TO CONTINUE
OBJECTIVE
DESTROY THE TOWER
NORMAL
CAMERA MODE [F2]
21.0
FIELD OF VIEW [F3/F4]
RESET CAMERA [F5]
1,263 XP
2
OBJECTIVE COMPLETE
OBJECTIVE COMPLETE
DESTROY THE TOWER
RUINS DESTRUCTION 46.1%
BUILDING DESTRUCTION 20.8%
ALTITUDE 19M
VEHICLE SPEED 27 KM/H
PLAY TIME 1:04
W D
S A

Water flows with a free surface and can interact with solid objects by forming vortices and small splashes

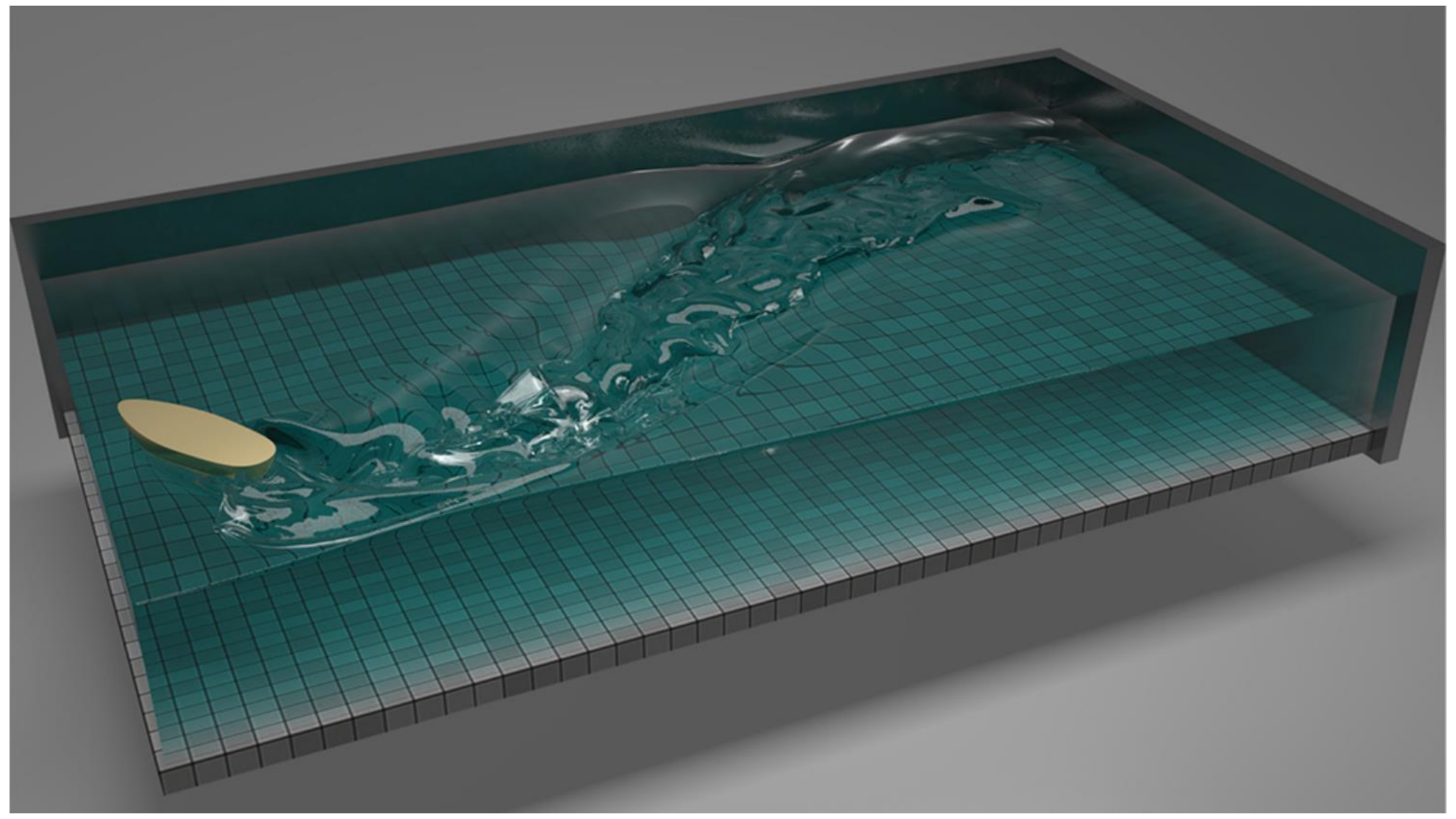


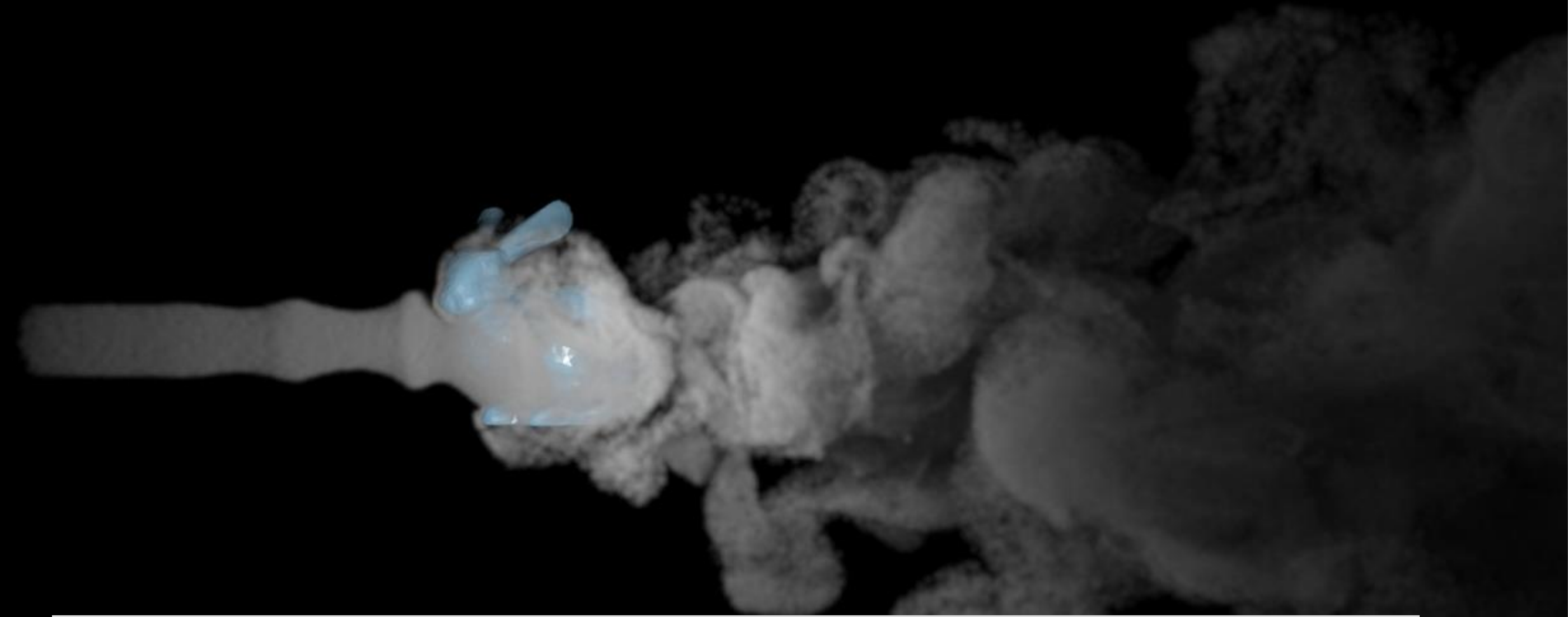
Mathematical Model for Fluids

- Fluid motion is simulated by solving Navier-Stokes equations with computer program
- Grid and particles are two fundamental data structures to solve these equations
- The simulators are usually expensive to run and require GPU hardware to accelerate

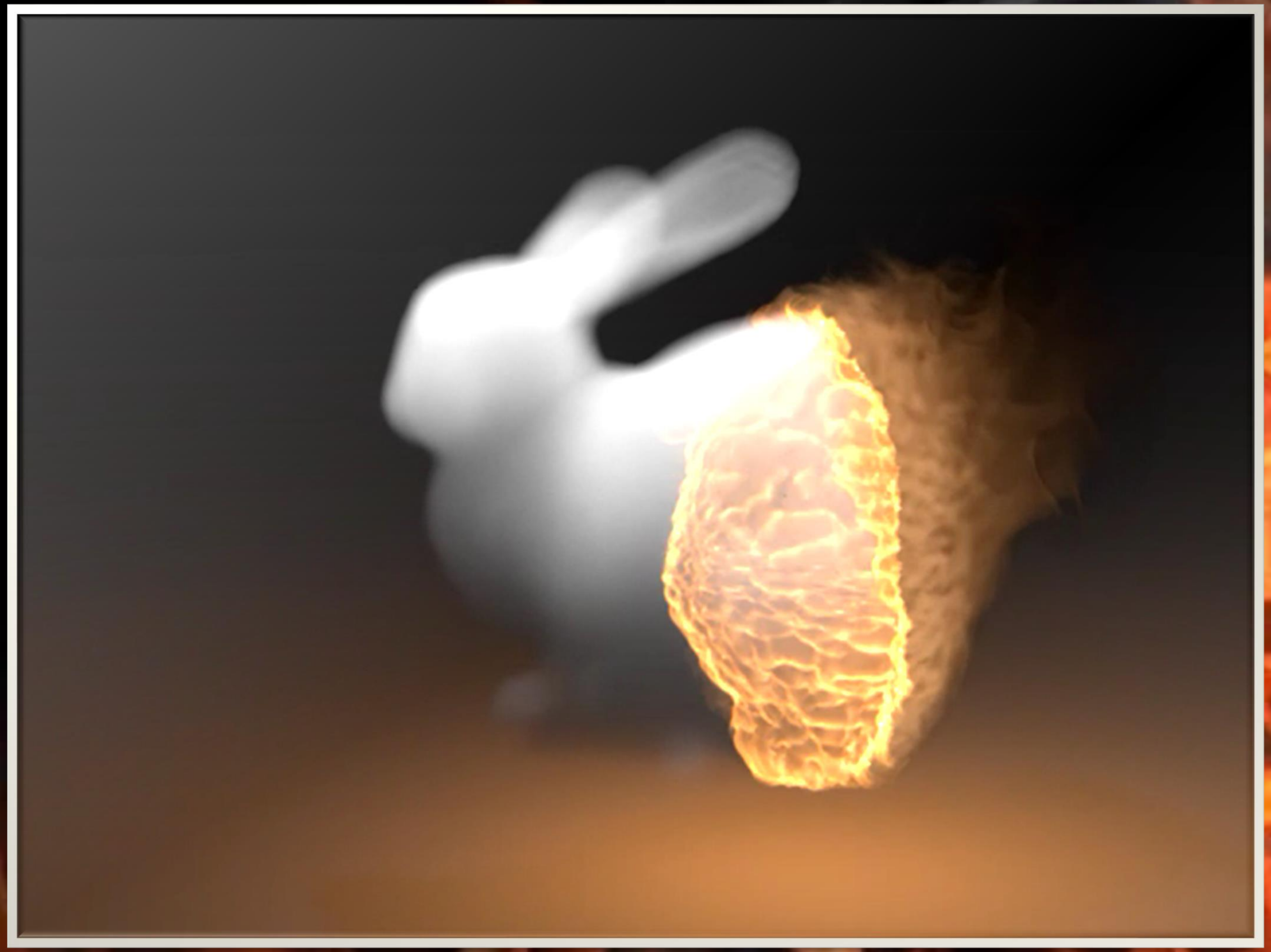


$$\begin{cases} \dfrac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\dfrac{1}{\rho} \nabla p + \nu \nabla \cdot \nabla \vec{u} + \vec{g} \\ \nabla \cdot \vec{u} = 0 \end{cases}$$



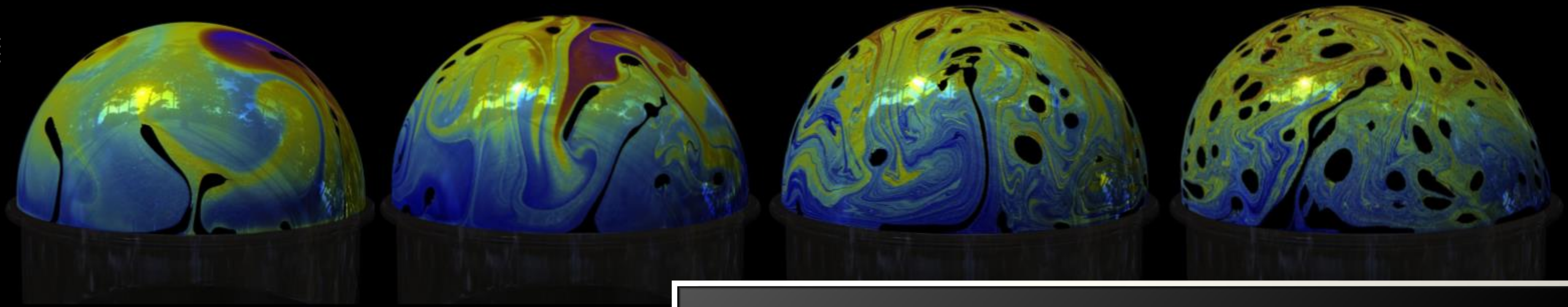


Smoke flows with air, displays a lot of vortices, and smoothly bypasses solid obstacles without being obstacled

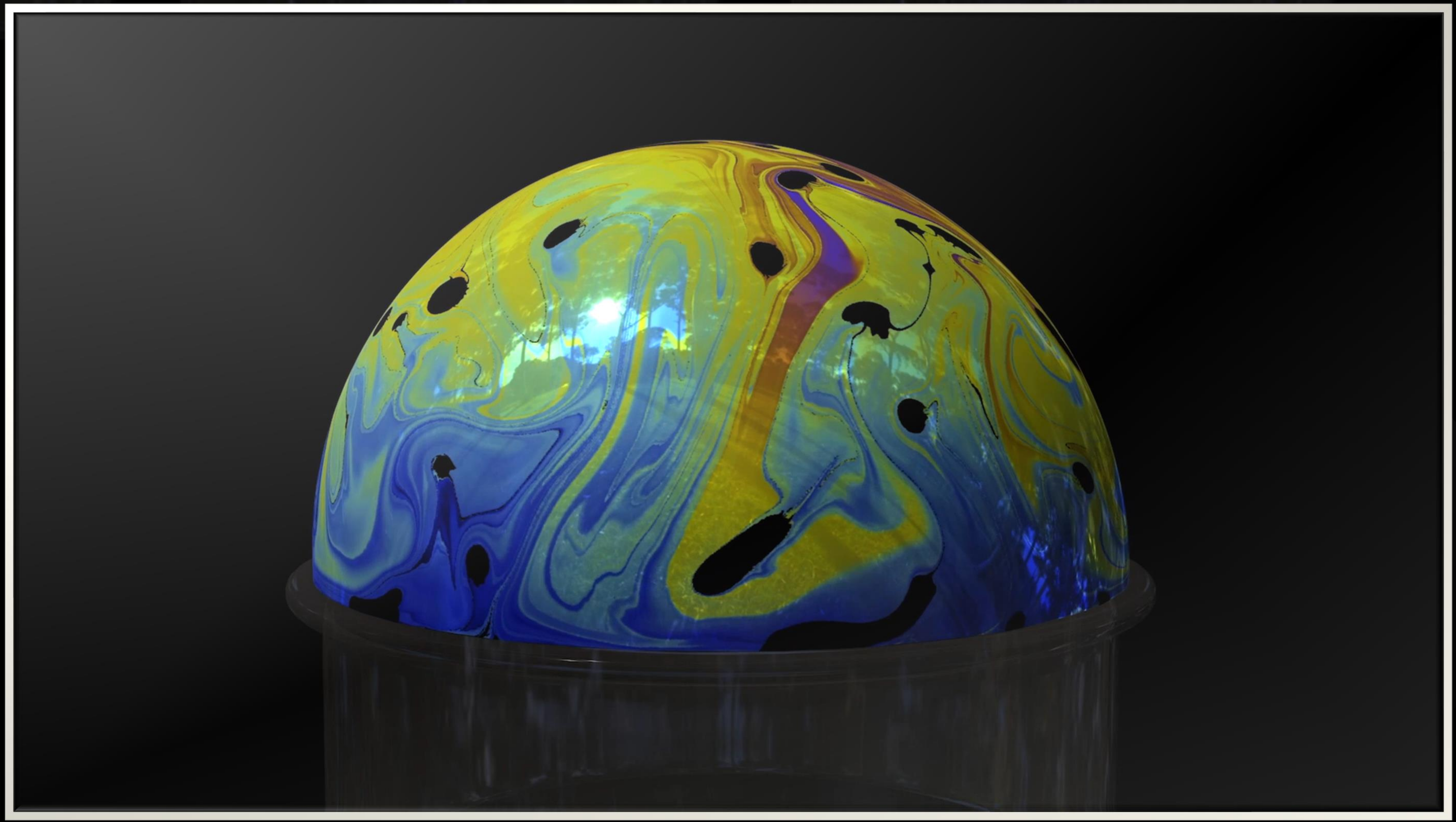


Fire flows with the surrounding air flow, which is driven by the unstable chemical reactions on interface between fuel and air

Snow is a kind of material that behaves half solid and half fluid --- it can flow under large external forces, and stay statically without force



Bubble and Foam:
fluid that flows on a deforming thin film




We can simulate all kinds of physics in computer graphics!
Floating leaves in brook
Compression
Joint rotation
Material Voxels
Mechanism

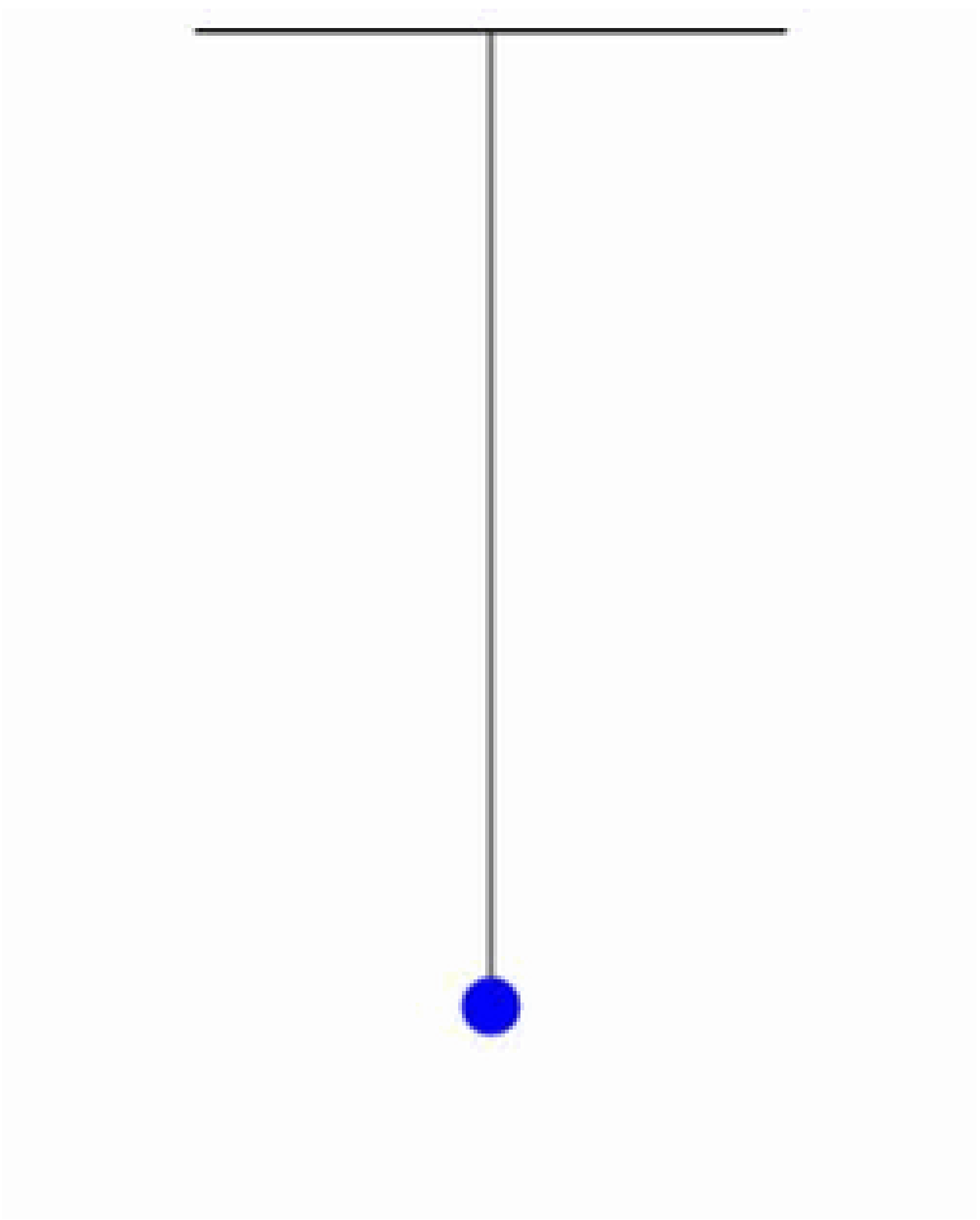
Physics-based Simulation

Simulation =

Physical Model + Geometric Representation + Temporal Evolution

Physical Models

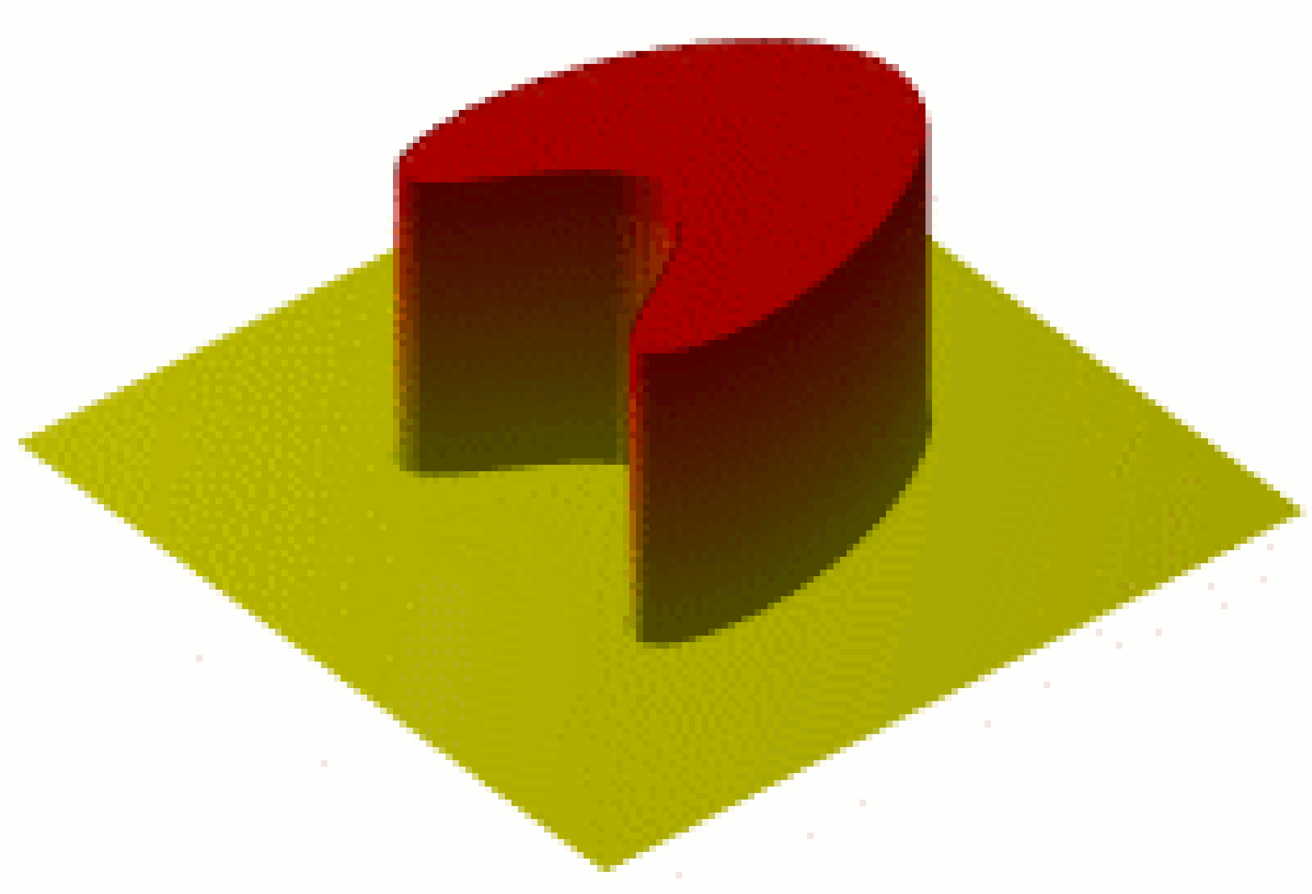
- Ordinary Differential Equations (ODE)



$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin\theta = 0$$

single variable function $\theta(t)$

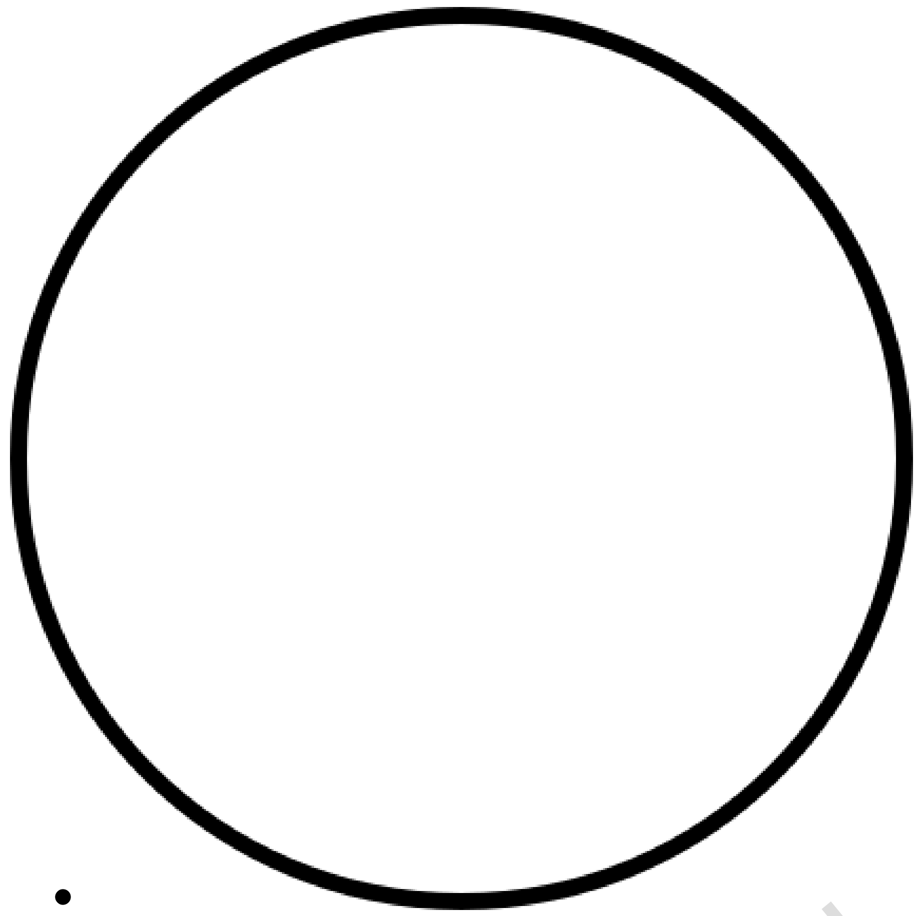
- Partial Differential Equations (PDE)



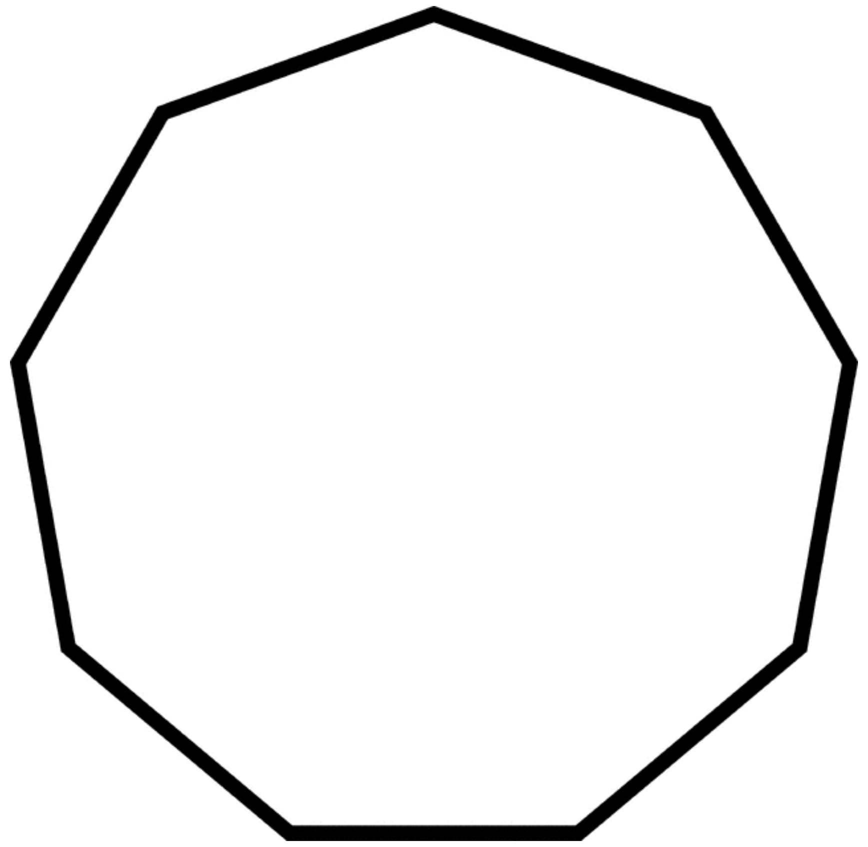
$$\frac{\partial u}{\partial t} - \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) = 0$$

multivariable function $u(t, x, y, z)$

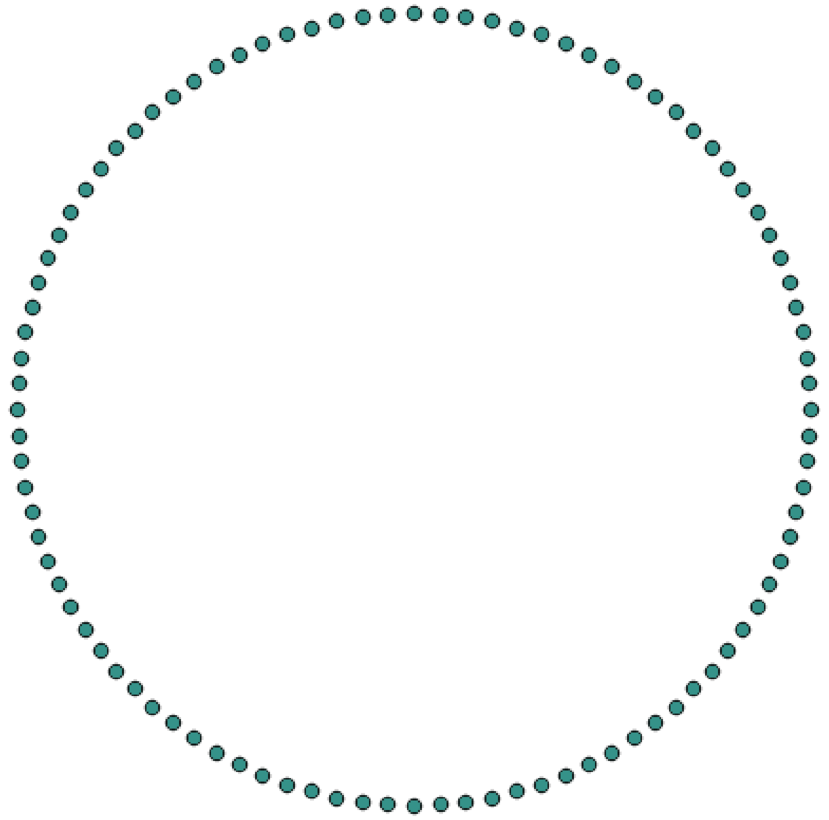
Geometric Representation



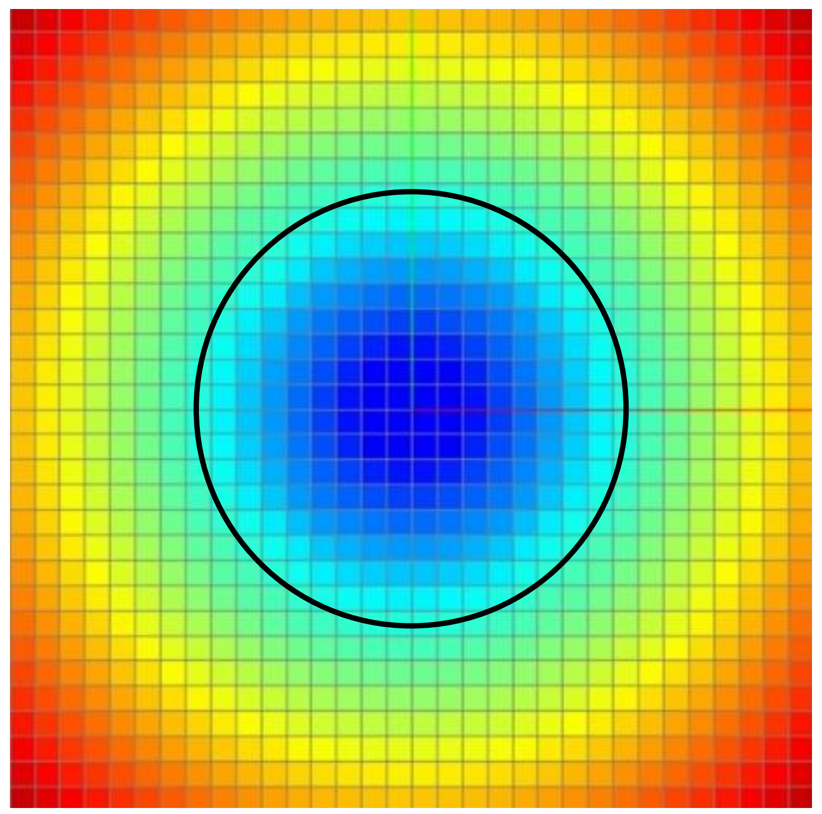
How to store a shape in computer?



Mesh

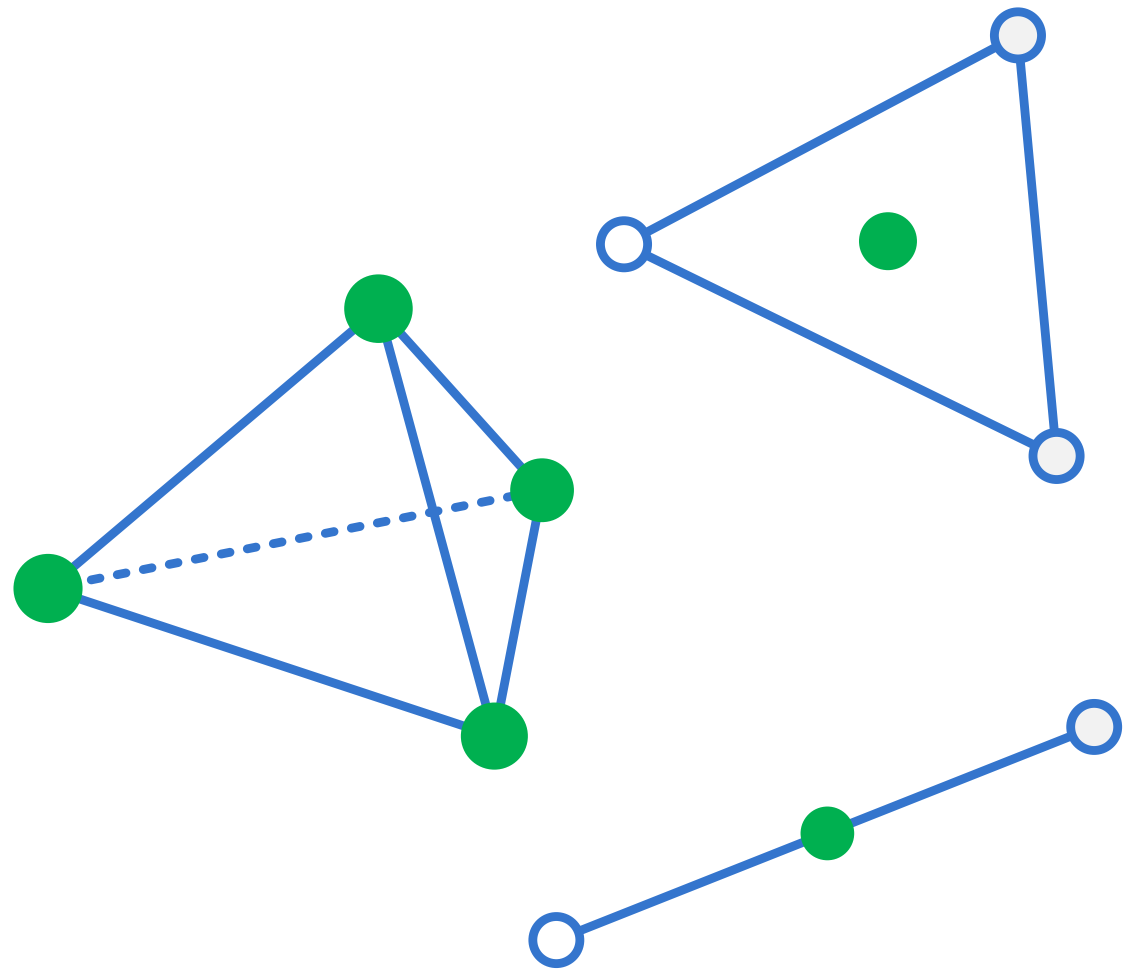


Particles

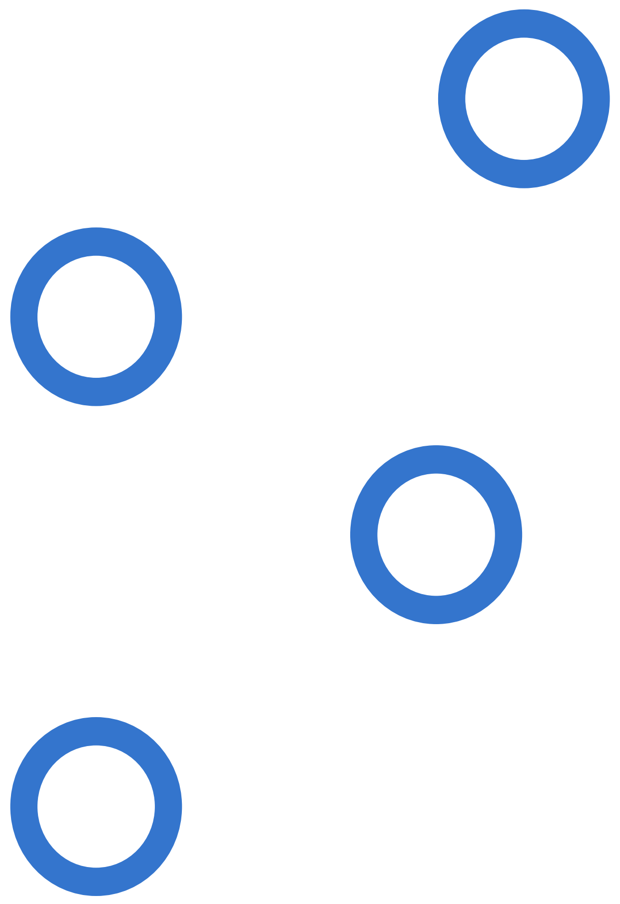


Grid

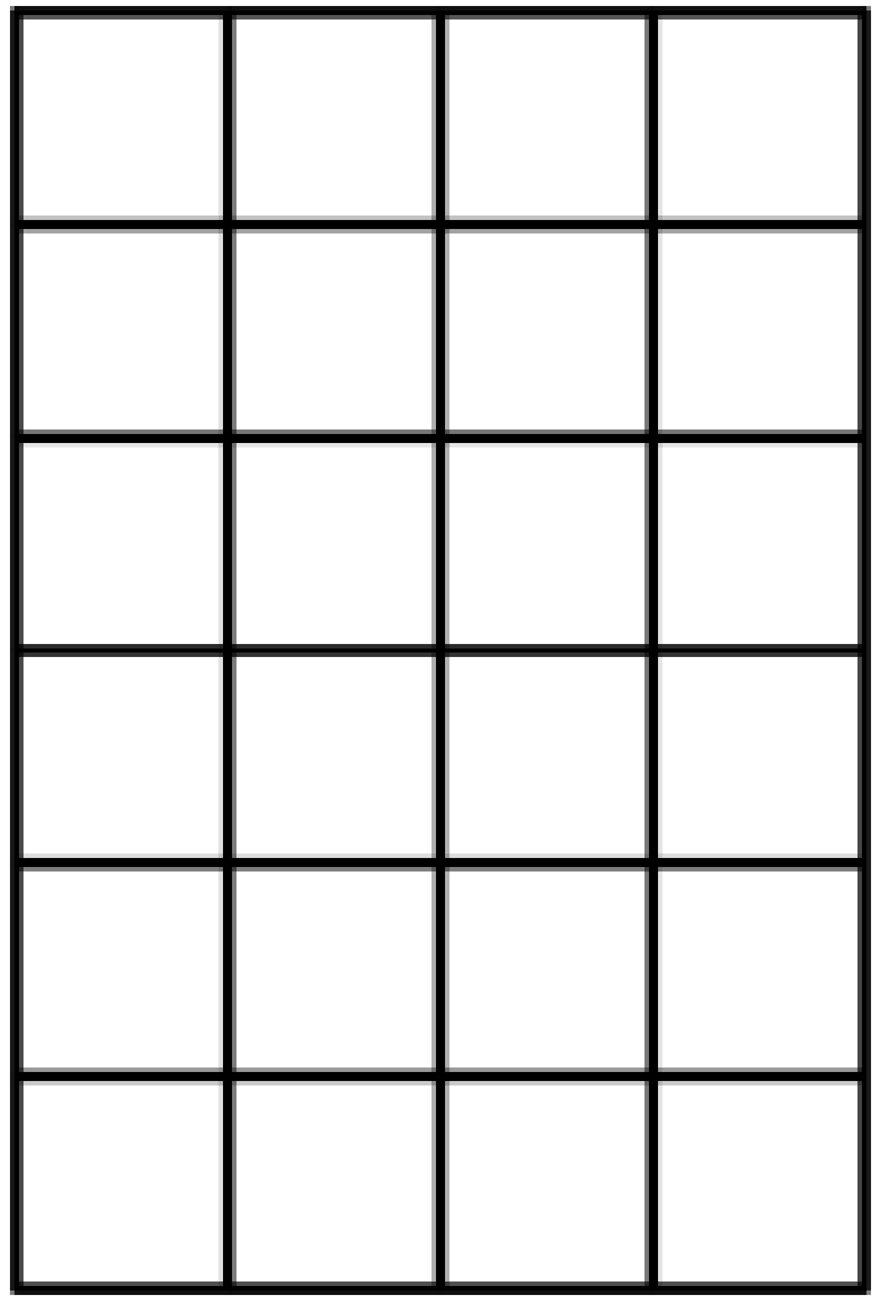
Geometric Data Structures



Mesh

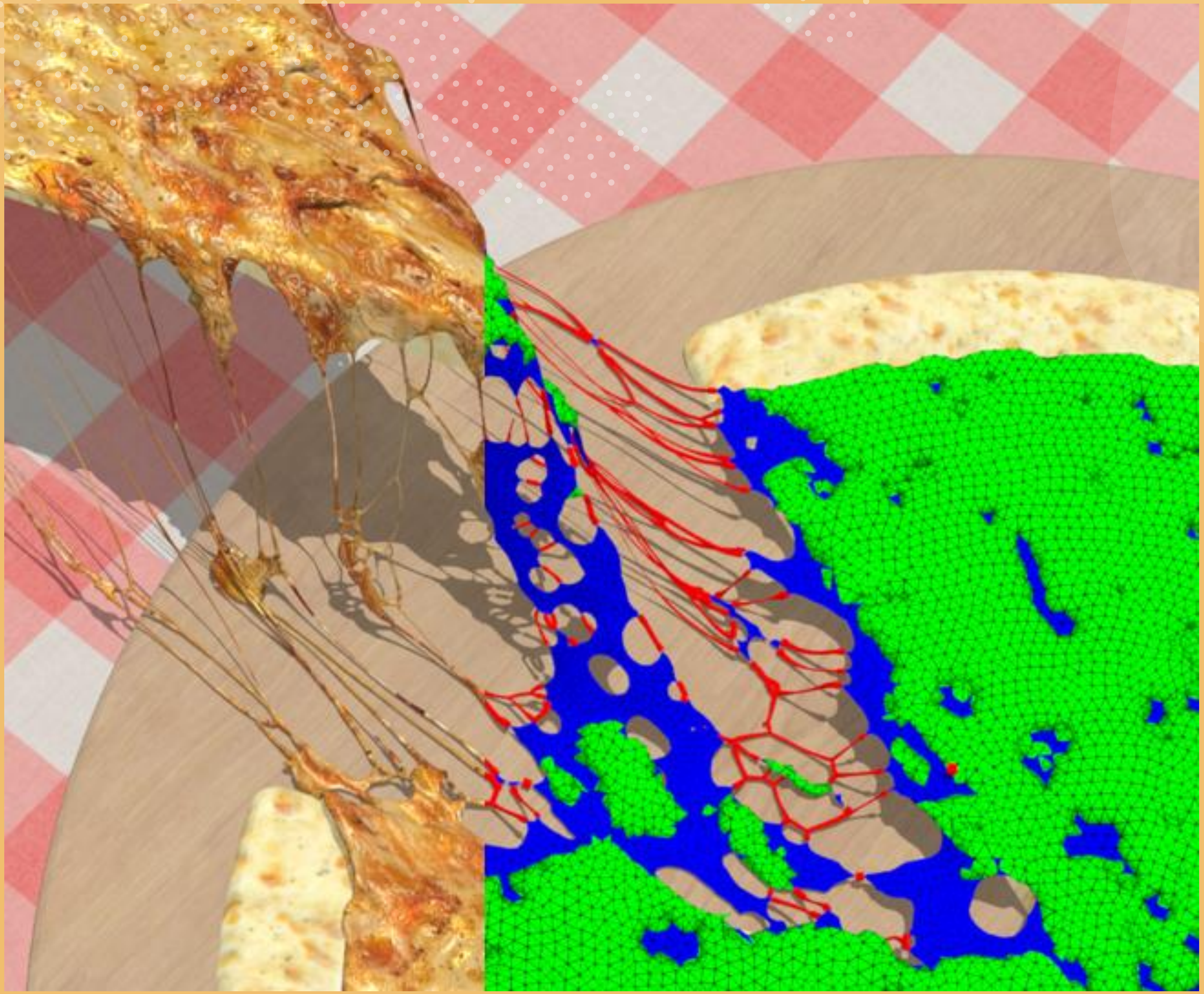


Particles

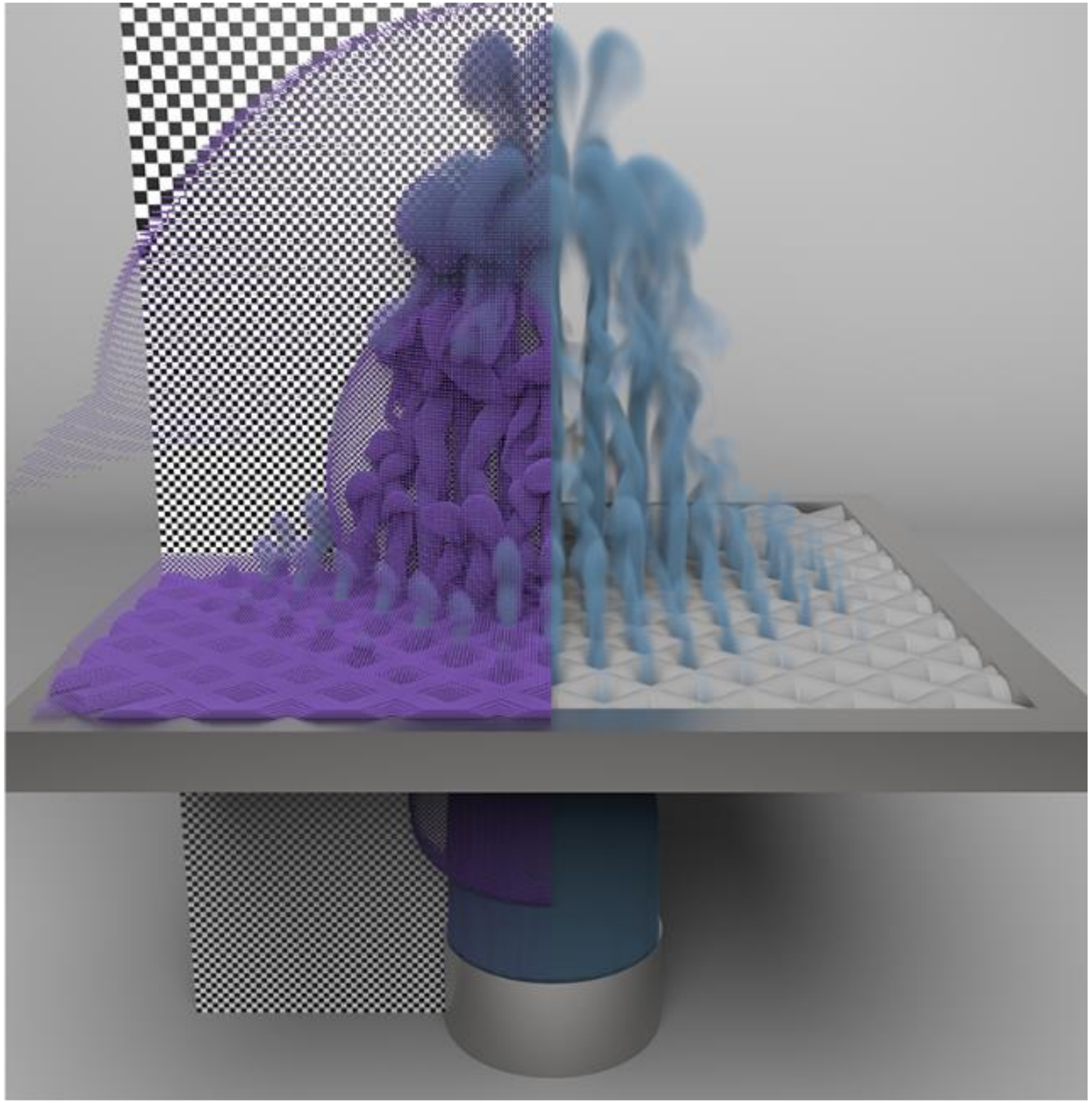


Grid

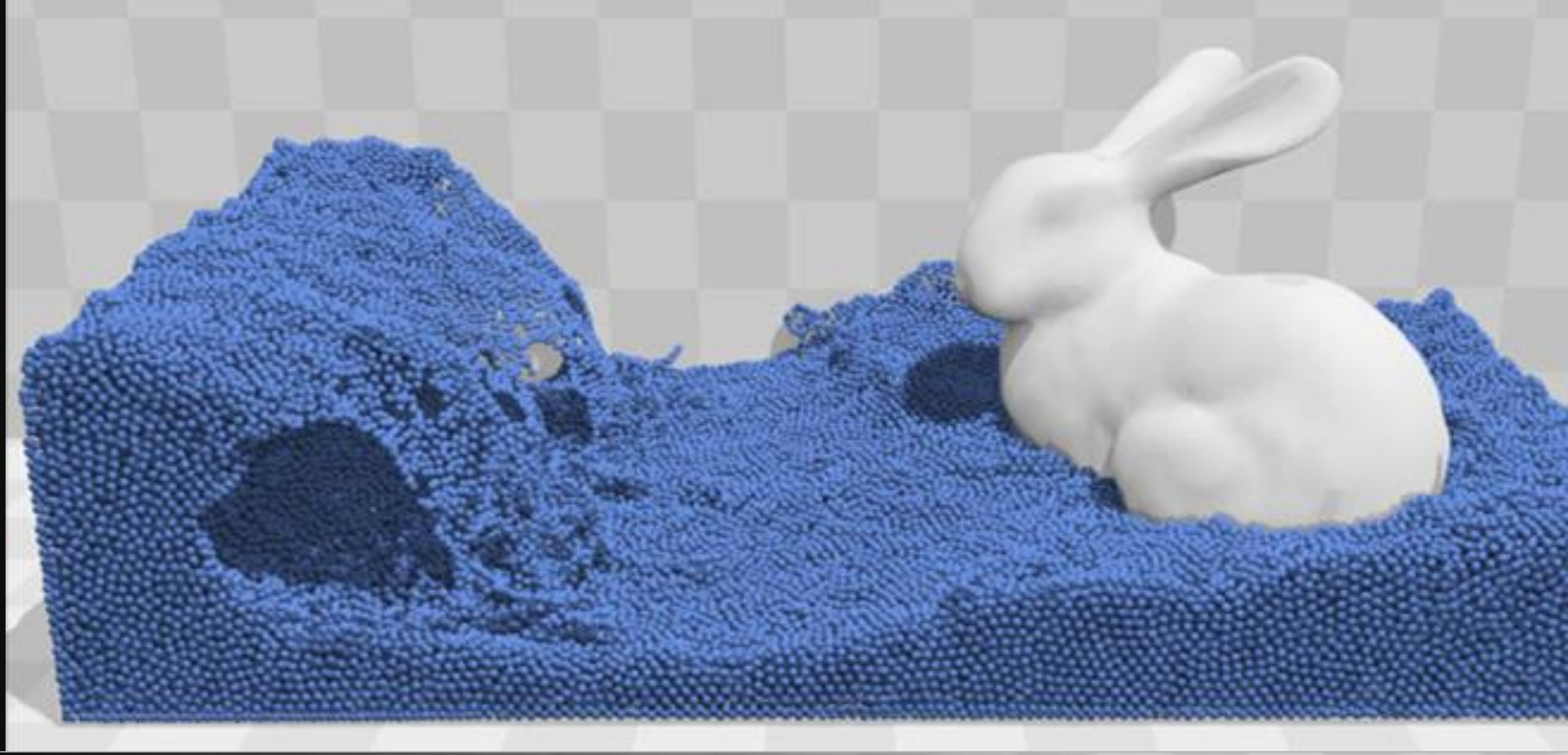
Mesh



Grid

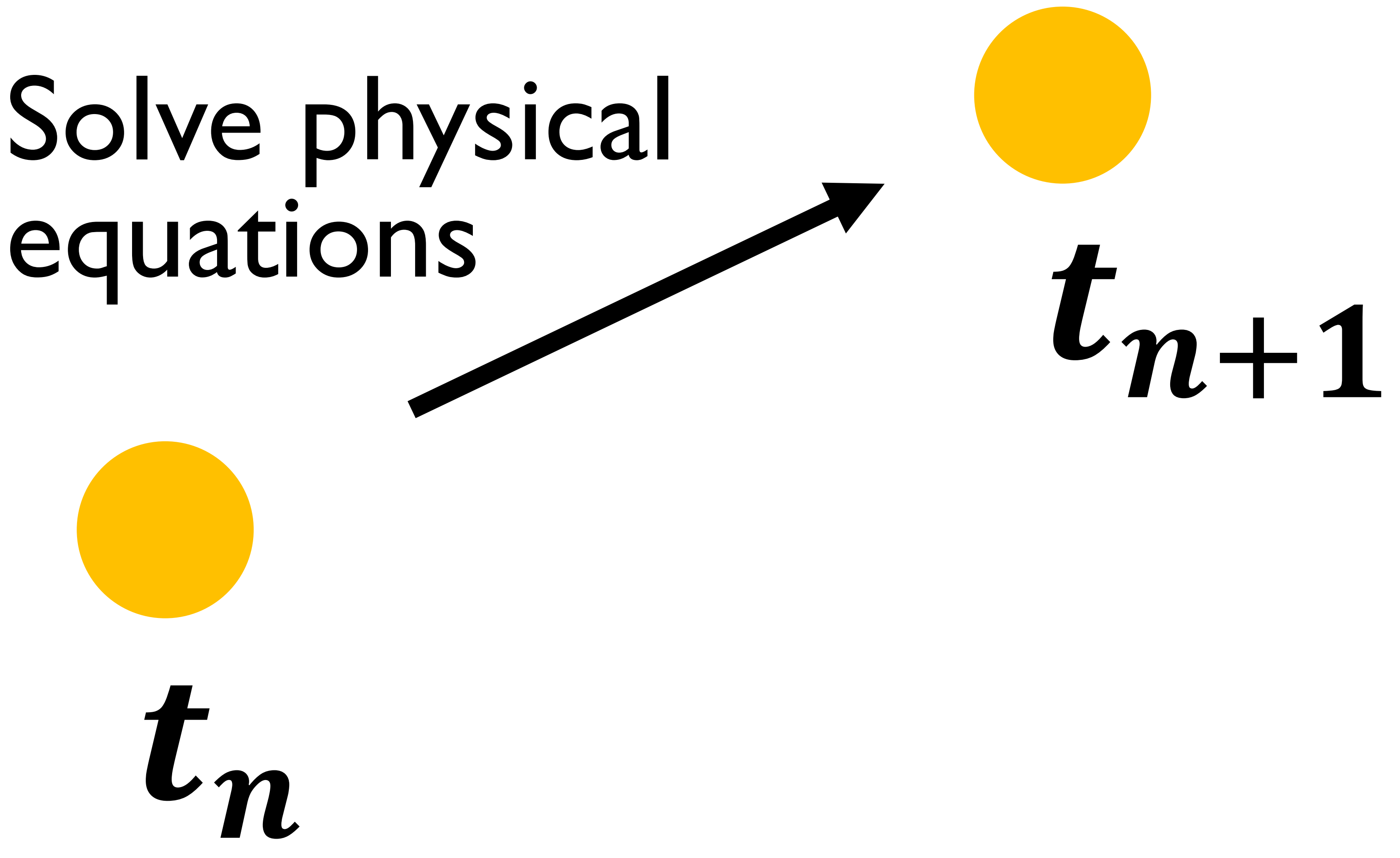


Particles



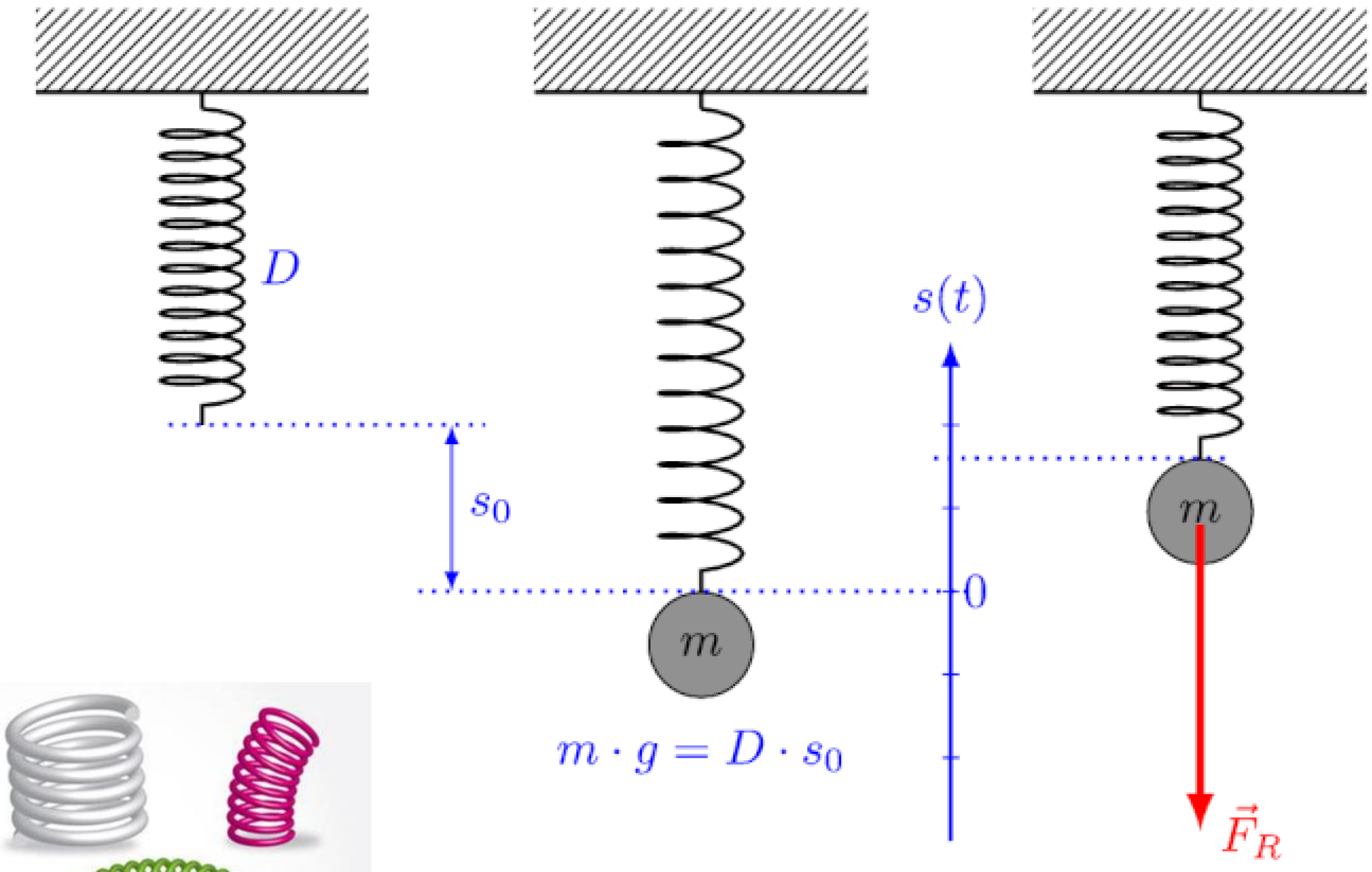
Temporal Evolution

- Simulate physical properties along the time axis as a sequence of states $p(t_0), p(t_1), p(t_2), \ldots$
- Update the next state based on the current state(s) by solving physical equations



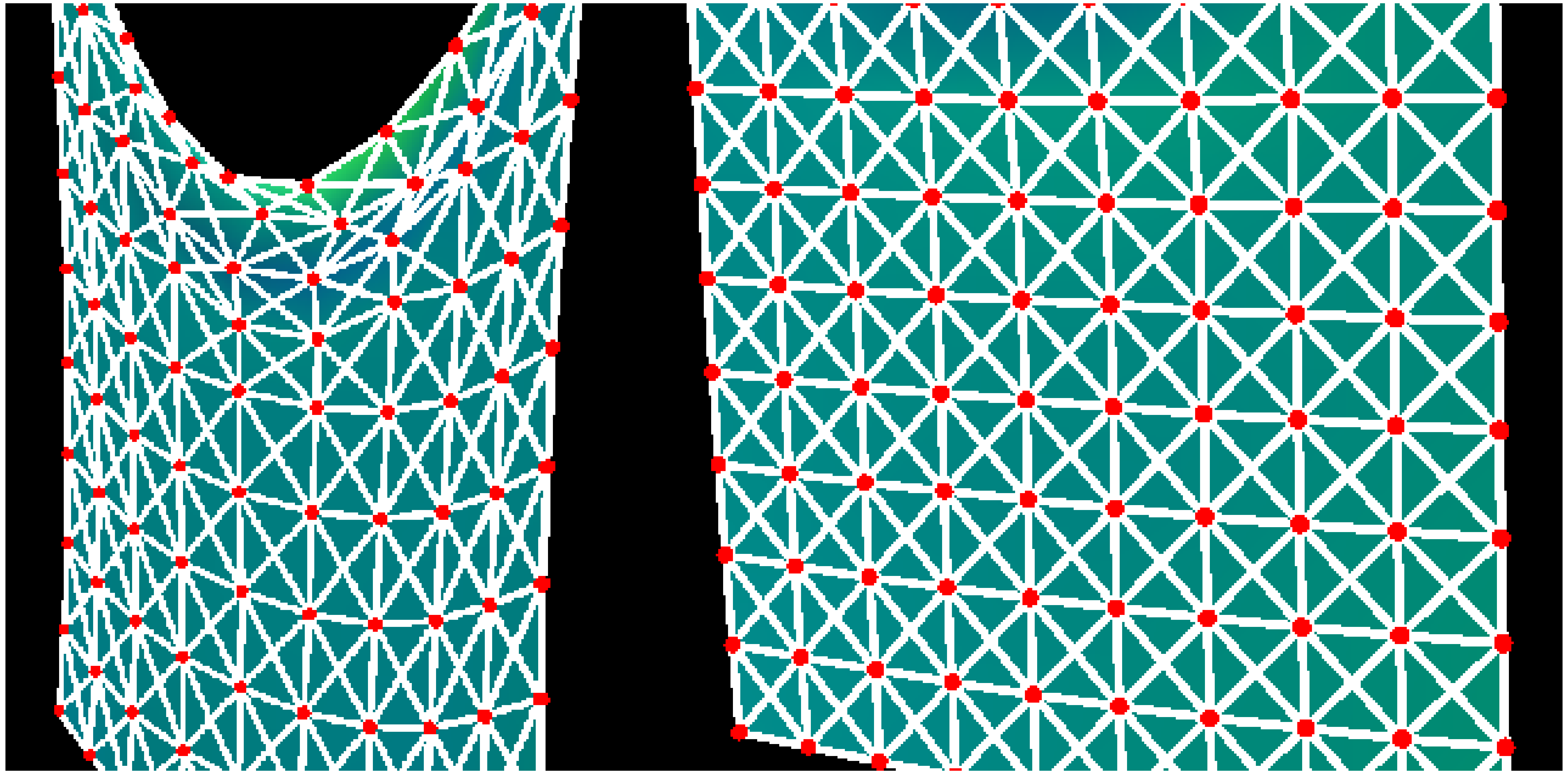
Mass-Spring Model

Mass-Spring Model



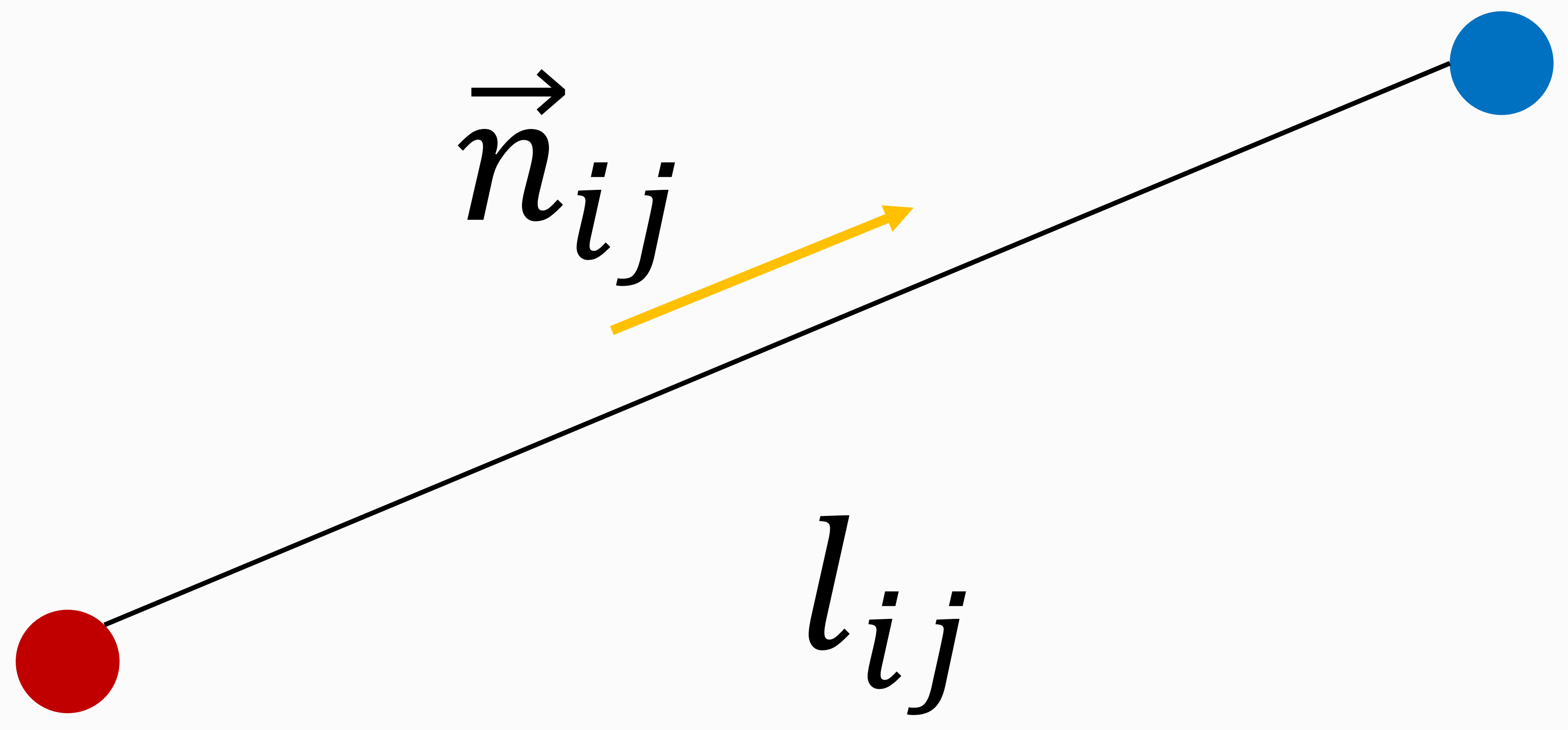
Mass-Spring Model

- The most popular simulation model for modeling soft body , cloth, and hair
- Key Idea: add a spring on each edge of the triangle mesh
- Simple, fast, and easy-to-implement



Physics Intuition of Spring

$$e = \frac{k_s}{2}(l - l_0)^2 = \frac{k_s}{2}\left(\left|\vec{l}_{ij}\right| - l_0\right)^2 = \frac{k_s}{2}\left(\sqrt{\left(\vec{x}_i - \vec{x}_j\right)^2} - l_0\right)^2$$

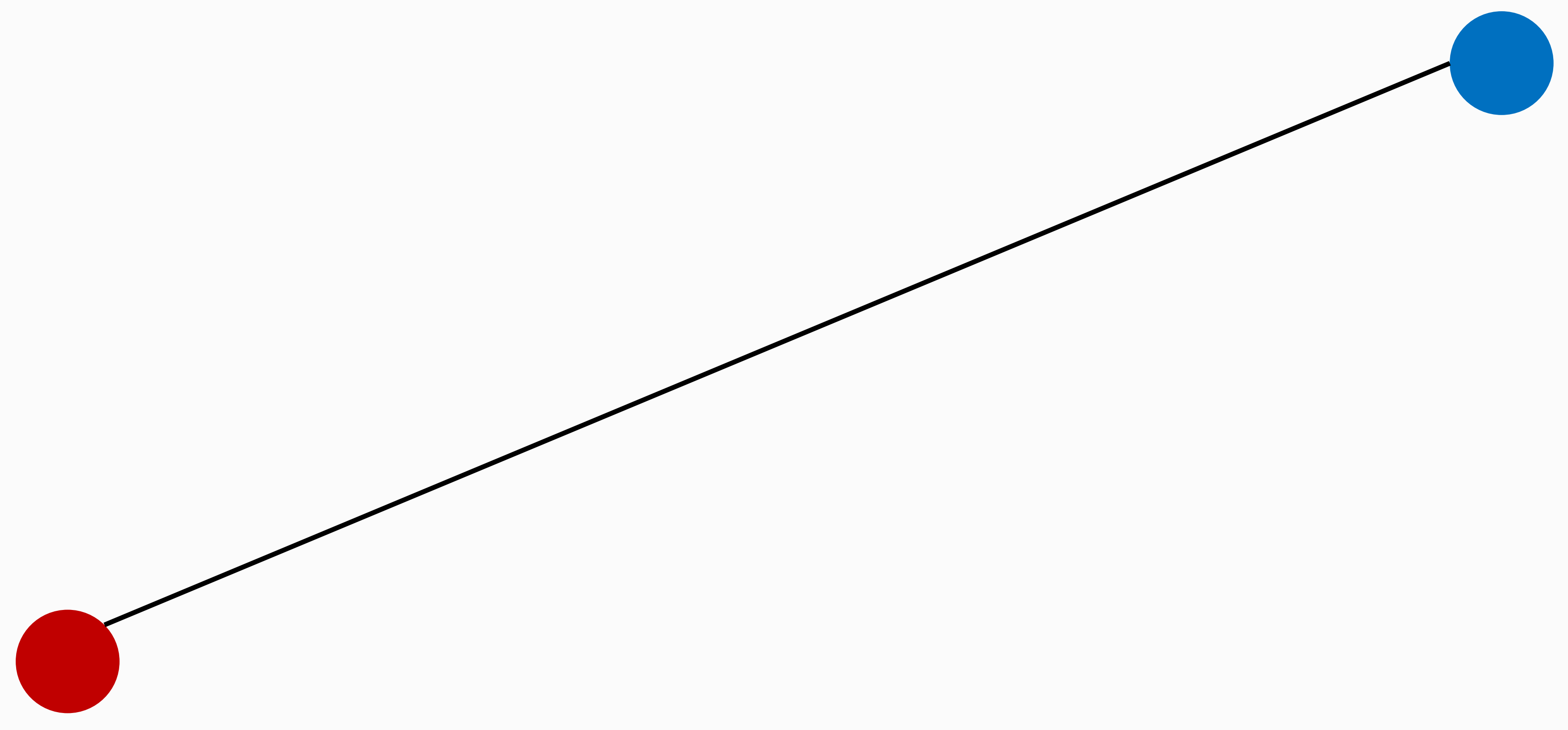


Spring Force

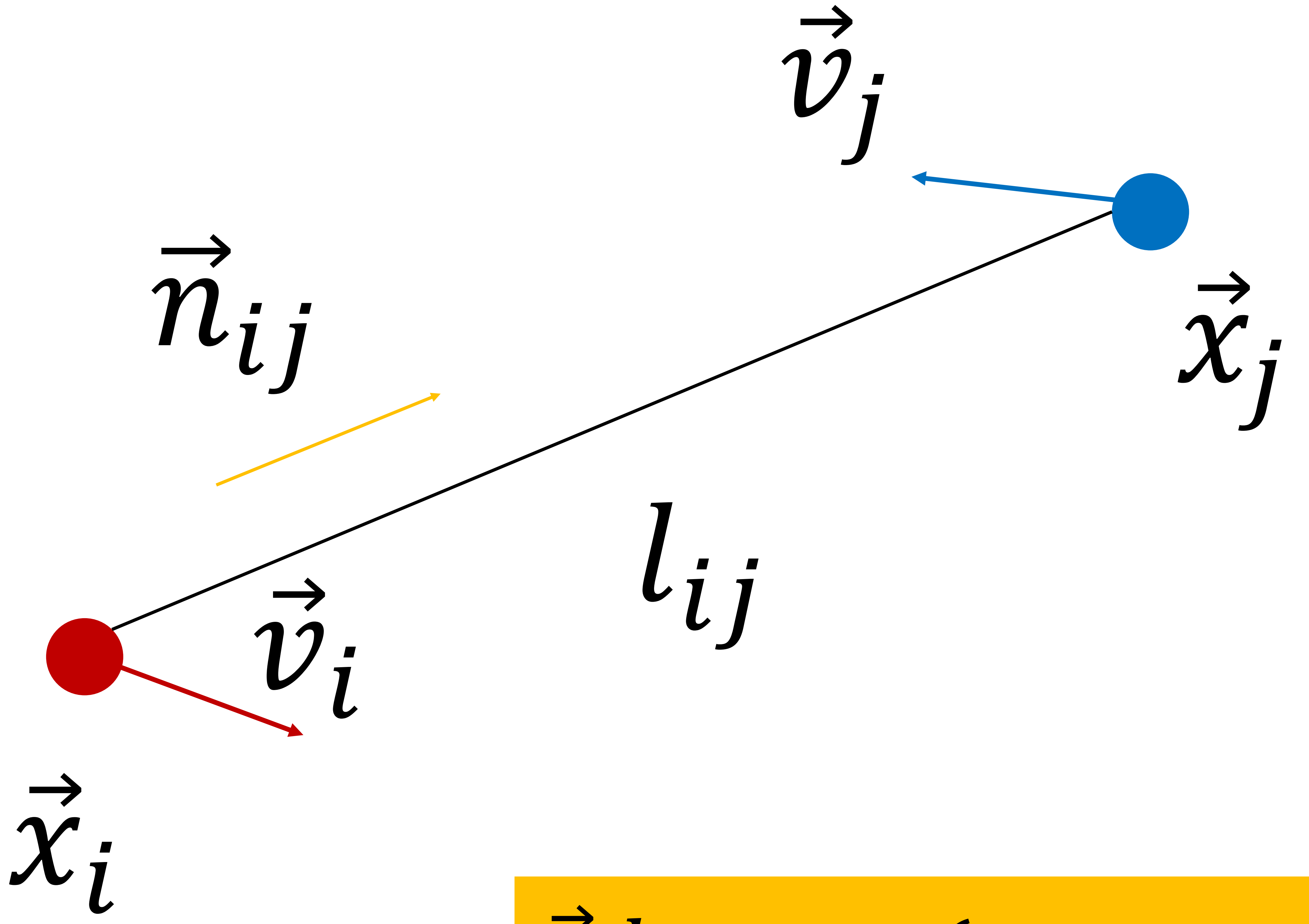
For the two connected particles, the spring will:

Keep the relative distance constant

Keep the relative velocity along the spring direction to be zero



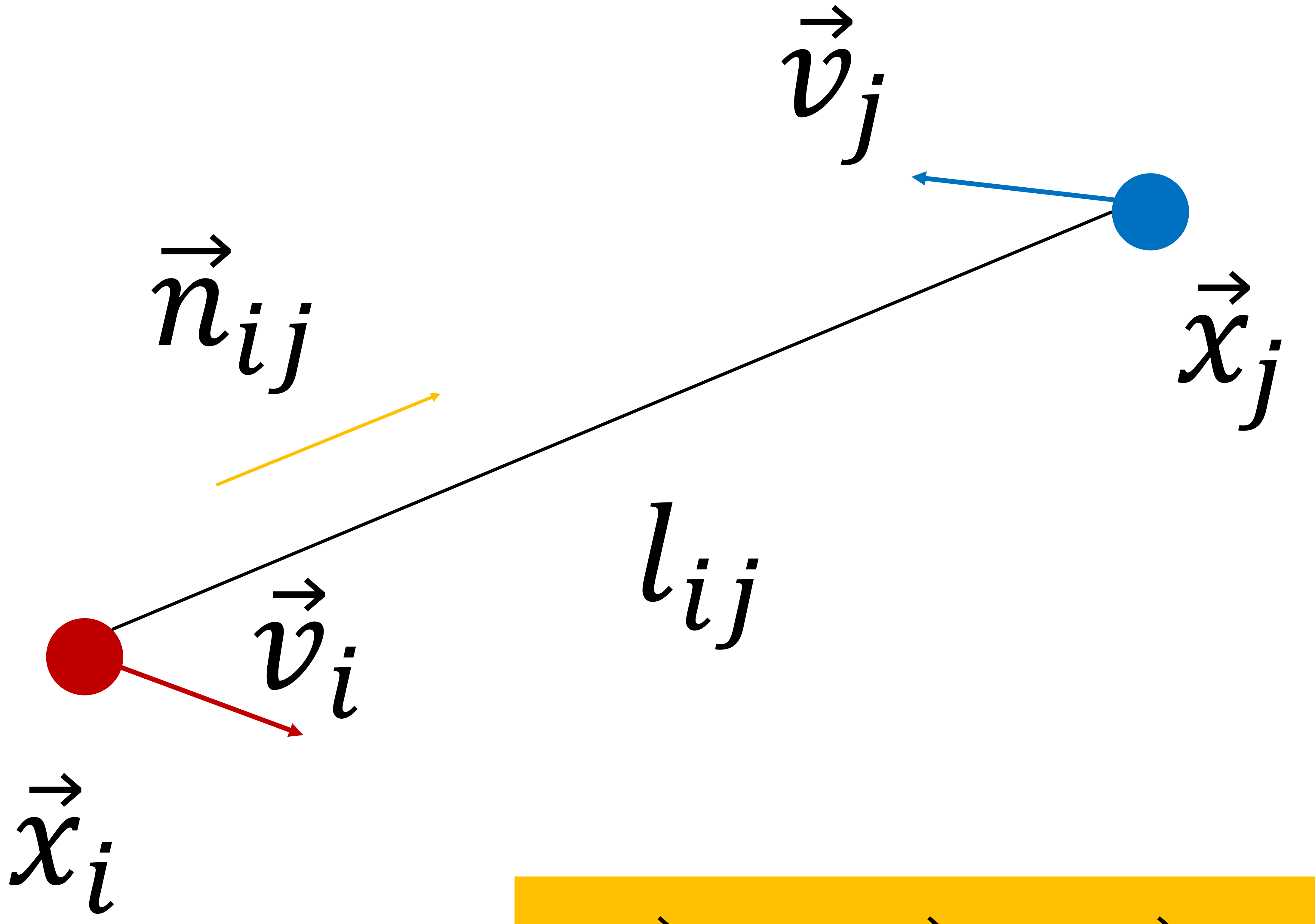
Physical Model: Damping Force



$$\vec{f}_i^{\,d} = k_d(\vec{v}_{ij} \cdot \vec{n}_{ij})\vec{n}_{ij}$$

$$\vec{f}_j^{\,d} = -\vec{f}_i$$

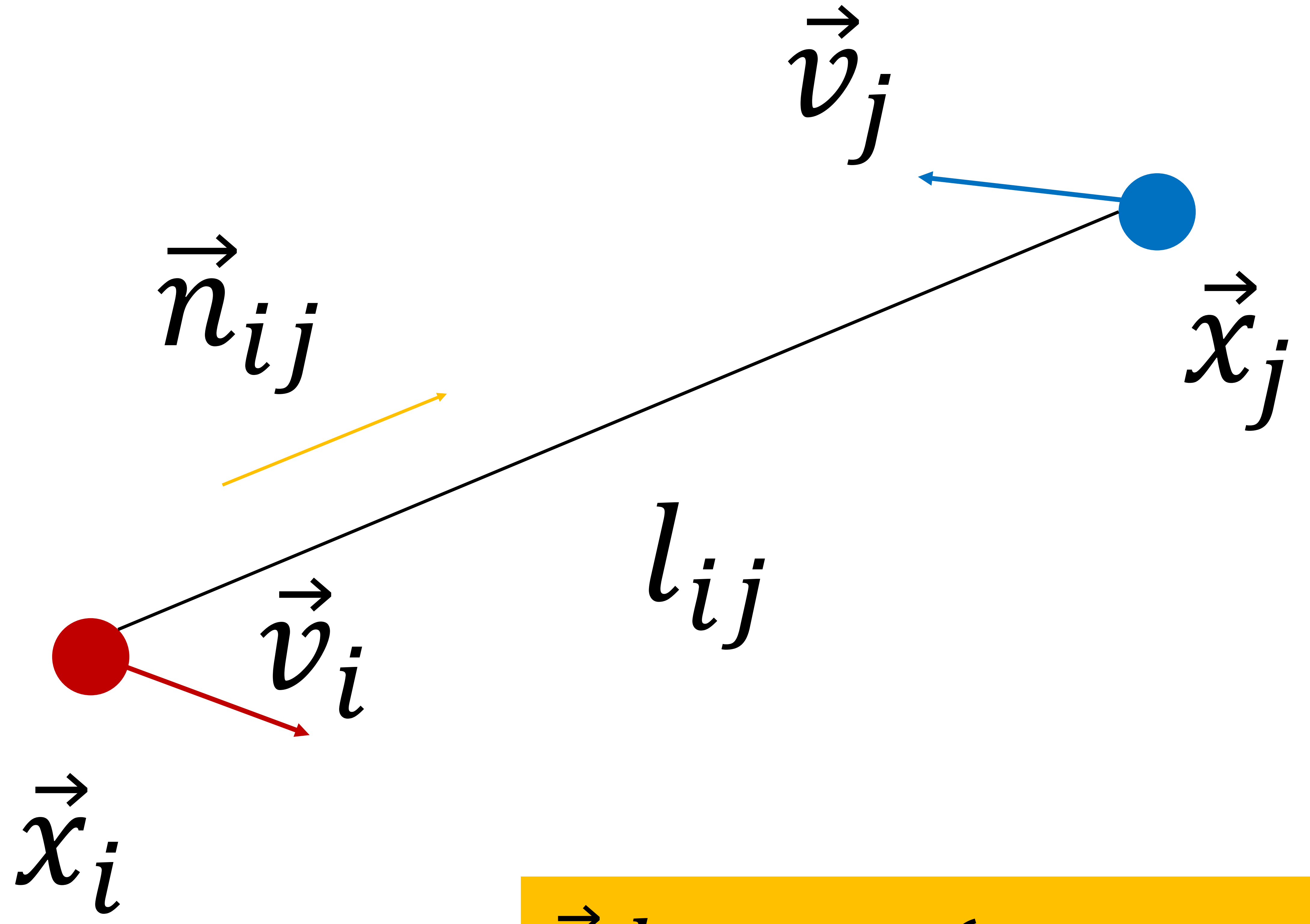
Physical Model: Putting Two Forces Together



$$\vec{f}_i = \vec{f}_i^s + \vec{f}_i^d$$

$$\vec{f}_j^d = -\vec{f}_i$$

Physical Model: Putting Two Forces Together



$$\vec{f}_i^d = k_d(\vec{v}_{ij} \cdot \vec{n}_{ij})\vec{n}_{ij}$$

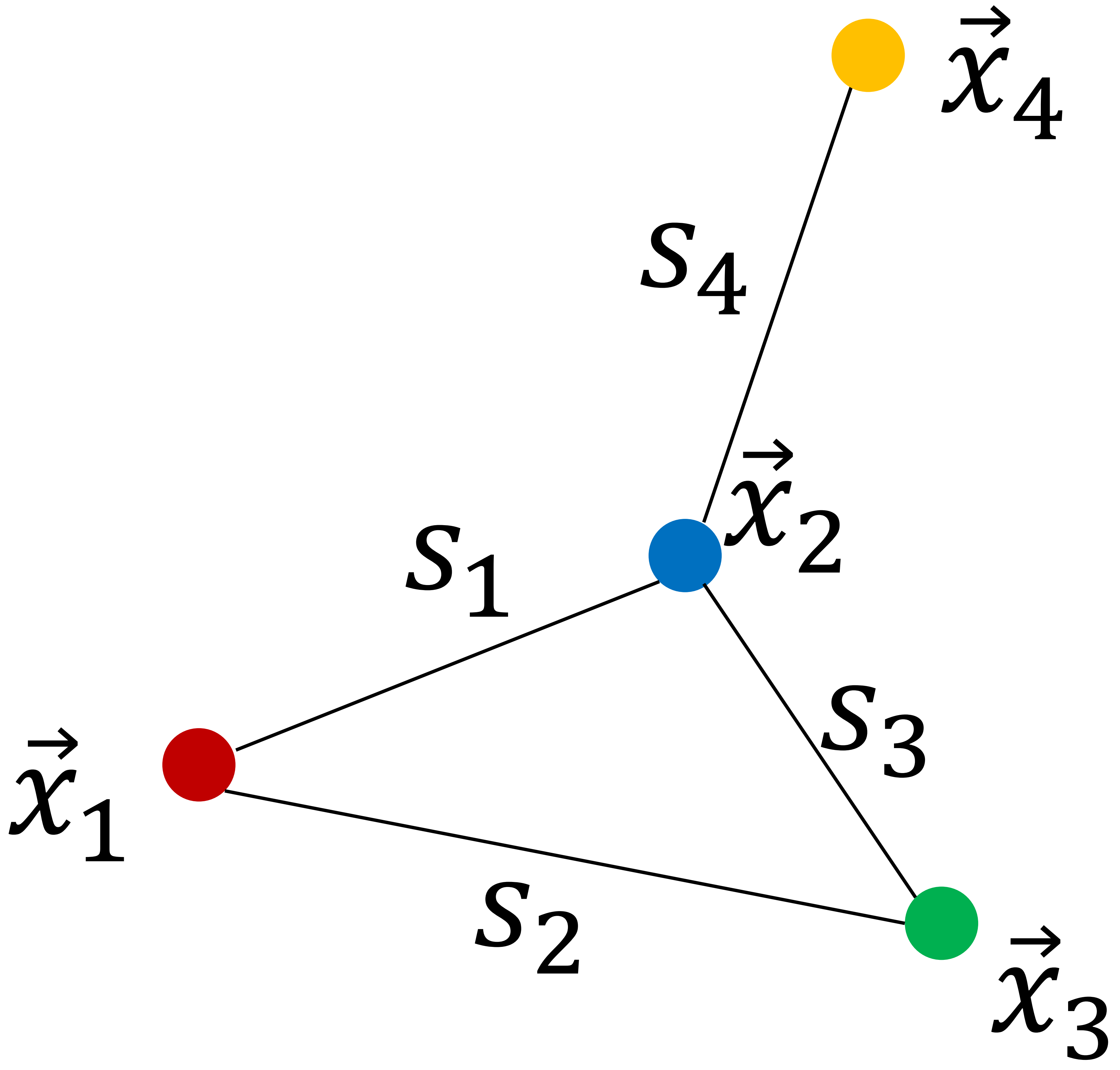
$$\vec{f}_j^d = -\vec{f}_i$$

Simulation Step

- Accumulate forces on a particle from all its incident springs

1. Set force on each particle to be zero
2. For each spring s connecting x_i and x_j:

   - Calculate spring force f_{ij}
   - Add f_{ij} to force on x_i
   - Add $-f_{ij}$ to force on x_j

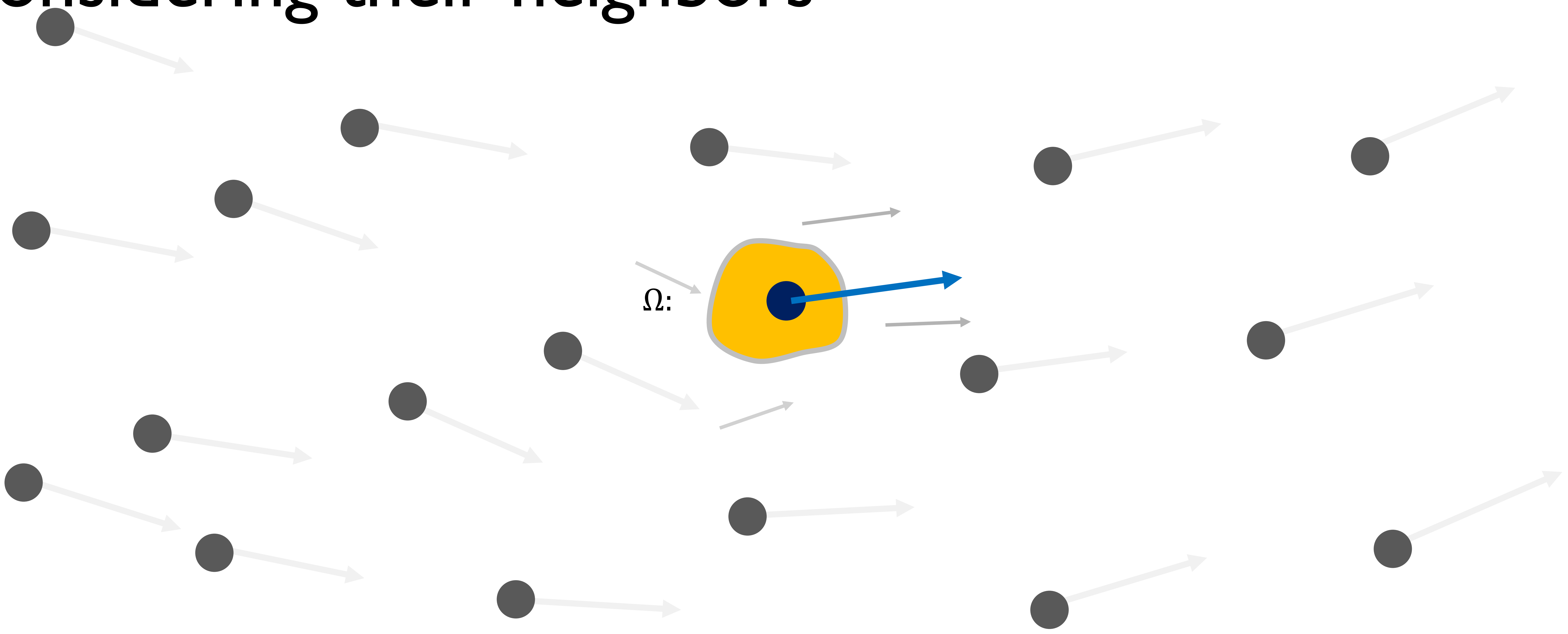


curly hair
(100K full strands, 250 guide strands, 45ms / frame)

Particle Fluid Model

Fluid Particle Model

- Each fluid particle carries a set of physical properties, including position, velocity, mass, density, pressure, etc.
- These properties will move with the particles and updated by considering their neighbors



Navier-Stokes Equation

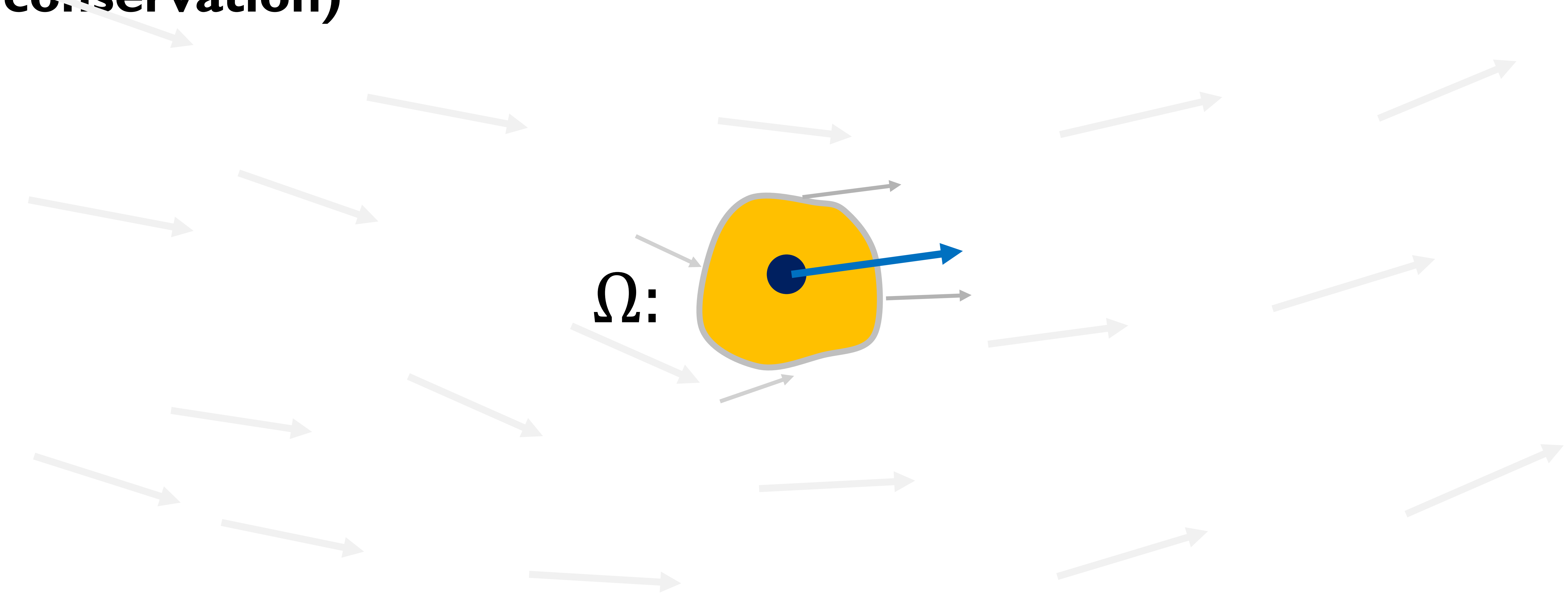
Acceleration | **Pressure Force** | **Viscosity Force** | **External Forces**

Momentum:

$$\rho \frac{D\vec{u}}{Dt} = -\nabla p + \mu \nabla \cdot \nabla \vec{u} + \rho \vec{g}$$

Incompressibility:
(mass conservation)

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



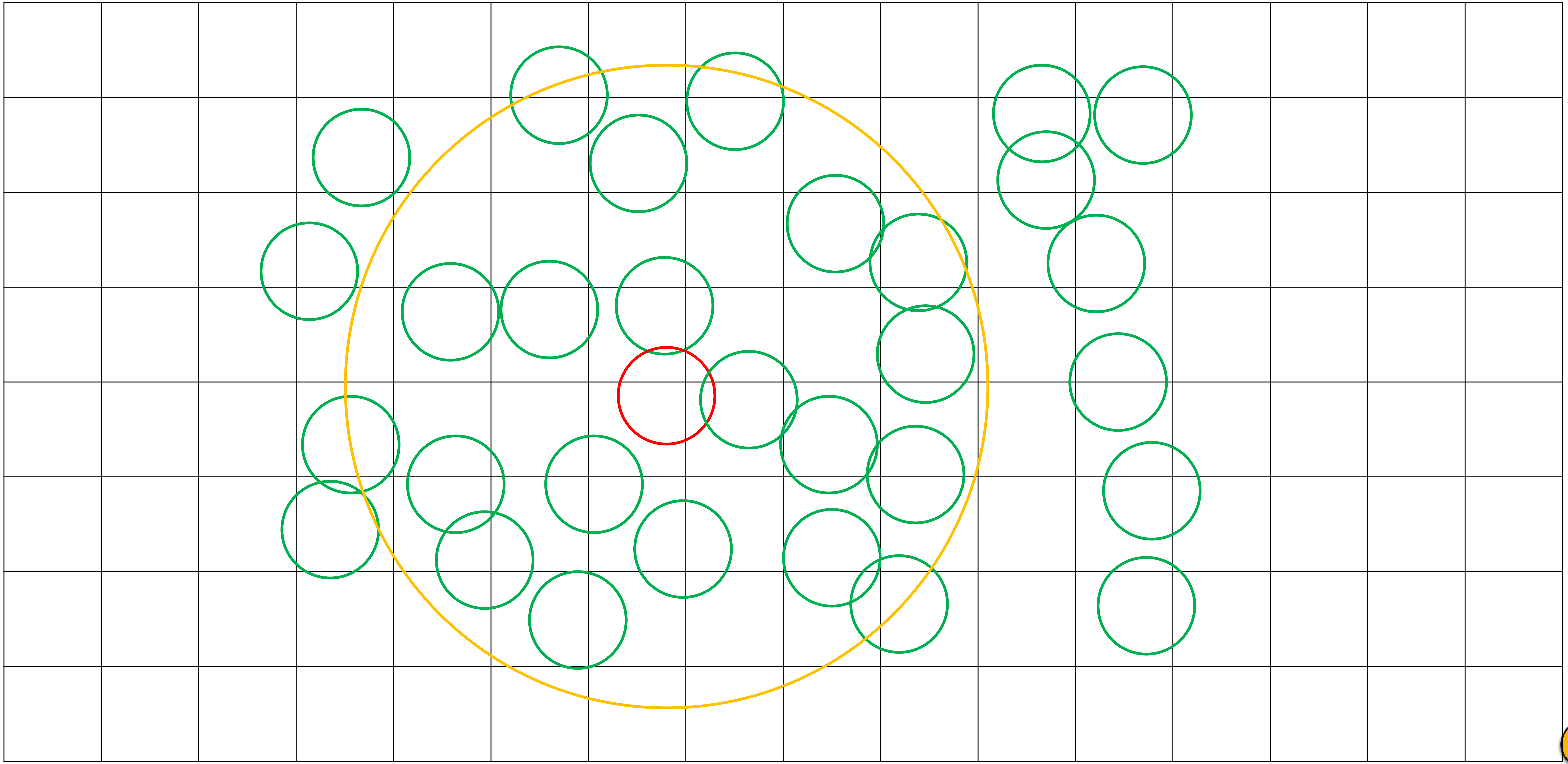
Pressure Force

- $\boldsymbol{f}_{pressure} = -\nabla p$
  - Push fluid from high-pressure region to low-pressure region

Viscosity Force

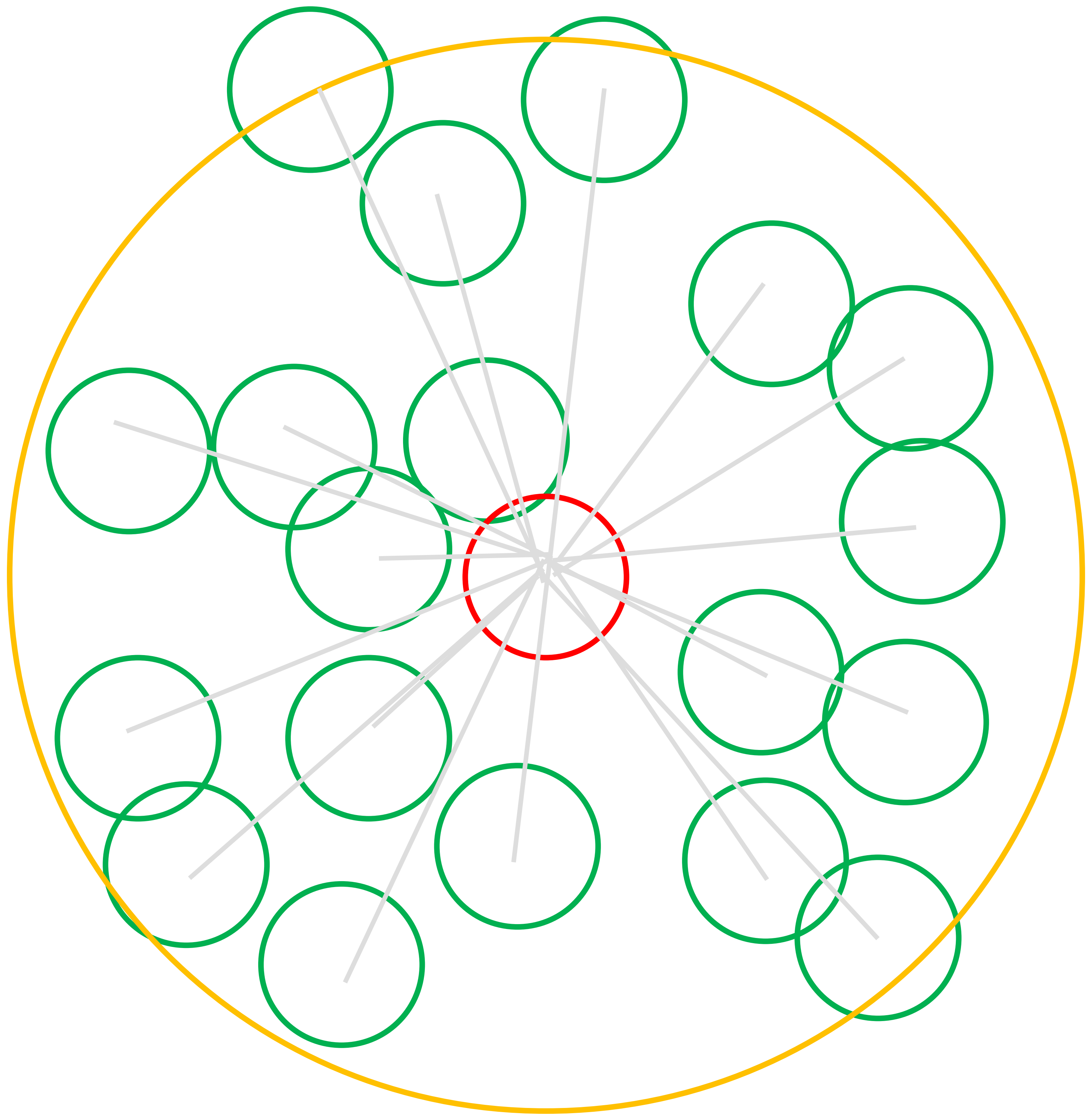
- $\boldsymbol{f}_{viscosity} = \mu \nabla \cdot \nabla \boldsymbol{u}$
  - Oppose the relative motion between fluid particles
  - Think of it as an internal "friction"
  - Smooth the velocity field

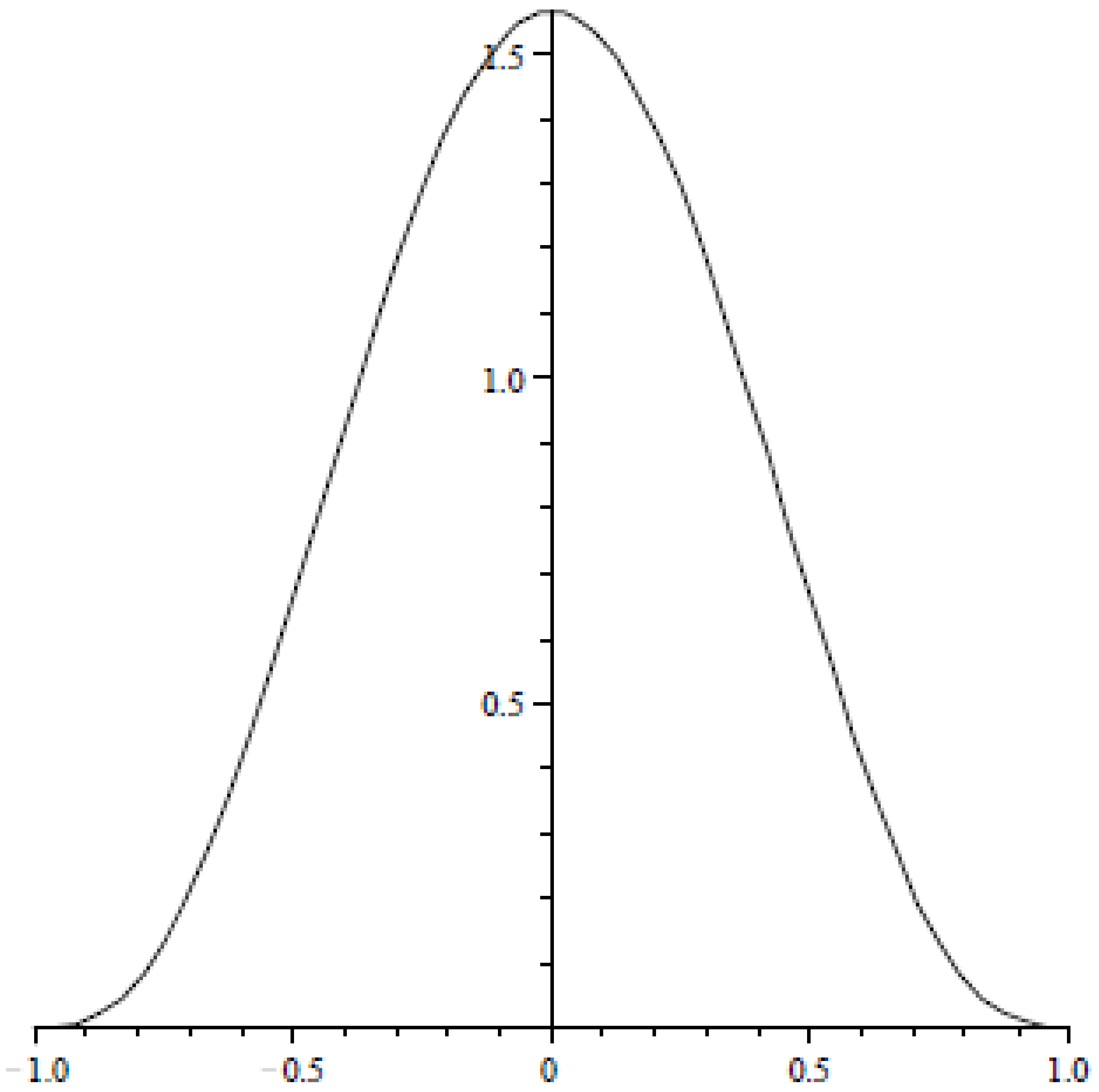
Key Idea: Use a Smoothing Kernel for Particle Force Calculation



Smoothing Kernel Function

- $A_s(\vec{x}_i) = \sum A_j V_j W(|\vec{x}_i - \vec{x}_j|)$
  - A smooth approximation equals the weighted sum of all the neighbors' quantities



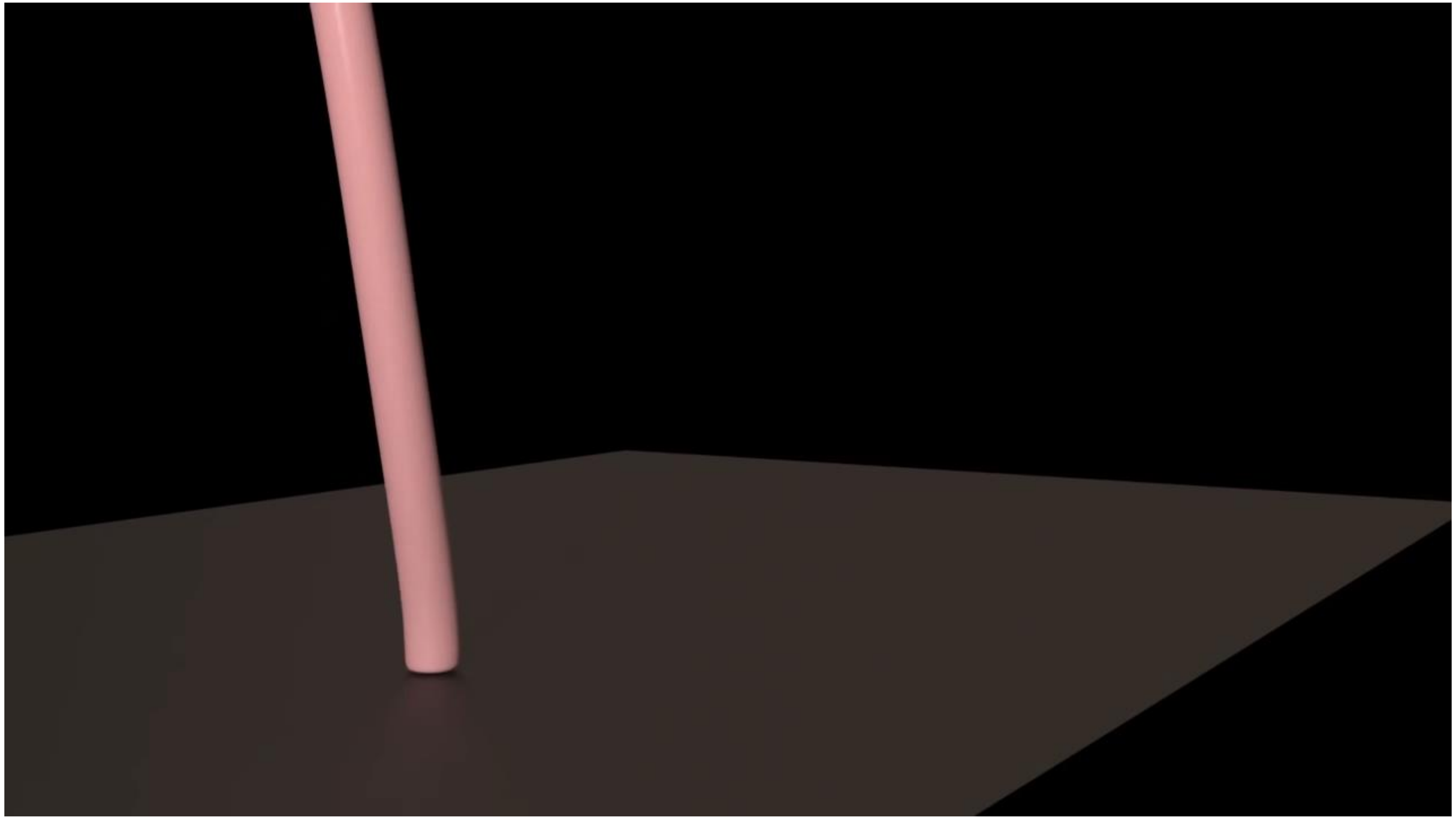


Simulation Loop

- In each time step:
  - Update the grid structure for neighbor inquiry
  - Calculate **density** for each particle
  - Calculate **pressure** for each particle
  - Calculate **pressure forces** for each particle
  - Calculate **viscosity forces** for each particle
  - Calculate **body forces** for each particle
  - Update particle position

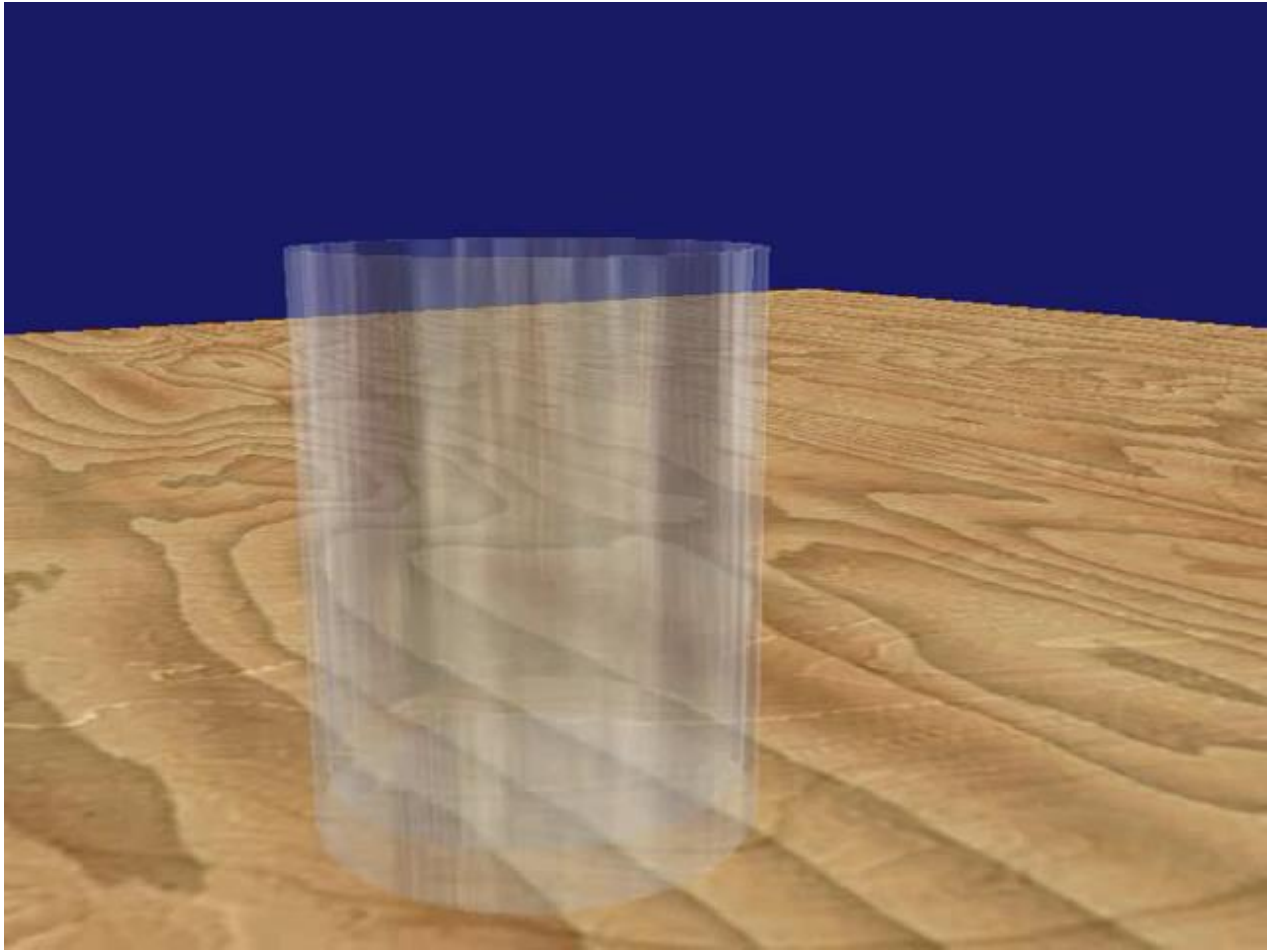
Large-Scale Particle Fluid

Viscous Particle Fluid

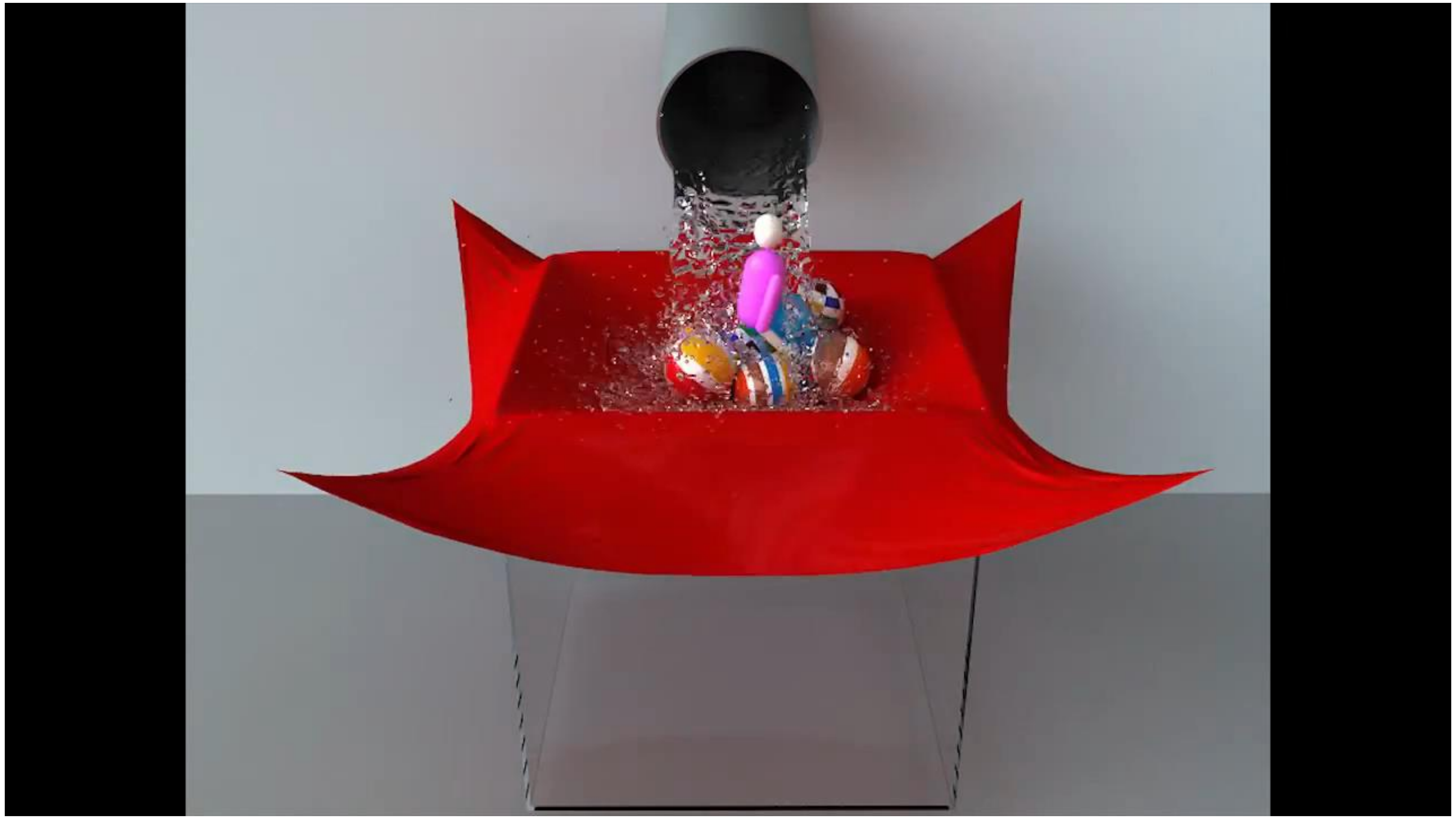


Surface Tension Particle Fluid

Real-Time Simulation

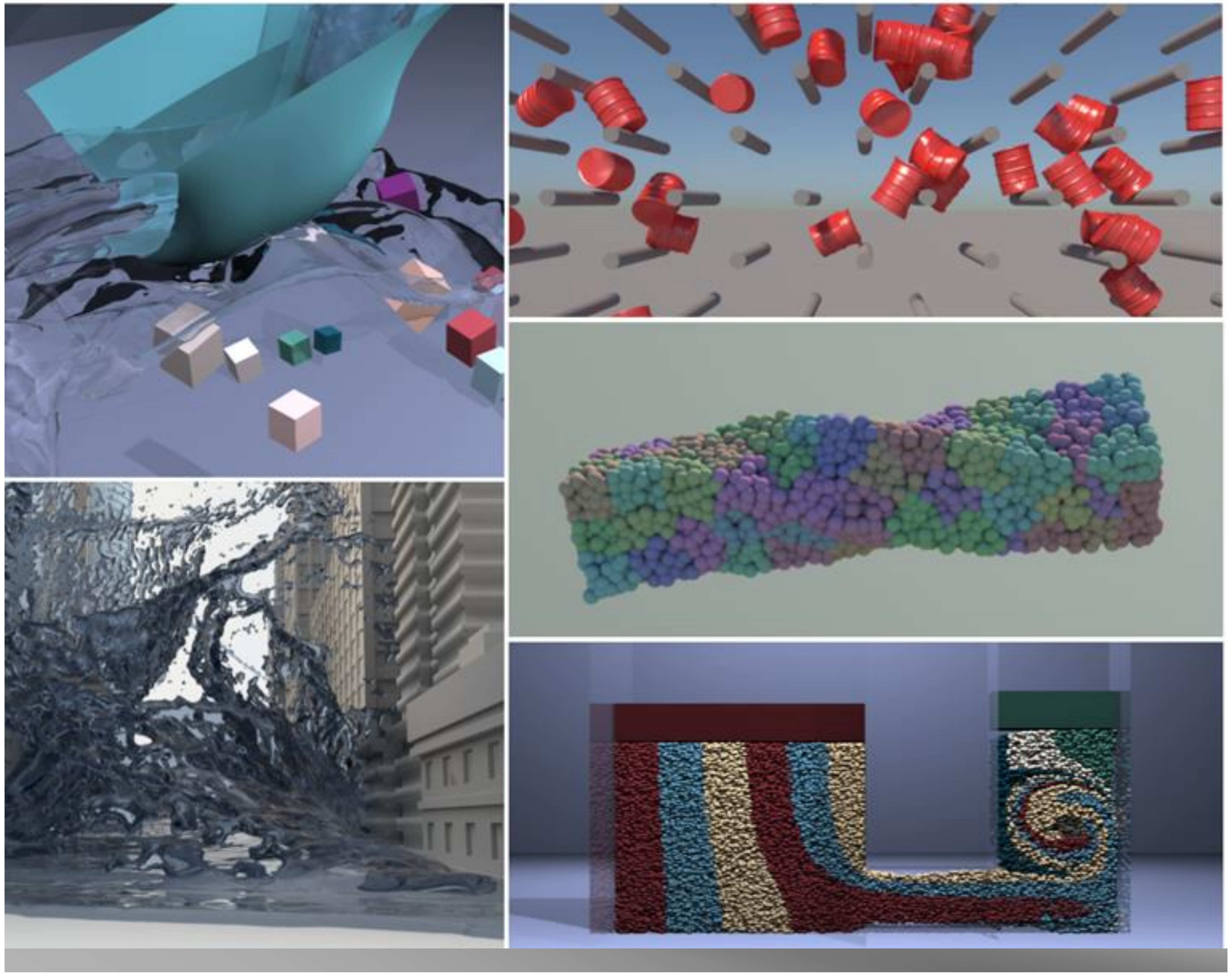


Solid-Fluid Interaction with Particles



Key Takeaways for Physically-based Animation

- Solve a physical equation to drive particle motion
- Interaction among particle neighbors is important
- Mass-spring model for solid simulation
- Smoothed particle model (SPH) for fluid simulation



Reading Materials

- **An introduction to physics-based animation**
  - https://education.siggraph.org/cgsource/content/introduction-physics-based-animation
- **Fluid Simulation for Computer Animation**
  - https://www.cs.ubc.ca/~rbridson/fluidsimulation/