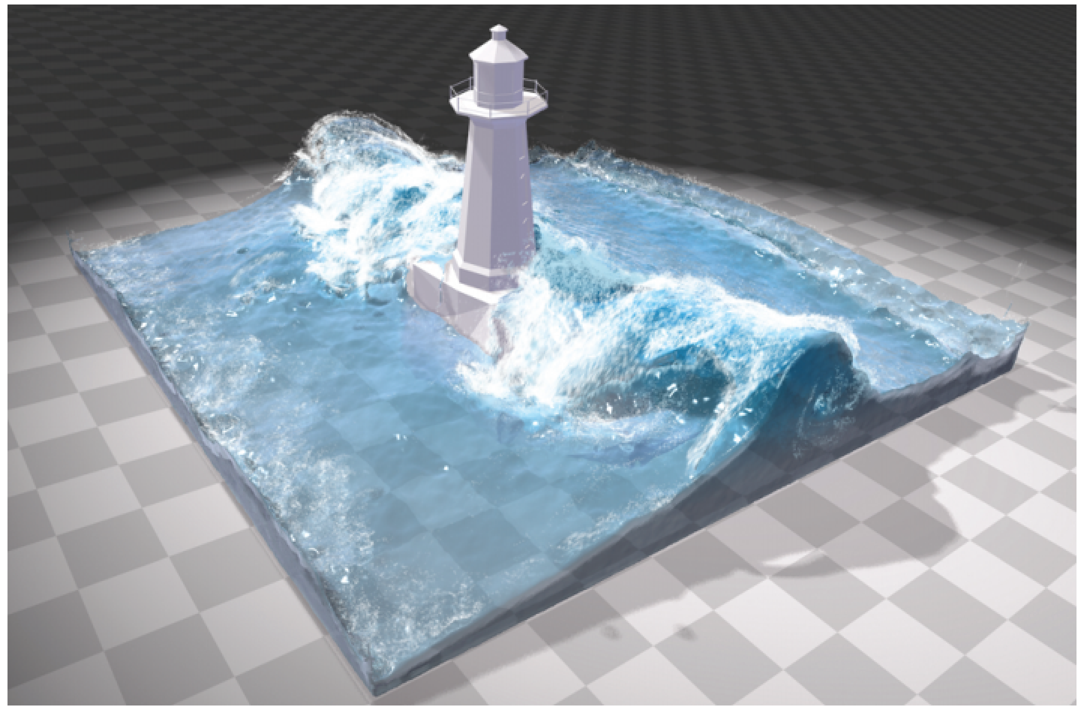
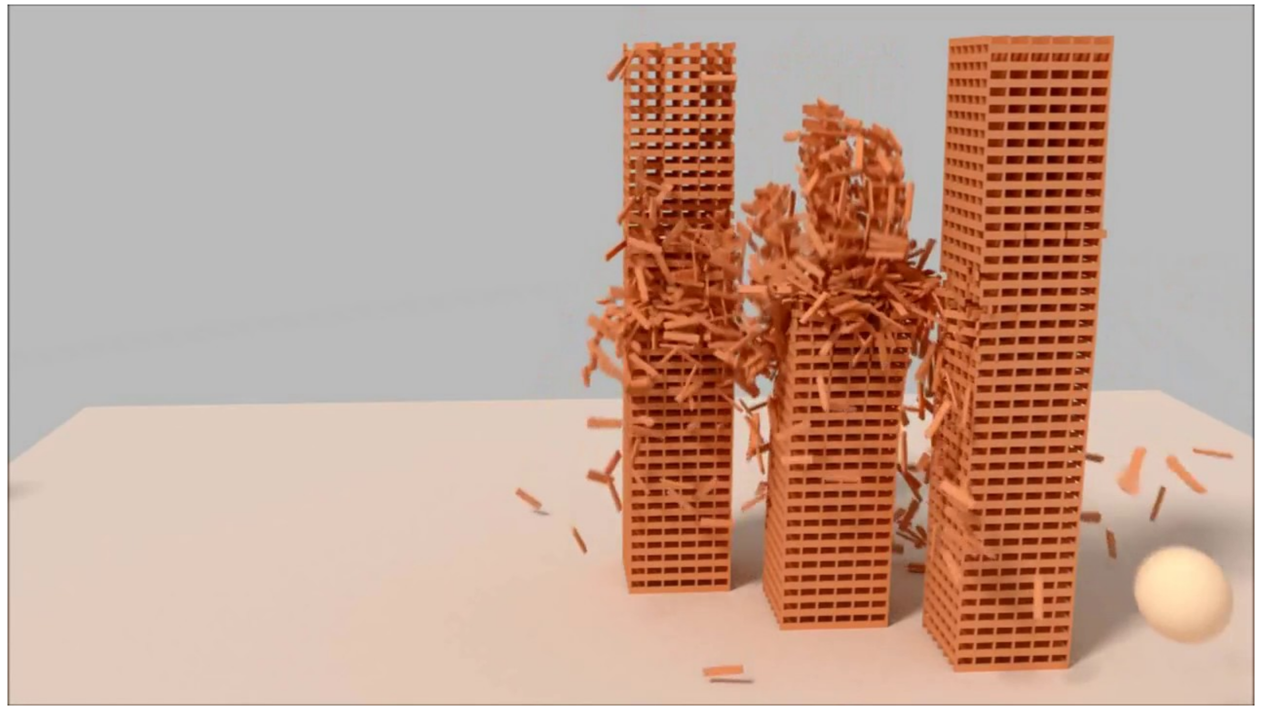
Physically based simulation - Models

When physically based simulation is needed

- Accurate dynamics
- Teadious to model by hand or procedurally
  - Multiple interacting elements: ex. Multiple collisions: rigid bodies, hairs, etc.
  - Complex animated geometry: Cloths, fluids





General methodology

1. Description of the system

Describe system by some parameters (positions, speed, orientation, etc).

- State of the system is known at time $t = 0$ - *Initial value problem in time*

- State of the system may be constrained in space - *Boundary value problem in space*

2. Evolution

Link the evolution of the system to forces or constraints using dynamic principles and conservation laws

$\Rightarrow$ Differential equation

3. Numerical Solution

Solve the differential equation using numerical iterative approaches.

Note: Fundamentally different that direct approach controling the trajectories at key-frames

- The system is set at an initial step

- We let the numerical solution build the space-time trajectory for us

(+) Allows to model complex behavior

(-) Lack of control on the result

Fundamental models

1- Particles
2- Rigid bodies
3- Continuum models

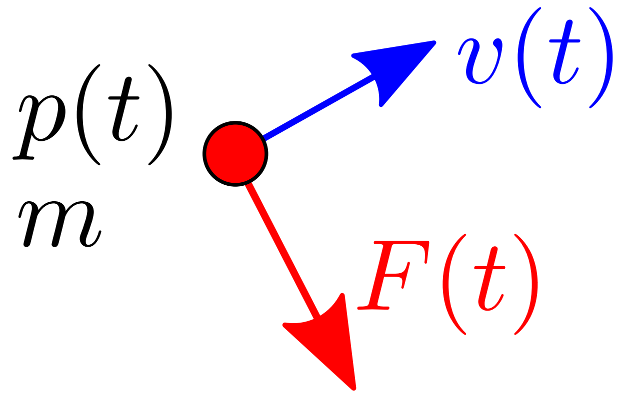
Physically-based particle system

1. Description

Particle is fully described by: Position p, Velocity v, Mass m

Fundamental quantities: position and linear momentum $P = m\,v$

Linear Momentum preserved in isolated system



2. Evolution

- Fundamental principle of dynamics

Force applied on particle $F(p, v, t)$

$$\begin{cases} p'(t) = v(t) \\ P'(t) = m\,v'(t) = F(p, v, t) \end{cases}$$

- Conservation of energy (ex. kinetic energy $(1/2\,m\,v^2)$+potential energy = const, etc.)

- Lagragian, or Hamiltonian (reduced coordinates)

Physically-based particle system

3. Numerical Solution

ODE (Ordinary Differential Equation) formulated as an **Initial Value Problem**

ex. $\left\{ \begin{array}{l} p'(t) = v(t) \\ mv'(t) = F(p, v, t) \end{array} \right.$, with $v(0) = v_0,\ p(0) = p_0$

- Discretize in time $t^k = k\,h, \quad h = \Delta t =$ time step.

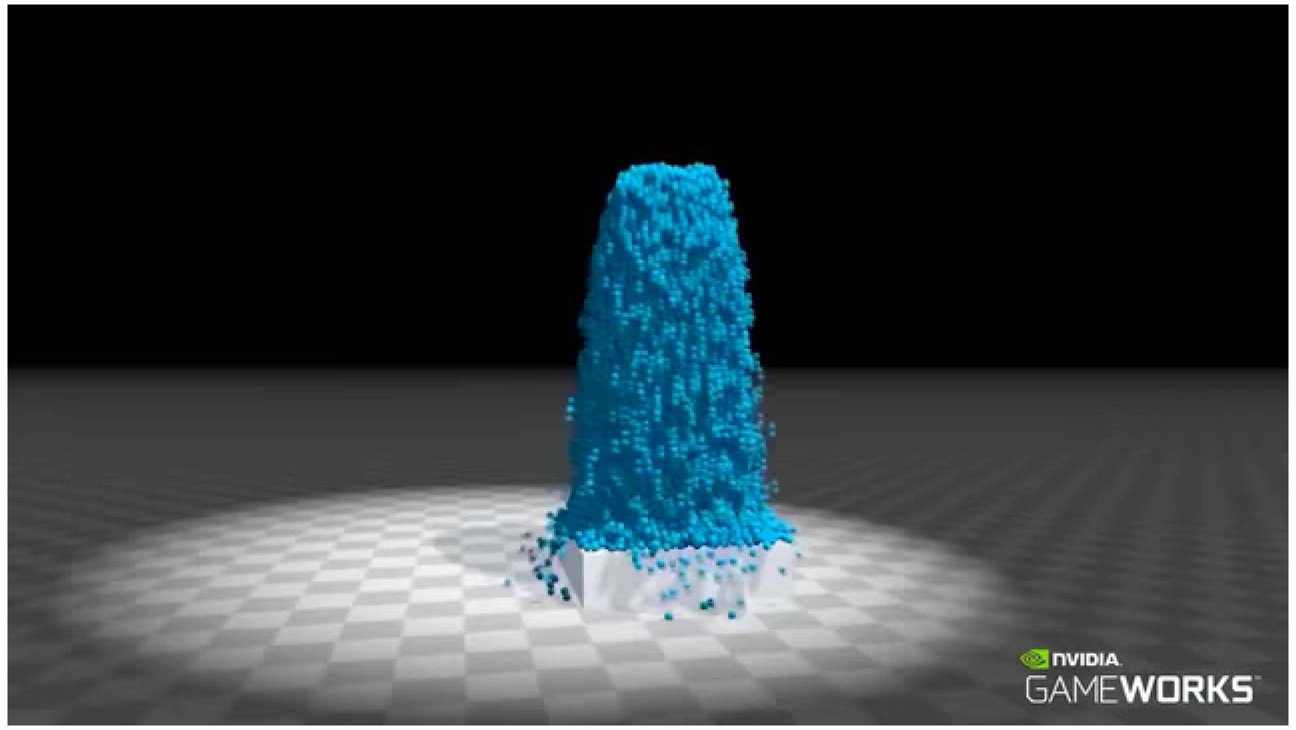
  $\Rightarrow$ Build a discrete numerical solution $p^k = p(t^k), v^k = v(t^k)$.

- We can consider initially the following iterative scheme

$$\left\{ \begin{array}{l} v^{k+1} = v^k + h\,F(p^k, v^k, t^k) \\ p^{k+1} = p^k + h\,v^{k+1} \end{array} \right.$$

Simple to implement, reasonably OK for simple examples (more details later).

Physically-based particle system



Pro

(+) Simple to implement, and control.

(+) Efficient to compute, scalable

(+) Highly adaptable from simple particle to rigid and deformable models

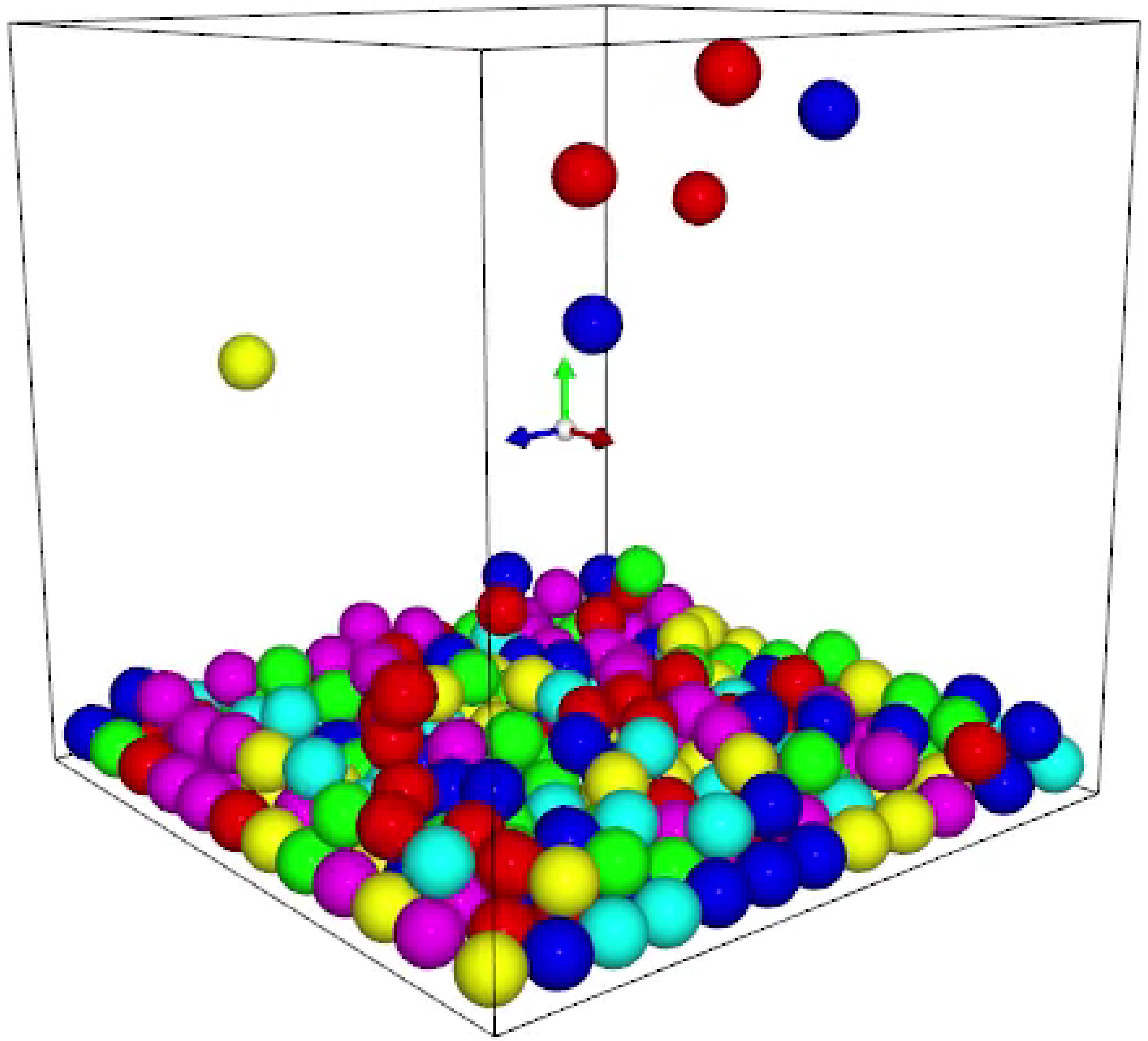
Cons

(-) Limited accuracy - highly simplified model from physical point of view.

$\Rightarrow$ Dominant model in CG production for general purpose deformable model animation.

Common use: Lots of sparsely interacting particles

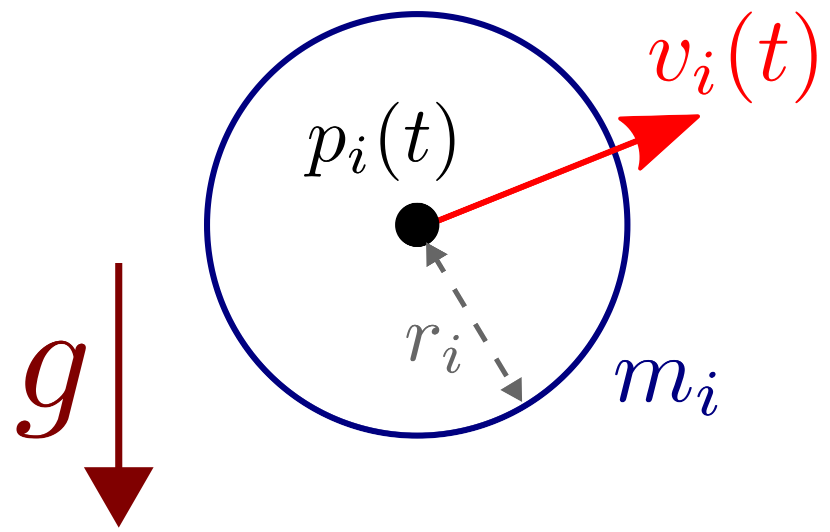
Rigid spheres



System modeling

Particles modeling the center of hard spheres.

- Spheres can collide with surrounding obstacles
- Spheres can collide with each others



- *System*: N particles with position p_i, velocity v_i, mass m_i, modeling a sphere of radius r_i.
  - Initial conditions $p_i(0) = p_i^0$, $v_i(0) = v_i^0$

- *Forces*: Single gravity forces $F_i = m_i\, g$. Collisions handled by *impulses*.

- *Temporal evolution*: Fundamental principle of dynamics $v_i(t) = p_i'(t)$, $v_i'(t) = g$

- *Numerical solution*

$$\left\{\begin{array}{l} v^{k+1} = v^k + h\, g \\ p^{k+1} = p^k + h\, v^{k+1} \end{array}\right.$$

Collision with a plane

Plane $\mathcal{P}$: parameterized using a point a and its normal n.

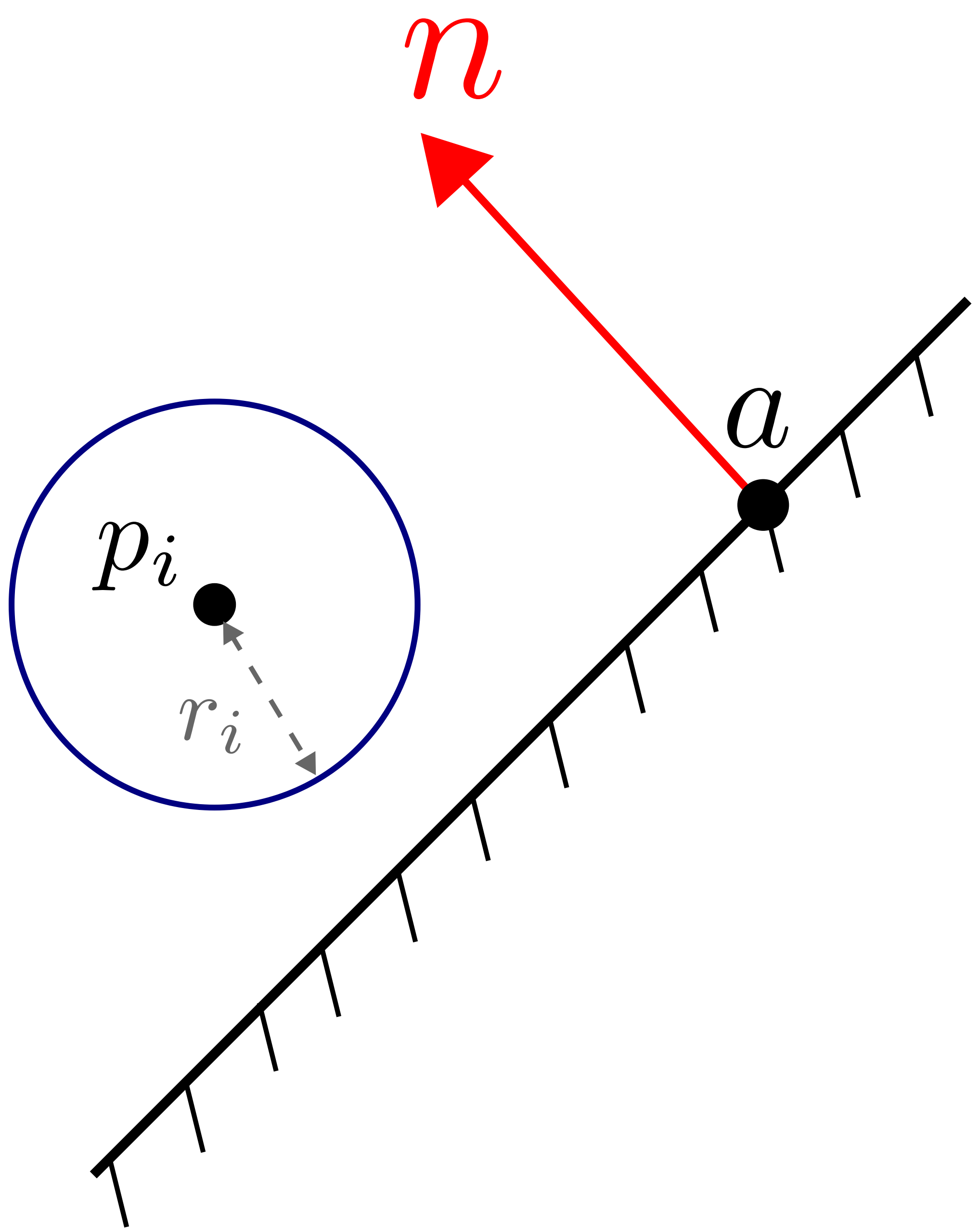
$$\{p \in \mathbb{R}^3 \in \mathcal{P} \Rightarrow (p-a) \cdot n = 0\}$$

- Sphere above plane : $(p_i - a) \cdot n > r_i$
- Sphere in collision: $(p_i - a) \cdot n \leq r_i$

- Collision detection algorithm

```
for(int i=0; i<N; ++i)
{
    float detection = dot(p[i]-a, n);
    if (detection <= r[i])
    {
        // ... collision response
    }
}
```

What should we do when a collision is detected



Collision response with plane

Suppose exact contact: $(p_i - a) \cdot n = r_i$

Collision response = **Update velocity**

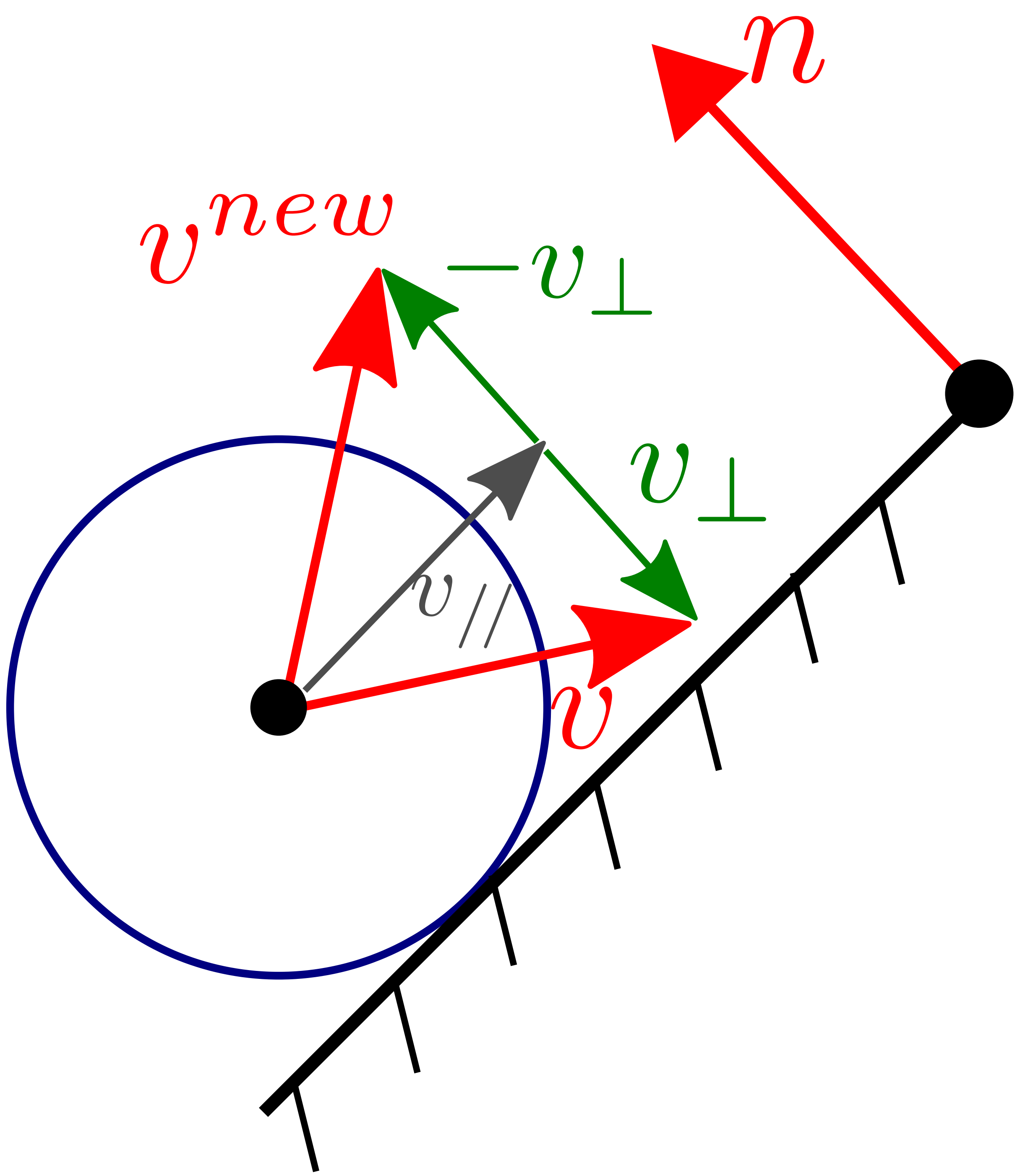
Split $v = v_{//} + v_\perp$

- $v_\perp = (v \cdot n)\, n$
- $v_{//} = v - (v \cdot n) n$

New velocity

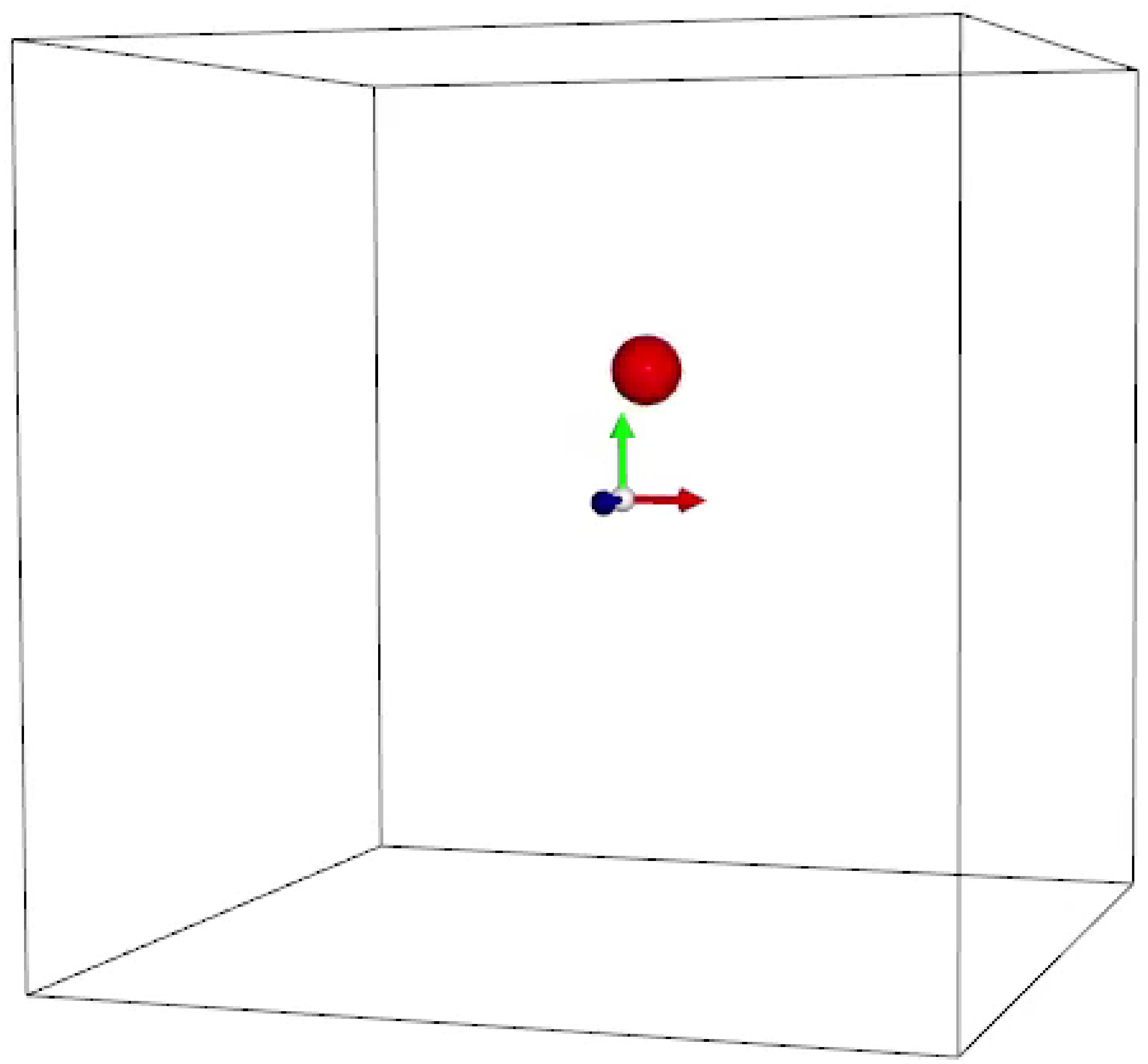
$v^{new} = \alpha\, v_{//} - \beta\, v_\perp$

- $\alpha \in [0, 1]$ Restitution coefficient in $//$ direction (friction)
- $\beta \in [0, 1]$ Restitution coefficient in $\perp$ direction (impact)



Result: Collision response

Applying collision response on speed only

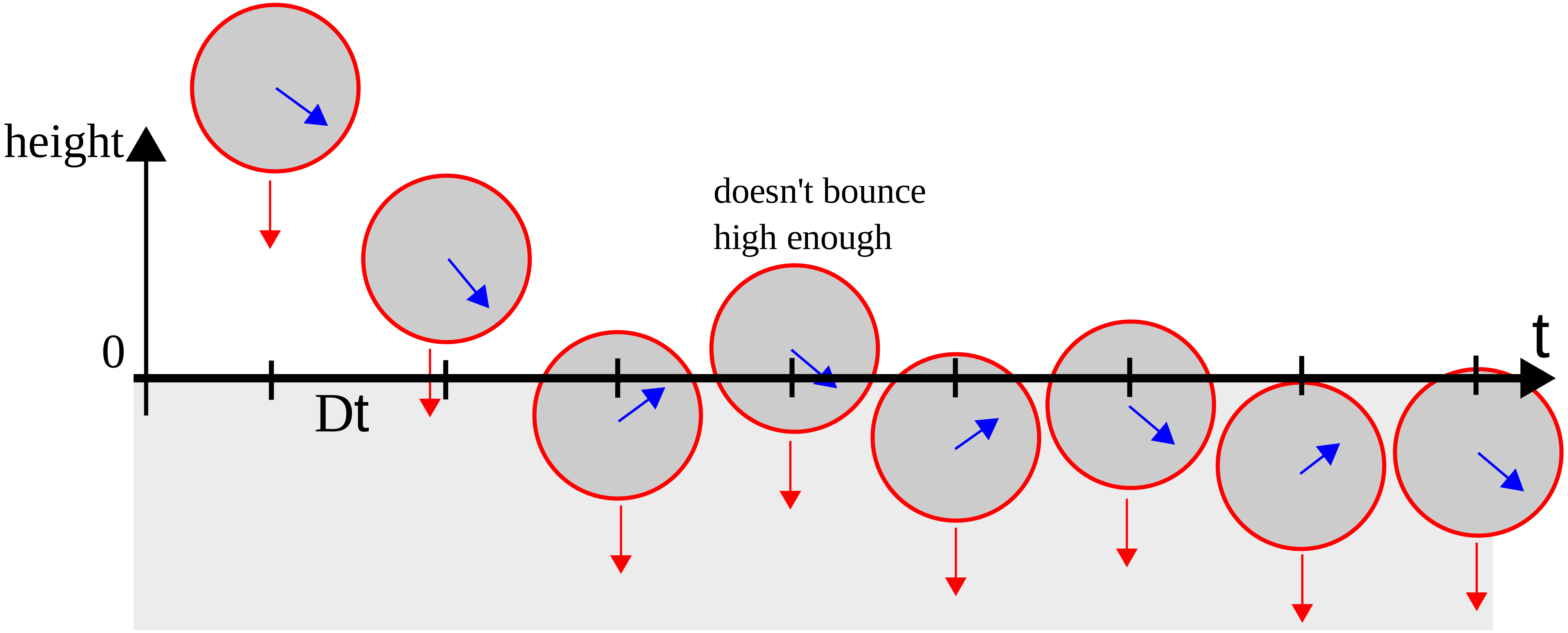


Result: Collision response - issue with discrete time

We assumed contact b/w sphere and plane

But: Exact contact never happens in discrete time

- *When collision is detected → already inside the wall*
- *Weight is still acting*



Collision response with plane : position

In real case (discrete time) no exact contact, but penetration $(p_i - a) \cdot n_i < r_i$
$\Rightarrow$ Need to compute collision response at contact point.

Three possibilities

(1) Correct position in projecting on the constaint

(+) Simple to implement

(-) Physically incorrect position

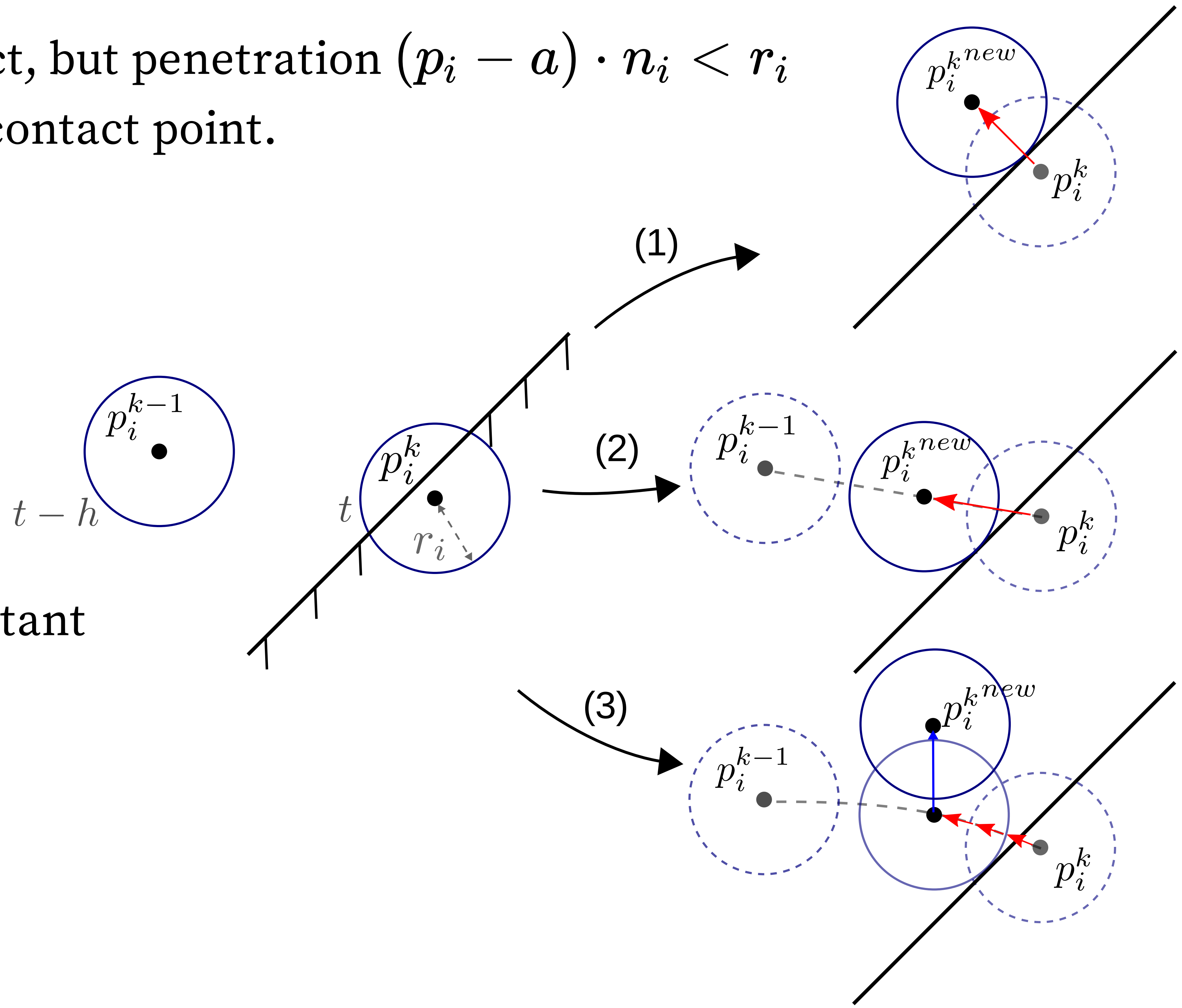
(2) Approximate the correct position

(3) Go backward in time to find exact instant of collision

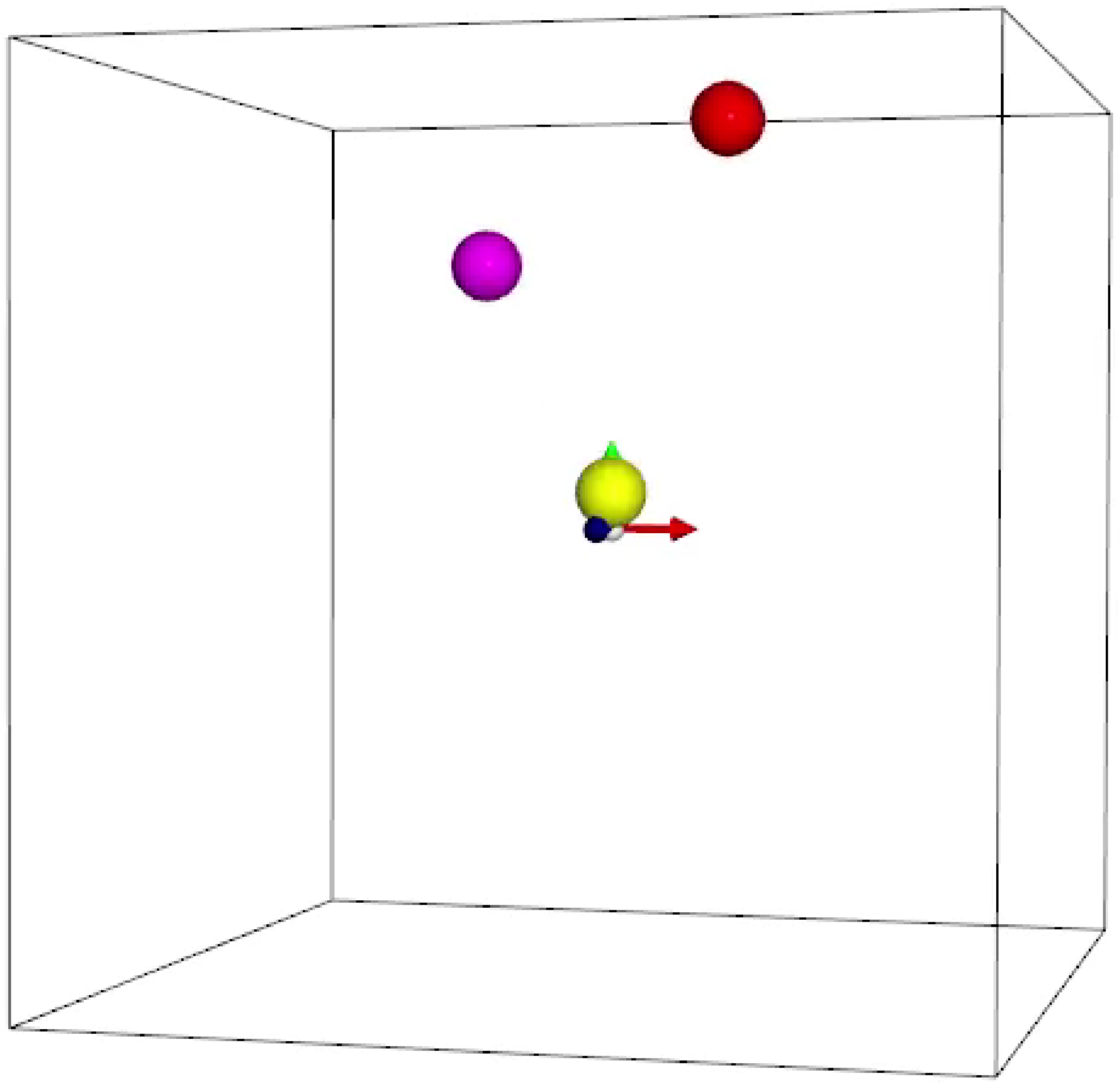
Continuous Collision Detection

(+) Physically correct

(-) Computationally heavy (binary search, etc.)



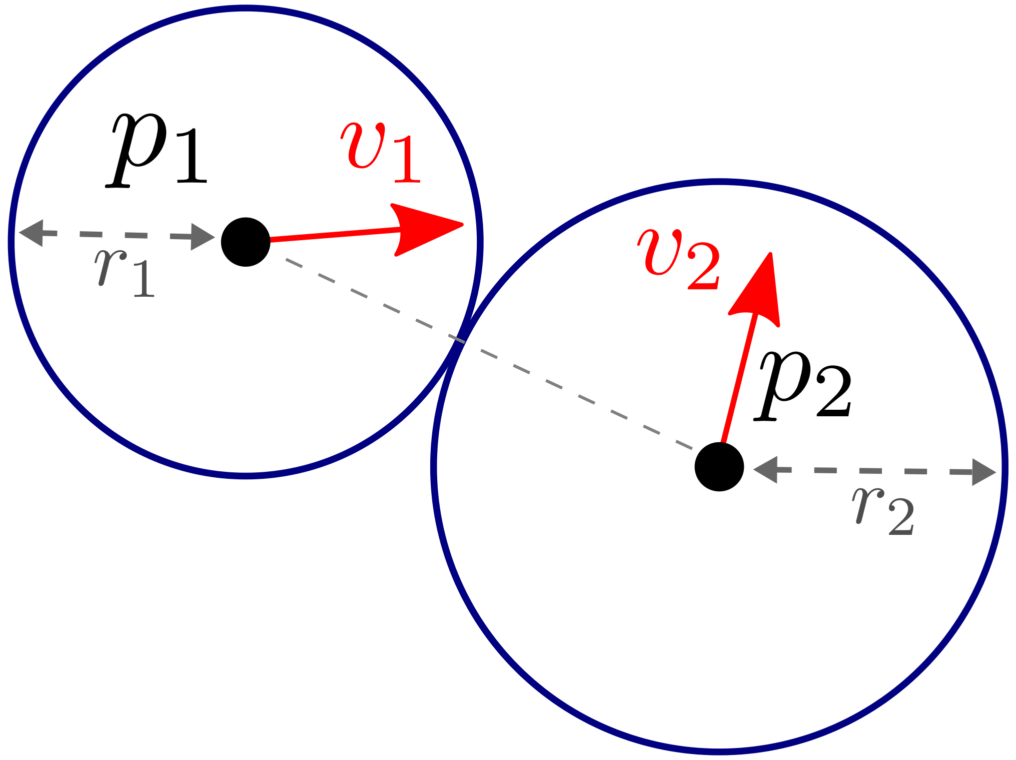
Result: Projecting position on plane



$p_i^{new} = p_i + d\, n$
$d = r_i - (p_i - a) \cdot n_i$: distance of penetration

Collision between spheres

Given 2 spheres (p_1, v_1, r_1, m_1), (p_2, v_2, r_2, m_2).
Collision when $\|p_1 - p_2\| \leq r_1 + r_2$



What happen with their velocities ?

$v_1 \rightarrow v_1^{new}, v_2 \rightarrow v_2^{new}$

Notion of impulse

An impulse J is the integrated force over time $J = \int_{t_1}^{t_2} F(t)\,\mathrm{d}t$

$\rightarrow$ results in a sudden change of speed (/momentum) in a discrete case

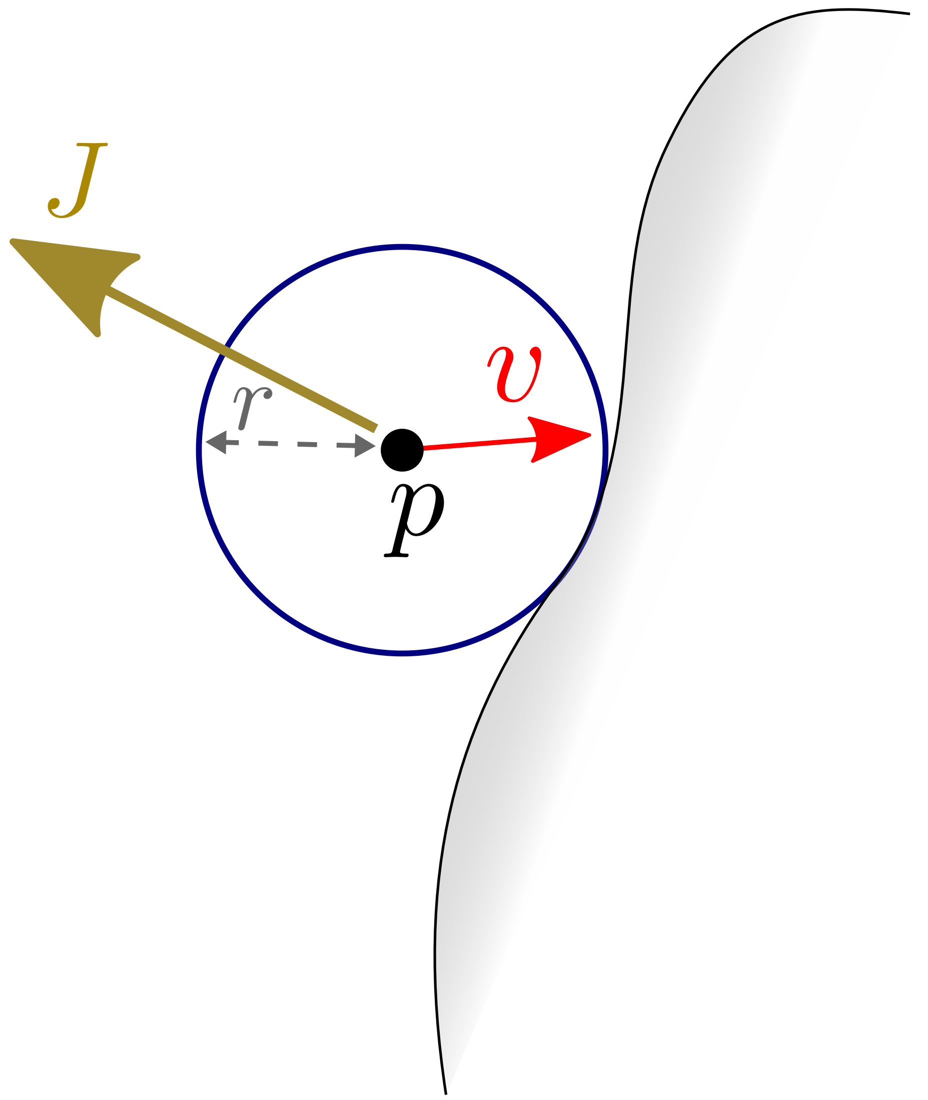
For a particle with constant mass

$$\int_{t_1}^{t_2} F(t)\,\mathrm{d}t = \int_{t_1}^{t_2} m\,a(t)\;\mathrm{d}t$$

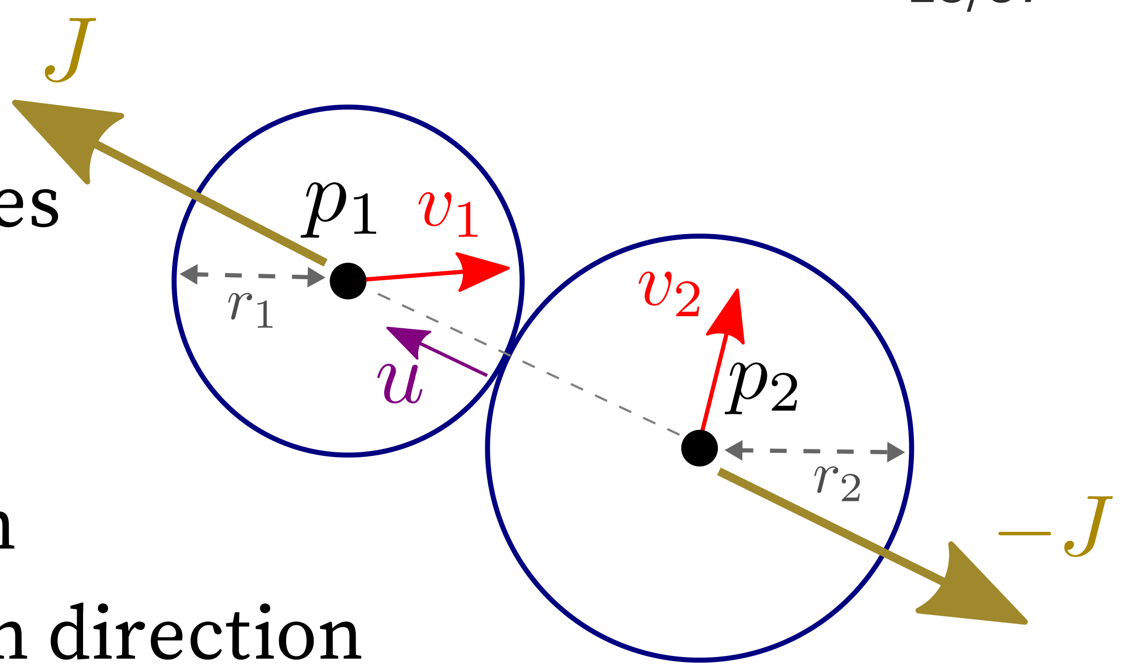
$$\Rightarrow J = m\,(v(t_2) - v(t_1))$$

For an impact $v \rightarrow v^{new}$

$$v^{new} = v + J/m$$



Two spheres in collision



Impulse orthogonal to the separating plane between the two surfaces

$$J = j\,u, \quad u = (p_1 - p_2)/\|p_1 - p_2\|$$

The system with the two spheres is preserving its linear momentum

$\Rightarrow$ Respective impulses j are equals in magnitude, and opposed in direction

$$m_1 v_1 + m_2 v_2 = m_1 v_1^{new} + m_2 v_2^{new} \Rightarrow m_1 (v_1^{new} - v_1) = -m_2 (v_2^{new} - v_2) \Rightarrow J_1 = -J_2$$

Assume collision of "hard spheres" $=$ "Elastic collision"

$=$ No loss of energy, conservation of kinetic energy of the system

$$\Rightarrow j = 2 \frac{m_1\, m_2}{m_1 + m_2} \,(v_2 - v_1) \cdot u$$

$$1/2\, m_1\, v_1^2 + 1/2\, m_2\, v_2^2 = 1/2\, m_1\, (v_1^{new})^2 + 1/2\, m_2\, (v_2^{new})^2$$

$$\Rightarrow m_1 v_1^2 + m_2 v_2^2 = m_1 \left(v_1 + \frac{j}{m_1} u\right)^2 + m_2 \left(v_2 - \frac{j}{m_2} u\right)^2$$

$$\Rightarrow 0 = 2\, j\, v_1 \cdot u + \frac{j^2}{m_1} - 2\, j\, v_2 \cdot u + \frac{j^2}{m_2}$$

$$\Rightarrow j = \frac{2}{1/m_1 + 1/m_2} \,(v_2 - v_1) \cdot u$$

Two spheres in collision

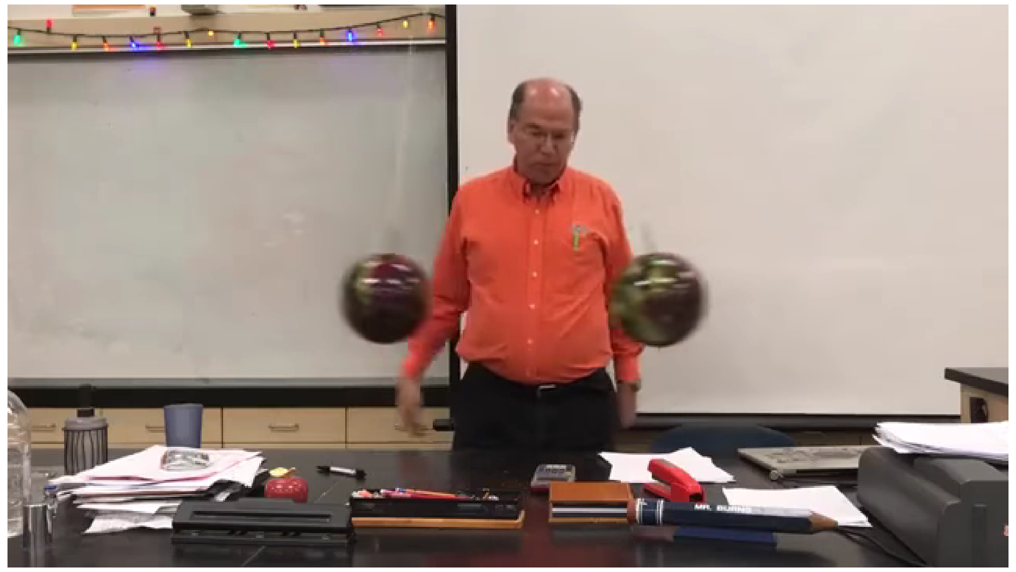
$$v_1^{new} = v_1 + j/m_1\, u = v_1 + 2\tfrac{m_2}{m_1+m_2}((v_2 - v_1)\cdot u)\, u$$
$$v_2^{new} = v_2 - j/m_2\, u = v_2 - 2\tfrac{m_1}{m_1+m_2}((v_2 - v_1)\cdot u)\, u$$

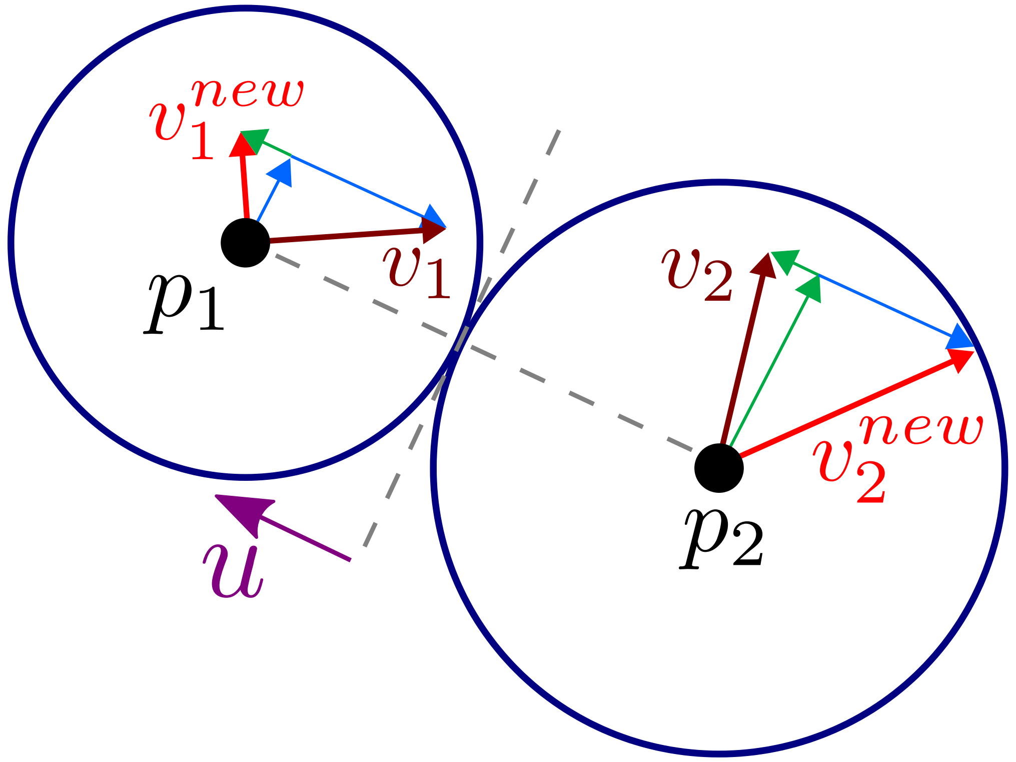
Rem. If $m_1 = m_2$: Switch their $\perp$ speeds

$$v_1^{new} = v_1 + ((v_2 - v_1)\cdot u)\, u = v_{1//} + v_{2\perp}$$
$$v_2^{new} = v_2 - ((v_2 - v_1)\cdot u)\, u = v_{2//} + v_{1\perp}$$

Can use restitution coefficient and attenuation $(\alpha, \beta) \in [0, 1]$

$$v_1^{new} = \alpha\, v_{1//} + \beta\, v_{2\perp}$$
$$v_2^{new} = \alpha\, v_{2//} + \beta\, v_{1\perp}$$





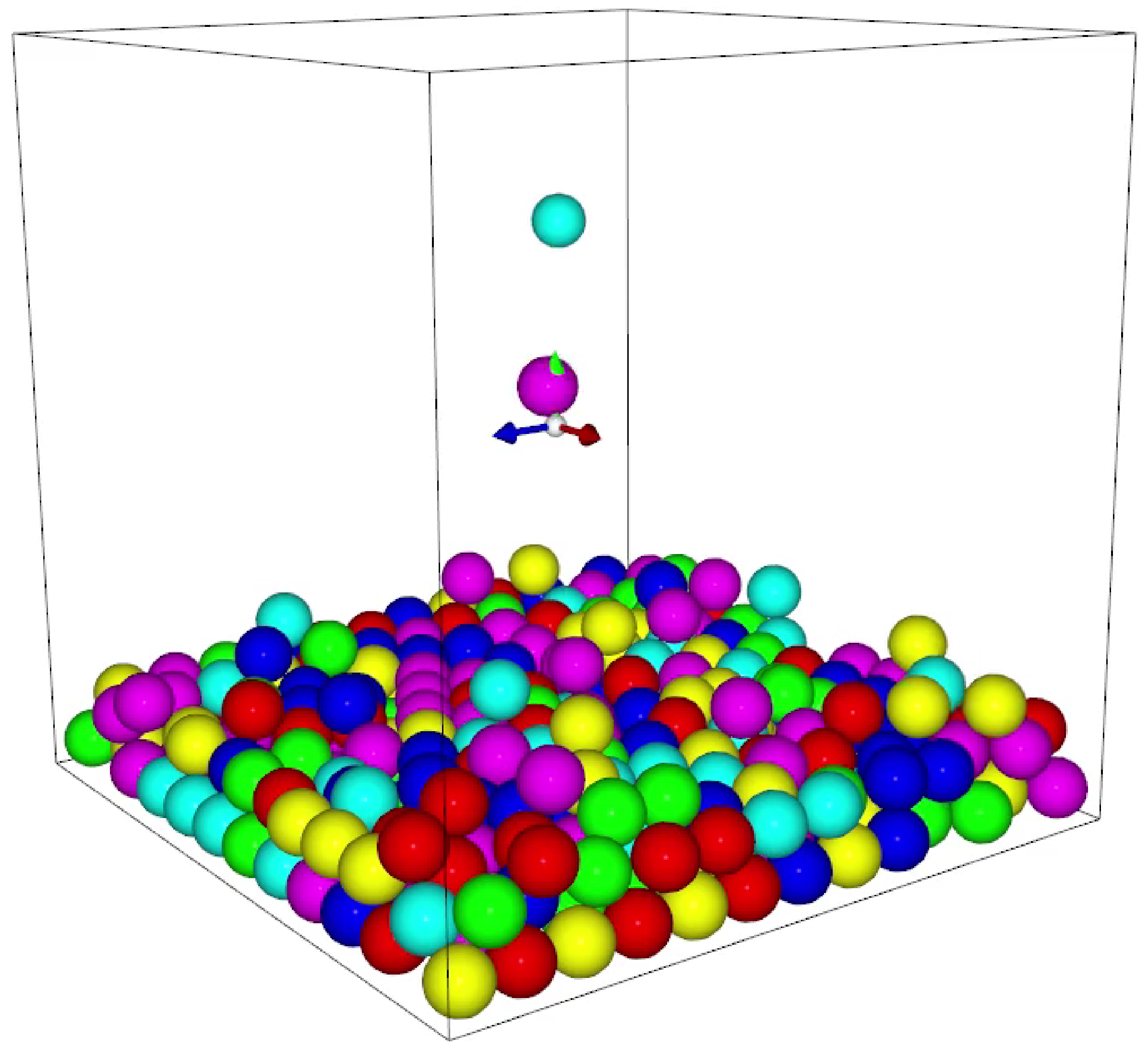
Summary

1. Detect collision $\|p_1 - p_2\| \leq r_1 + r_2$

2-a. If collision (relative speed $> \epsilon$)
   Elastic collision (/bouncing) $v_{1/2} = \alpha\, v_{1/2} \pm \beta\, J / m_{1/2}$

2-b. If *static* contact (relative speed $\leq \epsilon$)
   Friction $v_{1/2} = \mu\; v_{1/2}\,, \quad \mu \in [0, 1]$
   Avoids jittering

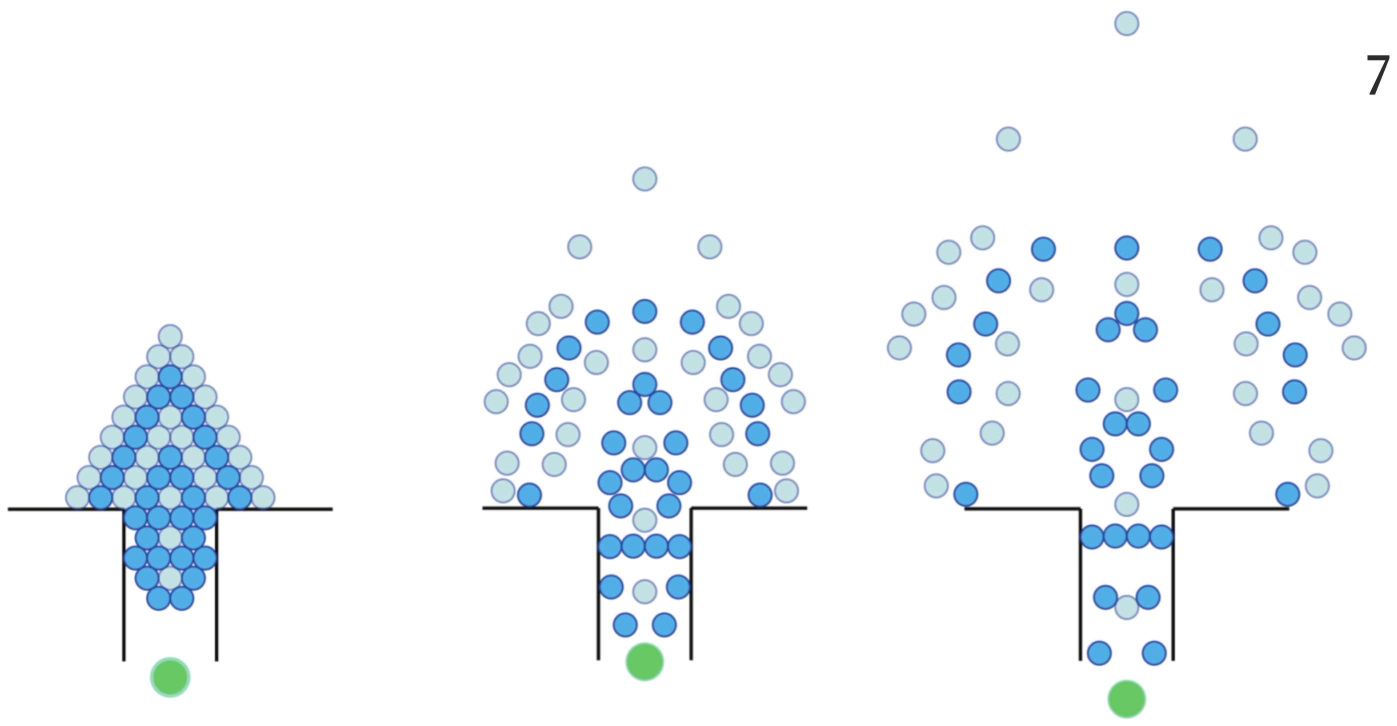
3. Correct position (project on contact surface)
   $p = p \pm d/2\, u$
   $d = r_1 + r_2 - \|p_1 - p_2\|$: Collision depth



Multiple collisions

Pairwise collisions $\Rightarrow$ no global collision free state

- Correcting one collision may induce new collisions.
- Order of correction does matter



Reducing time-step help, Iterating over mutliple pass help

But correct solution in all cases is complex $\rightarrow$ global approach

- Precompute contact graph

  explicit shock propagation management

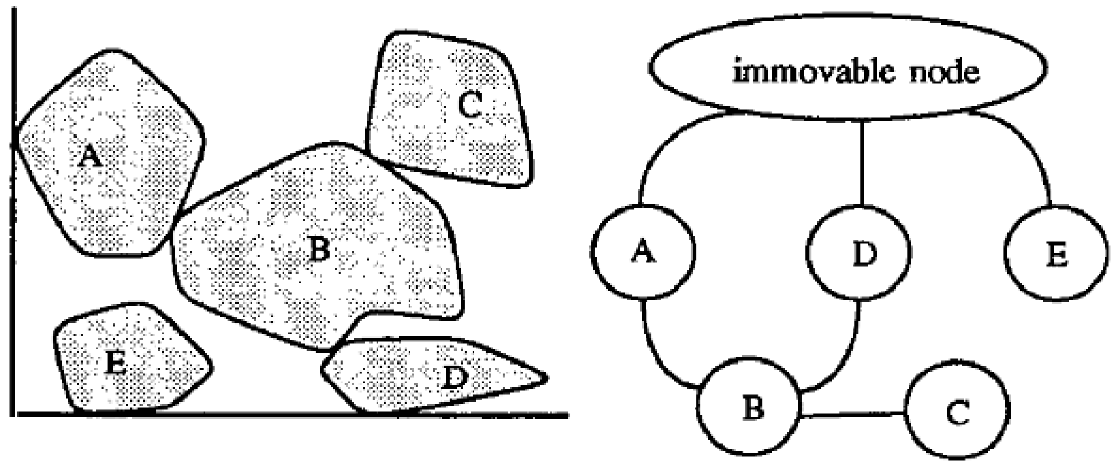
- Global constraint-based method

  Impulse: $n_i \cdot (v_i - v_j) \geq 0$

  Momentum preservation: $m_i v_i - m_j v_j = 0$

  Energy preservation/dissipation

  $\Rightarrow$ Linear Complementarity Program, Gauss Seidel, etc.



[Realistic Animation of Rigid Bodies. J. Hahn. SIGGRAPH 1988.]

[Collision Detection and Response for Computer Animation. M. Moore and J. Wilhelms. Computer Graphics 1988.]

[Reflections on Simultaneous Impact. B. Smith et al. SIGGRAPH 2012]

[Guaranteed Resolution of Simultaneous Rigid Body Impact. E. Vouga. ACM SIGGRAPH 2017]

Elastic models

Spring structure
Numerical solution of ODE
Cloth simulation

Material model

Elasticity: Shape goes back toward its original rest position when external forces are removed.

-Purely elastic models don't loose energy when deformed (potential $\leftrightarrow$ kinetic)

Plasticity: Opposite of elasticity. Plastic material don't come back to their original shape (/change their rest position during deformation).

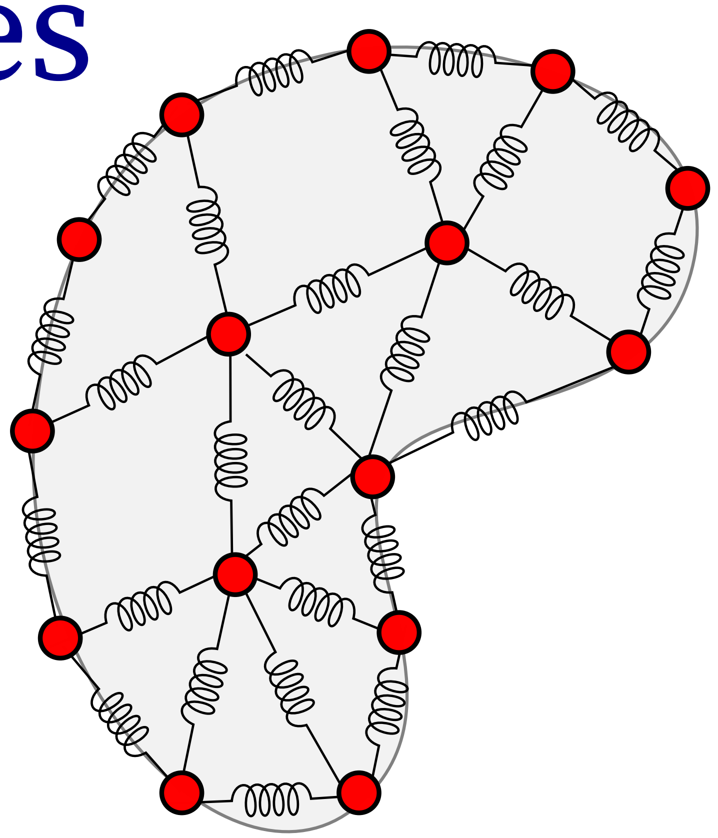
- Ductile material - can allow large amount of plastic deformation without breaking (plastic)
- Brittle - Opposite (glass, ceramics)

Viscosity: Resistance to flow (usually for fluid, ex. honey)

In reality

- *Elasto-plastic materials*: Allow elastic behavior for small deformation, and plastic at larger one.
- *Visco-elastic materials*: Elastic properties with delay.

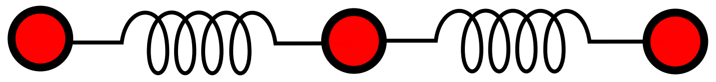
Modeling elastic shapes with particles



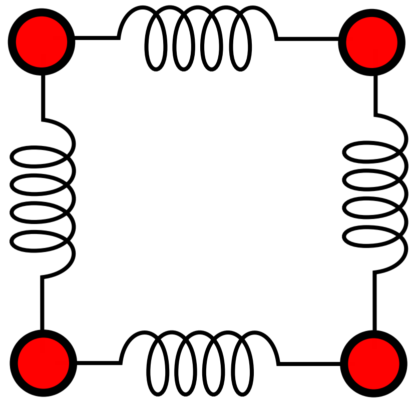
Spring mass systems

- Particles (position, velocity, mass): samples on shape
- Springs : link closed-by particles in the reference shape

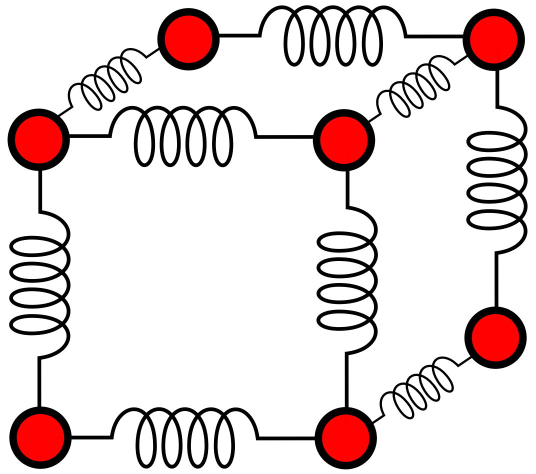
1D curve structure



2D surface structure



3D volume structure

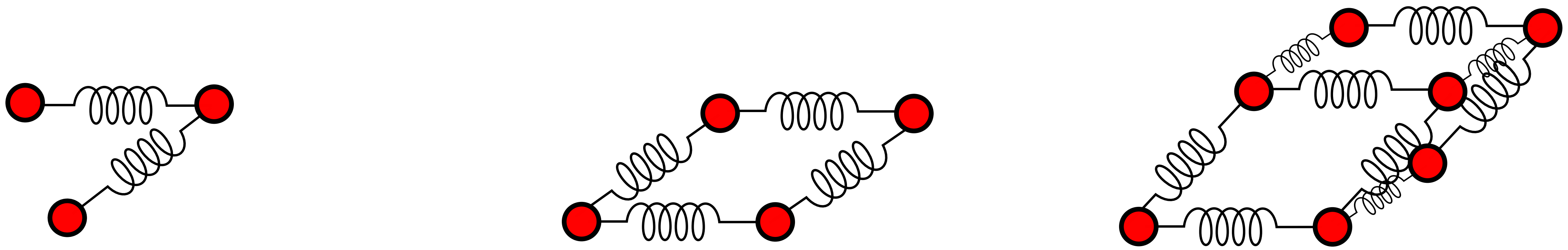


Spring structure

How to model spring connectivity ?

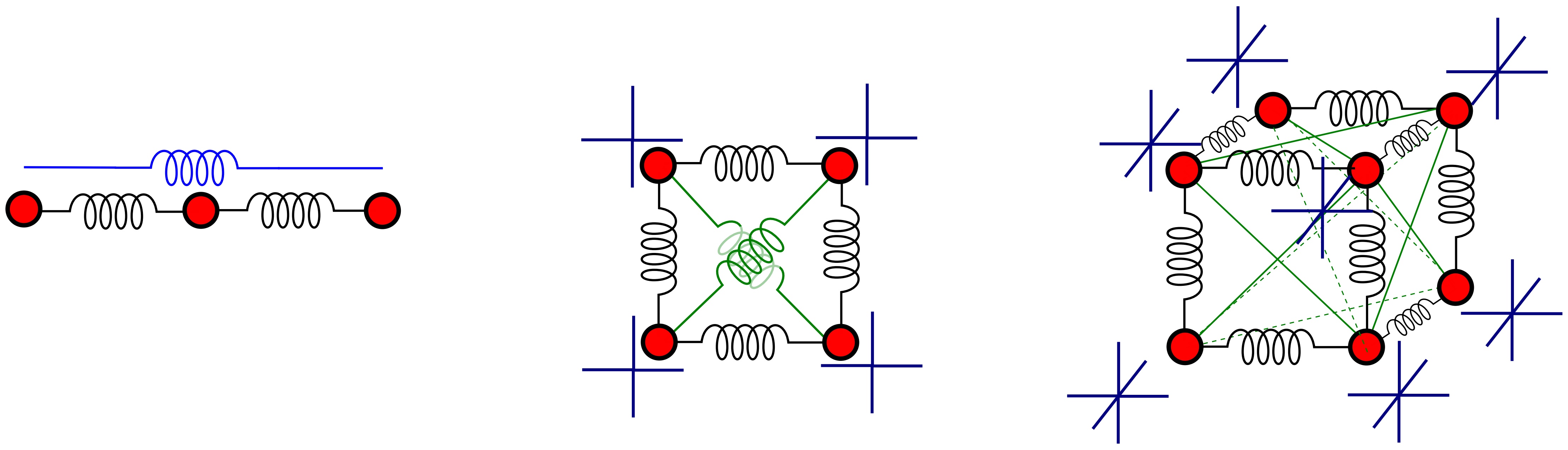
- **Structural springs**: 1-ring neighbors springs ($\simeq$ mesh edges)

  (+) Limit elongation/contraction, (-) Allows shearing, and bending



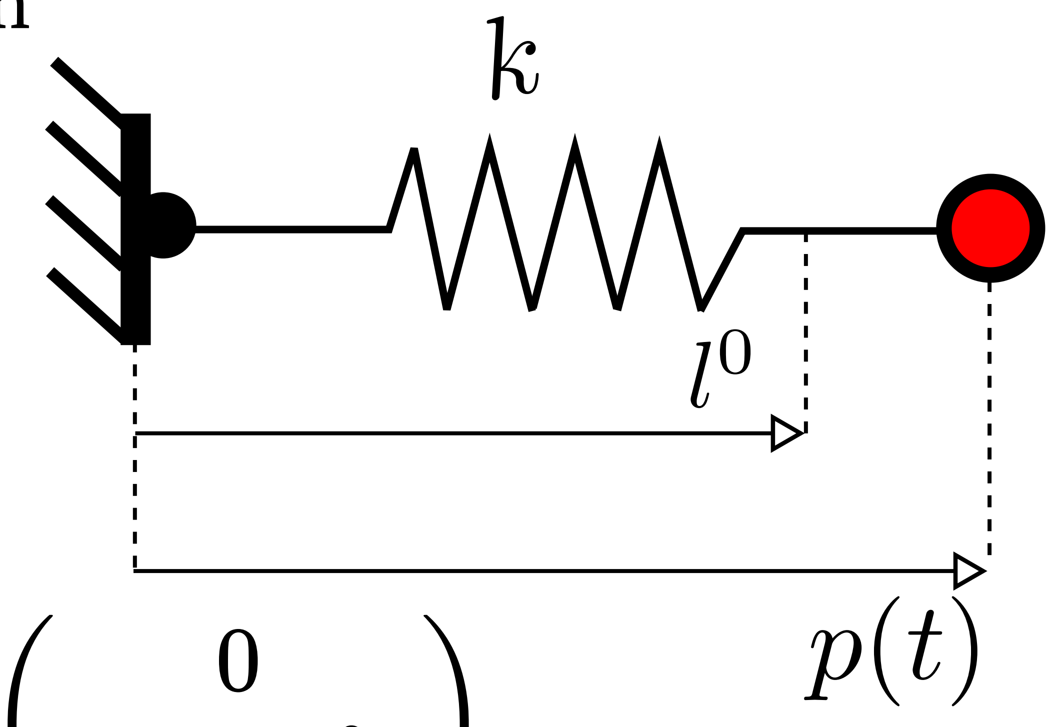
$\Rightarrow$ Add extra springs connectivity

- **Shearing springs**: Diagonal links
- **Bending springs**: 2-ring neighborhood



Example: 1D spring (/Harmonic oscillator)

- Force $F(t) = -k\,(p(t) - l^0)$, k spring stiffness, l^0 rest length
- Second order differential equation: $m\,p''(t) + k\,p(t) = k\,l^0$



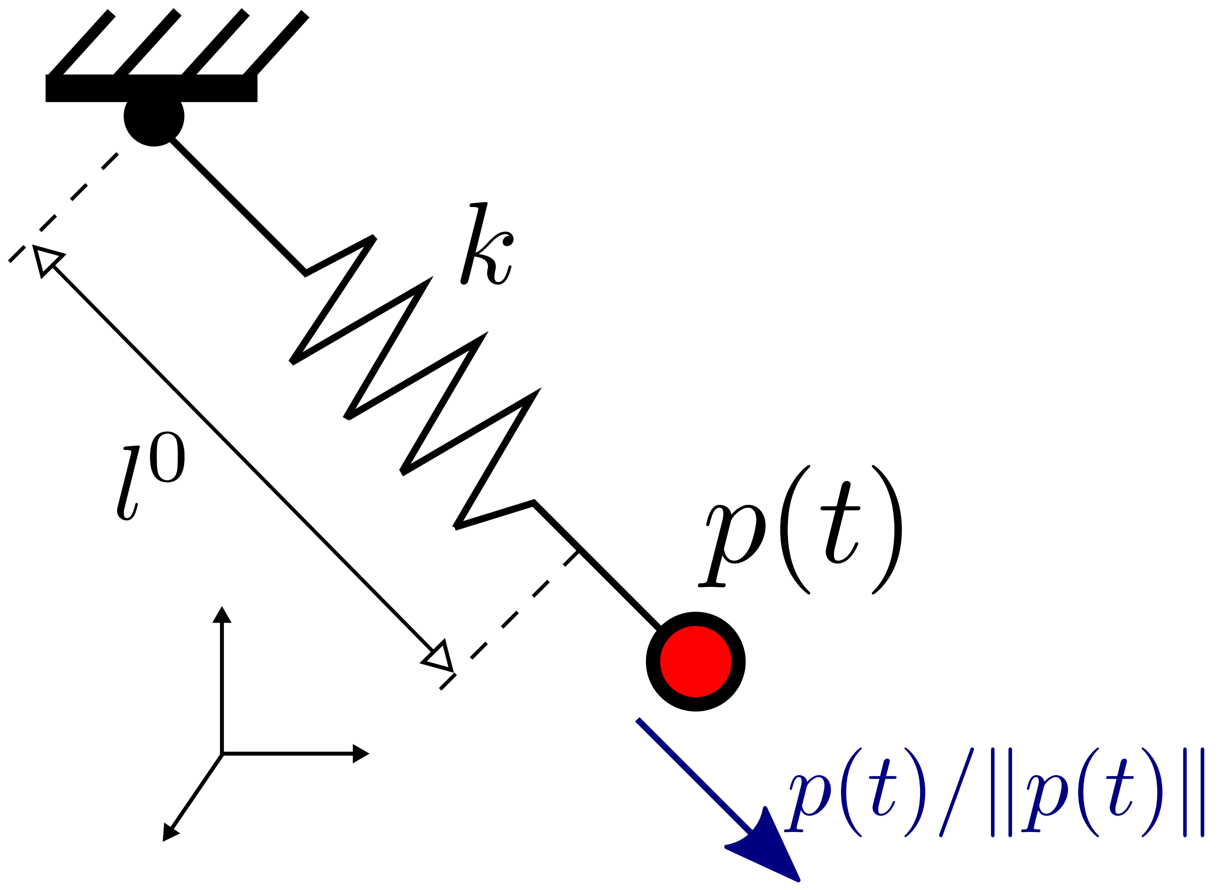
- First order system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} v(t) \\ -k/m\,\big(p(t) - l^0\big) \end{pmatrix}}_{\mathcal{F}(u,t)}$
- Linear system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} 0 & 1 \\ -k/m & 0 \end{pmatrix}}_{\text{A}} \underbrace{\begin{pmatrix} p \\ v \end{pmatrix}(t)}_{u(t)} + \underbrace{\begin{pmatrix} 0 \\ k/m\,l^0 \end{pmatrix}}_{b}$

- Exact solution known: $p(t) = A\,\sin(\omega\,t + \varphi) + l^0$

$$\omega = \sqrt{k/m},\, A^2 = (p^0 - l^0)^2 + \left(\frac{v_0}{\omega}\right)^2,\, \tan(\varphi) = \frac{p^0 - l^0}{v^0/\omega}$$

Example: 3D mass spring

- Force $F(t) = m\,g - k\,(\|p(t)\| - l^0)\,\frac{p(t)}{\|p(t)\|}$, k spring stiffness, l^0 rest length
- Second order differential equation: $m\,p''(t) = mg - k\,(\|p(t)\| - l^0)\,\frac{p(t)}{\|p(t)\|}$
- First order system $\underbrace{\begin{pmatrix} p \\ v \end{pmatrix}'(t)}_{u'(t)} = \underbrace{\begin{pmatrix} v(t) \\ g - k/m\,(\|p(t)\| - l^0)\,\frac{p(t)}{\|p(t)\|} \end{pmatrix}}_{\mathcal{F}(u,t)}$
- Not linear
- No simple explicit solution



Numerical integration of ODE

General formulation: $u'(t) = \mathcal{F}(u,t), u(t) = (p(t), v(t))$.

Explicit Euler

$u^{k+1} = u^k + \Delta t\, \mathcal{F}(u^k, t^k)$

(+) Easy to implement

(-) Worst scheme in all cases (divergence, low accuracy)

Explicit Runge-Kutta

$u^{k+1} = u^k + \Delta t\, \sum_j \alpha_j k_j$

(+) Good **accuracy**

(+) Efficient to apply

(+/-) Stability OK for non-stiff problem, diverge on stiff problem

(-) Artificial damping for constant energy system

Implicit methods

$u^{k+1} = u^k + \Delta t\, \mathcal{F}(u^{k+1}, t^{k+1})$

(+) Good to deal with **stiff problem** - very stable

(-) Add numerical damping (converge even if solution oscillates)

(-) Hard/computationally costly to apply on non linear problem

Symplectic integrator

$v^{k+1} = v^k + \Delta t\, F^k/m$

$p^{k+1} = p^k + \Delta t\, v^{k+1}$

(+) Handle well constant energy system, preserves energy (Hamiltonian systems)

(+) Simple and efficient to implement

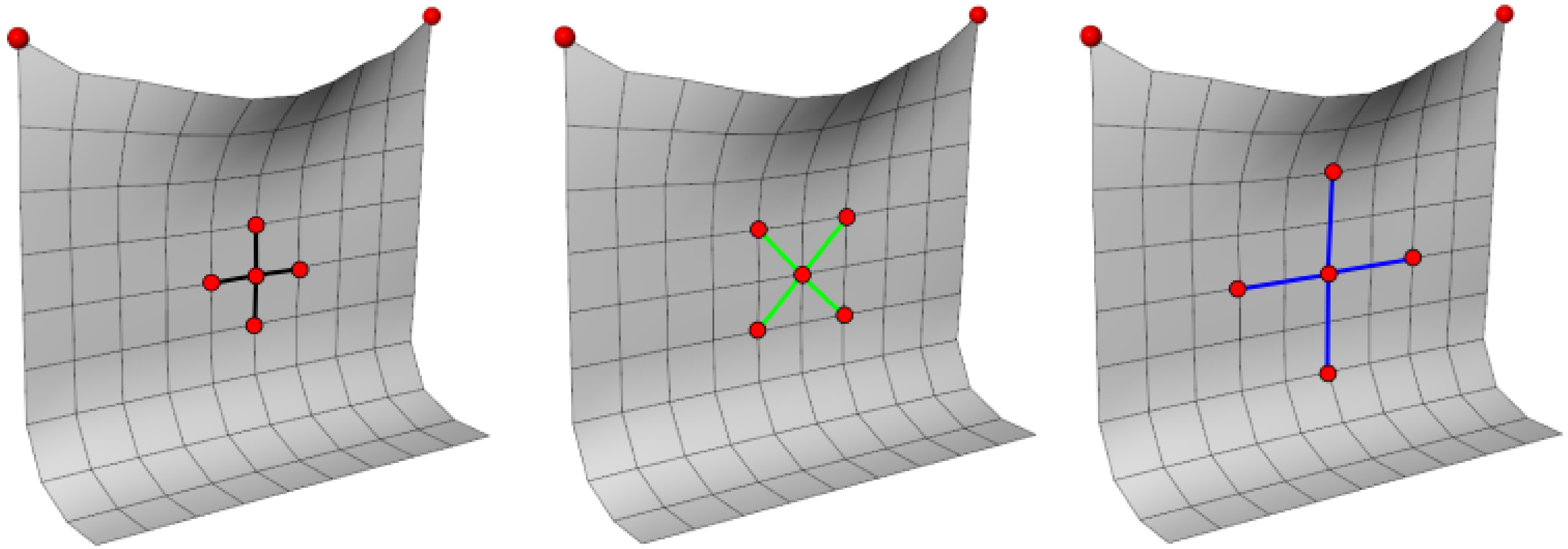
(-) Less accurate than RK

(-) Diverge on stiff problem

Cloth Simulation

Mass-spring cloth simulation

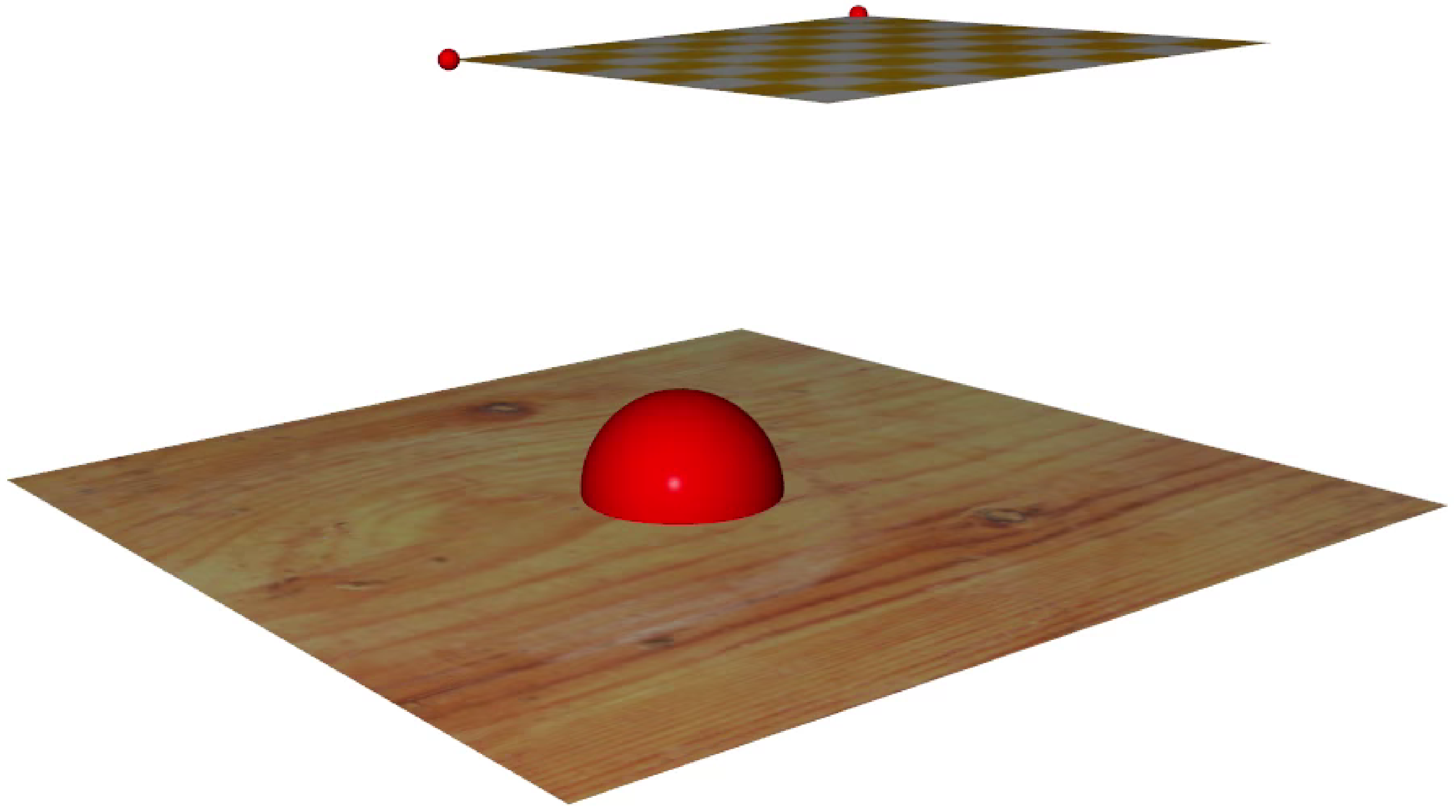
- Particles are sampled on a $N \times N$ grid.
  - Each particle has a mass m ($m_{cloth} = N^2\, m$)
- Set structural, shearing and bending springs.



Forces

- On each particle: gravity + drag + spring forces
- $F_i(p,v,t) = m_i\, g - \mu\, v_i(t) + \sum_{j \in \mathcal{V}_i} K_{ij}\ \left(\|p_j(t) - p_i(t)\| - L_{ij}^0\right) \dfrac{p_j(t) - p_i(t)}{\|p_j(t) - p_i(t)\|}$
  - $\mathcal{V}_i$: neighborhood of particle i
  - L_{ij}^0: rest length of spring ij

Associated ODE $\forall i, \begin{cases} p_i'(t) = v_i(t) \\ v_i'(t) = F_i(p,v,t)/m_i \end{cases}$



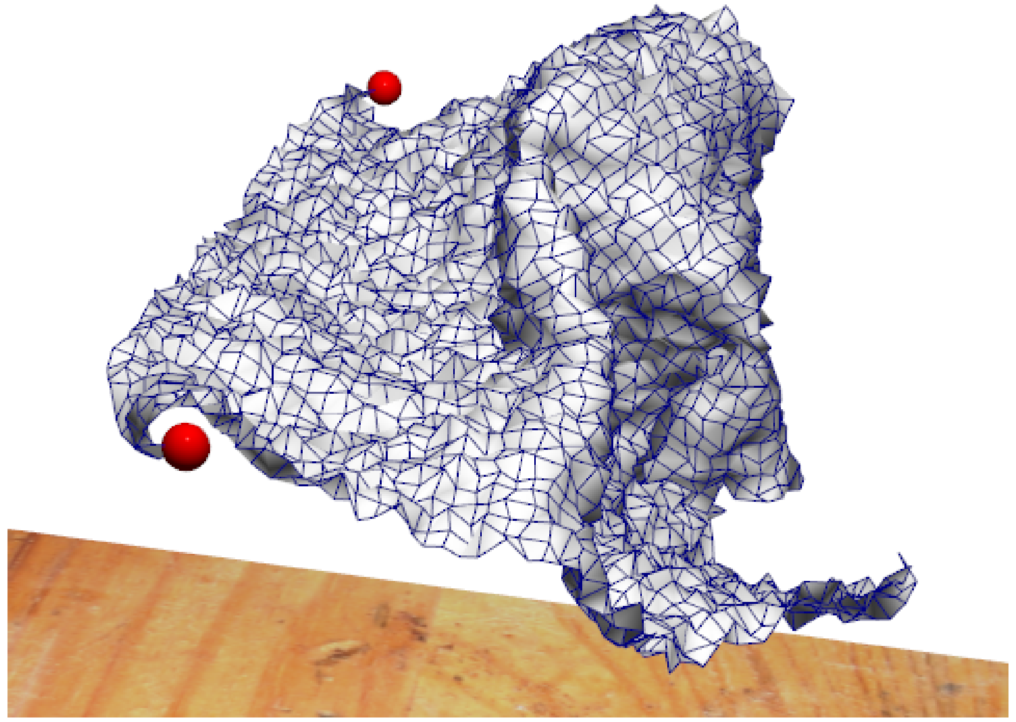
Q. How can we model the effect of the wind ?

Note on Mass-Spring numerical solution

- Non-linear ODE

- Large K_{ij} : good length preservation, but stiff ODE
  $\Rightarrow$ divergence of explicit schemes.
- Avoid explicit Euler (divergence)

- Semi-implicit Euler/Verlet works fine for low K_{ij}
  Semi-implicit Euler + PBD allows simple integration + stable stiff springs
  [Muller et al. PBD , Inextensible clothing in Computer Games]

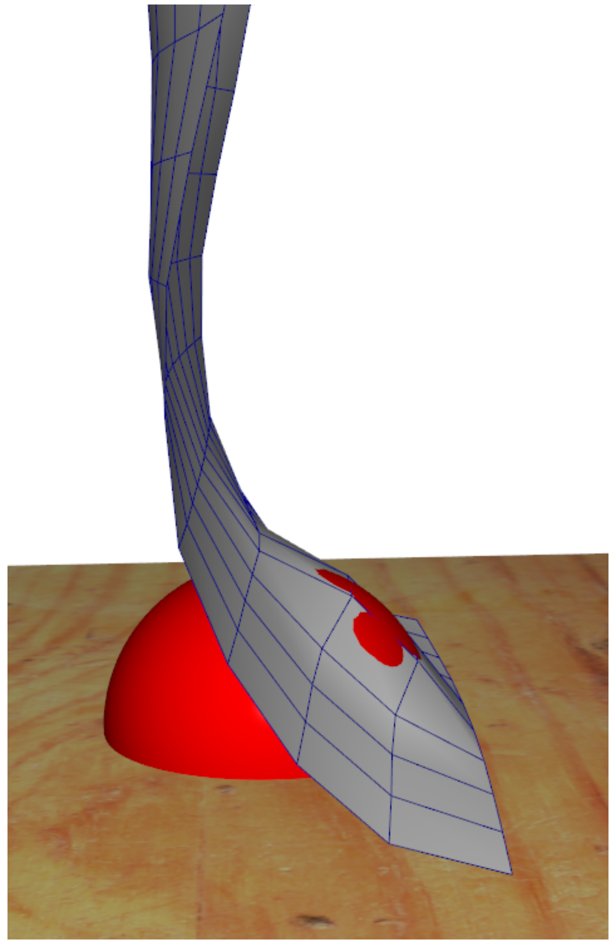


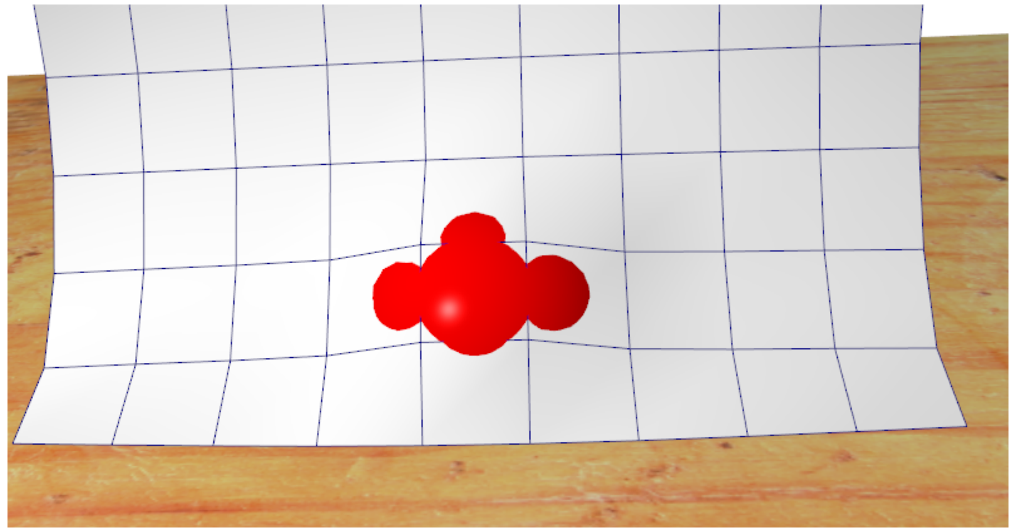
- RK4 more accurate (but higher complexity than Verlet)

- Implicit Euler : requires linearization, but very stable

Collisions

- Simple approach : Handled as collision between particles and shapes
  - (+) Simple and efficient
  - (-) Collision may still appears within a triangle
  - $\Rightarrow$ Exaustive approach: edges + faces





Limitation of mass spring model and continuous model

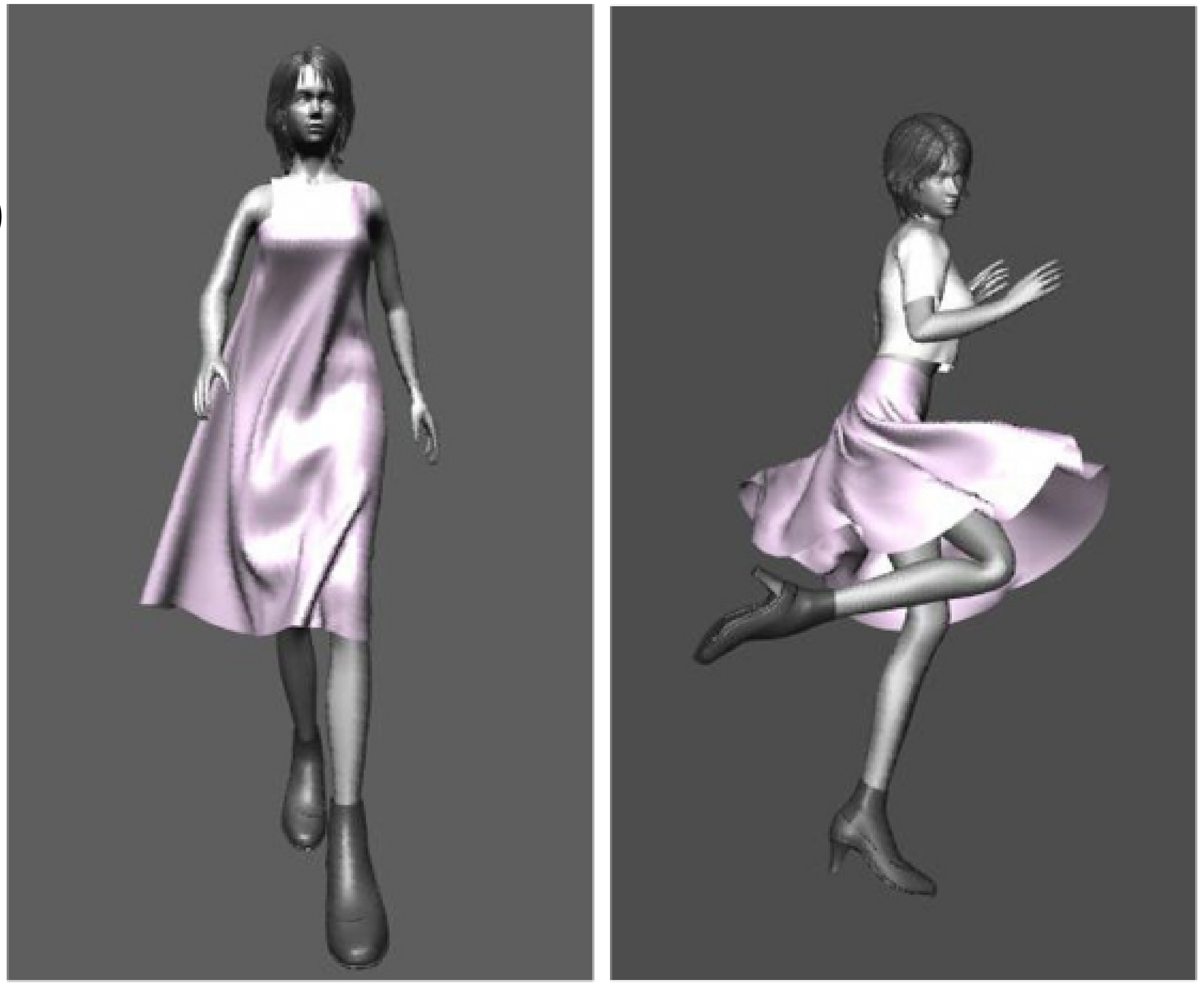
- Does mass-spring system converge toward a unique solution when sampling increase ?

  $\Rightarrow$ No :(

  Depends on the connectivity $\rightarrow$ bad for physical accuracy

Corollary

- Mass-springs work well for grid-mesh structure (draping)
- Less for arbitrary triangular meshes



1st improvment: Change toward energy formulation for bending springs (limits locking effect)

$$F = \frac{\partial E}{\partial p}$$

$$E = \frac{1}{2} K\, L\, \kappa^2,\ \kappa\text{: curvature}$$

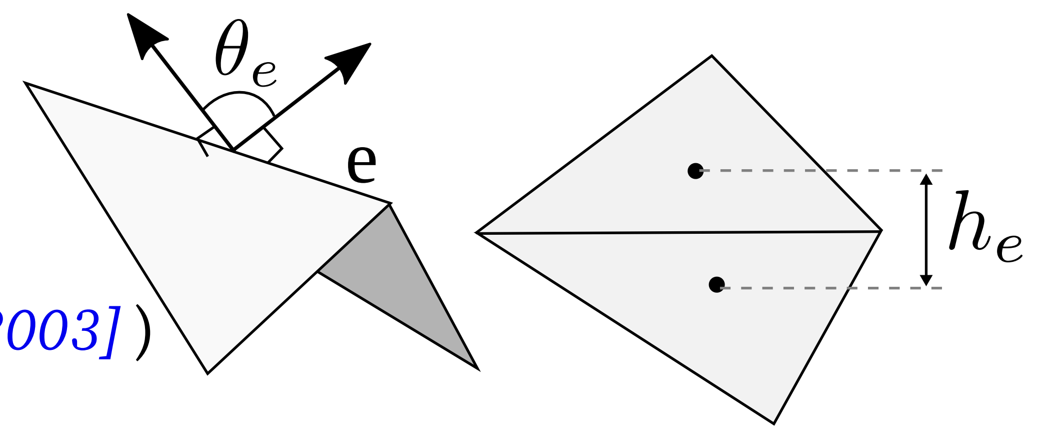
[Cho et al, Stable but Responsive Cloth, ACM SIGGRAPH 2002]

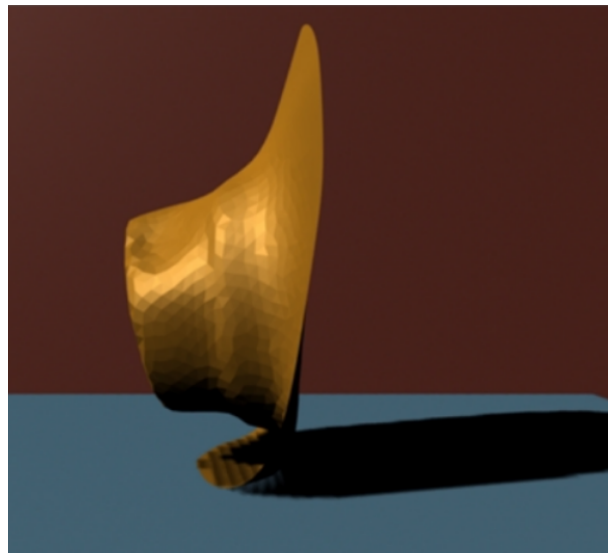
Triangle as continuous elements

- Defining Bending Energy between triangles
  - $W_B(x) = \sum_{edges\ e} (\theta_e - \theta_e^0) \frac{\|e^0\|}{h_e^0}$

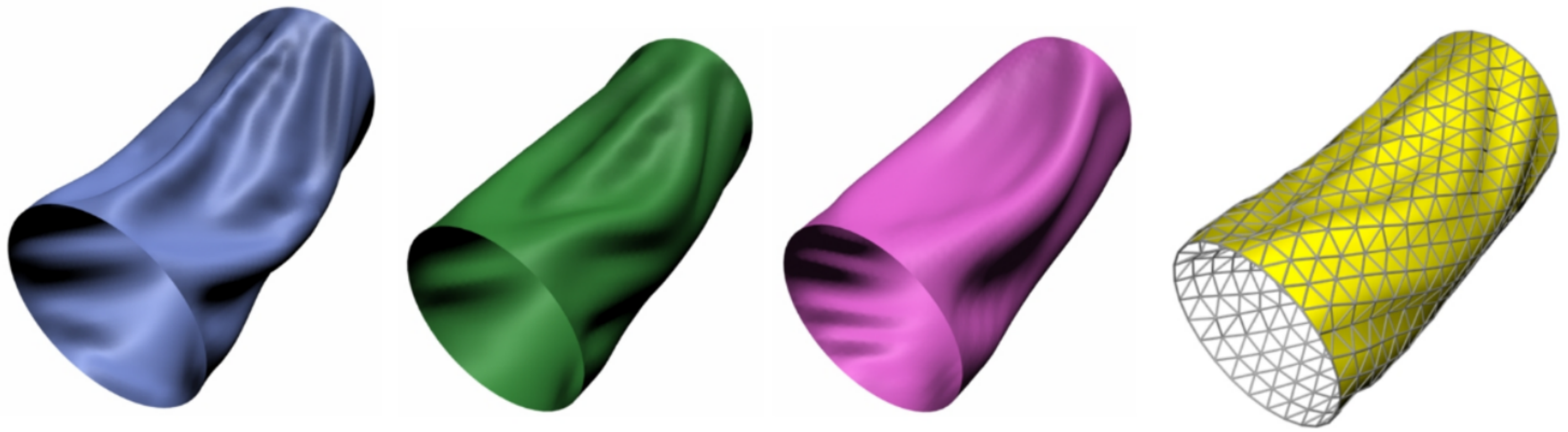
    [E. Grinspun et al., Discrete Shells, SCA 2003]

    (or expressed using forces in [R. Bridson et al., SCA 2003])



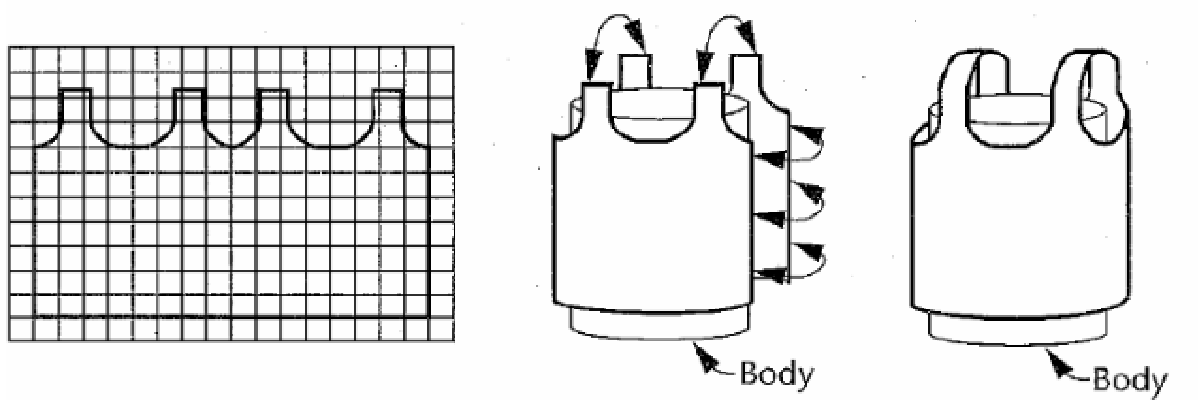


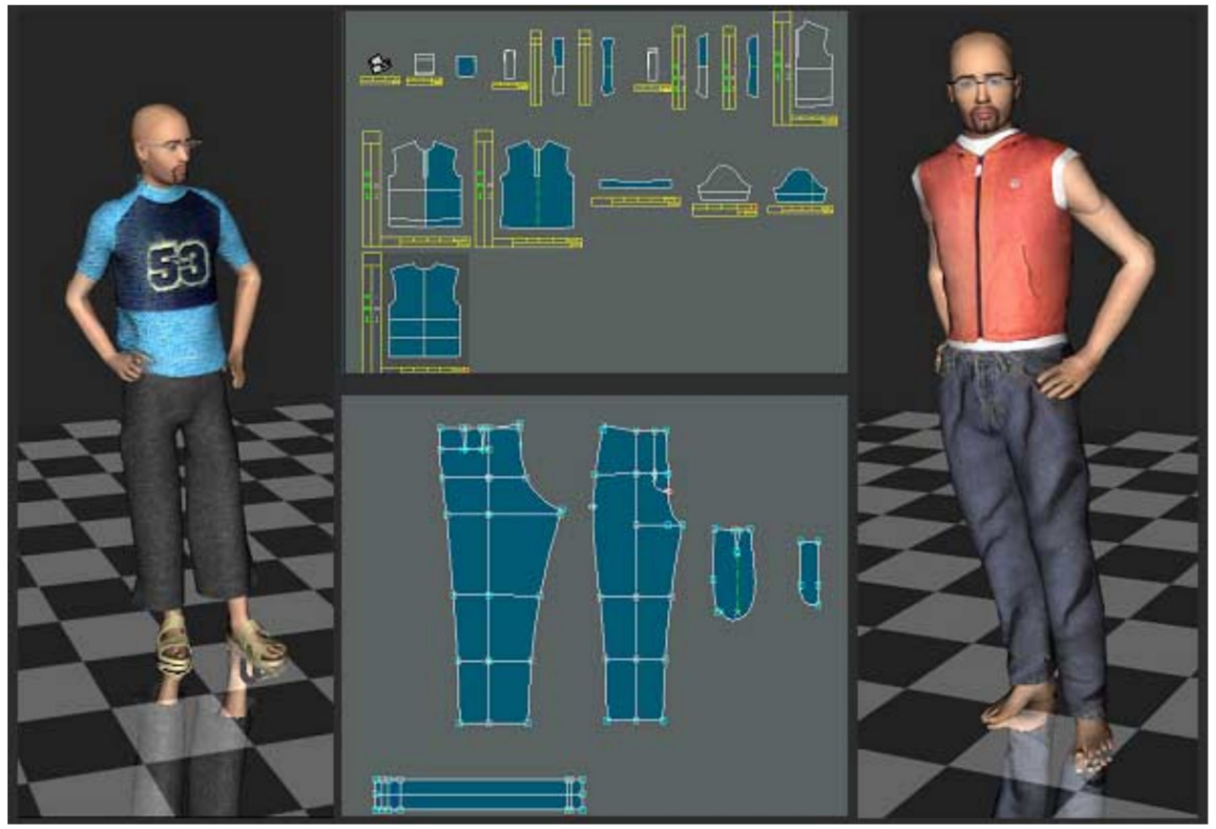
- Going toward full FEM numerical resolution
  - B. Thomaszewski et al. [SCA 2006], [VRIPHYS 2008], [EG 2009].



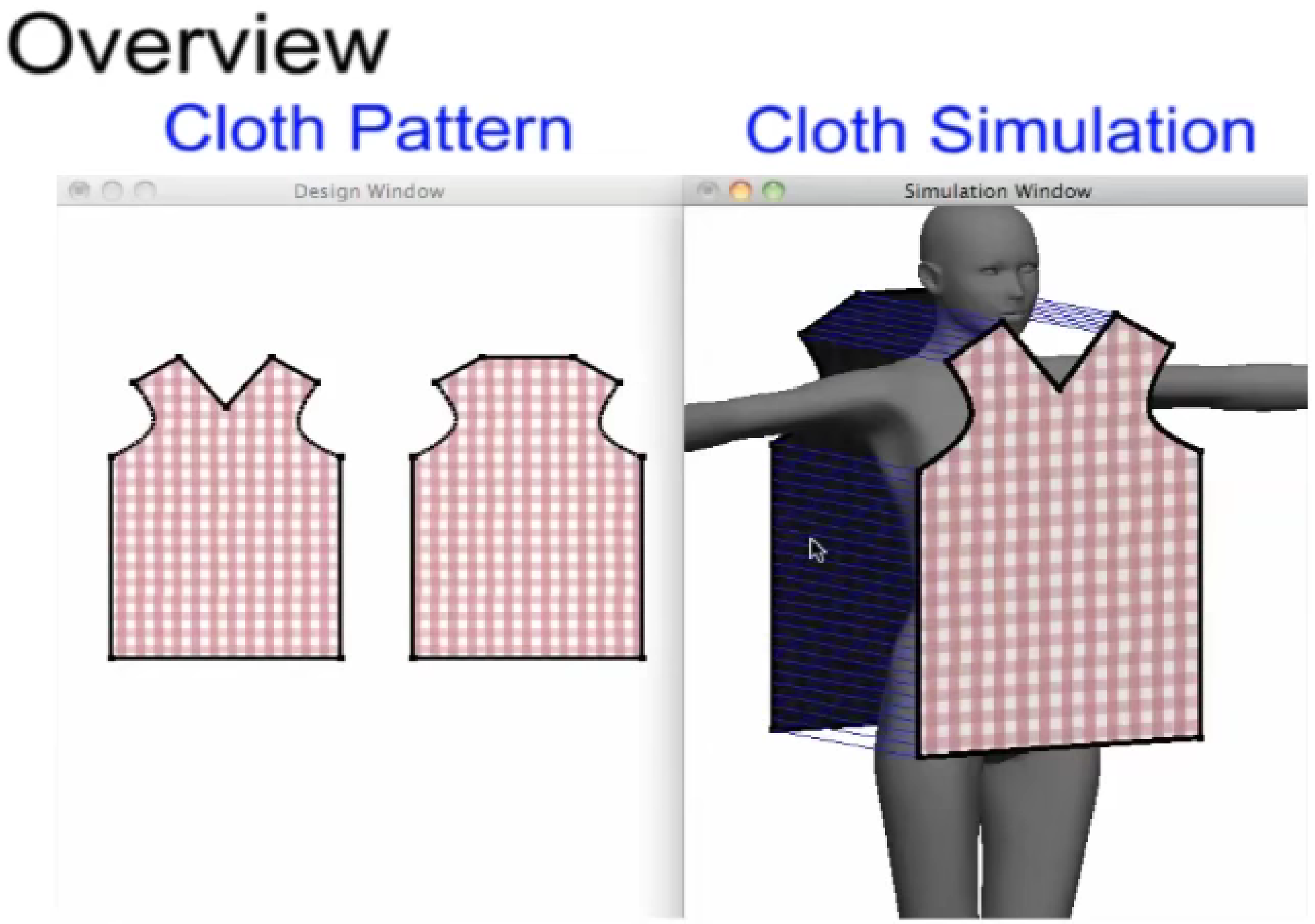
Clothing

- Stich 2D patterns together to generate full cloth
    - Cloths are developable material (preserve length w/r their 2D patterns)



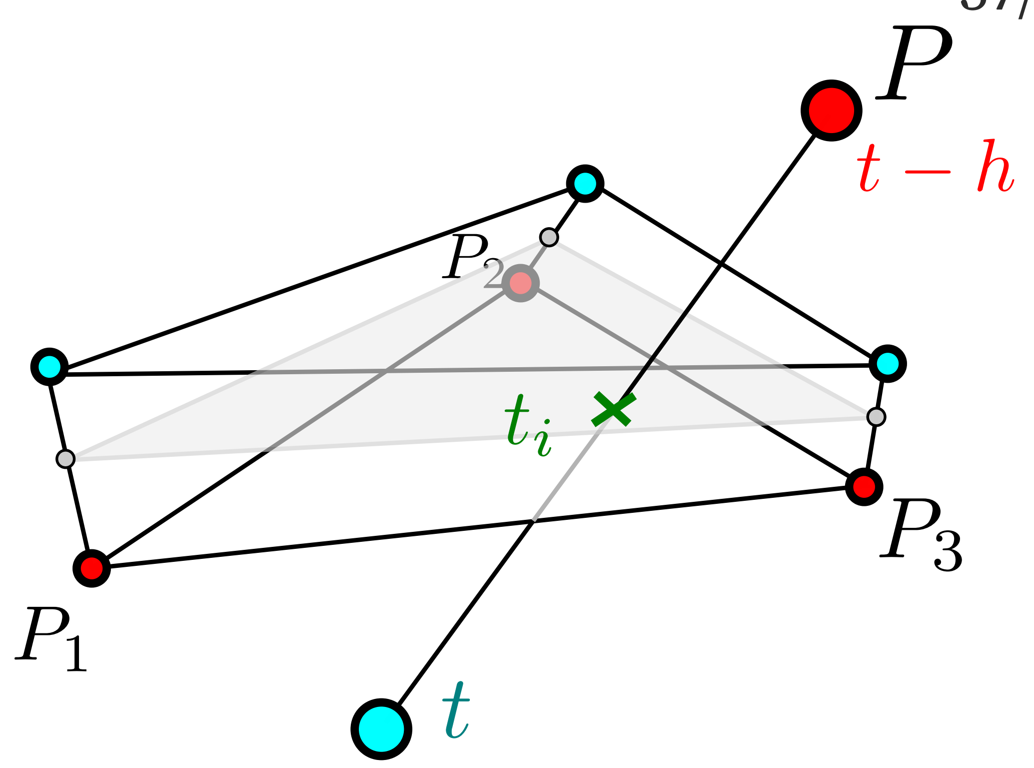


[Thalman et al. 2002]



[Umetani et al., 2011]

Detecting self collision



Handled as moving point in collision with moving triangle

Inputs

- Triangle $P_1(t)P_2(t)P_3(t)$, a point $P(t)$
- Each position $P_k(t) = P_k(0) + t\,v_{P_k}$

Computing intersection

Necessary condition

- Find $t_i \in [0, h]$ such that $P(t_i)$ is in triangle plane

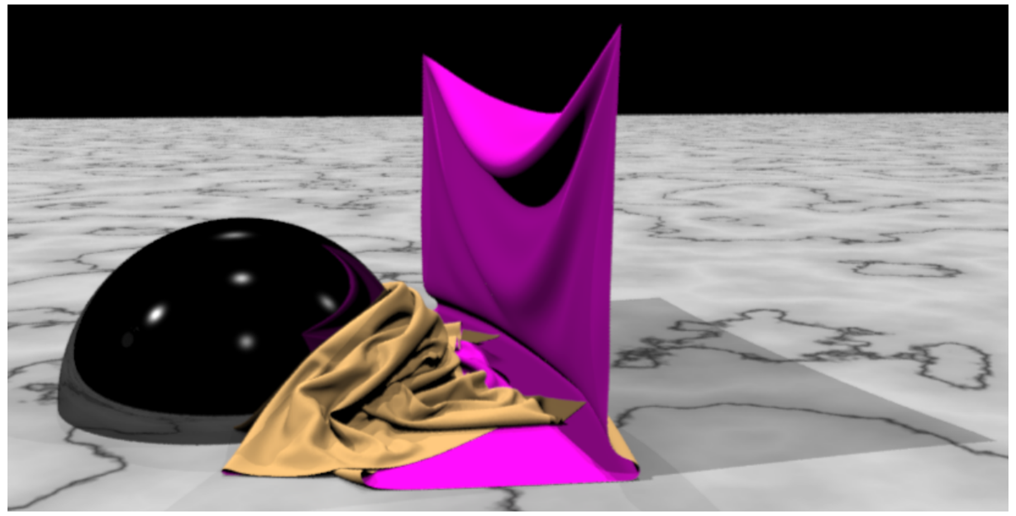
$$(P(t_i) - P_1(t_i)) \times n(t_i) = 0$$

$n(t_i)$: normal of the triangle at time t_i

Sufficient condition

- Check $P(t_i)$ is inside the triangle

$$P(t_i) = \alpha P_1(t_i) + \beta P_2(t_i) + \gamma P_3(t_i)$$
$$(\alpha, \beta, \gamma) \in [0, 1]^3, \alpha + \beta + \gamma = 1$$



[X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. Graphics Interface 1997.]

[R. Bridson et al. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. ACM SIGGRAPH 2002]