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).

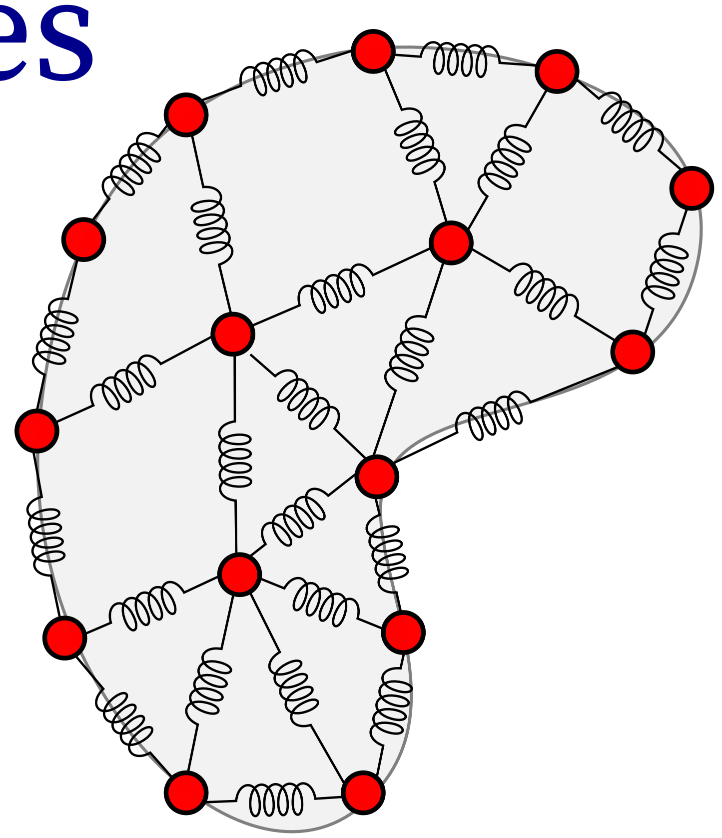
- Ductile material - can allow large amount of plastic deformation without breaking (plastic)
- Brittle - Opposite (glass, ceramics)

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.

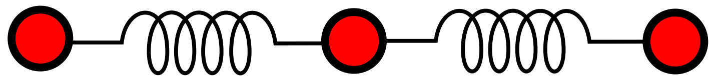
Modeling elastic shapes with particles



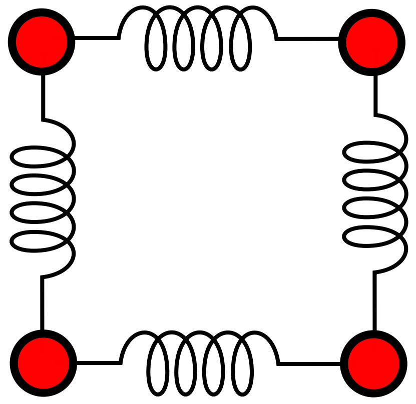
Spring mass systems

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

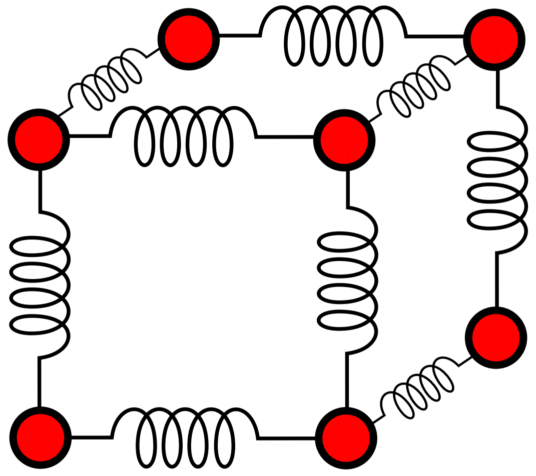
1D curve structure



2D surface structure



3D volume structure

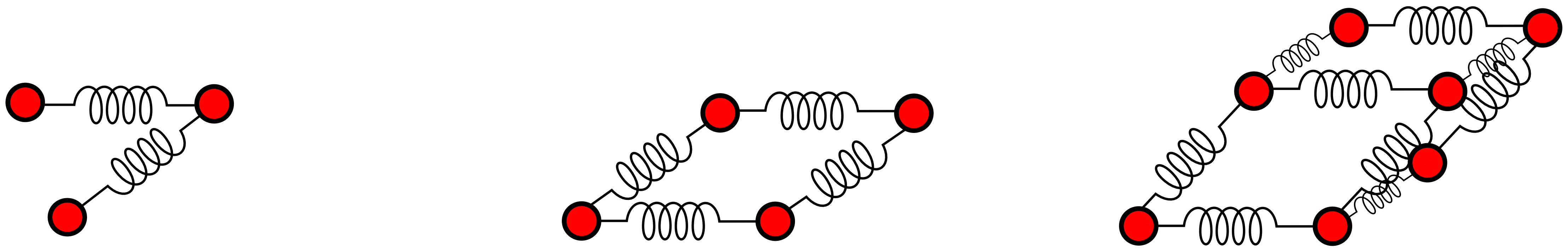


Spring structure

How to model spring connectivity ?

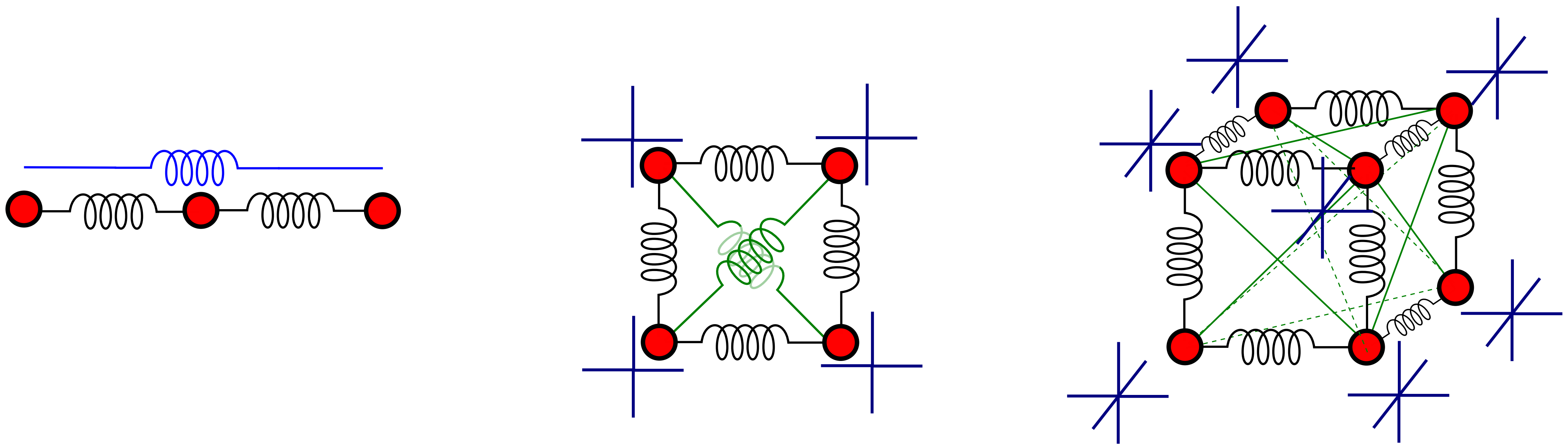
- **Structural springs**: 1-ring neighbors springs ($\simeq$ mesh edges)

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



$\Rightarrow$ Add extra springs connectivity

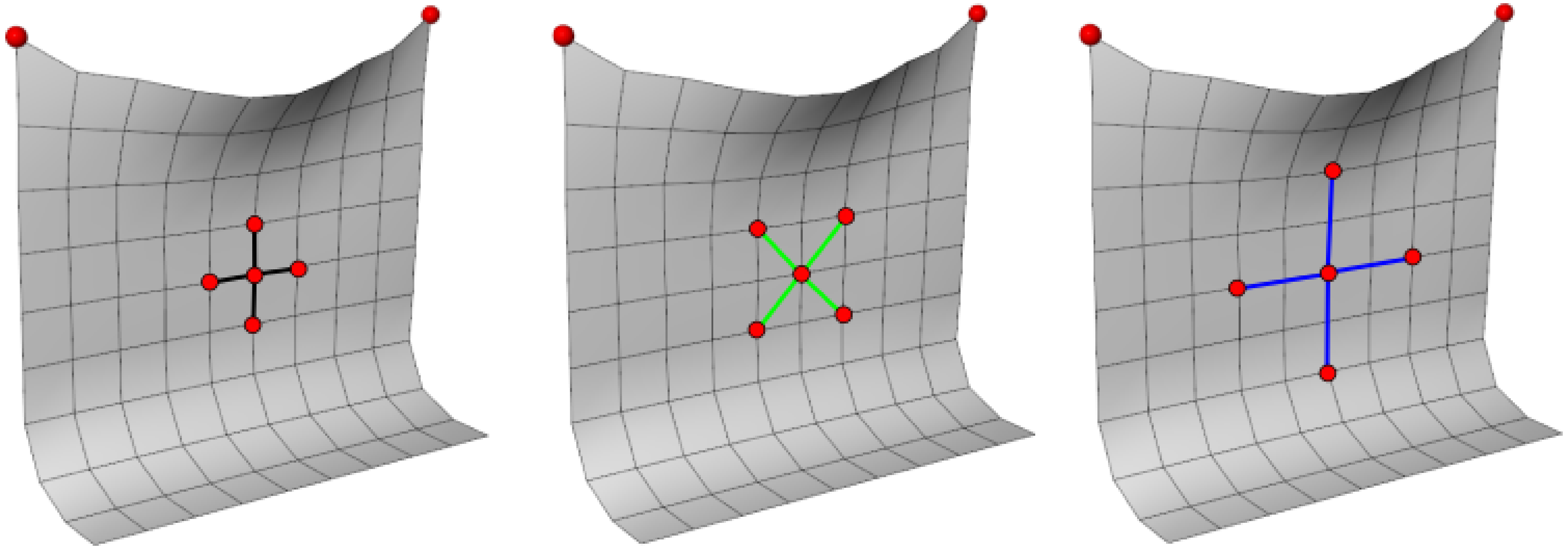
- **Shearing springs**: Diagonal links
- **Bending springs**: 2-ring neighborhood



Cloth Simulation

Mass-spring cloth simulation

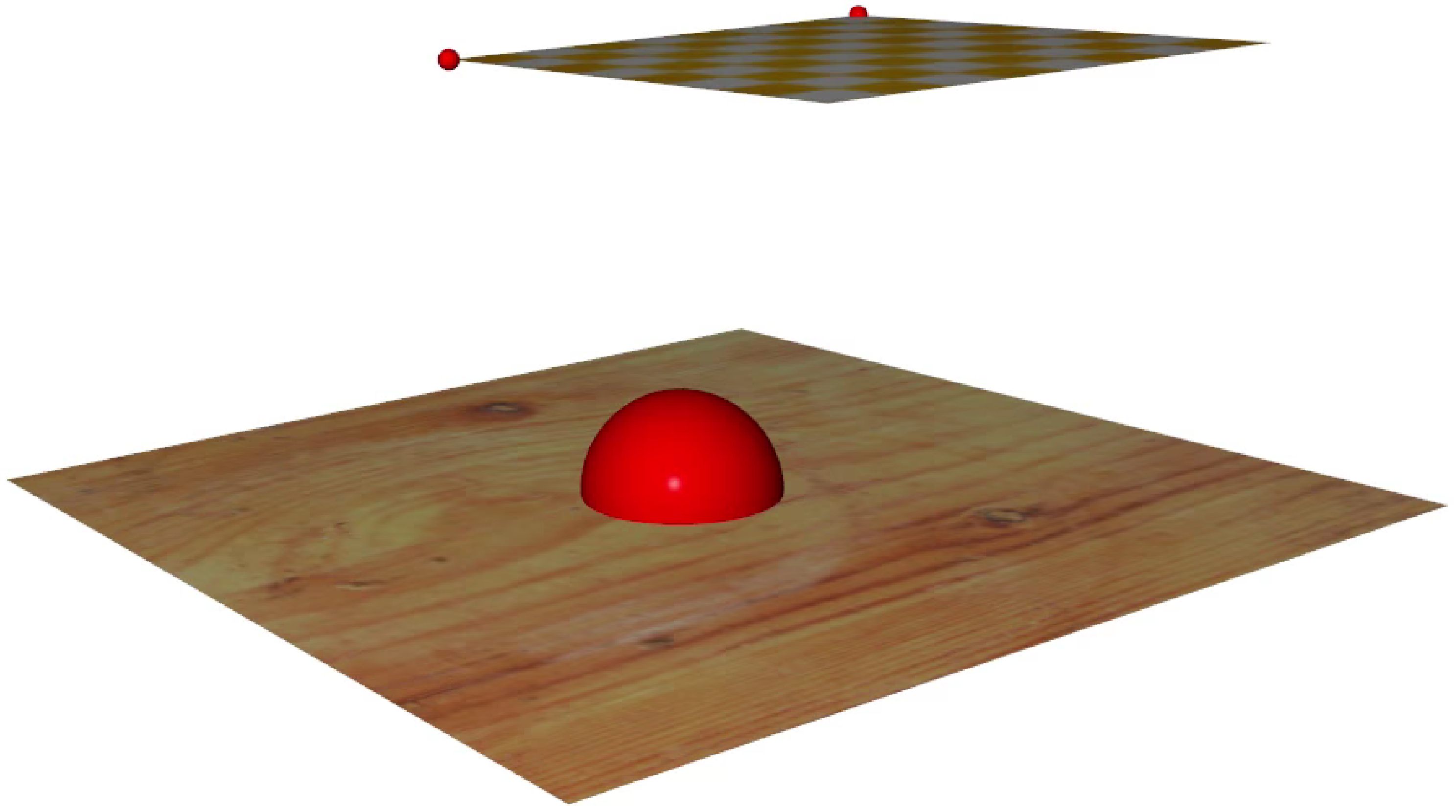
- Particles are sampled on a $N \times N$ grid.
  - Each particle has a mass m ($m_{cloth} = N^2 m$)
- Set structural, shearing and bending springs.



Forces

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

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

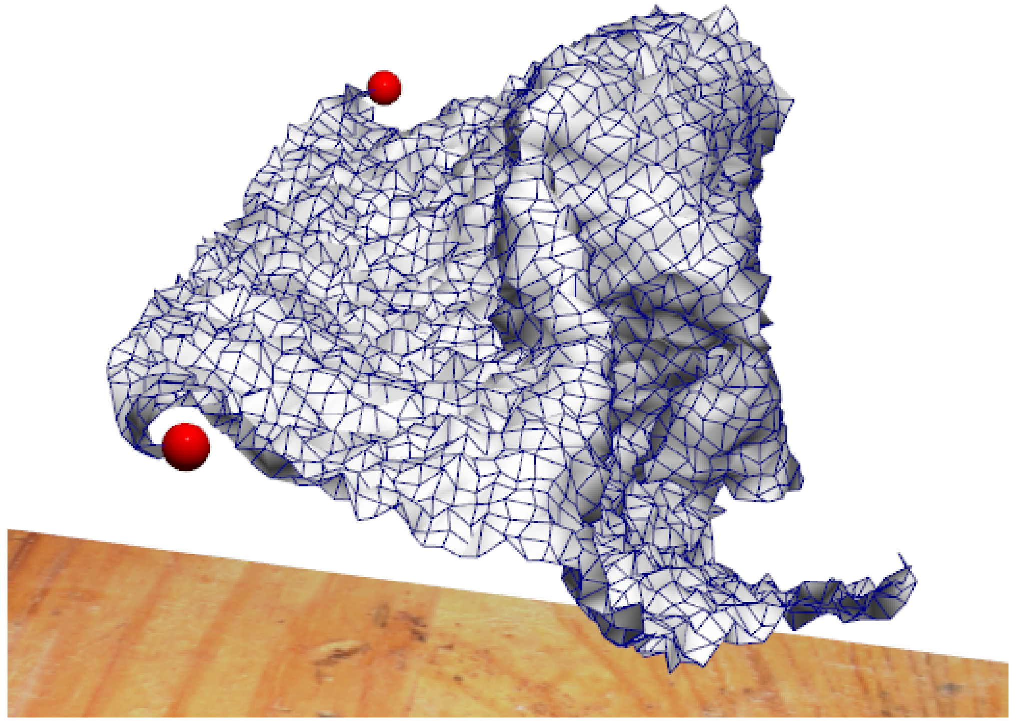


Q. How can we model the effect of the wind ?

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

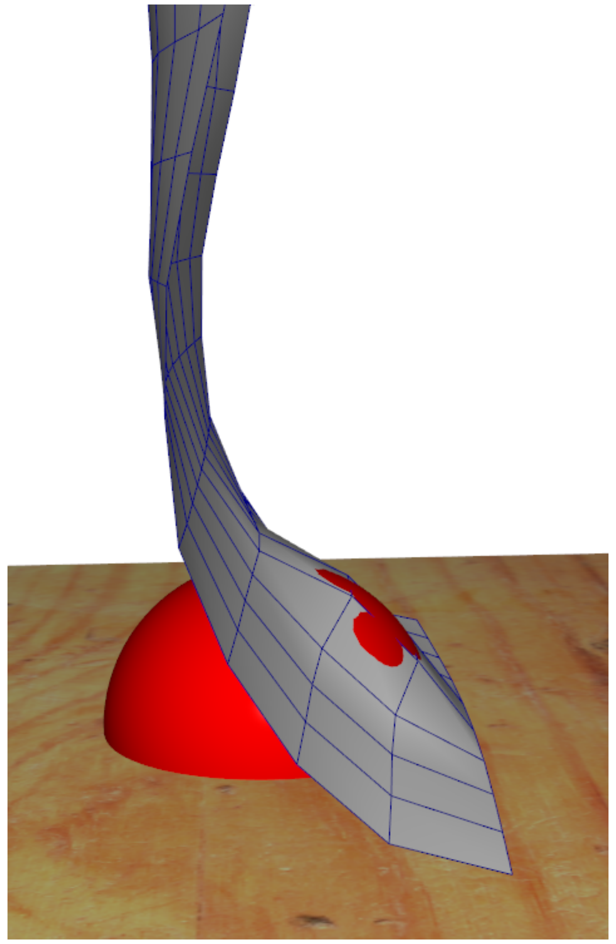
Collisions

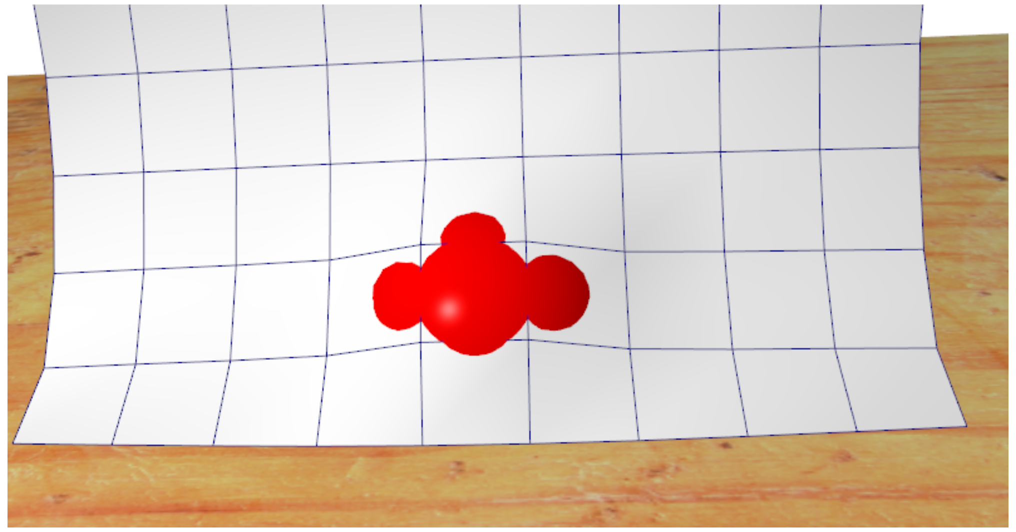
- Simple approach : Handled as collision between particles and shapes

  (+) Simple and efficient

  (-) Collision may still appears within a triangle

  ⇒ Exaustive approach: edges + faces





Limitation of mass spring model and continuous model

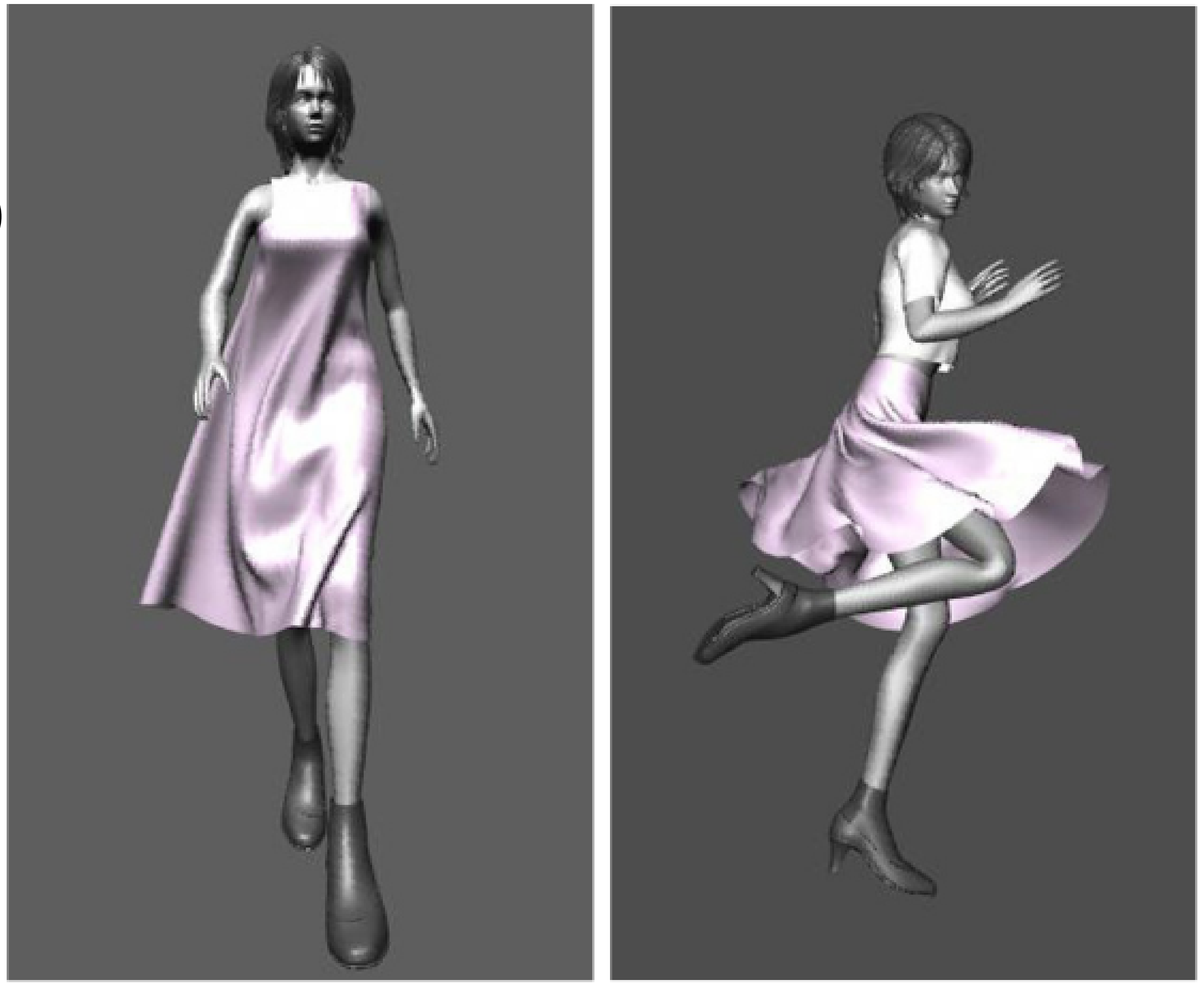
- Does mass-spring system converge toward a unique solution when sampling increase ?

  $\Rightarrow$ 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)

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

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

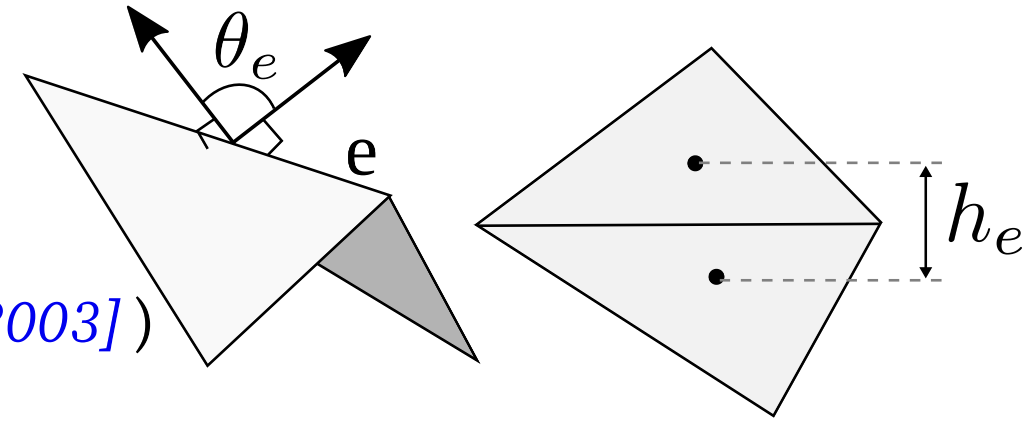
[Cho et al, Stable but Responsive Cloth, ACM SIGGRAPH 2002]

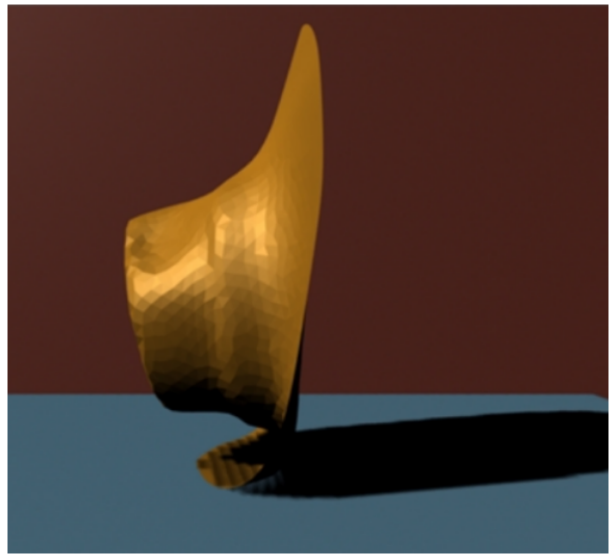
Triangle as continuous elements

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

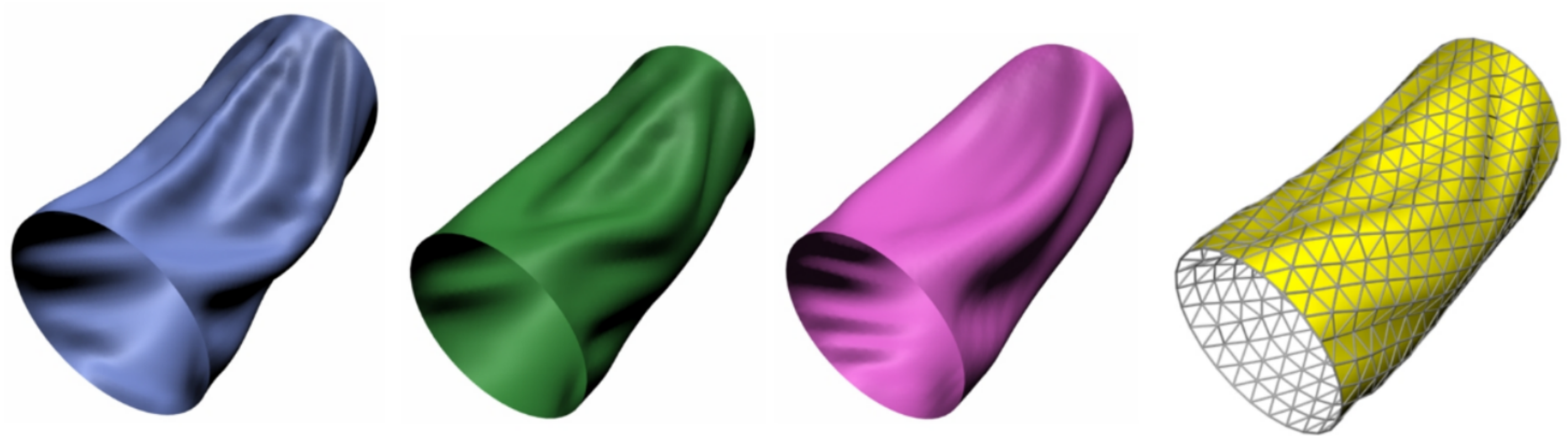
    [E. Grinspun et al., Discrete Shells, SCA 2003]

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



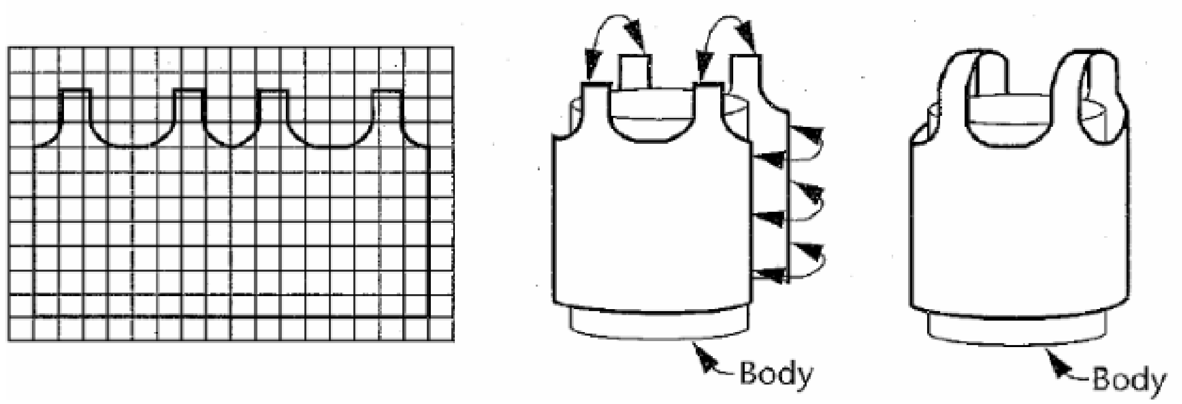


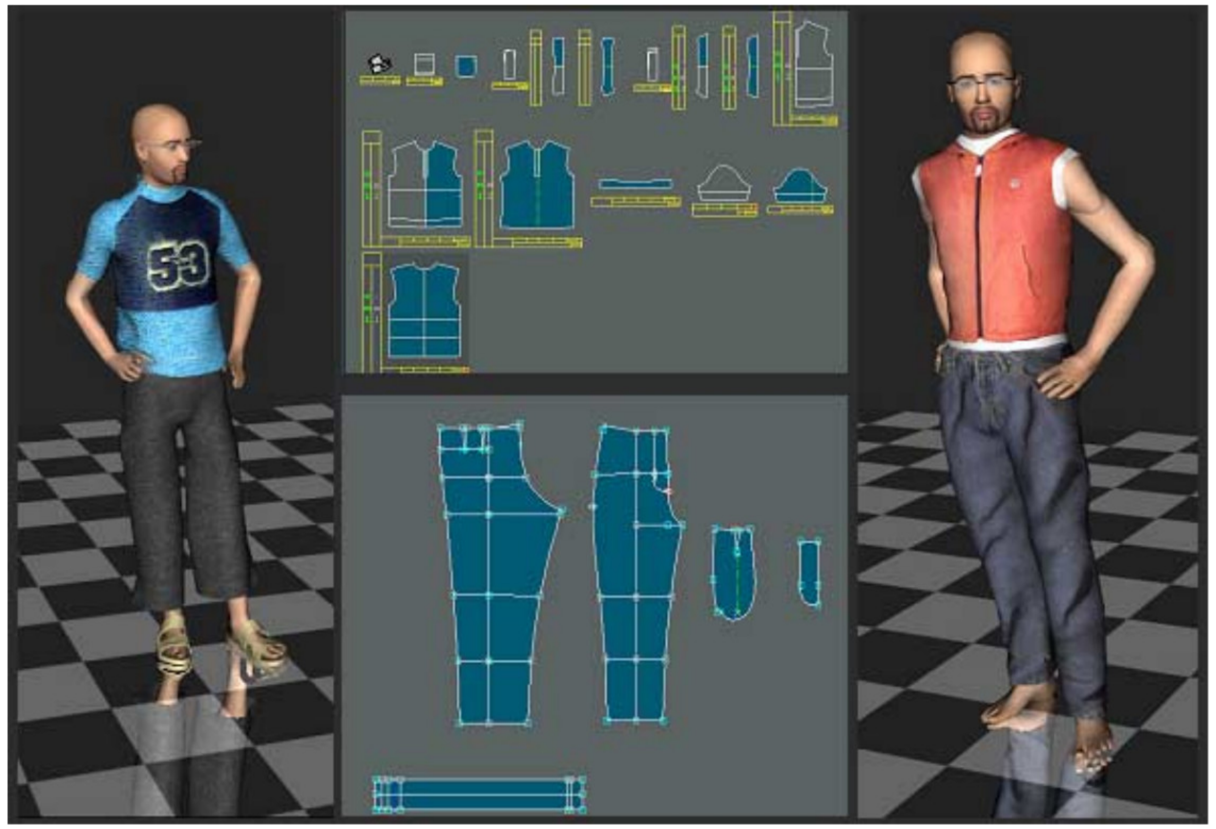
- Going toward full FEM numerical resolution
  - B. Thomaszewski et al. [SCA 2006], [VRIPHYS 2008], [EG 2009].



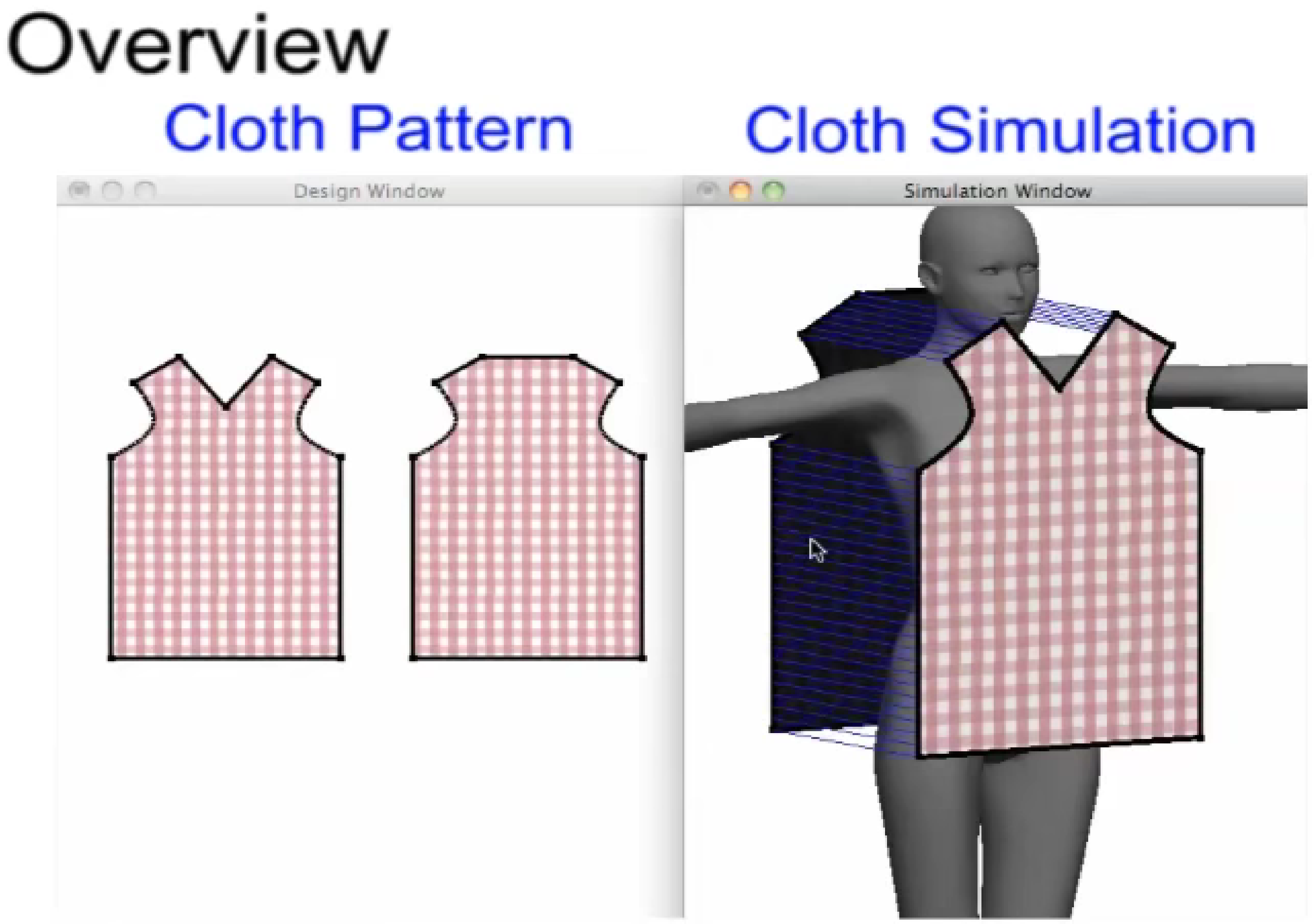
Clothing

- 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

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

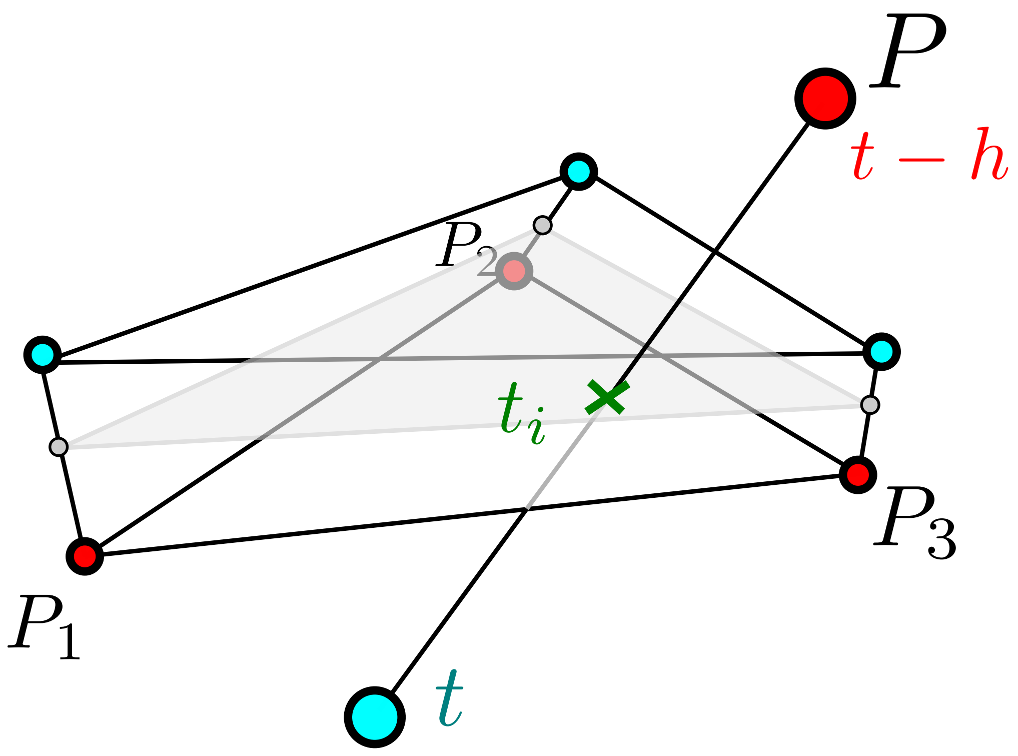
$n(t_i)$: normal of the triangle at time t_i

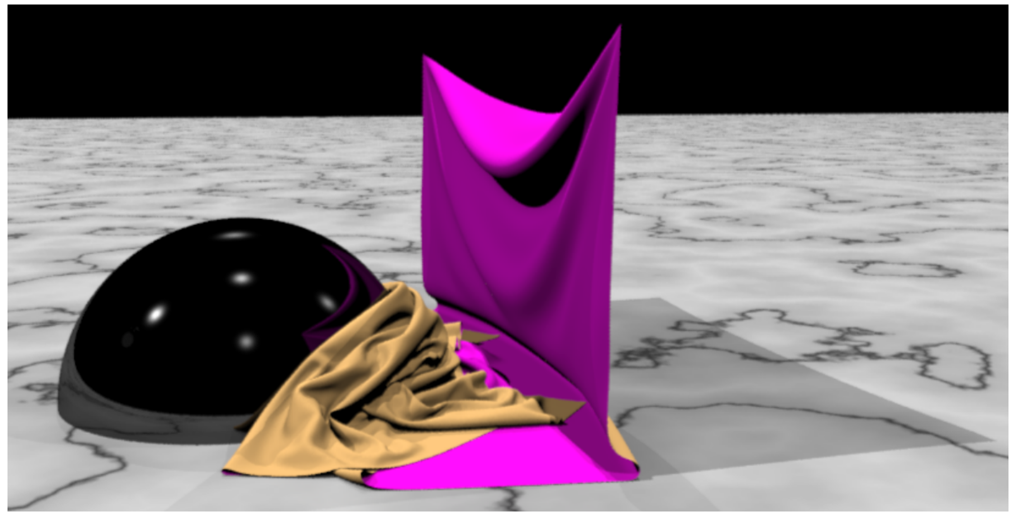
Sufficient condition

- 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]