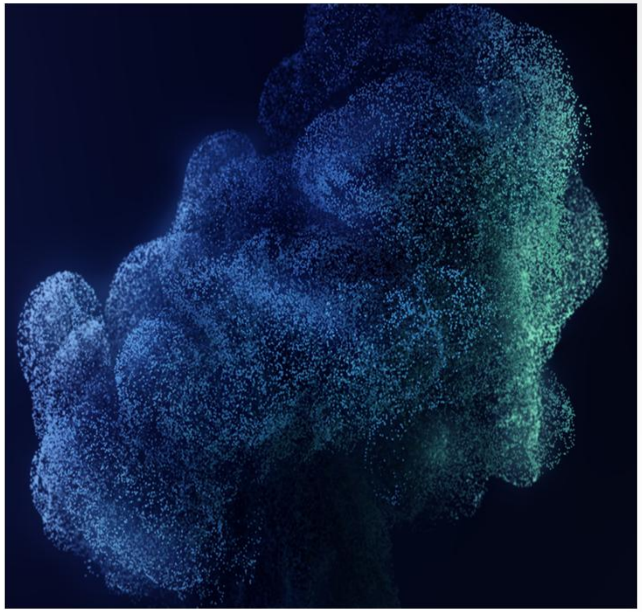
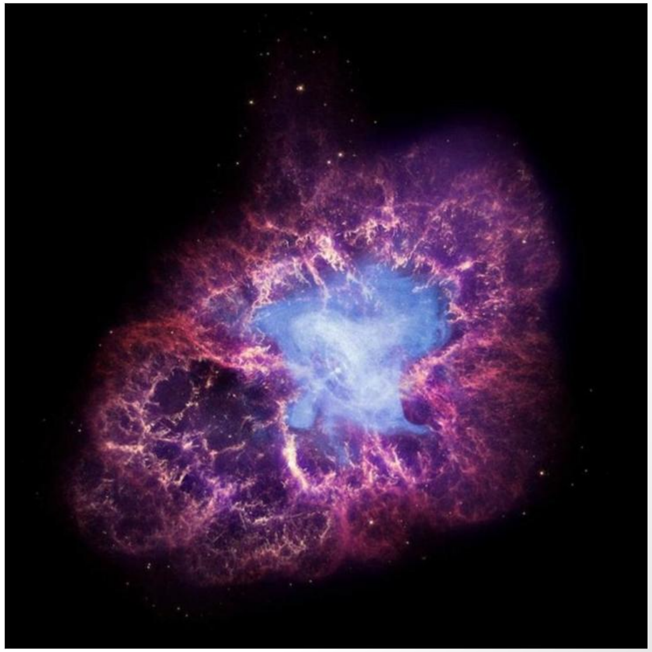
# Particle Systems





By Toni Susín (2015)

# Particle Systems

- Vectors of particles (big numParticles)
- **Initialization**:
  - **Fountain:**
    - Position: (0, 0, 0)
    - Velocity: ((rand01()-0.5), 10, (rand01()-0.5))

```
#include <random>
#define rand01() ((float)std::rand()/RAND_MAX)
```

# Particle Systems

- Vectors of particles (big numParticles)
- **Initialization**:
  - **Fountain:**
    - Position: (0, 0, 0)
    - Velocity: ((rand01()-0.5), 10, (rand01()-0.5))
  - **Waterfall:**
    - Position: (0, 10, 0)
    - Velocity: ((rand01()-0.5), 0, (rand01()-0.5))

```
#include <random>
#define rand01() ((float)std::rand()/RAND_MAX)
```

# Particle Systems

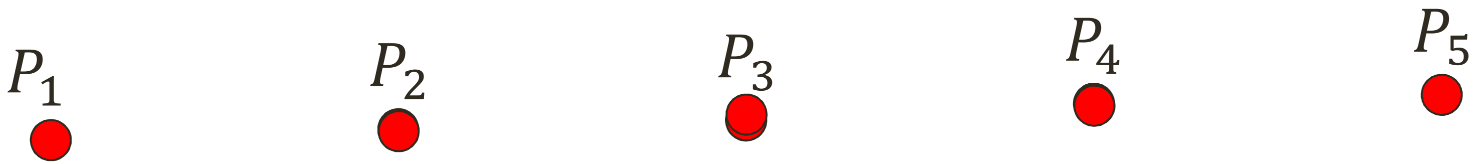
- Vectors of particles (big numParticles)
- **Initialization**:
  - **Fountain:**
    - Position: (0, 0, 0)
    - Velocity: ((rand01()-0.5), 10, (rand01()-0.5))
  - **Waterfall:**
    - Position: (0, 10, 0)
    - Velocity: ((rand01()-0.5), 10, (rand01()-0.5))
  - **Semi-Sphere:**
    - Azimut, $\alpha$ =360*(rand01()-0.5)
    - Altitude, $\beta$ = 90*rand01()
    - Position: $(\cos(\alpha) \cdot \cos(\beta),\ \ \sin(\beta),\ \ \sin(\alpha) \cdot \cos(\beta))$
    - Velocity: speed*(position.x, position.y, position.z)

# Particle Systems

- Vectors of particles (big numParticles)
- **Initialization**:
  - **Explosion:**
    - Azimut, $\alpha$ =360*(rand01()-0.5)
    - Altitude, $\beta$ = 180*(rand01()-0.5)
    - Position: $0.01 \cdot (\cos(\alpha) \cdot \cos(\beta),\ \ \sin(\beta)\,,\ \ \sin(\alpha) \cdot \cos(\beta))$
    - Velocity: speed*(position.x, position.y, position.z)

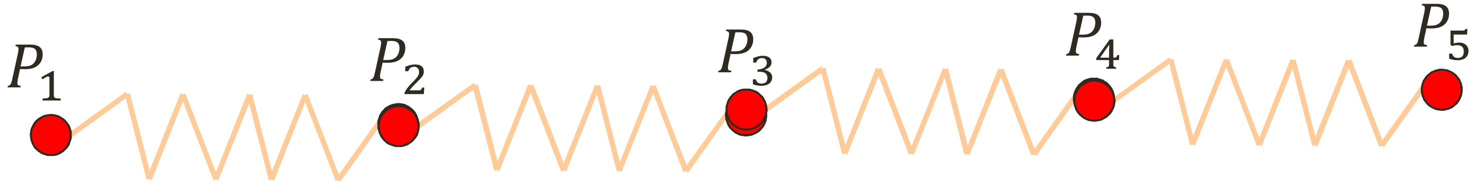
# Interaction between Particles

- Initial Particles



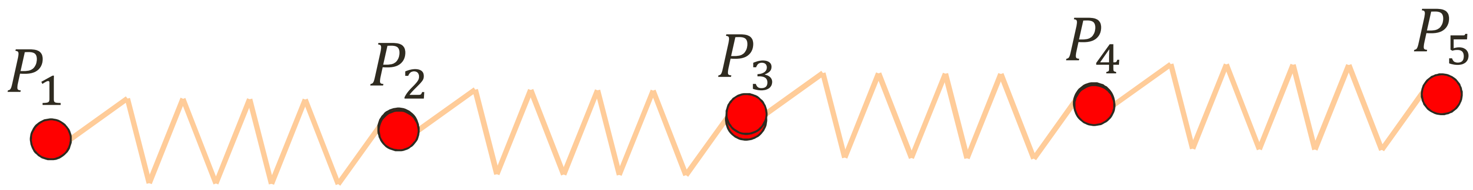
# Interaction between Particles

- Initial Particle interaction: add springs



# Mesh-Spring 1-Dim Models

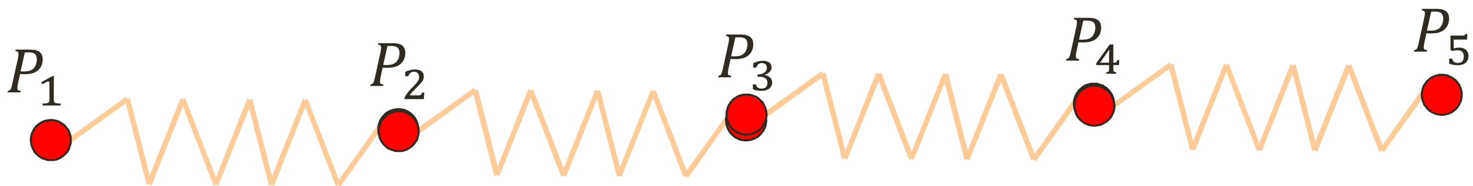
- **Exemple: 1D Mesh-Springs (Ropes)**




- Spring Forces between neigbourgs:
  - Num particles= Num Springs + 1
  - Spring parameters: Elasticity and damping

# Mesh-Spring 1-Dim Models

- **Exemple: 1D Mesh-Springs (Ropes)**



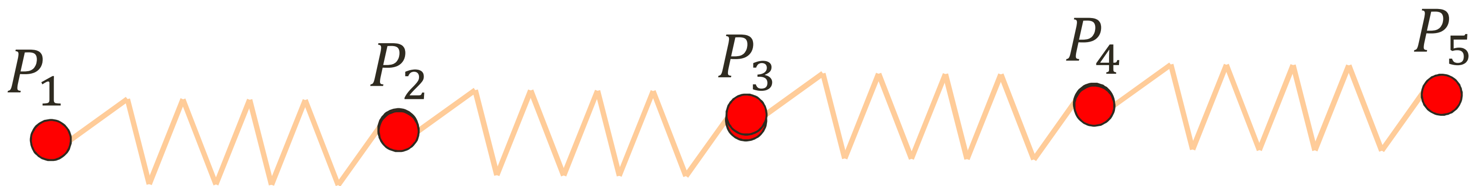

Particle 1

$$F_1^{m_1}(P_1, P_2) = \left(k_e \cdot \left(||P_2 - P_1|| - L_{12}\right) + k_d \cdot (v_2 - v_1) \cdot \frac{P_2 - P_1}{||P_2 - P_1||}\right) \cdot \frac{P_2 - P_1}{||P_2 - P_1||}$$

$$F_2^{m_1}(P_1, P_2) = -F_1^{m_1}$$

# Mesh-Spring 1-Dim Models

- **Exemple: 1D Mesh-Springs (Ropes)**




Particle 1

$$F_1^{m_1}(P_1,P_2) = \left(k_e \cdot \left(||P_2 - P_1|| - L_{12}\right) + k_d \cdot (v_2 - v_1) \cdot \frac{P_2 - P_1}{||P_2 - P_1||}\right) \cdot \frac{P_2 - P_1}{||P_2 - P_1||}$$

$$F_2^{m_1}(P_1,P_2) = -F_1^{m_1}$$

$$\boxed{F_1^{total} = F_1^{m_1}}$$

Particle 2

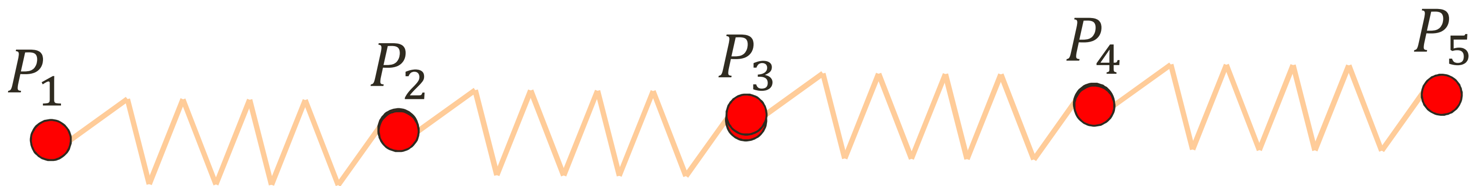
$$F_1^{m_2}(P_2,P_3) = \left(k_e \cdot \left(||P_3 - P_2|| - L_{23}\right) + k_d \cdot (v_3 - v_2) \cdot \frac{P_3 - P_2}{||P_3 - P_2||}\right) \cdot \frac{P_3 - P_2}{||P_3 - P_2||}$$

$$F_2^{m_2}(P_2,P_3) = -F_1^{m_2}$$

$$\boxed{F_2^{total} = F_2^{m_1} + F_1^{m_2}}$$

# Mesh-Spring 1-Dim Models

- **Exemple: 1D Mesh-Springs (Ropes)**




fixed

# Mass-Spring

- **Algorithm:**

// *initialization*
(1) **forall** particles $i$
(2) initialize $\mathbf{x}_i, \mathbf{v}_i$ and $m_i$
(3) **endfor**
// *simulation loop*
(4) **loop**
(5) **forall** particles $i$
(6) $\mathbf{f}_i \leftarrow \mathbf{f}^g + \mathbf{f}_i^{\text{coll}} + \sum_{j,(i,j)\in S} \mathbf{f}(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j)$
(7) **endfor**
(8) **forall** particles $i$
(9) $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t\, \mathbf{f}_i / m_i$
(10) $\mathbf{x}_i \leftarrow \mathbf{x}_i + \Delta t\, \mathbf{v}_i$
(11) **endfor**
(12) display the system every $n^{th}$ time
(13) **endloop**

## Spring Force:

$$\mathbf{f}(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) = \mathbf{f}^s(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{f}^d(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j)$$

## Elastic:

$$\mathbf{f}_i = \mathbf{f}^s(\mathbf{x}_i, \mathbf{x}_j) = k_s \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}(|\mathbf{x}_j - \mathbf{x}_i| - l_0)$$

$$\mathbf{f}_j = \mathbf{f}^s(\mathbf{x}_j, \mathbf{x}_i) = -\mathbf{f}^s(\mathbf{x}_i, \mathbf{x}_j) = -\mathbf{f}_i$$

## Damping:

$$\mathbf{f}_i = \mathbf{f}^d(\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}_j, \mathbf{v}_j) = k_d(\mathbf{v}_j - \mathbf{v}_i) \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}$$

$$\mathbf{f}_j = \mathbf{f}^d(\mathbf{x}_j, \mathbf{v}_j, \mathbf{x}_i, \mathbf{v}_i) = -\mathbf{f}_i$$

Direction missing!!

$$\frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}$$

# Verlet Solver

- Taylor:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{x}}(t)\Delta t^2 + \frac{1}{6}\dddot{\mathbf{x}}(t)\Delta t^3 + O(\Delta t^4)$$

$$\mathbf{x}(t-\Delta t) = \mathbf{x}(t) - \dot{\mathbf{x}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{x}}(t)\Delta t^2 - \frac{1}{6}\dddot{\mathbf{x}}(t)\Delta t^3 + O(\Delta t^4)$$

# Verlet Solver

- Taylor:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{x}}(t)\Delta t^2 + \frac{1}{6}\dddot{\mathbf{x}}(t)\Delta t^3 + O(\Delta t^4)$$

$$\mathbf{x}(t-\Delta t) = \mathbf{x}(t) - \dot{\mathbf{x}}(t)\Delta t + \frac{1}{2}\ddot{\mathbf{x}}(t)\Delta t^2 - \frac{1}{6}\dddot{\mathbf{x}}(t)\Delta t^3 + O(\Delta t^4)$$

- Adding:

$$\begin{aligned}\mathbf{x}(t+\Delta t) &= 2\mathbf{x}(t) - \mathbf{x}(t-\Delta t) + \ddot{\mathbf{x}}(t)\Delta t^2 + O(\Delta t^4) \\ &= \mathbf{x}(t) + [\mathbf{x}(t) - \mathbf{x}(t-\Delta t)] + \mathbf{f}(t)\,\Delta t^2/m + O(\Delta t^4)\end{aligned}$$

**Fast, stable, low precision**

$$\boxed{x_{t+dt} = x_t + k_d(x_t - x_{t-dt}) + \Delta t^2\,\frac{\mathrm{f}(t)}{m}}$$

# Implicit Euler Method

- Euler Method

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \Delta t\, \mathbf{f}(\mathbf{x}^t, \mathbf{v}^t)/m$$
$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t\, \mathbf{v}^t.$$

- Implicit Euler Method

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \Delta t\, \mathbf{f}(\mathbf{x}^{t+1})/m$$
$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t\, \mathbf{v}^{t+1}.$$

# Implicit Euler Method

$$\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T]^T$$
$$\mathbf{v} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_n^T]^T$$
$$\mathbf{f}(\mathbf{x}) = [\mathbf{f}_1(\mathbf{x}_1, \ldots, \mathbf{x}_n)^T, \ldots \mathbf{f}_n(\mathbf{x}_1, \ldots \mathbf{x}_n)^T]^T$$

$$\mathbf{M} \in \mathbb{R}^{3n \times 3n} \quad m_1, m_1, m_1,\ m_2, m_2, m_2,\ \ldots, m_n, m_n, m_n$$

$$\mathbf{M}\mathbf{v}^{t+1} = \mathbf{M}\mathbf{v}^t + \Delta t \mathbf{f}(\mathbf{x}^{t+1})$$
$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta \mathbf{v}^{t+1}$$

$$\mathbf{M}\mathbf{v}^{t+1} = \mathbf{M}\mathbf{v}^t + \Delta t \mathbf{f}(\mathbf{x}^t + \Delta t \mathbf{v}^{t+1})$$

# Implicit Euler Method

$$\mathbf{M}\mathbf{v}^{t+1} = \mathbf{M}\mathbf{v}^t + \Delta t \mathbf{f}(\mathbf{x}^t + \Delta t \mathbf{v}^{t+1})$$

Taylor

$$\mathbf{M}\mathbf{v}^{t+1} = \mathbf{M}\mathbf{v}^t + \Delta t \left[ \mathbf{f}(\mathbf{x}^t) + \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}^t) \cdot (\Delta t \, \mathbf{v}^{t+1}) \right]$$

$$= \mathbf{M}\mathbf{v}^t + \Delta t \mathbf{f}(\mathbf{x}^t) + \Delta t^2 \mathbf{K}\mathbf{v}^{t+1},$$

$\mathbf{K} \in \mathbb{R}^{3n \times 3n}$ is the Jacobian of $\mathbf{f}$.

Tangent Stiffness Matrix

$$\left[\mathbf{M} - \Delta t^2 \mathbf{K}\right] \mathbf{v}^{t+1} = \mathbf{M}\mathbf{v}^t + \Delta t \mathbf{f}(\mathbf{x}^t)$$

$$\mathbf{A}\mathbf{v}^{t+1} = \mathbf{b},$$

Linear System

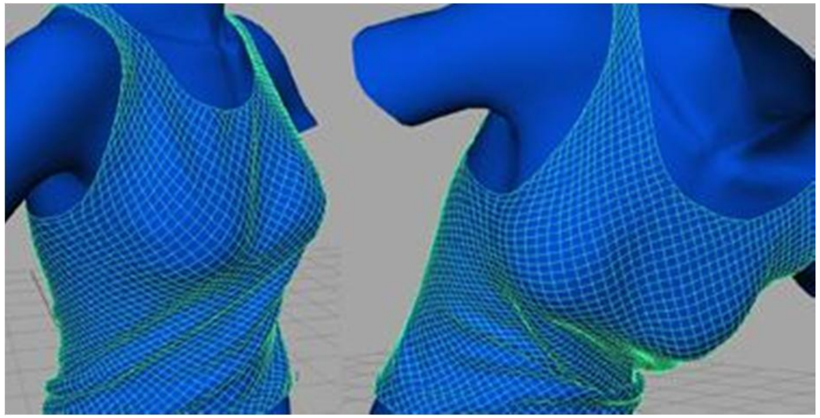
# Implicit Euler Method

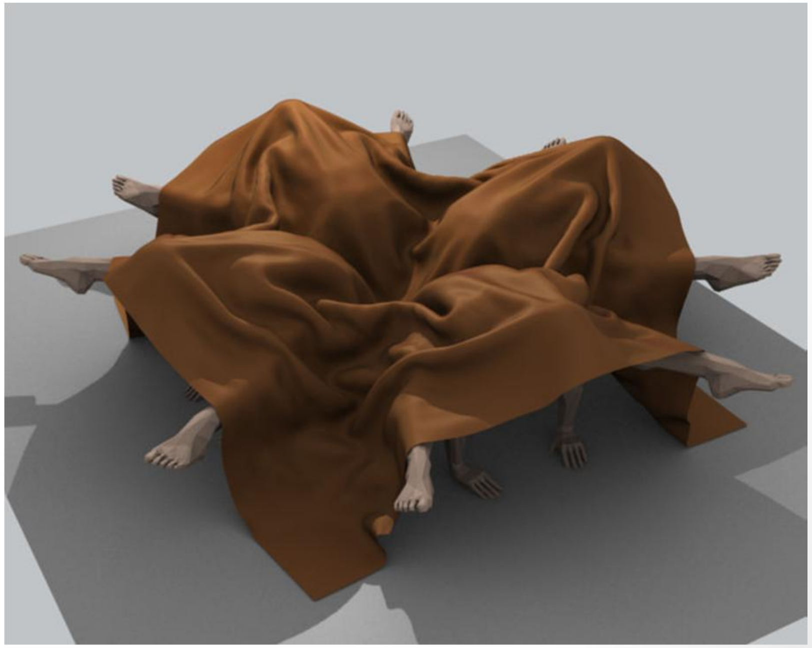
Now let us have a look at $\mathbf{K}$. A spring force between particles $i$ and $j$ adds the four $3\times 3$ sub-matrices $\mathbf{K}_{i,i}$, $\mathbf{K}_{i,j}$, $\mathbf{K}_{j,i}$ and $\mathbf{K}_{j,j}$ to the global matrix $\mathbf{K}$ at positions $(3i,3i)$, $(3i,3j)$, $(3j,3i)$ and $(3j,3j)$ respectively. In order to evaluate these sub-matrices, we need to deduce

$$l = |\mathbf{x}_j - \mathbf{x}_i|$$

$$\begin{aligned}
\mathbf{K}_{i,i} &= \frac{\partial}{\partial \mathbf{x}_i}\mathbf{f}^s(\mathbf{x}_i,\mathbf{x}_j) \\
&= k_s \frac{\partial}{\partial \mathbf{x}_i}\left((\mathbf{x}_j-\mathbf{x}_i) - l_0 \frac{\mathbf{x}_j-\mathbf{x}_i}{|\mathbf{x}_j-\mathbf{x}_i|}\right) \\
&= k_s\left(-\mathbf{I} + \frac{l_0}{l}\left[\mathbf{I} - \frac{(\mathbf{x}_j-\mathbf{x}_i)(\mathbf{x}_j-\mathbf{x}_i)^T}{l^2}\right]\right) \\
&= -\mathbf{K}_{i,j} = \mathbf{K}_{j,j} = -\mathbf{K}_{j,i}
\end{aligned}$$

# Cloth Simulation

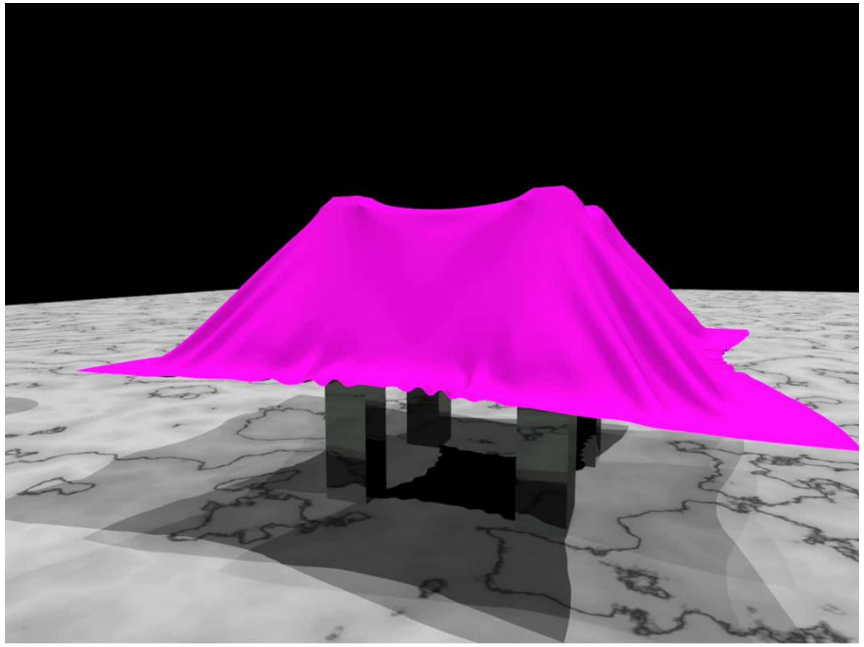


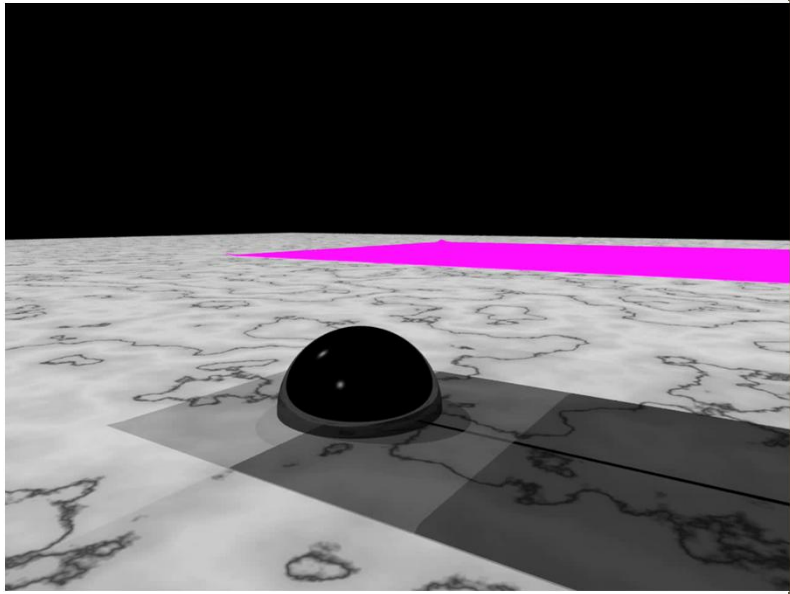


# Cloth Simulation

- Research:

(Fedkiw et al.-Stanford)





# Cloth Simulation

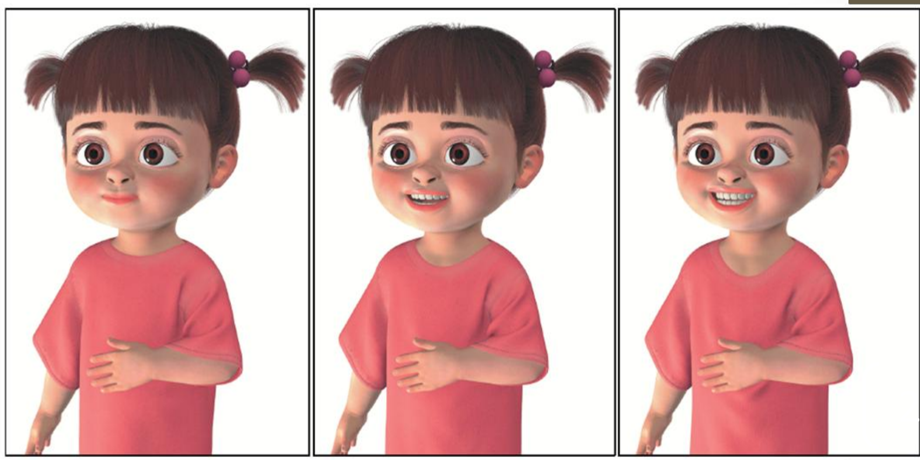
- Realism:

Nadia M.Thalman

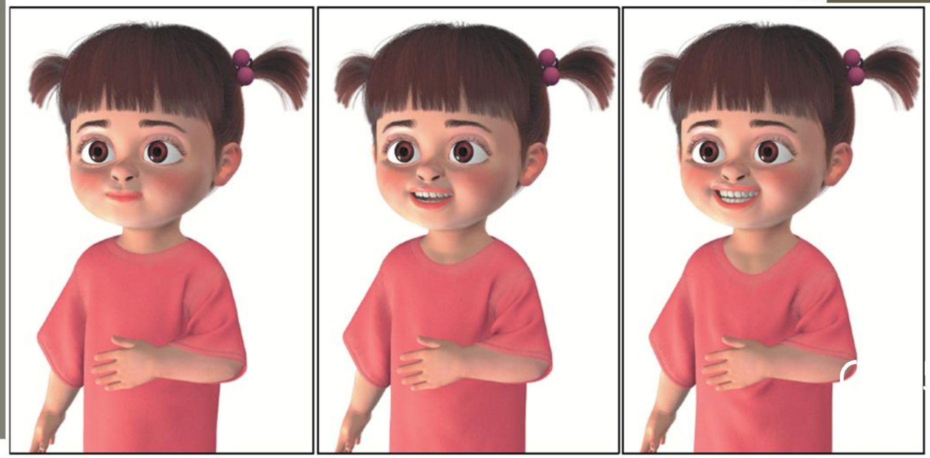
(MiRALAB-Geneve)

# Cloth Simulation

- Geometric Structure
- Dynamic Model
- Simulation on the GPU



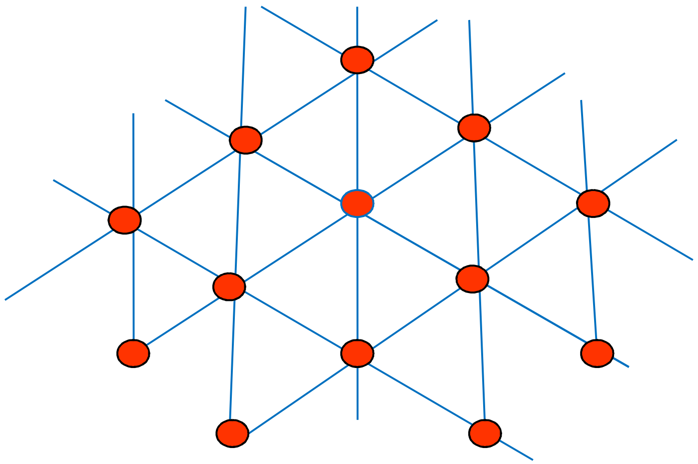
# Cloth Simulation

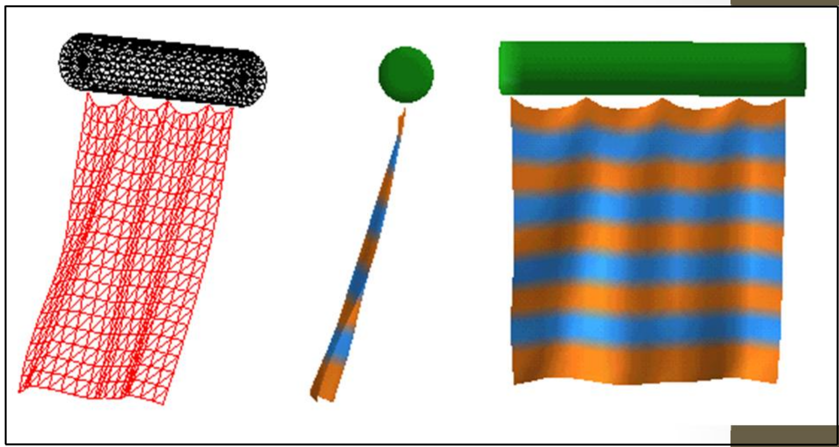


# Mesh-Spring 2-Dim Models

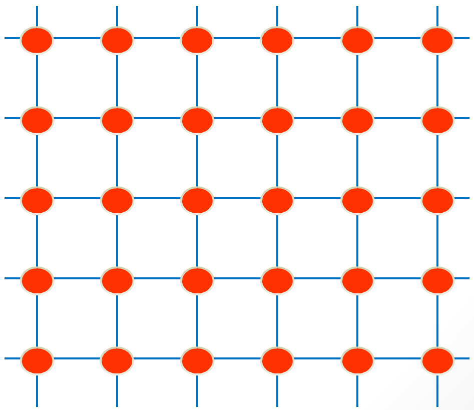
- **Exemple: 2D Mesh-Springs (Cloth)**

Triangular Mesh





Regular Mesh



# Force Types

$$F_i = \sum (F_{int} + F_{ext} + F_r)\ , \qquad i = 1, \ldots n$$

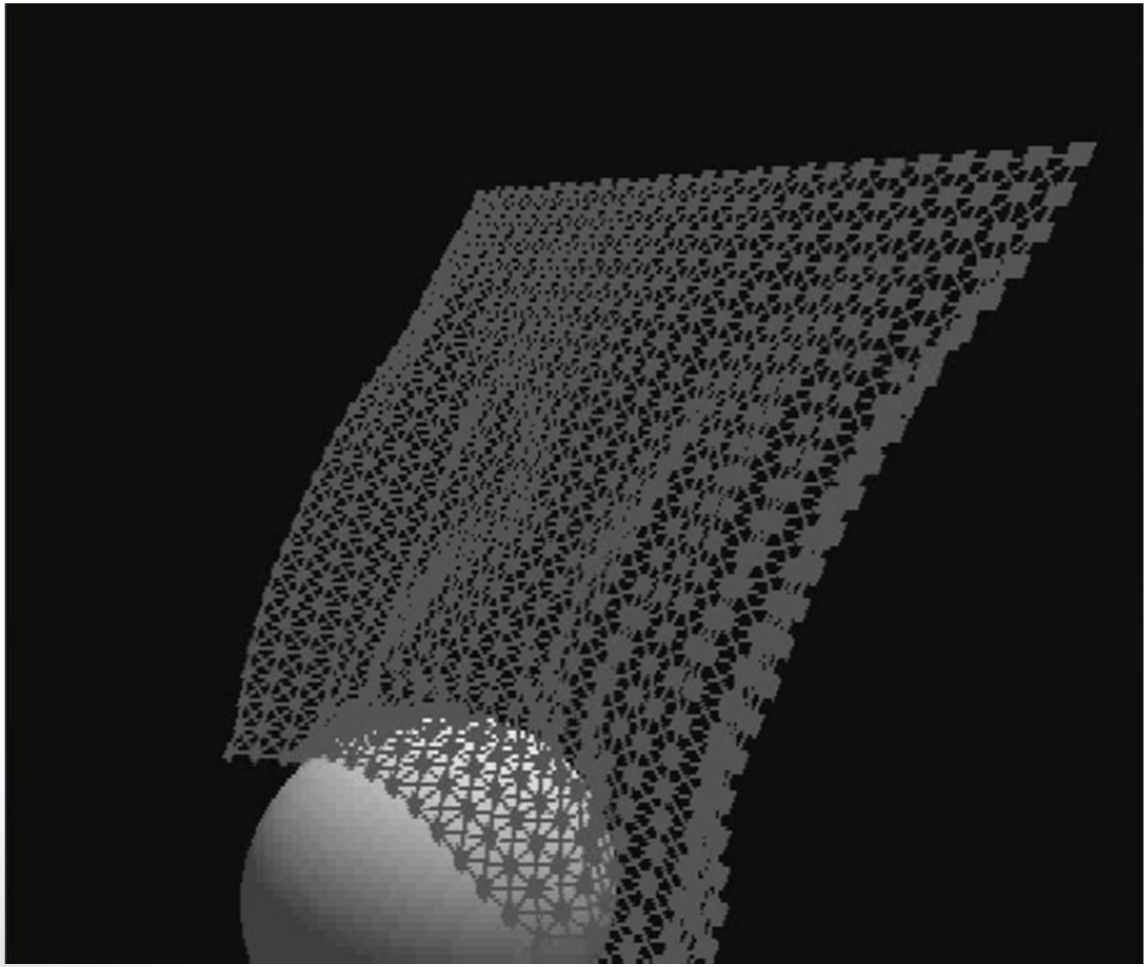
$F_{int}$ : **Internal Cloth Forces**

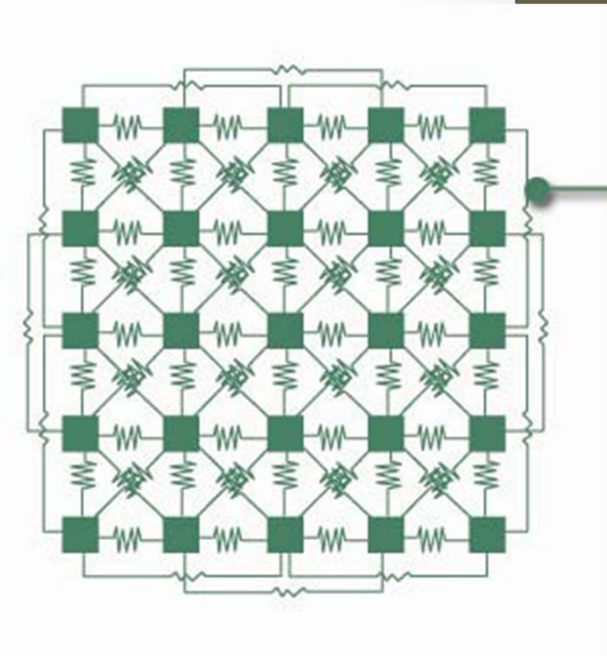
$F_{ext}$ : **External Forces**

$F_r$ : **Restriction Forces**

# Internal Force Computation: Provot

- Provot's Spring Model (95)

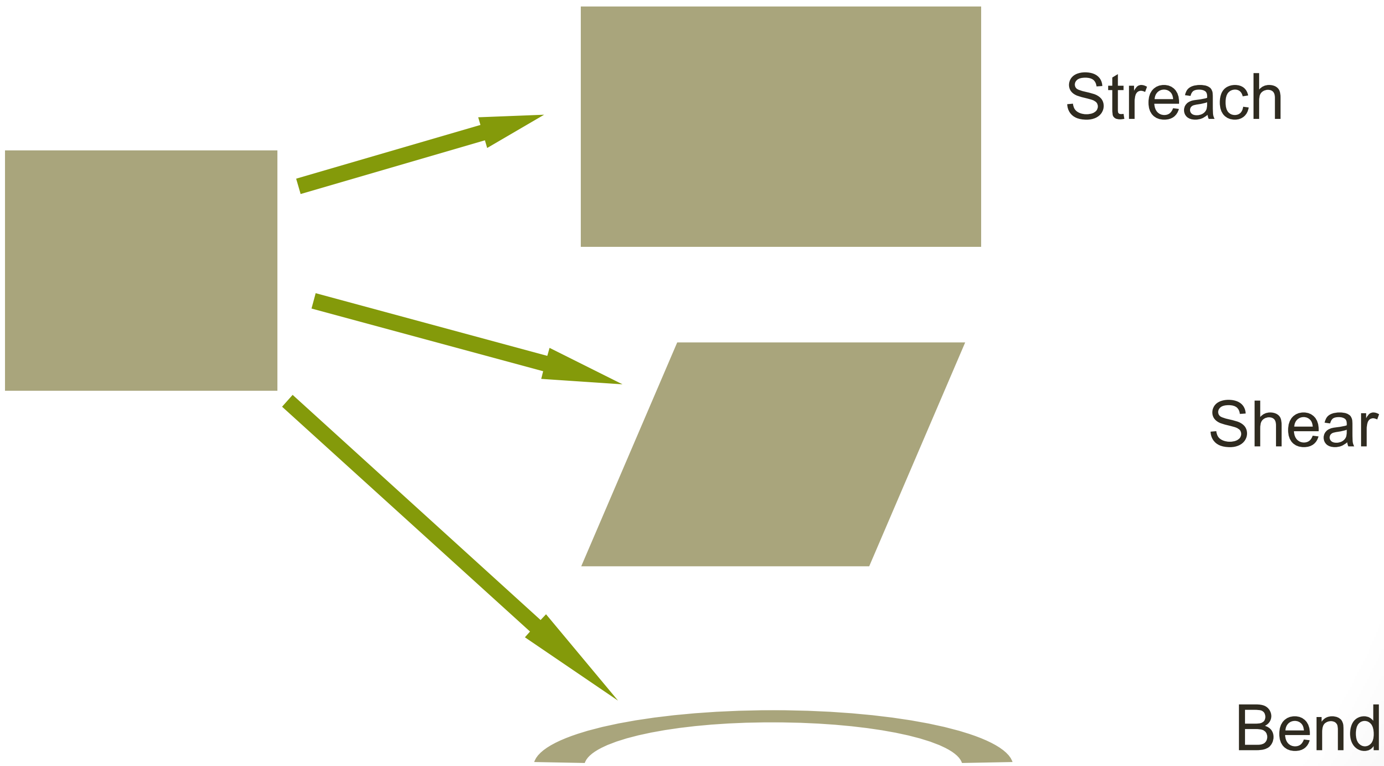




Particles are connected by Stretch Springs, Shear Springs and Bend Springs.

# Internal Elastic Forces

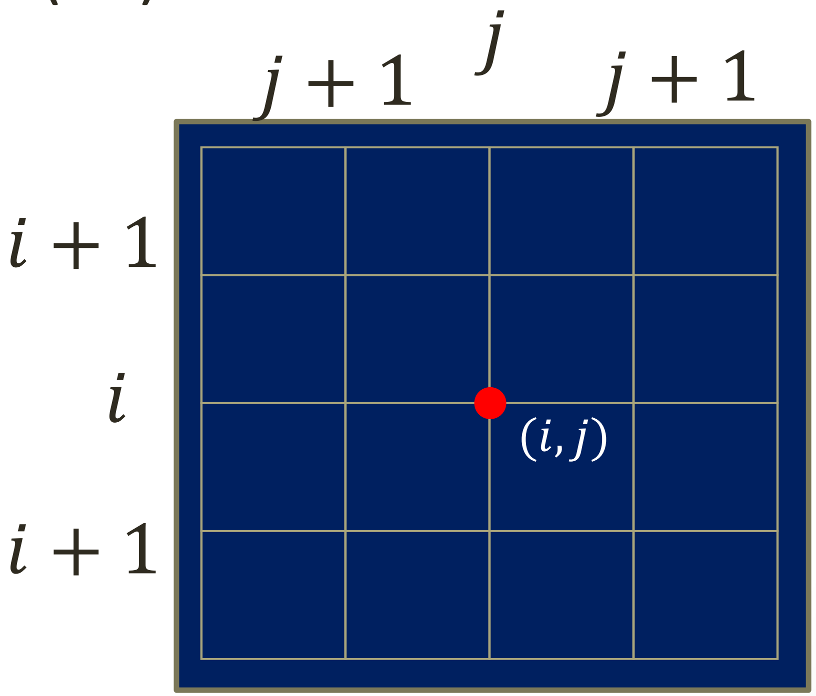
- 2D Deformation:



# Internal Force Computation: Mass-Spring

- Provot’s Spring Model (95)

(regular meshes)




Streach

Shear

Bend

# Internal Force Computation: Mass-Spring

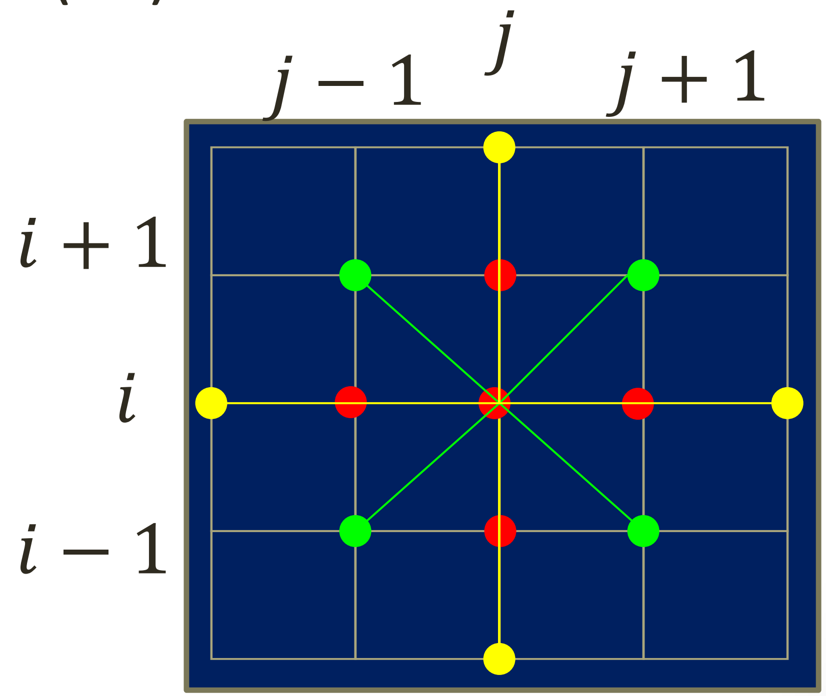
- Provot’s Spring Model (95)

(regular meshes)

Streach

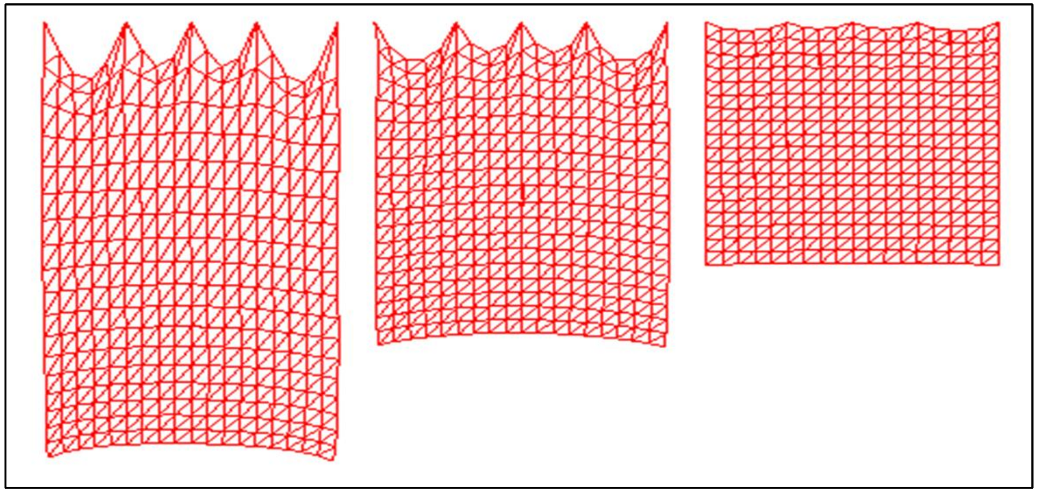
Shear

Bend



# Inverse Kinematics

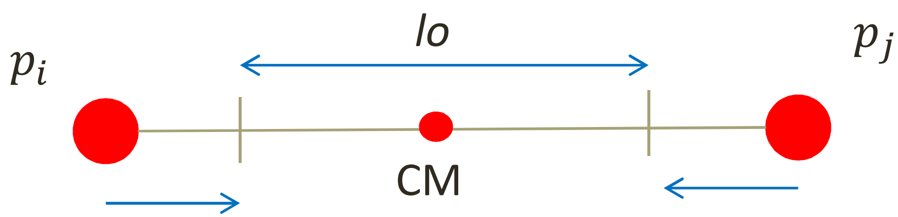
- Excessive elongation that must be corrected

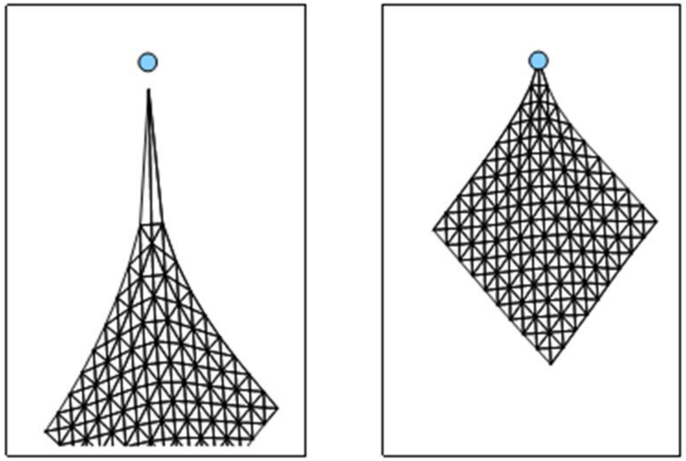


# Inverse Kinematics

- Excessive elongation correction

Loop for all springs (relaxation) until all distances are near de initial resting one.

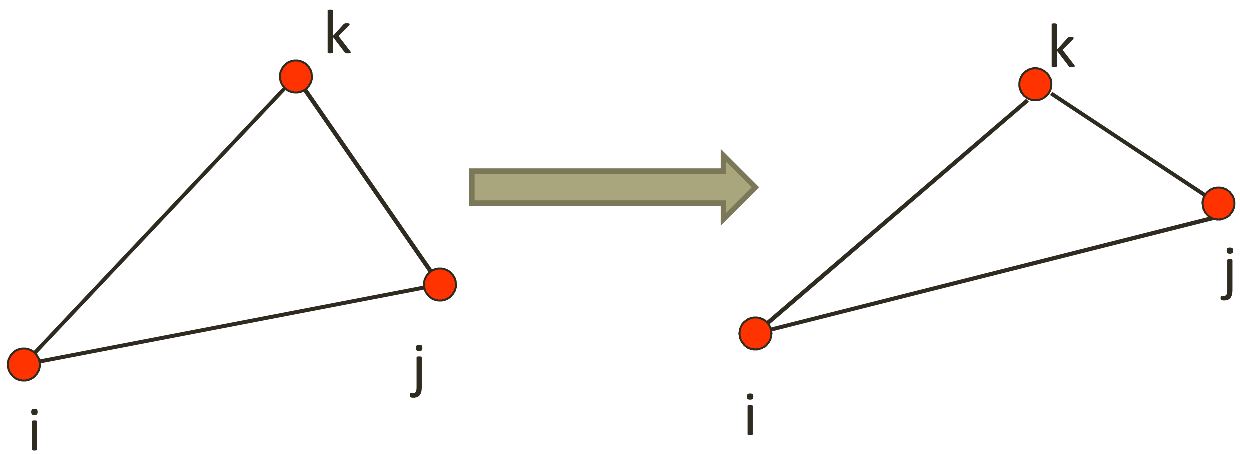


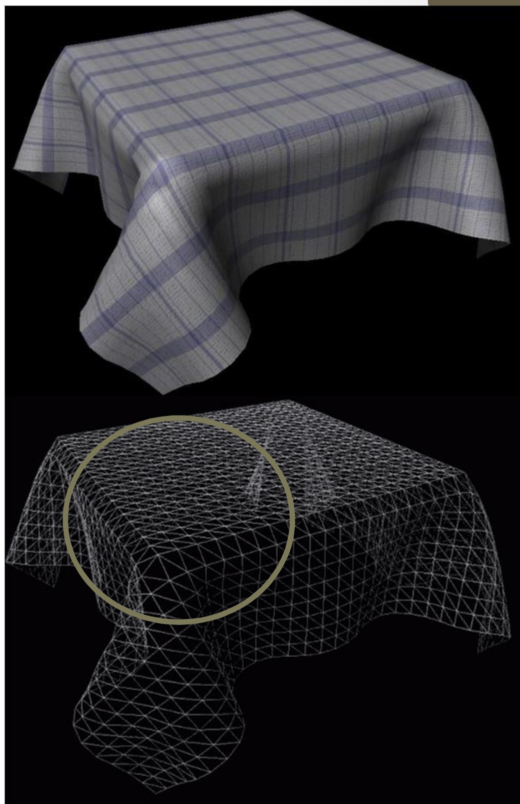



# Internal Force Computation: B-W

- General Triangular Meshes:

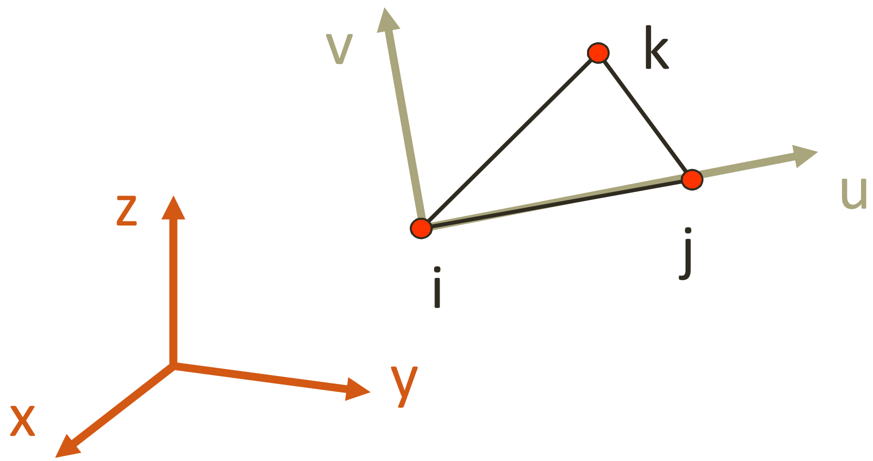
Constrain model (Baraff-Witkin 98):



# Internal Force Computation: Plane Deformation

Set w(u,v) the map from local to world coordinates



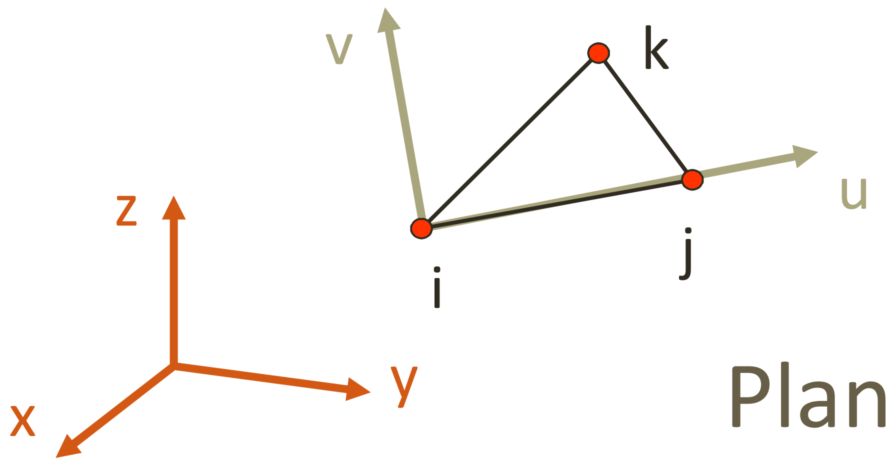

$$\Delta\, x_1 = x_j - x_i$$
$$\Delta\, x_2 = x_k - x_i$$

$$\Delta\, u_1 = u_j - u_i$$
$$\Delta\, u_2 = u_k - u_i$$

$$(\mathbf{w}_u \quad \mathbf{w}_v) = (\Delta\mathbf{x}_1 \quad \Delta\mathbf{x}_2)\begin{pmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{pmatrix}^{-1}$$

# Internal Force Computation: Plane Deformation

- Funció w(u,v) de canvi de coordenades




## Plane Deformation Matrix

$$\varepsilon = \begin{pmatrix} \|w_u\| - b_u & w_u^T \cdot w_v \\ w_u^T \cdot w_v & \|w_v\| - b_v \end{pmatrix}$$

$b_u, b_v \cong 1$

# Internal Force Computation: B-W

- The internal forces are associated to the internal Energy

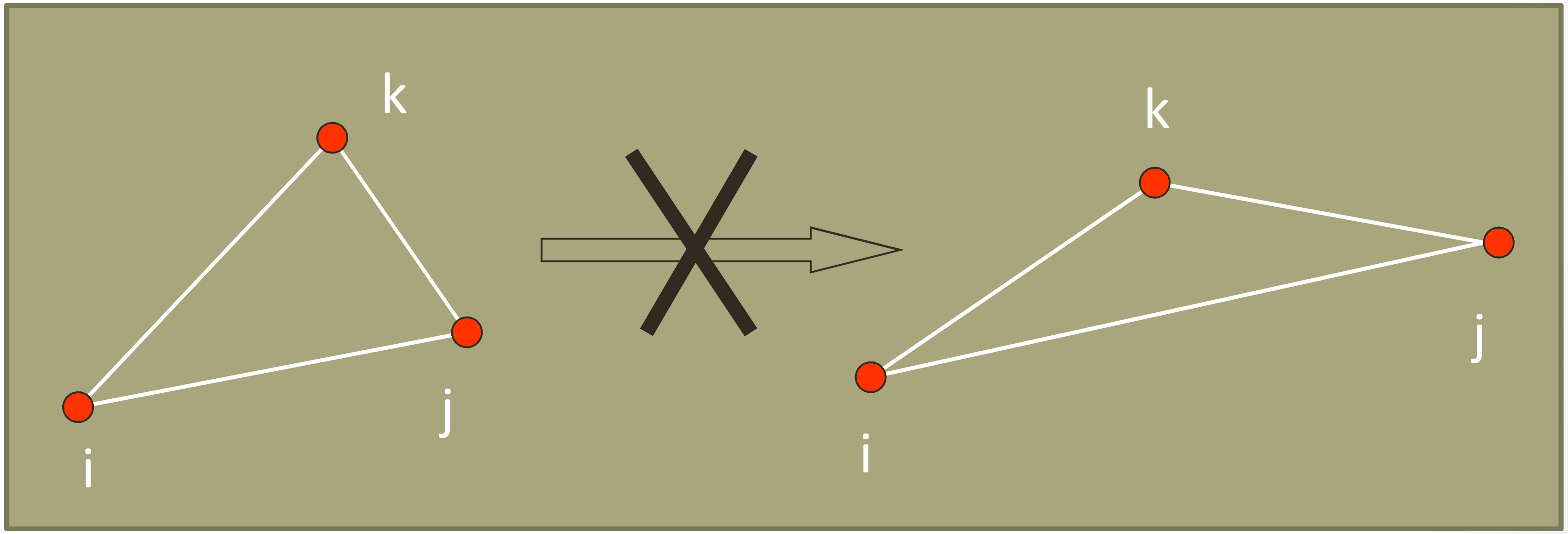
$$\boxed{F_{int} = -\frac{\partial E}{\partial x}}$$

- This Energy comes from the restriction conditions C(x) that controls the mesh deformation

$$\boxed{E_c = \frac{k_s}{2} \cdot C^T \cdot C}$$

# Internal Force Computation: STREACH

- Streach forces takes care of the variation on triangle edges.

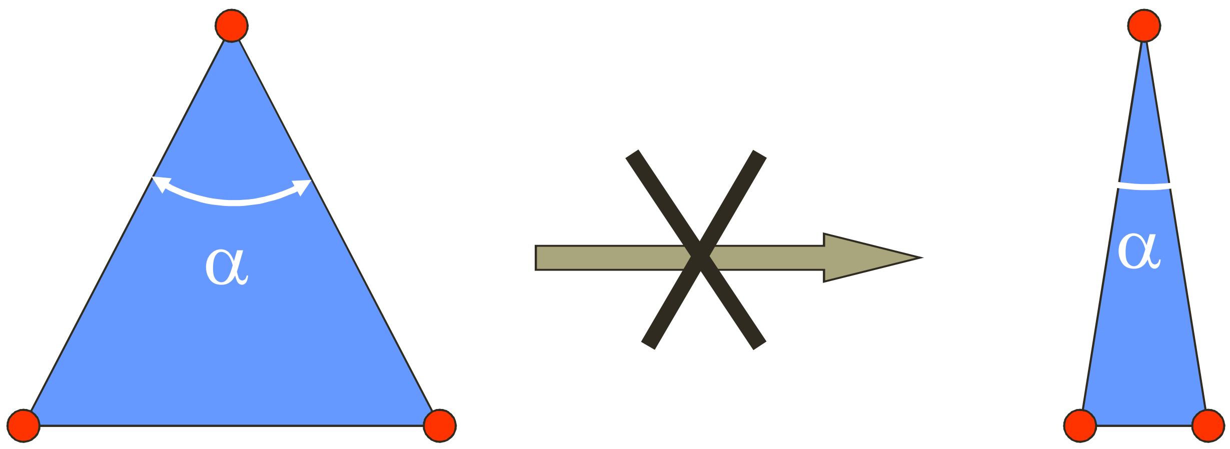



$$\mathbf{C}(\mathbf{x}) = a \begin{pmatrix} \|\mathbf{w}_u(\mathbf{x})\| - b_u \\ \|\mathbf{w}_v(\mathbf{x})\| - b_v \end{pmatrix}$$

a =Triangle area

# Internal Force Computation: SHEAR

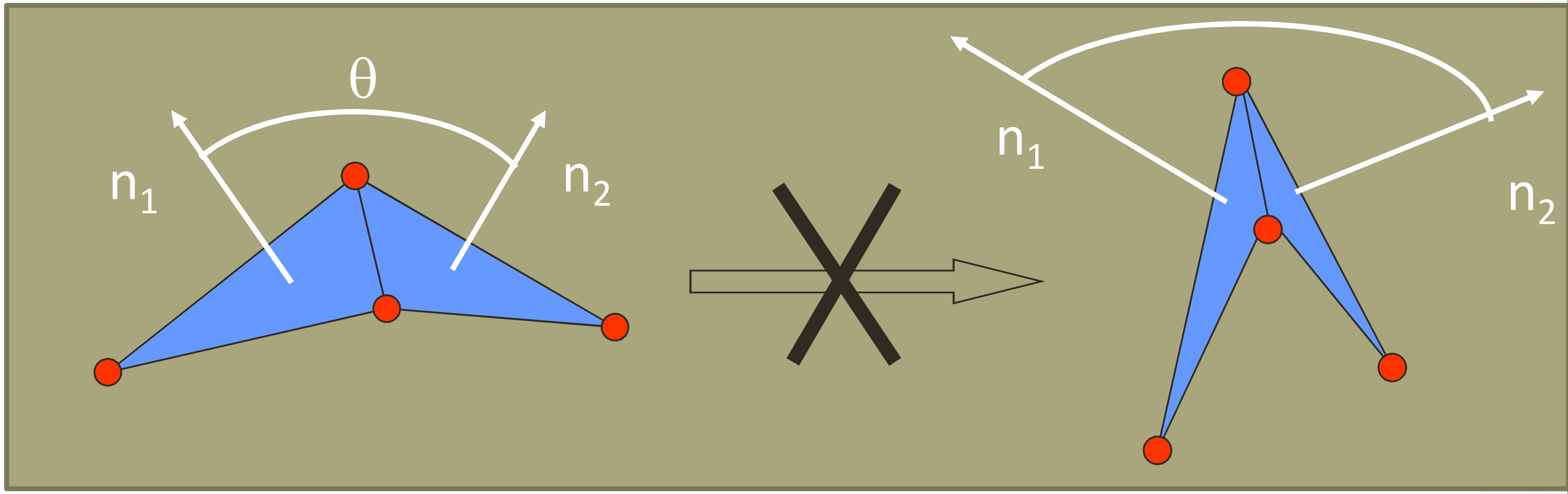
- Shear forces maintain the internal angles




$$C(\mathbf{x}) = a\mathbf{w}_u(\mathbf{x})^T\mathbf{w}_v(\mathbf{x})$$

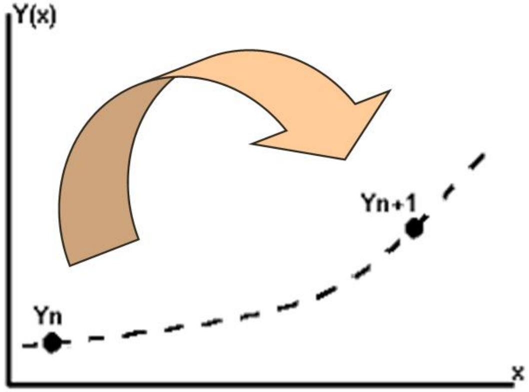
# Internal Force Computation: Bend

- Forces against cloth bending




$$C(x) = \theta$$

# Numerical Solver: Implicit Euler



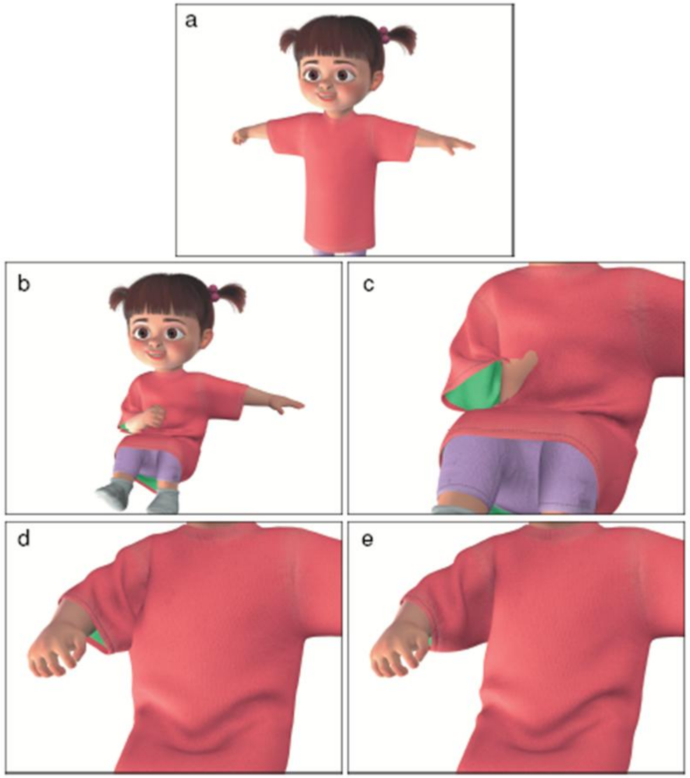

Implicit Euler Method

$$(M - h\frac{\partial f}{\partial v} - h^2\frac{\partial f}{\partial x})\Delta v = h(f_o + h\frac{\partial f}{\partial x}v_o)$$

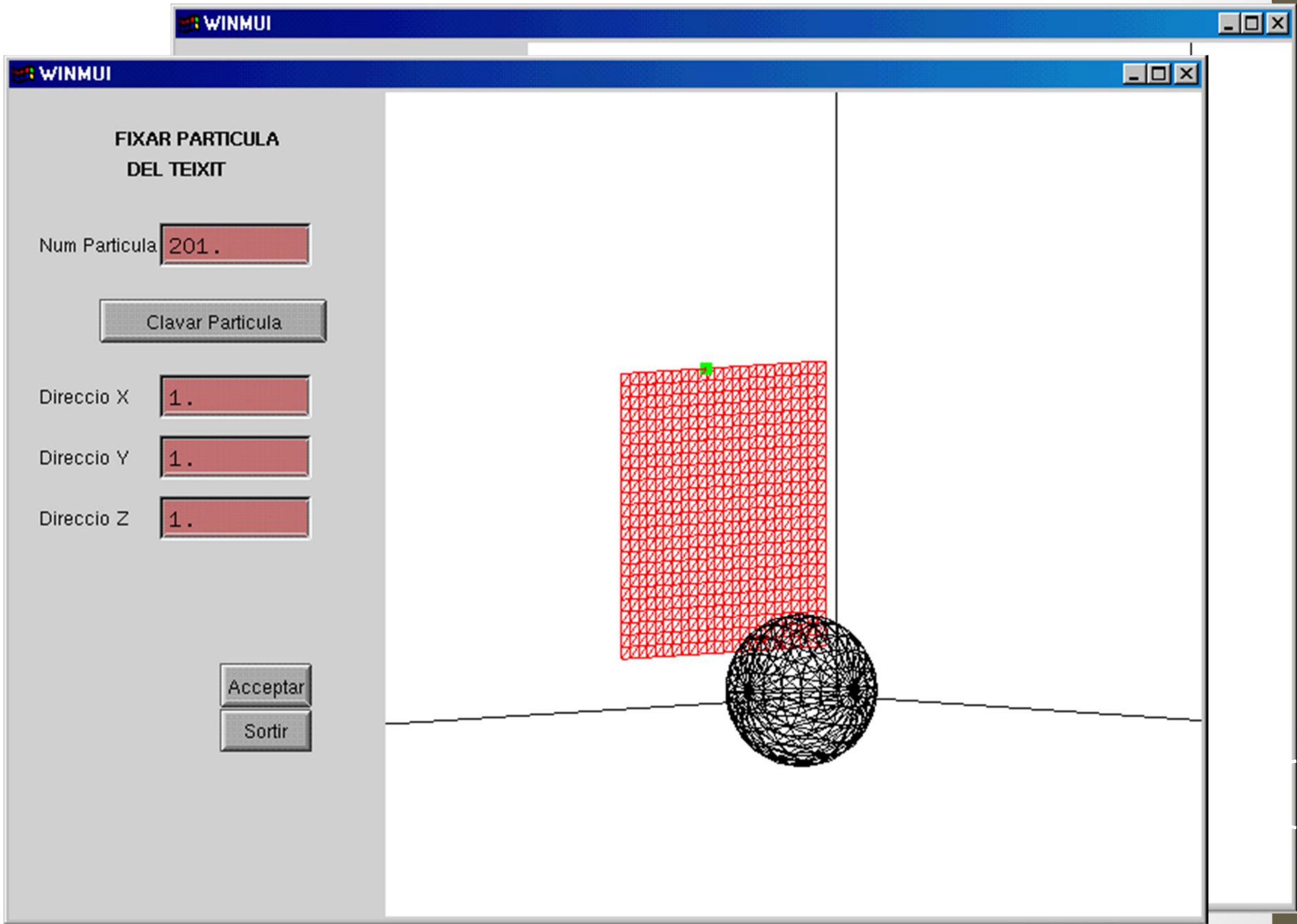
$$K_{ij} = \frac{\partial f_i}{\partial x_j} = -k\left(\frac{\partial C(x)}{\partial x_i}\frac{\partial C(x)}{\partial x_j}^T + \frac{\partial^2 C(x)}{\partial x_i \partial x_j}C\right)$$

K block matrix 3n x 3n dimensional
solved using **Conjugate Gradient** method

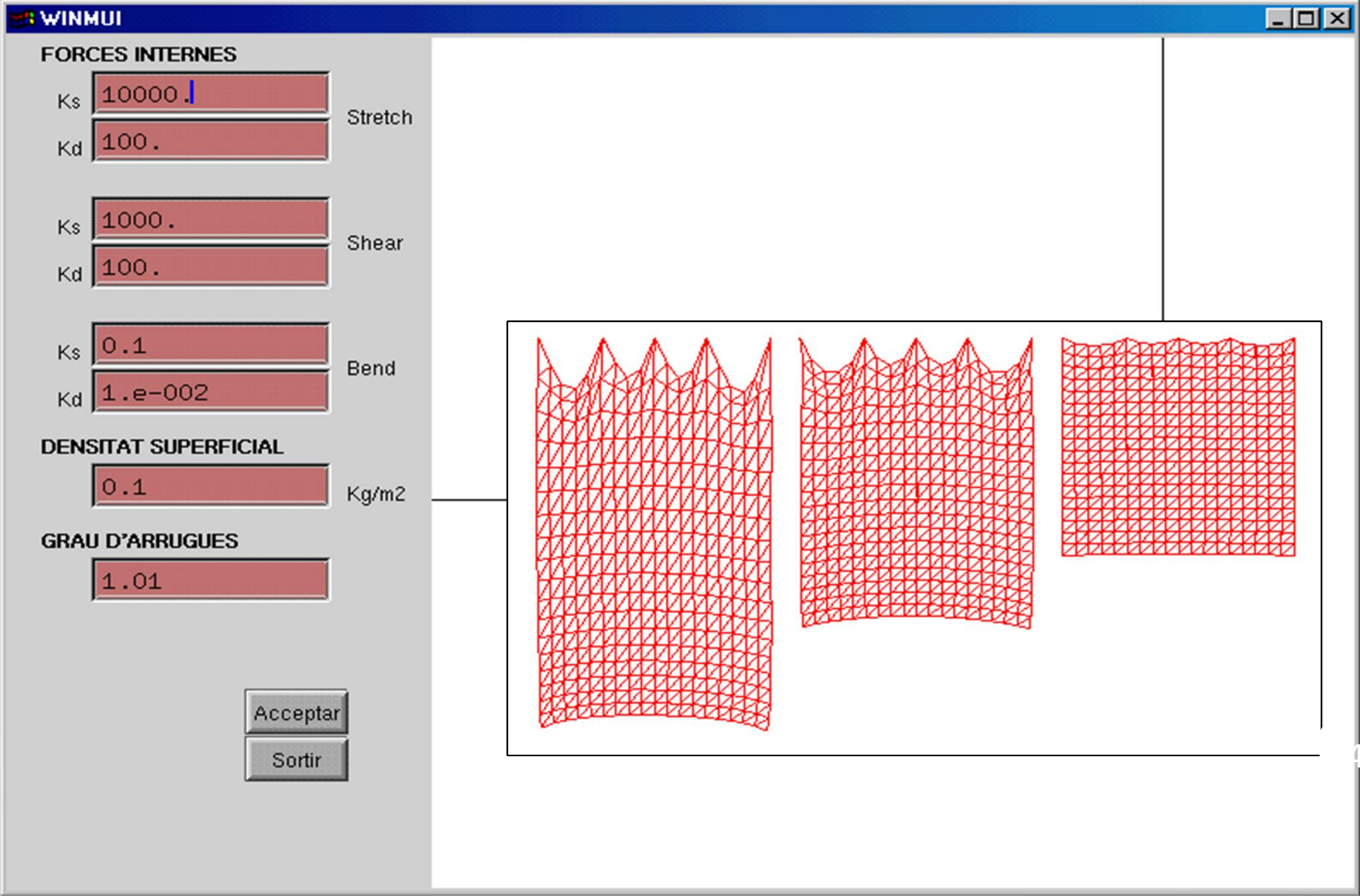
# B-W Cloth Model: Images



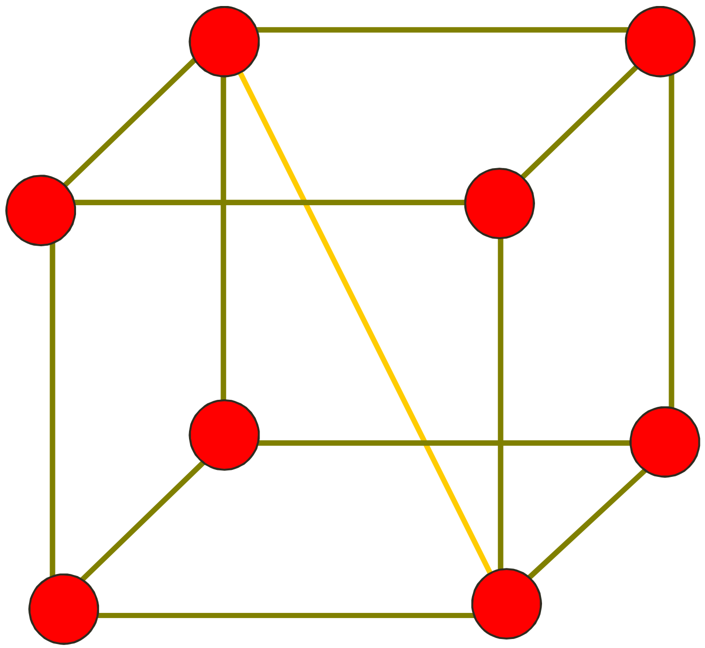
# B-W Cloth Model: Forces



# B-W Cloth Model: Forces

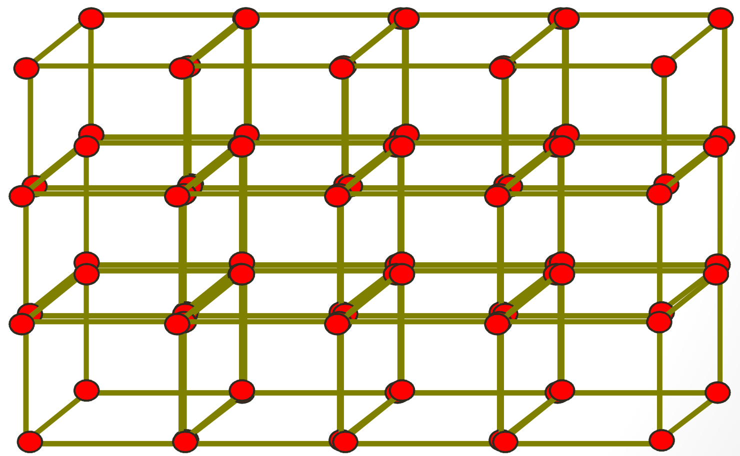


# Mesh-Spring 3-Dim Models



<u>3D Geometry</u>:

Jelly objects



# Mesh-Spring 3-Dim Models

<u>3D Geometry</u>:

Jelly objects