

# Master 1 Informatique - UE M1if37

## Animation en synthèse d'image

### Partie - Simulation par modèles physiques

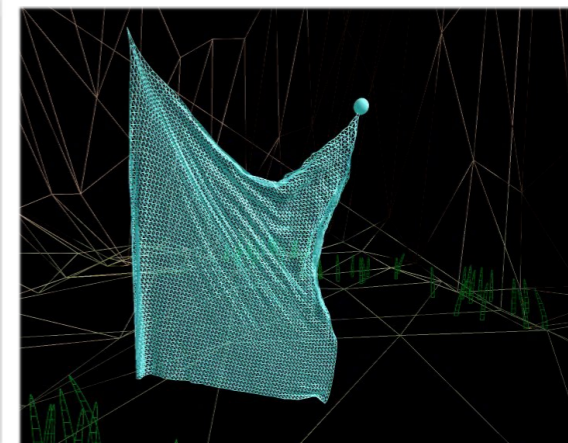
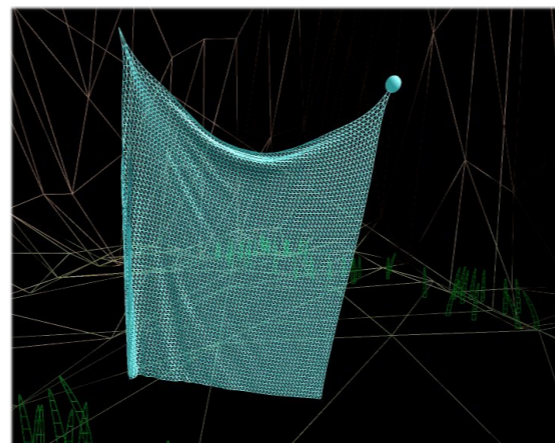
---

#### Cours 4 - Système masses-ressorts

Florence Zara

LIRIS - Université Lyon 1

<http://liris.cnrs.fr/florence.zara>



---

# Applications

---

- Simulation de cheveux
- Simulation de textiles
- Simulation de tissus mous

---

# Applications - simulation de cheveux

---

## A Mass Spring Model for Hair Simulation

Andrew Selle\*  
Stanford University  
Industrial Light + Magic

Michael Lentine  
Stanford University

Ronald Fedkiw  
Stanford University  
Industrial Light + Magic



---

# Applications - artistiques

---



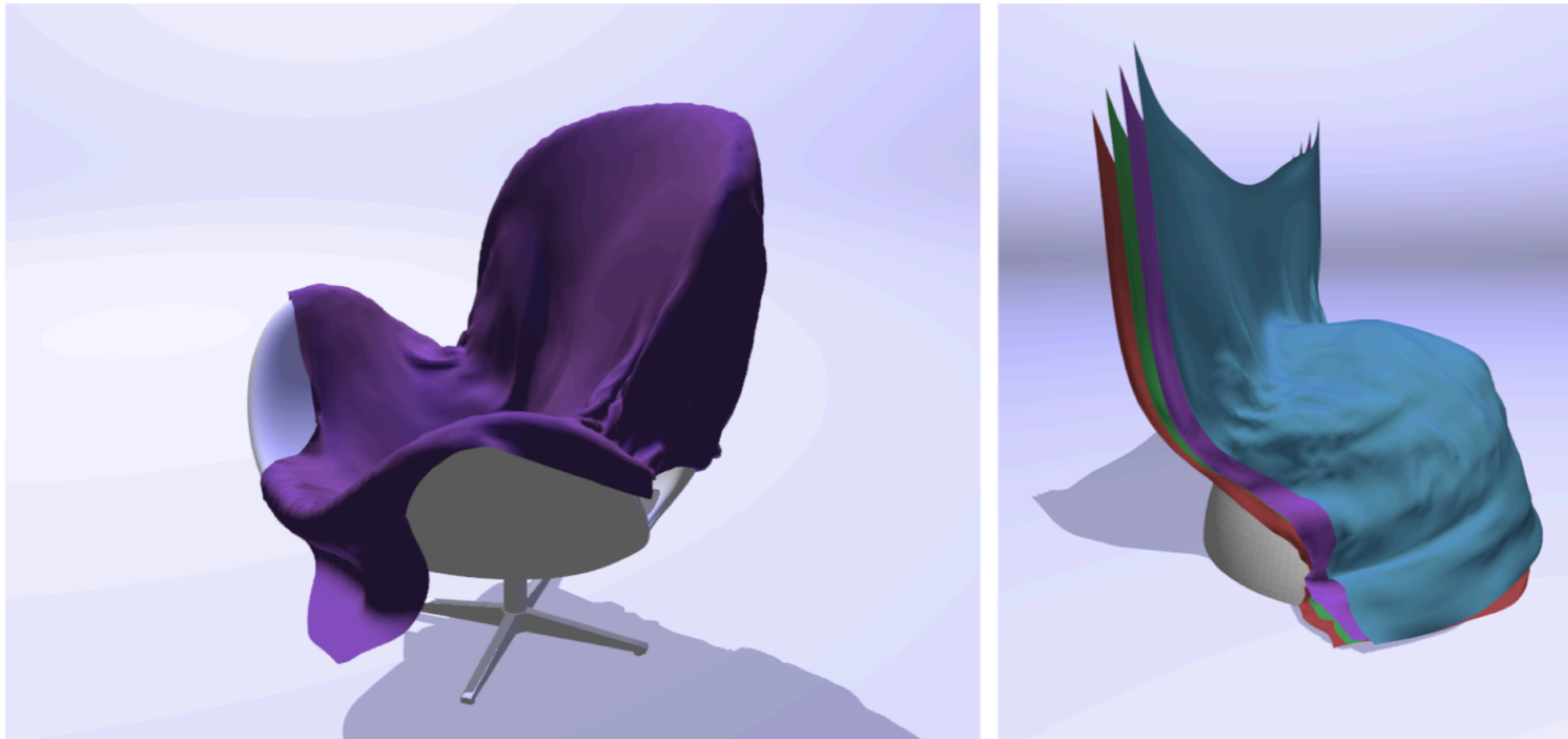
Stanford bunny

[images sur internet]

---

# Applications - simulation de textiles

---



Pontus Pall, Oskar Nylén, Marco Fratarcangeli.

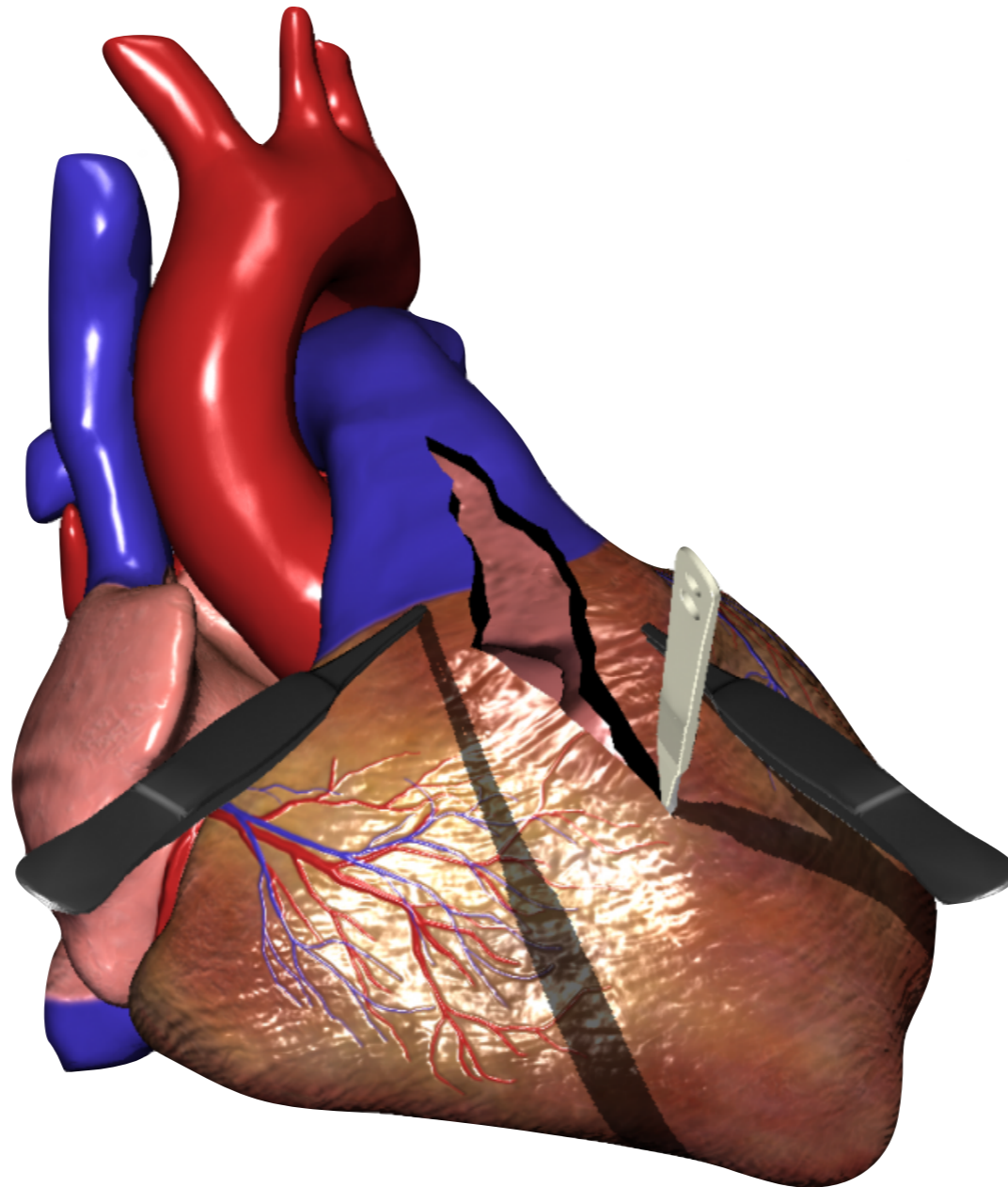
Fast Quadrangular Mass-Spring Systems using Red-Black Ordering, VRIPHYS (2018)

---

# Applications - médical

---

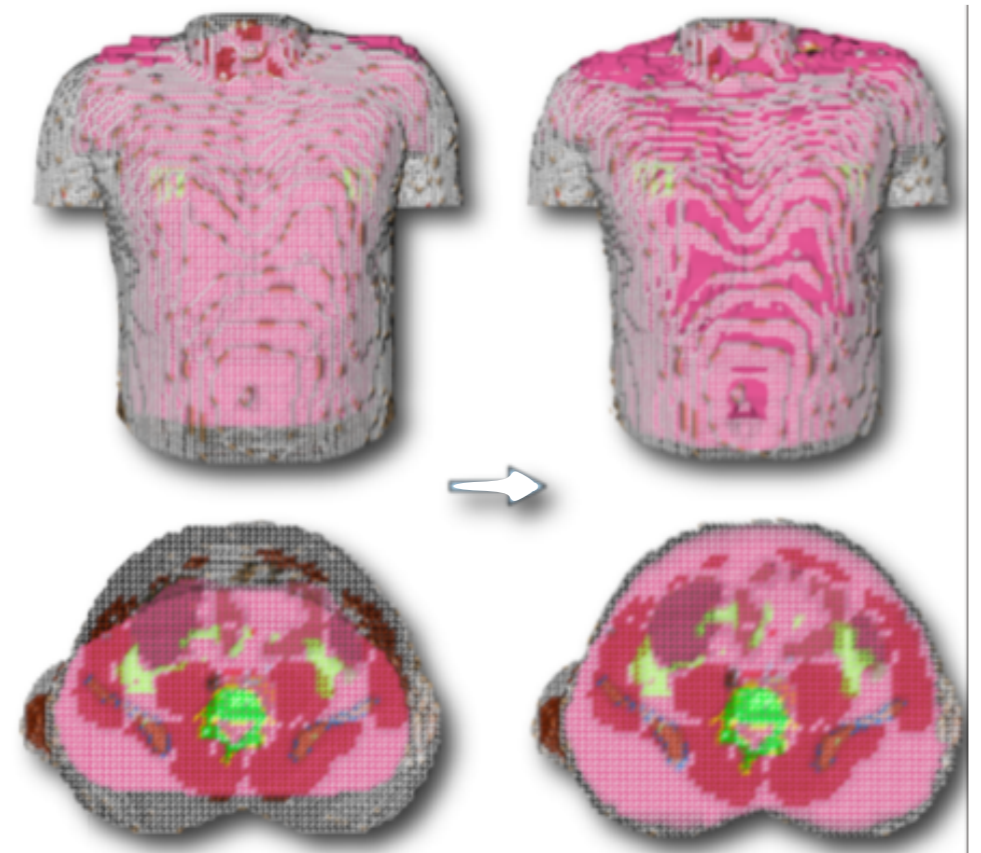
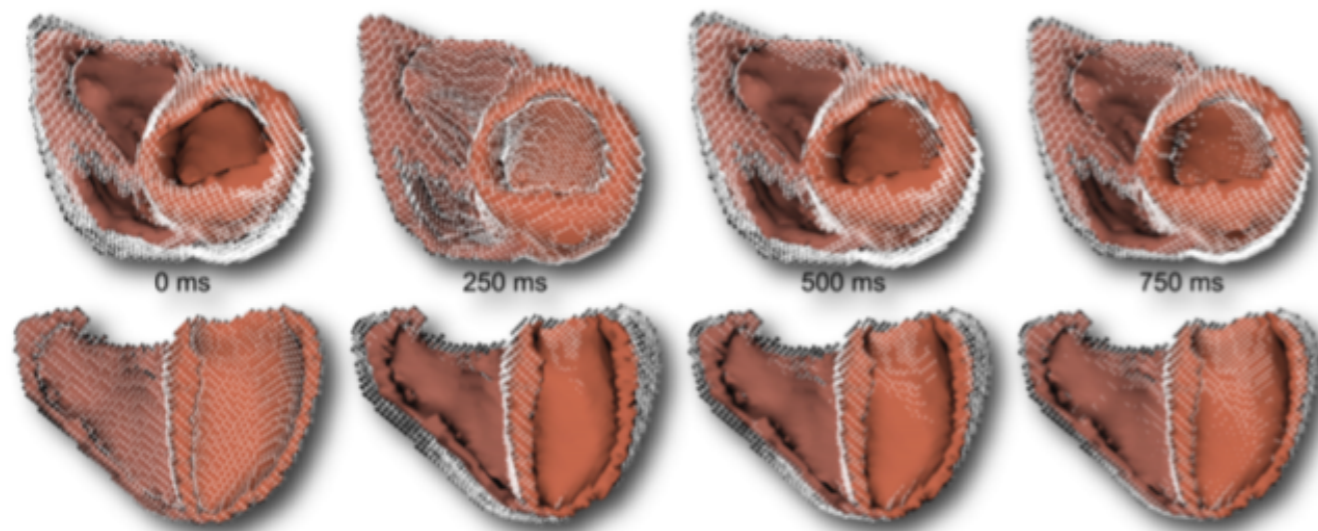
Modèle physique moins coûteux en temps de calcul que simulation par éléments finis



[Jesper Mosegaard]

# Systeme masses-ressorts - rappel du principe

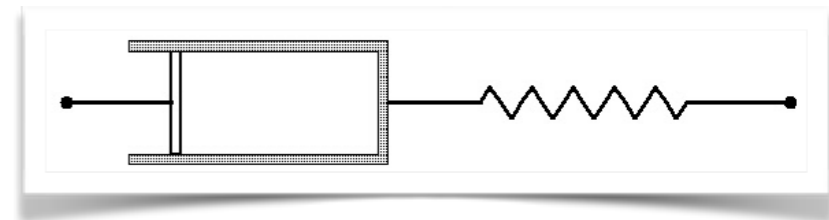
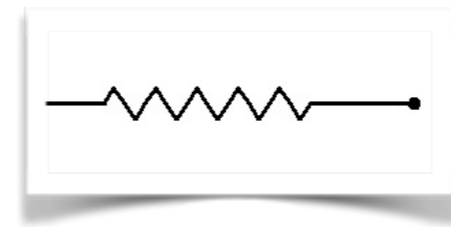
- **Objet est discrétisé en un ensemble de particules connectées entre elles par des ressorts**



# Ressort

Ressort est caractérisé par

- longueur au repos ( $l_0$ )
- coefficient de raideur ( $k$ )
- coefficient d'amortissement ( $\nu$ )
- ralentit le mouvement dans la direction du ressort





---

# Masse de l'objet

---

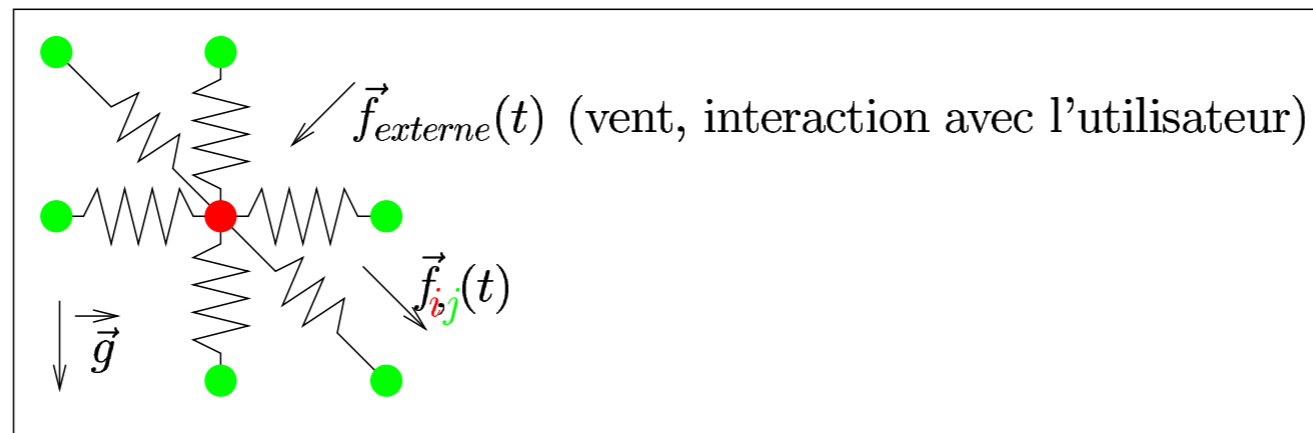
Répartition uniforme de la masse  $m$  de l'objet sur les particules

- Particule  $i$  de masse  $m_i$
- $m = \sum_i m_i$

# Forces des ressorts

- Calcul des forces exercées sur chaque particule

- $\vec{f}_i(t) = \sum_{j|j \text{ voisin de } i} [\vec{f}_{i,j}^e(t) + \vec{f}_{i,j}^v(t)] + \text{gravité} + \text{interactions}$



$$\left\{ \begin{array}{l} \vec{f}_{i,j}^e(t) = k_{ij} (\|x_i(t) - x_j(t)\| - l_{ij}) \frac{x_i(t) - x_j(t)}{\|x_i(t) - x_j(t)\|} \quad \text{élasticité} \\ \vec{f}_{i,j}^v(t) = \left( \nu_{ij} (\dot{x}_i(t) - \dot{x}_j(t)) \cdot \frac{x_i(t) - x_j(t)}{\|x_i(t) - x_j(t)\|} \right) \frac{x_i(t) - x_j(t)}{\|x_i(t) - x_j(t)\|} \quad \text{viscosité} \end{array} \right.$$

*Revoir les slides du cours précédent*

---

# Boucle de simulation

---

- Simulation basée sur la boucle de la **dynamique Newtonienne**

- A chaque pas de temps :

- Calcul des **forces** appliquées sur les particules
  - Forces dues aux ressorts, force de gravité
- Calcul des **accélérations** (principe fondamental de la dynamique)
- Calcul des **vitesse**s (intégration numérique des accélérations)
- Calcul des **positions** (intégration numérique des vitesses)

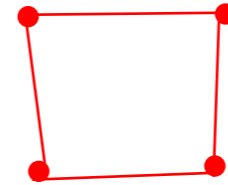
---

---

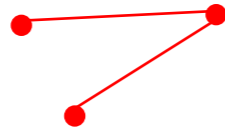
*Allons un peu plus loin*

# Positionnement des ressorts

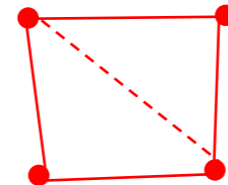
Où positionner les ressorts connectant les particules?



- Les objets peuvent s'écraser, se plier, se tordre



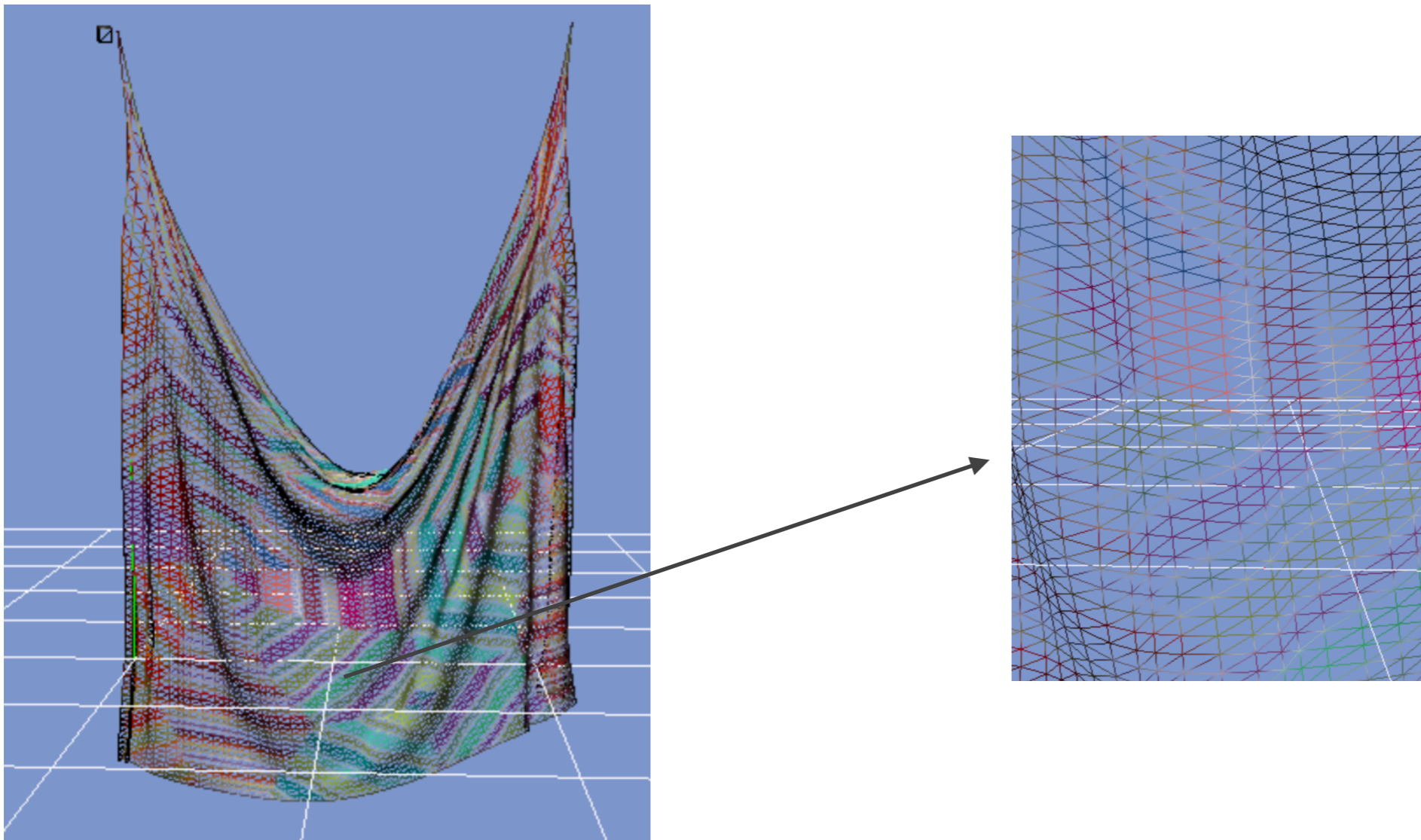
- Liaisons supplémentaires ajoutées pour plus de rigidité



# Positionnement des ressorts

Configuration qui est utilisée dans le TP :

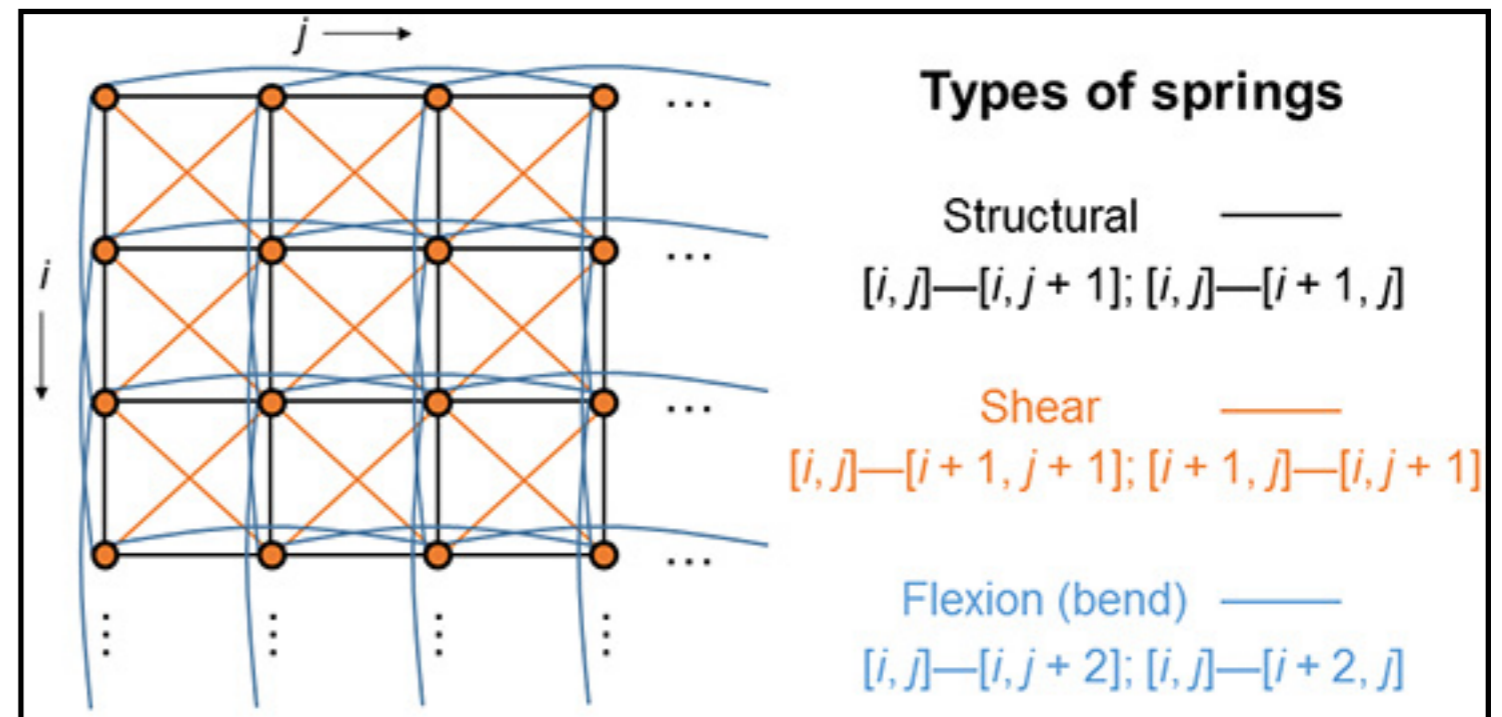
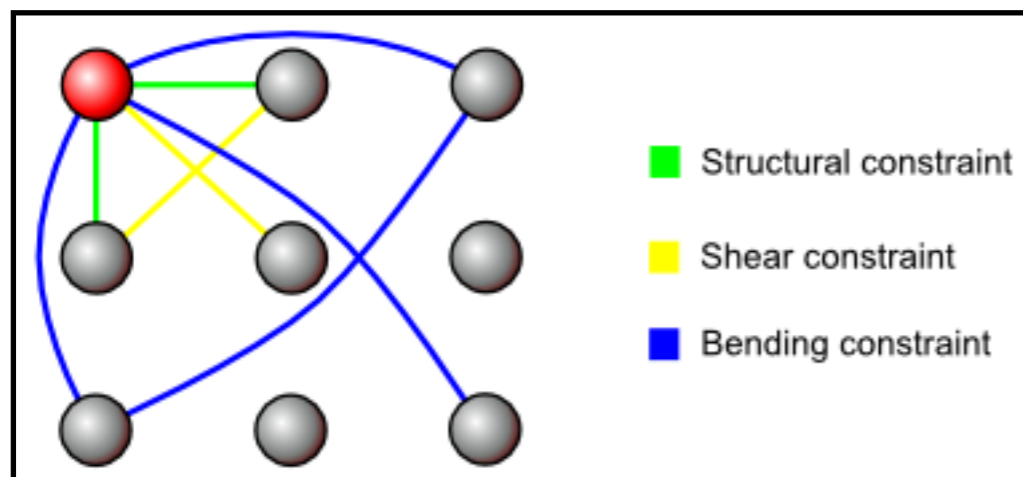
maillage triangulaire



Déformation de l'objet dépend de la position des ressorts

# Positionnement des ressorts

Il est possible de rajouter d'autres connections / ressorts

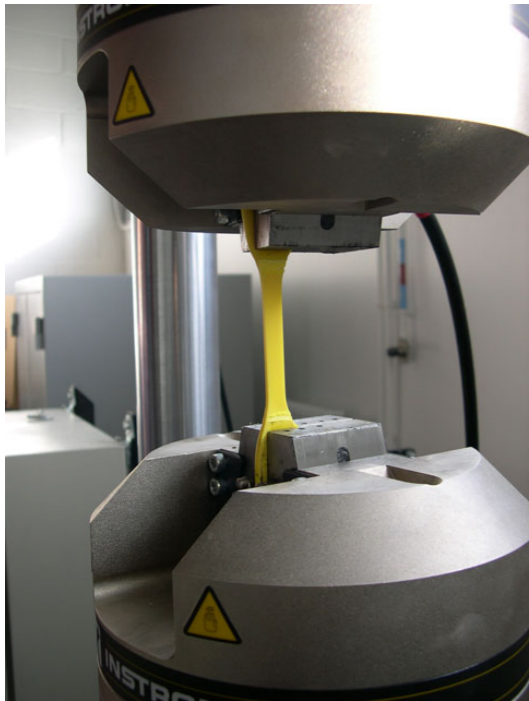


*shear = cisaillement*  
*bending = flexion*

Vous pouvez modifier le code du TP pour changer la configuration des ressorts  
(qui ne correspondra plus à celle du maillage géométrique)

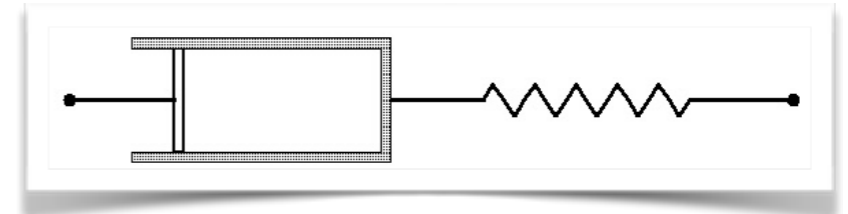
# Définition des raideurs

## Lien entre la raideur et les coefficients du matériau



**Biomatériaux**

Matériaux	Module (GPa)
Émail dentaire	82,5
Bec de poussin	50
Cartilage	0,024
Cheveu	10
Collagène	0,006
Laine	14
Piquant d'oursin	15 à 65
Radius	18,6
Soie d'araignée	25
Soie du ver à soie	17
Vertèbre cervicale	0,23



Si  $E$  et  $\nu$  connus



$k = ?$



# Définition des raideurs

## Lien entre la raideur et les coefficients du matériau

N°	Auteurs	Formulation des raideurs	Propriétés géométriques et mécaniques du matériel	Bases physiques
1	Wilhelms and Van Gelder (1997)	$k_{tri} = \frac{A_1 + A_2}{l_0^2}$ $k_{tri}^a = c \sum_i \frac{A_i}{3}$	maillage triangulaire, membrane élastique	approximation géométrique
2	Gelder (1998)	$k_{tri}^c = \frac{E_2 \sum_e A_e}{ l_0 ^2}$ $k_{tetra}^c = \frac{E \sum_e V_e}{ l_0 ^2}$	maillage triangulaire ou maillage tétraédrique, isotropique, élastique linéaire	approximation géométrique

N°	Auteurs	Formulation des raideurs	Propriétés géométriques et mécaniques du matériel	Bases physiques
8	Baudet et al. (2009)	$k_{rect}^e = \frac{E(j^2(3\nu+2)-i^2)}{4 l_0 h_0(1+\nu)}$ $k_{rect}^d = \frac{E(l_0^2+h_0^2)}{4 l_0 h_0(1+\nu)}$ $k_{hexa}^e = \frac{E l_0(4\nu+1)}{8(1+\nu)}$ $k_{hexa}^d = \frac{E i \sum_{j \in \{x_0, y_0, z_0\}} j^2}{8(1+\nu) \prod_{l \in \{x_0, y_0, z_0\}, l \neq i} l}$	matériau élastique linéaire avec maillage rectangulaire et hexaédrique	paramétrisation basée sur déformations en cisaillement et étirement
9	Natsupakpong and Çavuşoğlu (2010)	$k_{rect}^e = \frac{\lambda}{3} + \frac{\mu}{2}$ $k_{rect}^d = \left( \frac{5\lambda}{12} + \frac{3\mu}{4} \right)$	élastique linéaire, maillage rectangulaire	approximation d'un modèle FEM

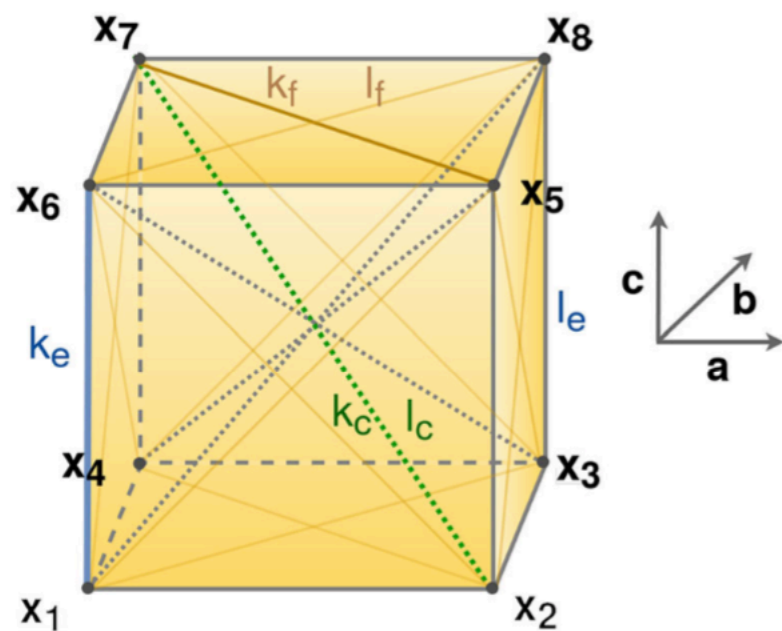
TABLE 1.1 Suite du tableau récapitulatif des formulations des raideurs.

N°	Auteurs	Formulation des raideurs	Propriétés géométriques et mécaniques du matériel	Bases physiques
3	Maciel et al. (2003)	$k_{hexa} = \frac{k_0 \cos_0 + \sum_i k_i \cos_i}{l_0 + \sum_i l_i \cos_i}$	objets élastiques linéaires, maillage hexaédrique avec ressorts diagonaux	ajustement angulaire de la formulation de Gelder (1998)
4	Lloyd et al. (2007)	$k_{tri} = \sum_e E h \frac{\sqrt{3}}{4}$ $k_{rect}^e = \sum_e \frac{5}{16} h E$ $k_{rect}^d = \frac{7}{16} h E$ $k_{tetra} = \sum_e \frac{2\sqrt{2}}{25} l E$ $k_{tetra}^V = \sum_e \frac{2\sqrt{2}}{21} l E \frac{4}{5}$ $k_V = \frac{\sqrt{2}}{84} l^3 E \frac{2}{5}$	élastique linéaire, raideurs pour la préservation du volume pour maillage tétraédrique	basé sur modèle FEM de référence
5	Zerbato et al. (2007)	$k_{tetra} = \frac{E_1 + E_2}{2} \frac{\sum_i V_i}{l_0^2}$	échantillons de foie	approximation géométrique
6	Delingette (2008a,b)	$k_{tri} = \frac{2 \cot^2 \alpha_i (\lambda + \mu) + \mu}{16 A_p} = \frac{E (2 \cot^2 \alpha_i + 1 - \nu)}{16(1 - \nu^2) A_p}$ $c_{tri} = \frac{2 \cot \alpha_i \cot \alpha_j (\lambda + \mu) - \mu}{16 A_p} = \frac{E (2 \cot \alpha_i \cot \alpha_j + \nu - 1)}{16(1 - \nu^2) A_p}$ $k_{tetra} = \frac{(\lambda + \mu) V}{2} (D_i \cdot D_j)^2 + \frac{\mu V}{2} \ D_i\ ^2 \ D_j\ ^2$ $c_{tetra} = \frac{(\lambda + \mu) V}{2} (D_i \cdot D_j)(D_i \cdot D_k) + \frac{\mu V}{2} \ D_i\ ^2 (D_j \cdot D_k)$	élastique linéaire, membrane non-linéaire (Saint Venant Kirchhoff), maillage triangulaire ou maillage tétraédrique	basé sur l'énergie de déformation d'un triangle ou d'un tétraèdre
7	Arnab and Raja (2008)	$k_{tri}^E = E \sum_{i \in j} \frac{V_j}{l_i^2}$ $k_{tri}^G = E \sum_{i \in j} \frac{V_j}{2 l_i^2 (1 - \nu)}$ $k_{tri}^B = \frac{E}{3(1 - 2\nu)} \sum_{i \in j} \frac{V_j}{l_i^2}$	objets 3D élastiques	modèle géométrique avec propriétés élastiques

# Définition des raideurs

## Hybrid 3D mass-spring system for simulation of isotropic materials with any Poisson's ratio

K. Golec<sup>1</sup> · J.-F. Palierne<sup>2</sup> · F. Zara<sup>1</sup>  · S. Nicolle<sup>3</sup> · G. Damiand<sup>1</sup>

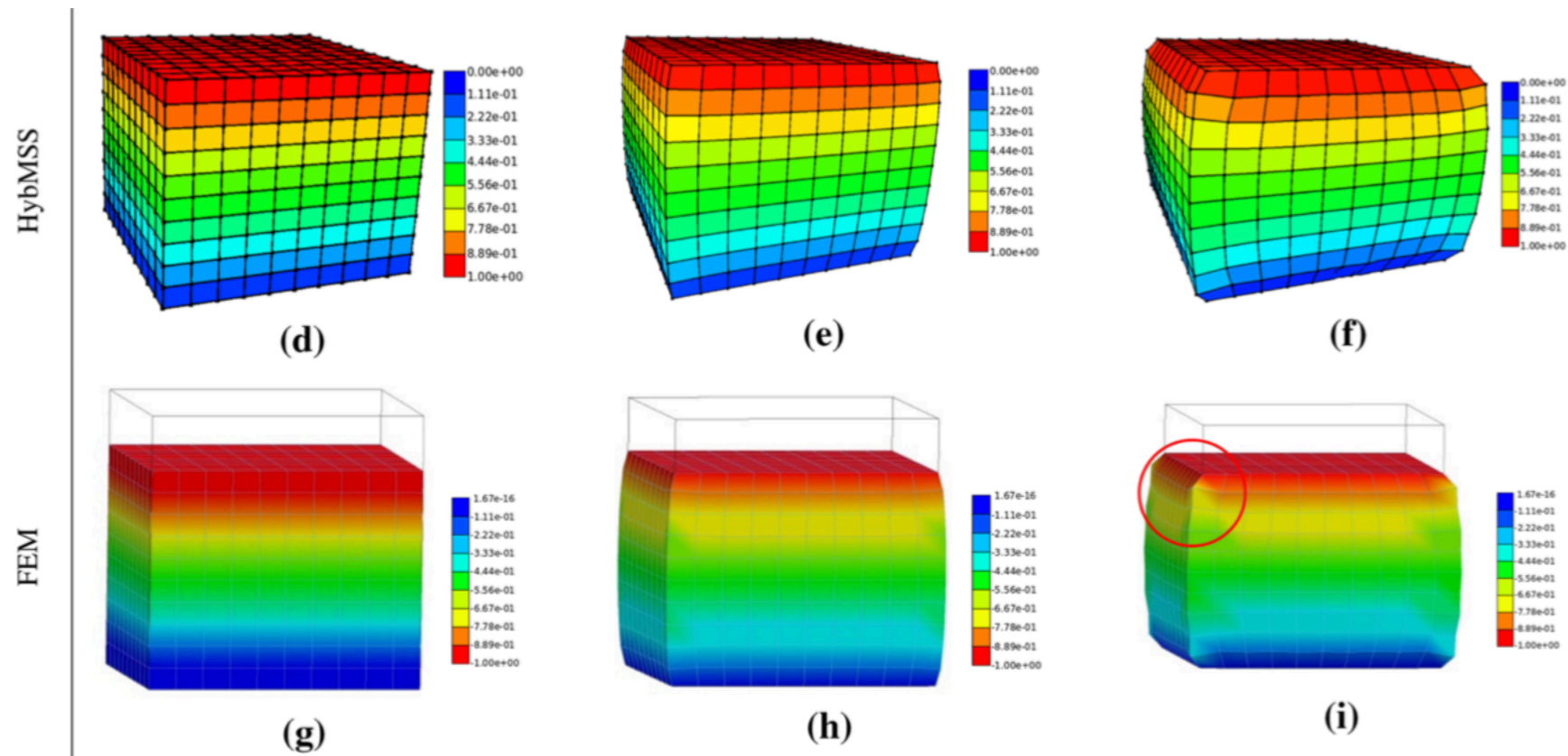
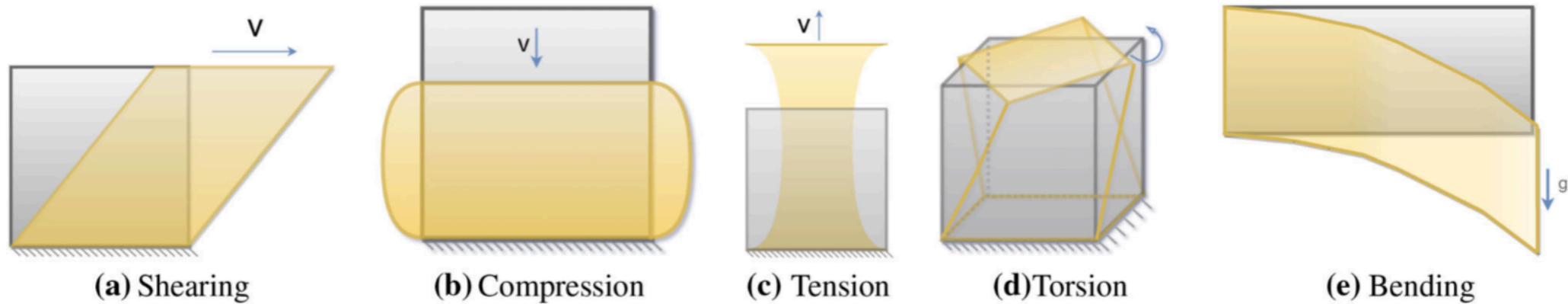


$$A = k_f / k_c.$$

$$\begin{cases} k_c = \frac{3 G l_e}{4 + 3A} = \frac{6}{5} \frac{E l_e}{4 + 3A} \\ k_f = A k_c \\ k_e = G l_e \frac{8 + 3A}{4 + 3A} = \frac{2}{5} \frac{E l_e}{4 + 3A} \end{cases}$$

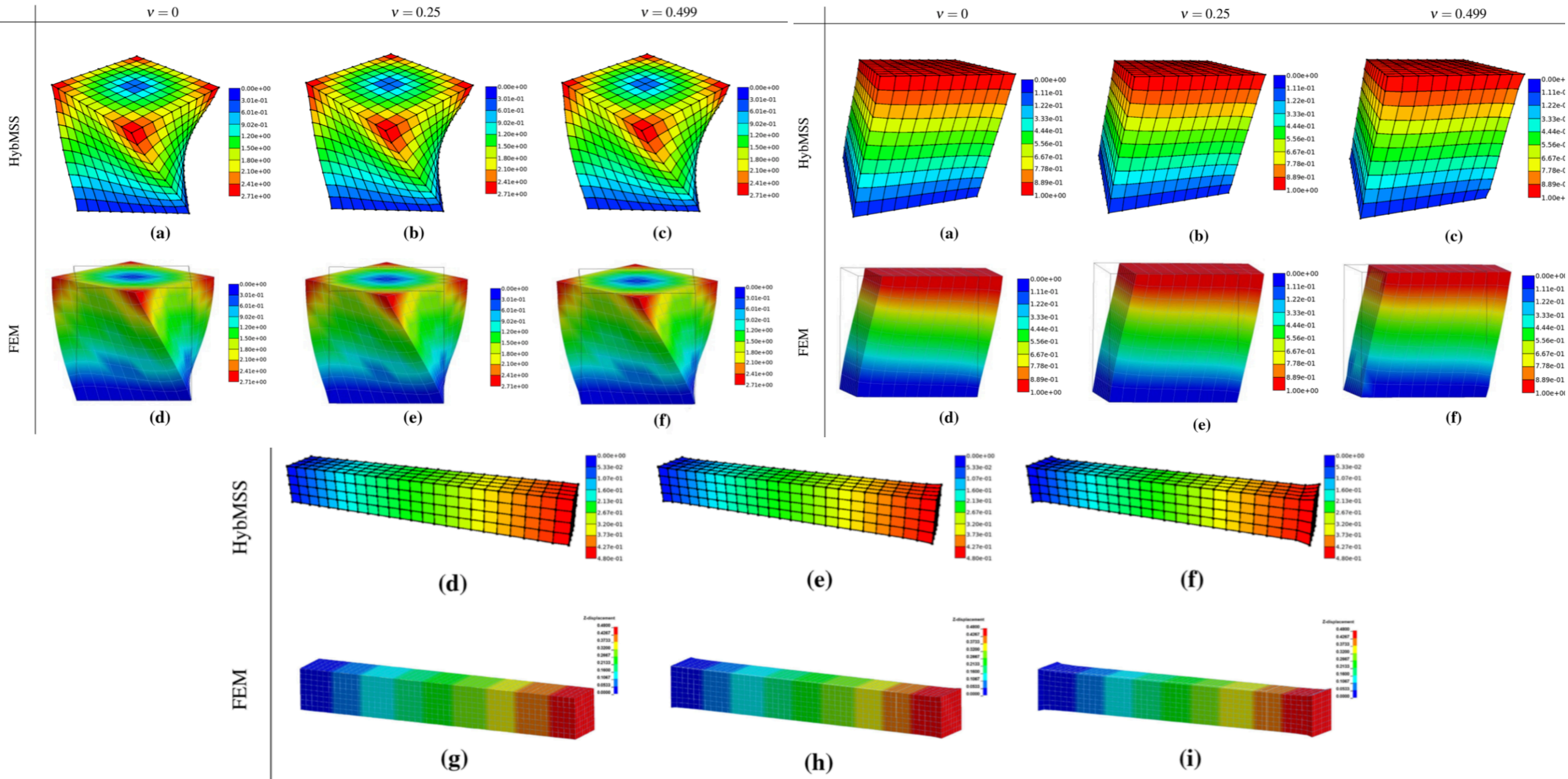
[Karolina Golec 2019]

# Validation des solutions



[K. Golec 2019]

# Validation des solutions

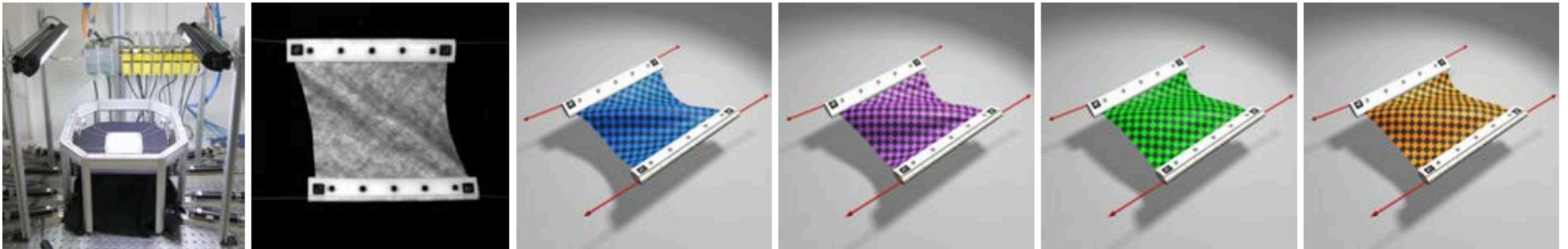


# Définition des raideurs

## Data-Driven Estimation of Cloth Simulation Models

E. Miguel<sup>1,2</sup> D. Bradley<sup>2</sup> B. Thomaszewski<sup>2</sup> B. Bickel<sup>2</sup> W. Matusik<sup>2,3</sup> M. A. Otaduy<sup>1</sup> S. Marschner<sup>4</sup>

1. URJC Madrid 2. Disney Research Zurich 3. MIT CSAIL 4. Cornell University



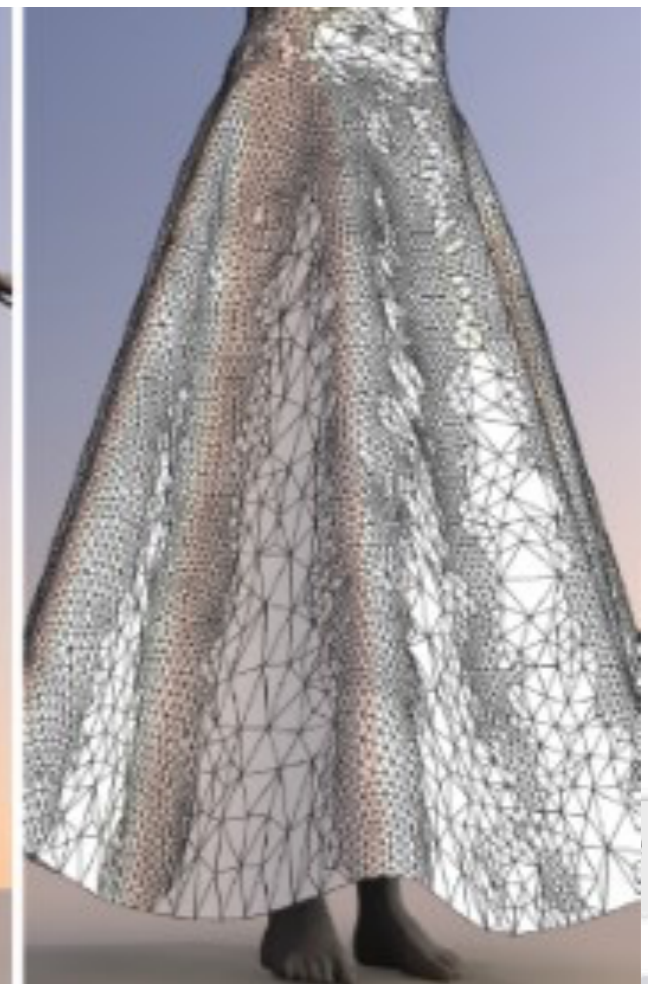
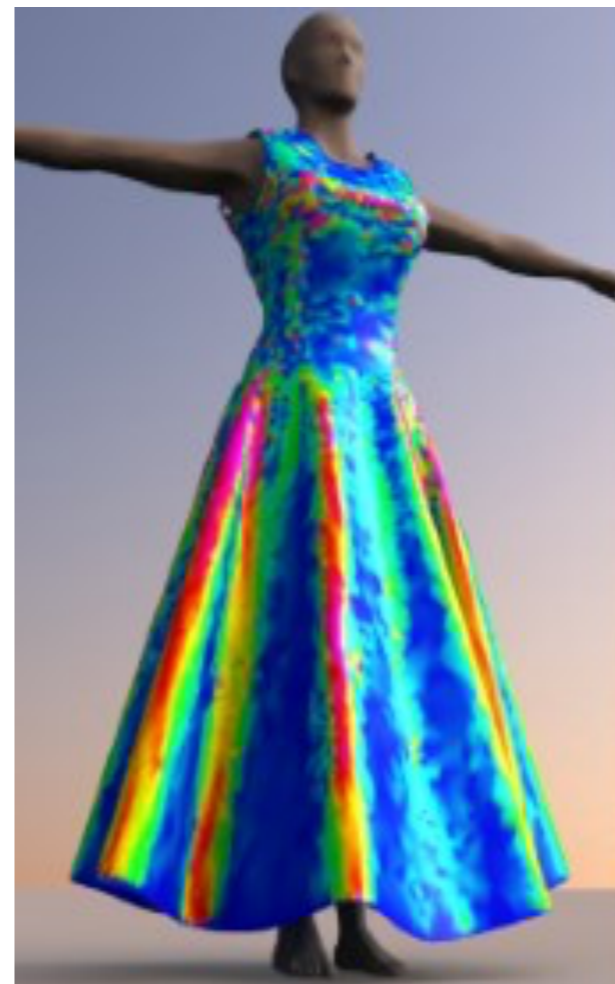
[vidéo]

**Autre solution pour définir les raideurs :  
comparaison et ajustement en fonction d'enregistrement vidéos du comportement**

# Raffinement de maillage

Connections peuvent évoluer au cours de la simulation pour s'ajuster au besoin (ajustement de la courbure)

*J. Bender & C. Deul / Adaptive cloth simulation using corotational finite elements*



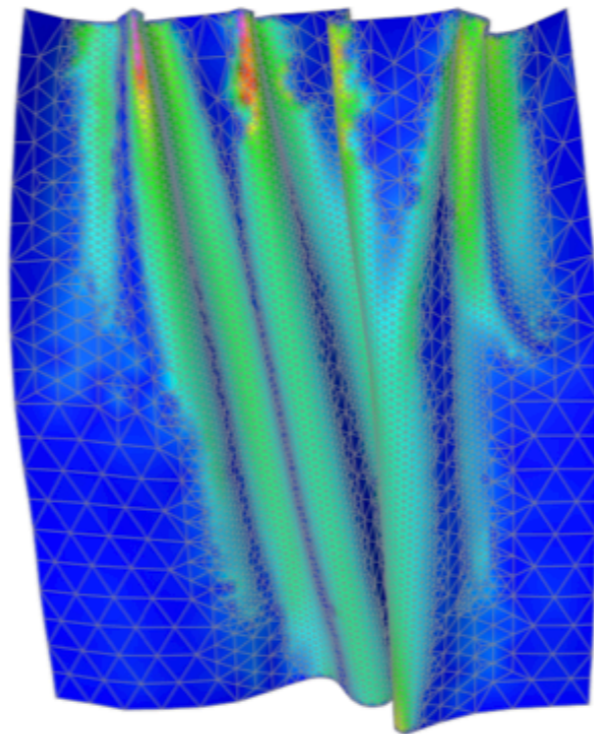
*Papier utilisant les EF*

# Raffinement de maillage

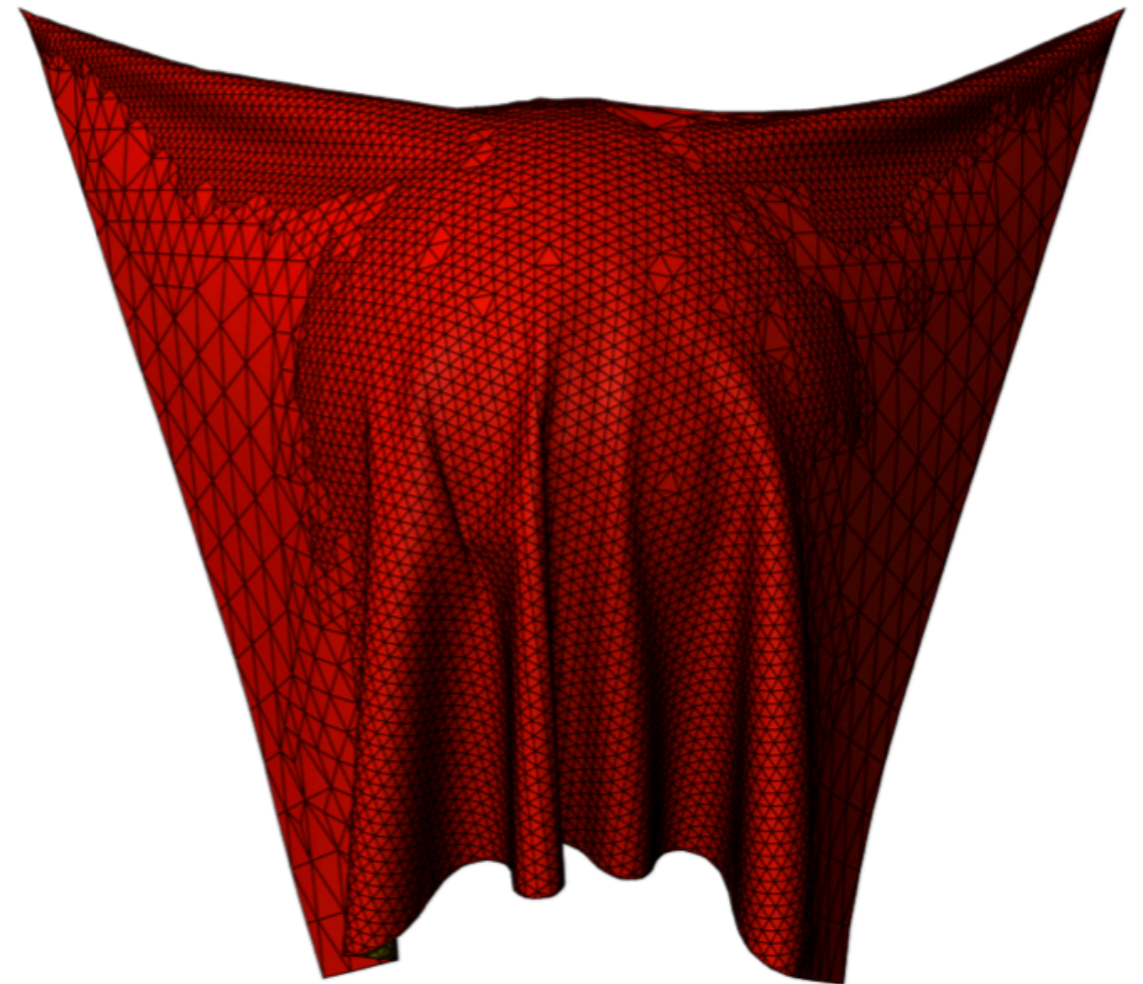
*J. Bender & C. Deul / Adaptive cloth simulation using corotational finite elements*



(a) Curtain model



(b) Curvature and triangulation



---

# Raffinement de maillage

---

## Multi-Resolution Cloth Simulation

Yongjoon Lee<sup>1</sup>

Sung-eui Yoon<sup>1</sup>

Seungwoo Oh<sup>2</sup>

Duksu Kim<sup>1</sup>

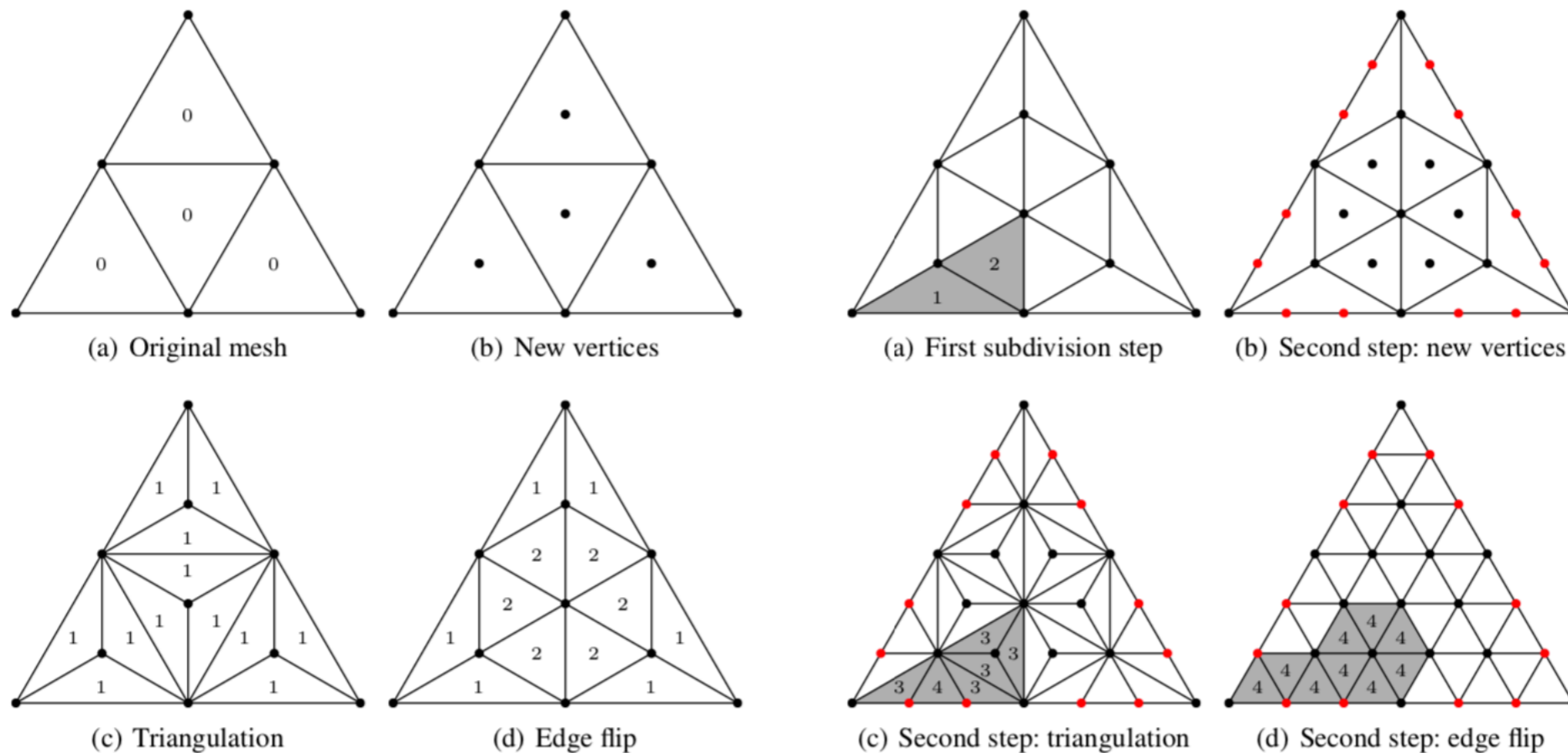
Sunghee Choi<sup>1</sup>





# Raffinement de maillage

*J. Bender & C. Deul / Adaptive cloth simulation using corotational finite elements*



**Première étape : modification du maillage géométrique  
(effectuée en fonction de critères géométriques par exemple)**

---

# Raffinement de maillage

---

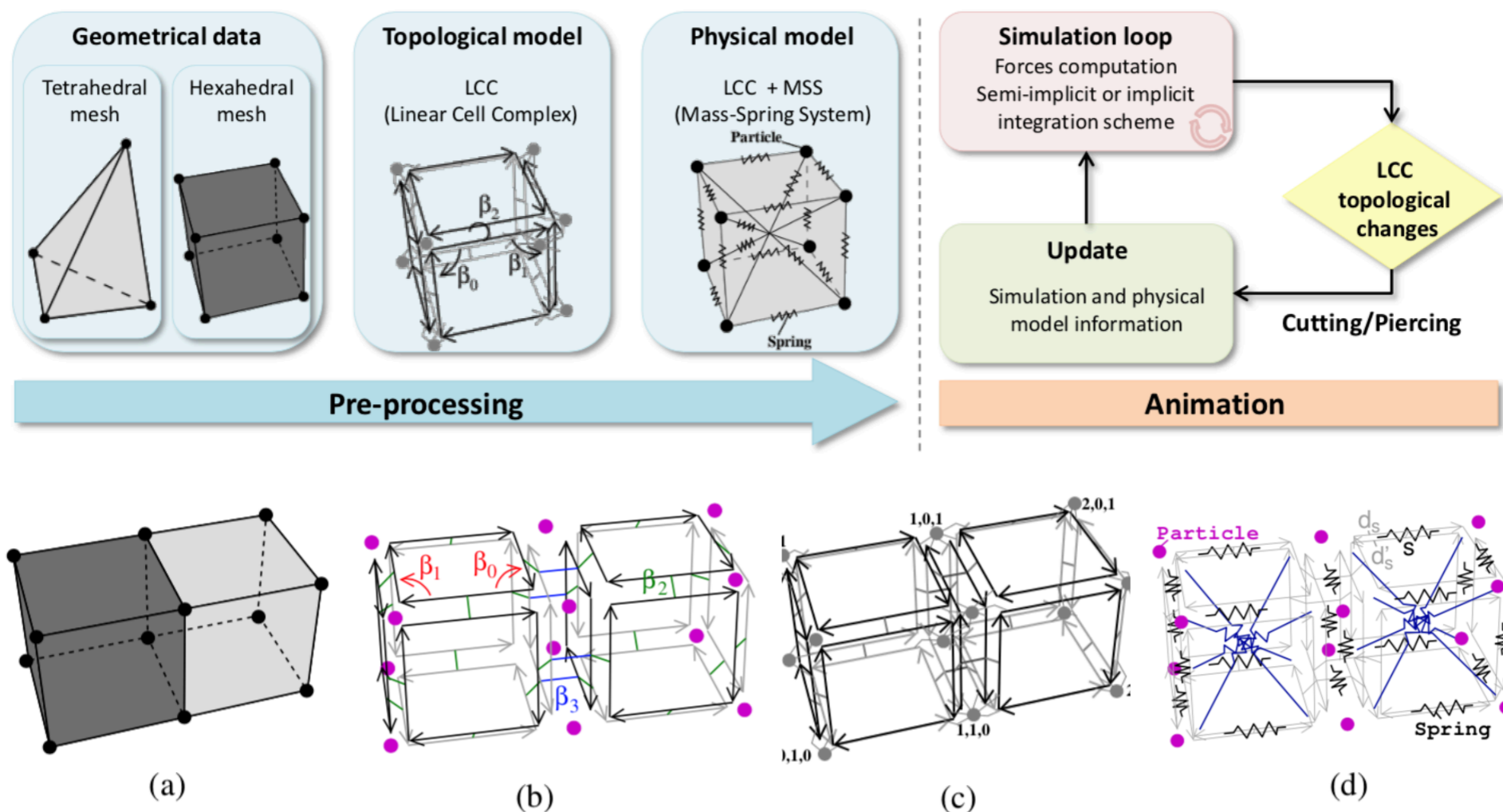
## Deuxième étape :

### Modification du maillage géométrique

- **induit une modification des connections entre les particules**
  - modification du voisinage des particules (pour le calcul des forces)
- **induit une modification des paramètres mécaniques**
  - on doit recalculer les masses des particules (nouvelle répartition)
  - on doit recalculer les raideurs des ressorts

# Structure de données

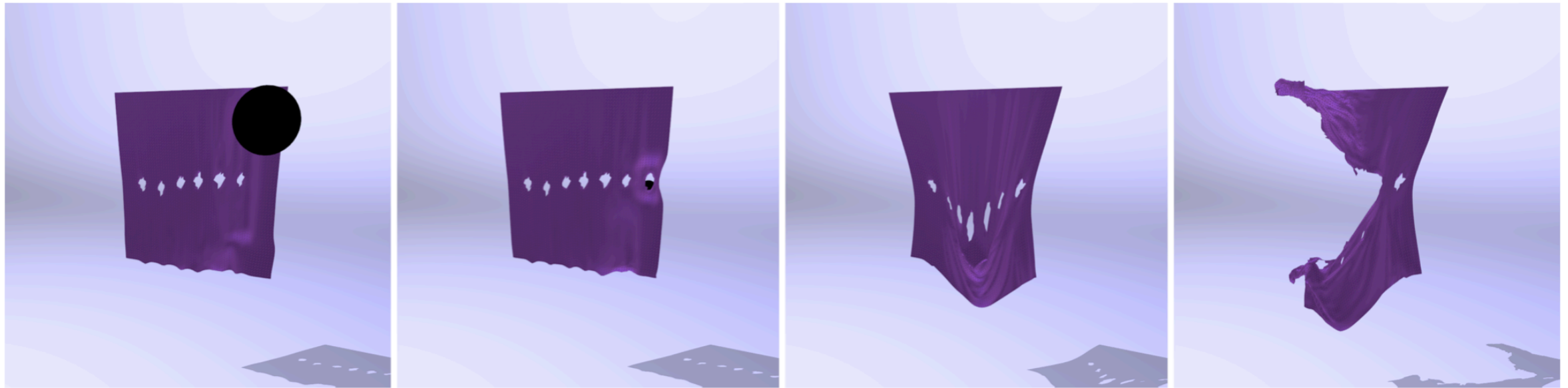
Une solution possible : emploi d'une structure de données topologiques (LCC) incluant les paramètres mécaniques



[E. Flechon 2013]

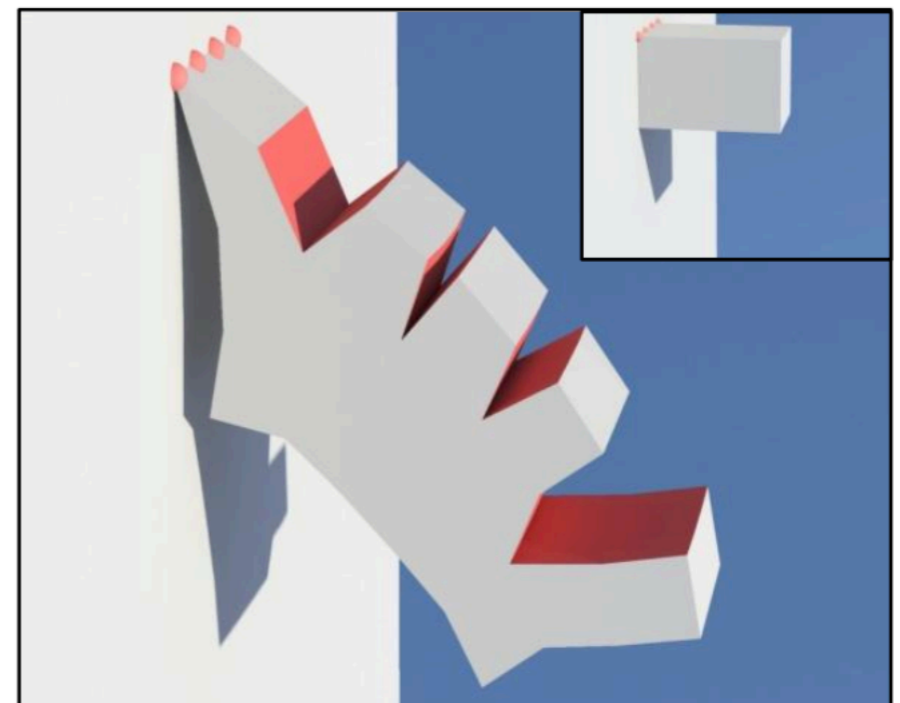
# Changement topologique

Autre type de changement topologique : découpe, déchirure



Pontus Pall, Oskar Nylén, Marco Fratarcangeli.

Fast Quadrangular Mass-Spring Systems using Red-Black Ordering, VRIPHYS (2018)



[E. Flechon 2013]

# Parallélisation

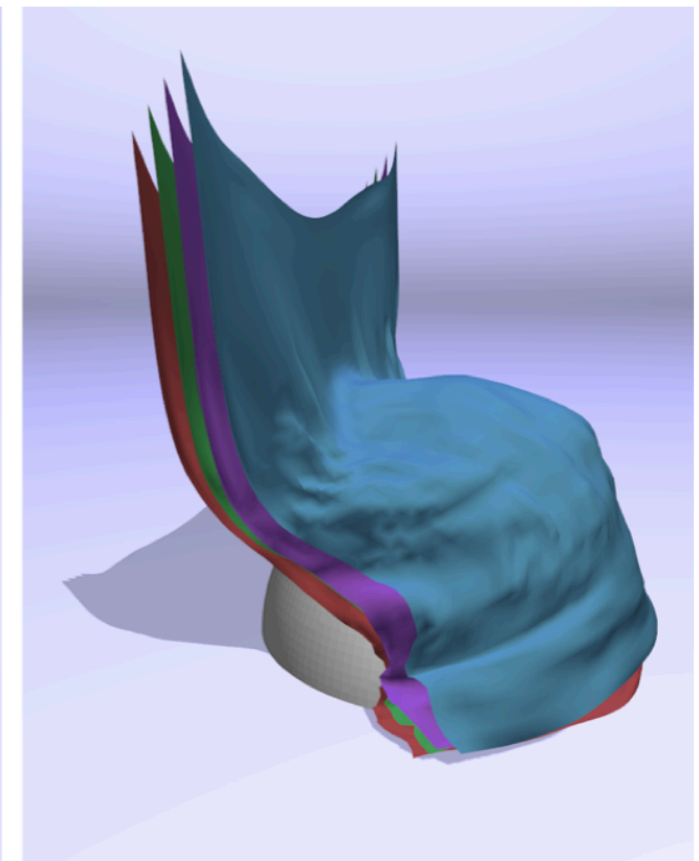
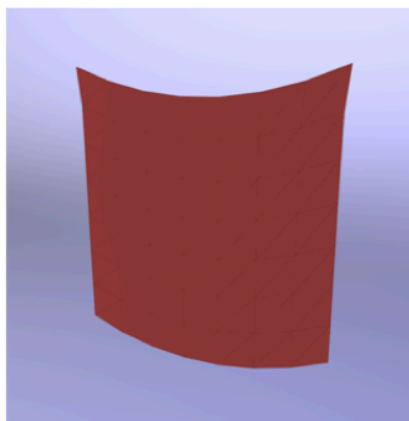
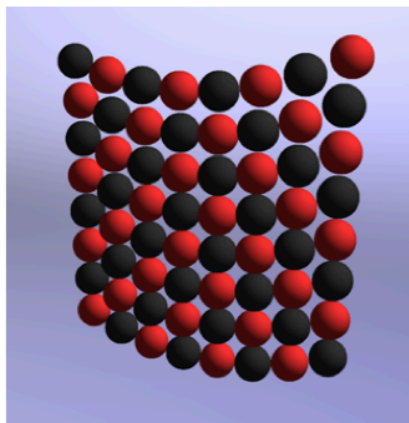
## Fast Quadrangular Mass-Spring Systems using Red-Black Ordering

Pontus Pall

Oskar Nylén

Marco Fratarcangeli

Chalmers University of Technology, Sweden



[vidéo]

---

# Projective Dynamics

---

- **Méthode qui se situe entre Position Based Dynamics et la méthode des éléments finis**
- **Contrainte est décrite en fonction de l'énergie potentielle**
- **Pour un ressort (loi de Hooke)**
  - reliant particules  $i$  et  $j$  (ayant comme positions  $p_i$  et  $p_j$ )
  - de raideur  $k$  (noté  $s$  dans les algos) et de longueur au repos  $l_{ij}$
  - l'énergie potentielle est définie par :

$$W(p_i, p_j) = \frac{1}{2} k \left( \|p_i - p_j\| - l_{ij} \right)^2$$

---

# Boucle de simulation - Projective Dynamics

---

- La contrainte due au ressort est supposée satisfaite quand l'énergie est minimale (i.e. nulle)
- On considère l'ensemble  $U$  de tous les ressorts du système identifiables par les indices  $i$  et  $j$  des particules les reliant
- On obtient le problème de minimisation suivant :

$$W(p) = \min_{d \in U} \|(p_i - p_j) - l_{ij}\|^2$$

# Projective Dynamics - algorithme

---

**Algorithm 1:** Projective Dynamics - parallel Red-Black Gauss-Seidel solver

---

```
1 Numerical integration (Verlet)
2 loop numNonLinearIterations times
3   Local step
4   loop numLinearIterations times
5     for each  $P \in (P_R, P_B)$  do
6       for each  $p_i \in P$  do in parallel
7         Global step
8       end
9     end
10  end
11 end
```

chaque ressort est  
résolu de manière  
indépendante

2 ensembles indépendants  
de particules  
( $P_R$  et  $P_B$ ) sont définis

configuration locale sert à  
construire système linéaire  
afin de projeter les positions  
des particules le plus proche  
de la solution locale trouvée



# Projective Dynamics en parallèle

---

**Algorithm 2:** Verlet integration

---

```
1 for each particle  $\mathbf{p}_i$  do in parallel
2   |  $\mathbf{f}_{ext} \leftarrow \mathbf{f}_g - \mathbf{v}_i \mathbf{f}_{damping}$ 
3   |  $\mathbf{p}_i(t) \leftarrow 2\mathbf{p}_i(t) - \mathbf{p}_i(t-h) + h^2 \mathbf{f}_{ext}$ 
4 end
```

---

Dynamique : intégration de Verlet

---

**Algorithm 3:** Projective Dynamics, local step

---

```
1 for each spring  $s_{ij} \in U$  do in parallel
2   |  $\mathbf{d}_{ij} = L_{ij}(\mathbf{p}_i - \mathbf{p}_j) / \|\mathbf{p}_i - \mathbf{p}_j\|$ 
3 end
```

---

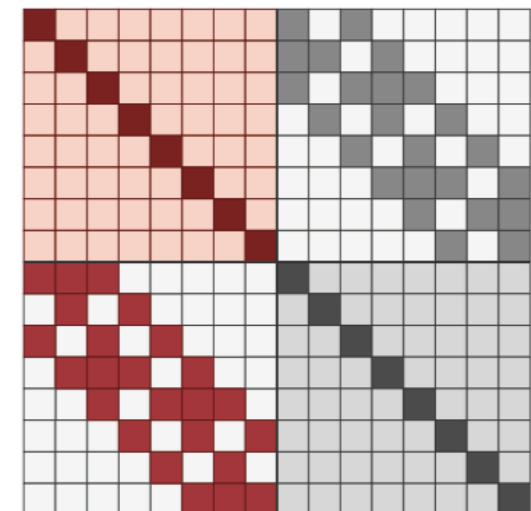
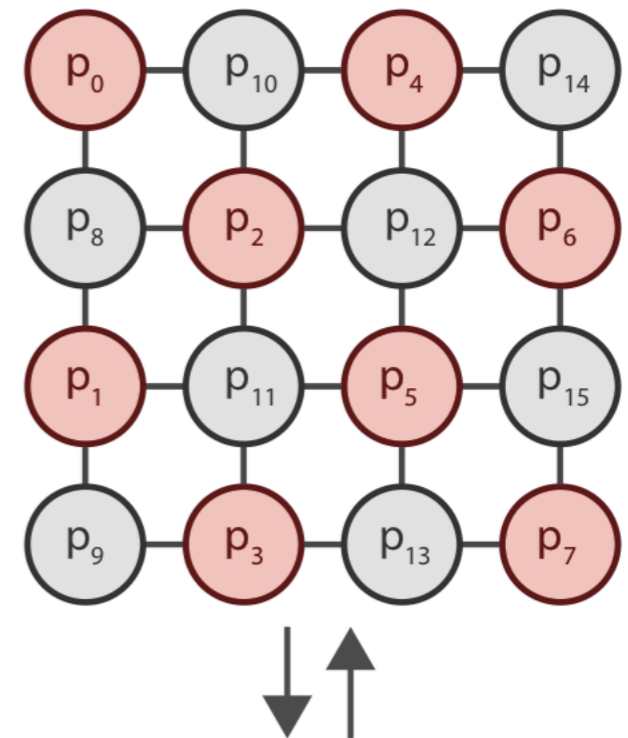
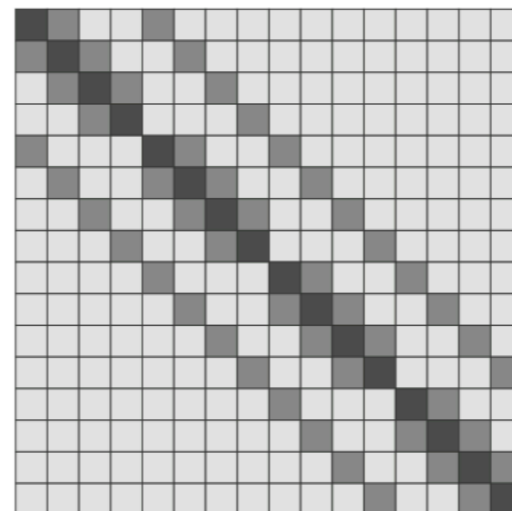
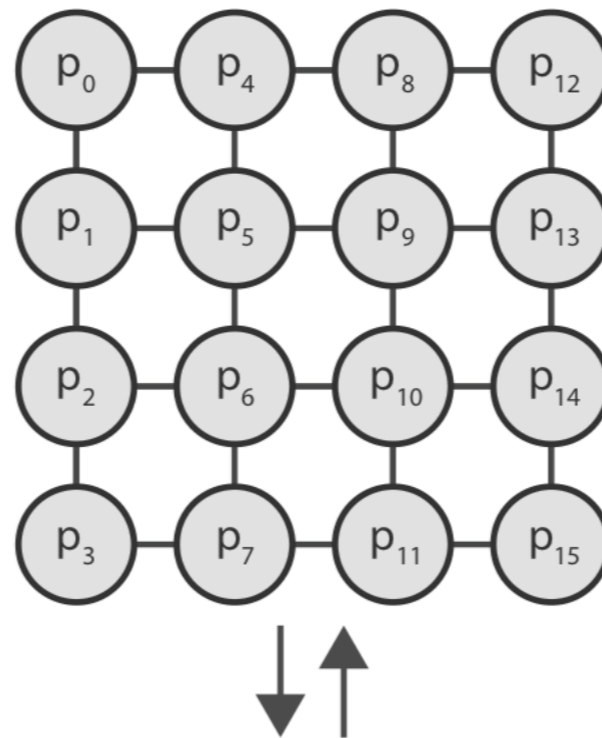
Local step : donne solution  
locale pour ressort ij

$$\mathbf{p}_i^{k+1} = \frac{\frac{m_i}{h^2} \mathbf{y}_i + \sum_{j < i} s_{ij} (\mathbf{p}_j^{k+1} + \mathbf{d}_{ij}) + \sum_{j > i} s_{ij} (\mathbf{p}_j^k + \mathbf{d}_{ij})}{\frac{m_i}{h^2} + \sum_{j \neq i} s_{ij}}$$

Global step

# Projective Dynamics en parallèle

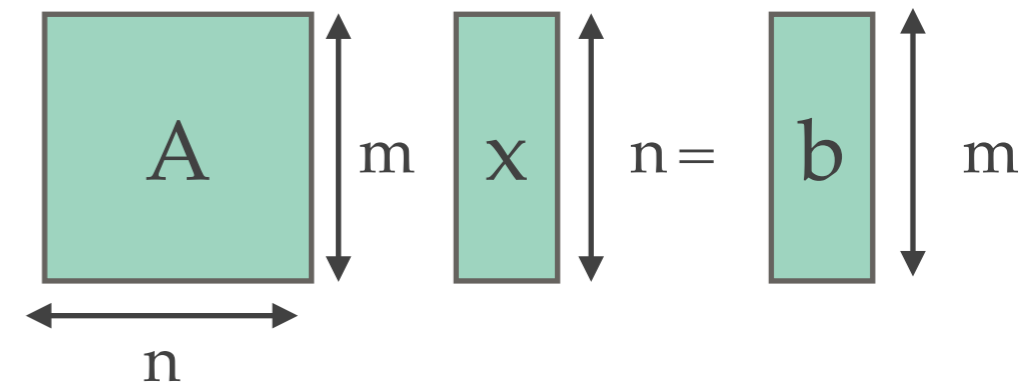
- Système masses-ressorts peut être vu comme un graphe
- Réorganisation par gagner en efficacité



# Résolution système linéaire

## Résolution système linéaire de type $Ax = b$

- avec  $A$  matrice de taille  $m \times n$  de scalaires
- $x$  est un vecteur à  $n$  inconnus
- $b$  est un vecteur avec  $m$  entrées



### Résolution possible avec des solveurs linaires

- méthode Jacobi
- méthode Gauss-Seidel

### Principe de ces méthodes itératives

- Méthodes débutent avec une solution initiale :  $x_0$
- Puis, solveurs calculent en itérant une approximation :  $x^{k+1} = f(x^k)$

Solution peut être en temps réel

# Résolution système linéaire

## Méthode Jacobi

- Suppose que les valeurs diagonales  $a_{ii}$  ne sont pas nulles
- Solutions itératives donnée par :

$$x_i^{k+1} = \frac{\left(b_i - \sum_{j \neq i} a_{ij} x_j^k\right)}{a_{ii}}$$

## Méthode Gauss-Seidel

- Résolution séquentielle de chaque équation
- Résolution basée sur l'état courant et mise à jour directe du résultat

$$x_i^{k+1} = \frac{\left(b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k\right)}{a_{ii}}$$

valeurs courantes

$x^{k+1}$

valeurs de l'itération précédente

$x^k$

# Master 1 Informatique - UE M1if37

## Animation en synthèse d'image

### Partie - Simulation par modèles physiques

---

#### Cours 4 - Système masses-ressorts

Florence Zara

LIRIS - Université Lyon 1

<http://liris.cnrs.fr/florence.zara>

