

# Méthode des Masses Tenseurs : un outil efficace en simulation bio-mécanique ?

Xavier FAURE, Florence ZARA, Fabrice JAILLET

Équipe SAARA - LIRIS

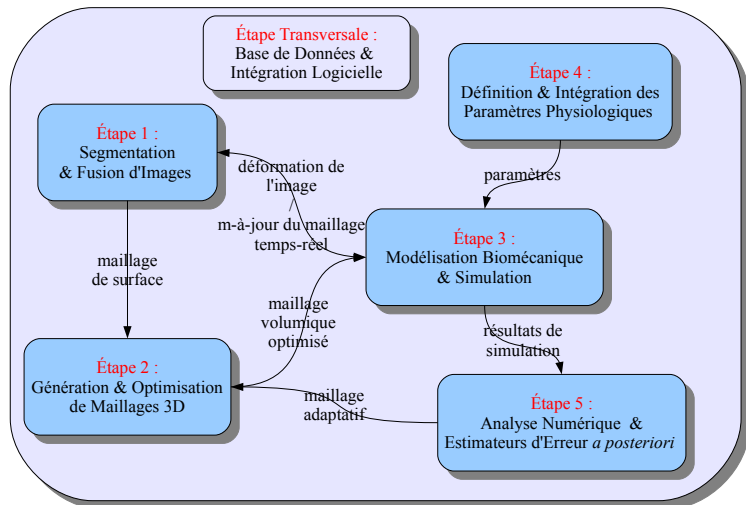
<http://liris.cnrs.fr/saara>

Séminaire PRIMES

15 avril 2014



# Contexte : Informatique Graphique appliquée à la Santé



## Vers un modèle bio-mécanique idéal :

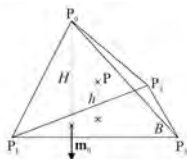
- suffisamment rapide (hors chocs) ;
  - planning (h) -> diagnostic (min) -> entraînement (TR !)
- suffisamment précis ;
  - acquisition souvent invasive
  - validation complexe (ex vivo, in vivo experimental)
- et patient spécifique.

# Comment gagner en rapidité :

## Une combinaison entre ?

- modélisation géométrique simplifiée ;
- loi comportement simplifiée ou adaptée ;
- choix du type d'éléments, du nombre de nœuds ;
- calculs préliminaires ;
- raffinement/déraffinement du maillage ;
- calcul parallèle (multi CPU, multi GPU) ;
- ...

# Système Masses-Tenseurs



Modèle **discret** basé sur **continu**, similaire à la MEF :

- Énergie de déformation  $W = (I_1, I_2, I_3)$
- Invariants du tenseur des déformations de Cauchy-Green  
 $C = F^T F = U^2$

Propriétés :

- Précis, mais structure de données conséquente
- Lois de comportement variées, paramétrisation facile
- Forces discrètes (similaire Masses-Ressorts)
- Changements de topologie

[Delingette 1999] [Picinbono 2000] [Schwarz 2003]

## Calcul de la force

Dérivation de l'énergie sur la position nodale :

$$\vec{F} = \frac{\partial W}{\partial \vec{U}}$$

Pour le sommet  $I$ , décomposable sur un élément :

$$\vec{F}_I = -\frac{\partial W}{\partial \vec{U}_I} = -\sum_{T_j} \vec{F}_{i-E_j} \quad \text{avec} \quad \vec{F}_{i-E_j} = \frac{\partial W_{E_j}}{\partial \vec{U}_i}$$

Énergie élastique linéaire, pour un matériau isotrope homogène :

Tenseur Green-St Venant linéarisé :  $\varepsilon = \frac{1}{2}(\nabla U + \nabla U^T)$

Loi de Hooke : Énergie de déformation quadratique selon le déplacement  $U$  du point  $\vec{X}$  :

$$\begin{aligned} W &= \frac{\lambda}{2}(\text{tr}(\varepsilon))^2 + \mu \text{tr}(\varepsilon^2) \\ &= \frac{\lambda}{2}(\text{div}U)^2 + \mu \|\nabla U\|^2 - \frac{\mu}{2} \|\text{rot } U\|^2 \end{aligned}$$

Intégration de l'énergie sur l'élément

$$W_E = \int_E W(\vec{X}) dX = \int_R W(\vec{R}) \det([J(\vec{R})]) dR$$

utilisation des points de Gauss

# Sur un Tétraèdre P1 :

Sous hypothèse P1 : gradient du champ de déplacement constant sur l'élément

=> Fonctions d'interpolations linéaires sur l'élément (attention cas général : élément de référence)

$$U(\vec{X}) = \sum_{j=0}^3 \Lambda_j(\vec{X}) U_j \text{ et } \Lambda_j(\vec{X}) = \alpha_j \vec{X} + \beta_j$$

Énergie et force sur tétra P1 :

$$W_{E_i} = \sum_{j,k=0}^3 U_j^t B_{jk}^{E_i} U_k$$

$$\text{avec } B_{jk}^{E_i} = \frac{\lambda}{2}(\alpha_j \otimes \alpha_k) + \frac{\mu}{2}((\alpha_k \otimes \alpha_j) + (\alpha_j \cdot \alpha_k) Id_3)$$

la contribution du tétra  $i$  au tenseur de raideur de l'arête  $(j, k)$

et par dérivation : force de l'élément sur le point  $p$

$$F_{p-E_i} = 2 \sum_j^3 [B_{pj}^{E_i}] U_j$$



# Extension à l'élasticité non linéaire : Saint Venant - Kirchhoff

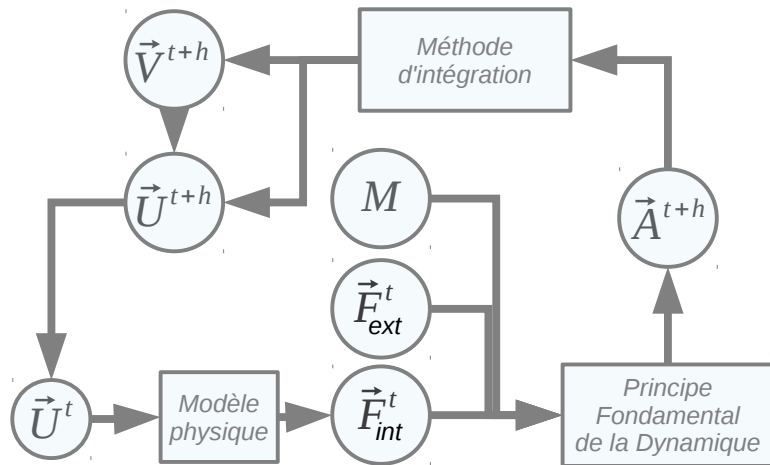
Tenseur Green-St Venant complet :

$$\varepsilon = \frac{1}{2}(\nabla U + \nabla U^T + \nabla U^T \nabla U)$$

et répartition des constantes pour StVK :

	Matrices	Vecteurs	Scalars
Sommet $p$	$B^{pp}$	$C^{ppp}$	$D^{pppp}$
Arête $pj$	$B^{pj}$	$C^{ppj}$ $C^{ppj}$ $C^{jjp}$ $C^{pjj}$	$D^{jppp}$ $D^{jjjp}$ $D^{jppj}$ $D^{pjjp}$ $D^{jjpp}$
Triangle $pjk$		$C^{jkp}$ $C^{kjp}$ $C^{pjk}$	$D^{jkpp}$ $D^{jpkp}$ $D^{pjkp}$ $D^{jjkp}$ $D^{jkjp}$ $D^{kjpp}$ $D^{kkjp}$ $D^{kjkp}$ $D^{jkkp}$
Tetraèdre $pjkl$			$D^{jklp}$ $D^{jlkp}$ $D^{kjlp}$ $D^{kljp}$ $D^{ljkp}$ $D^{lkjp}$

# La boucle de simulation

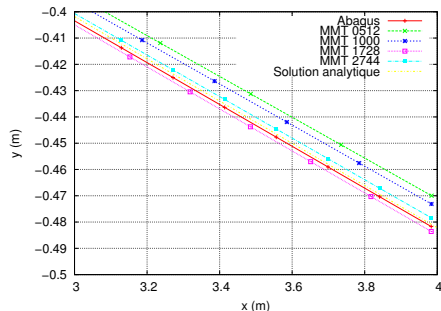
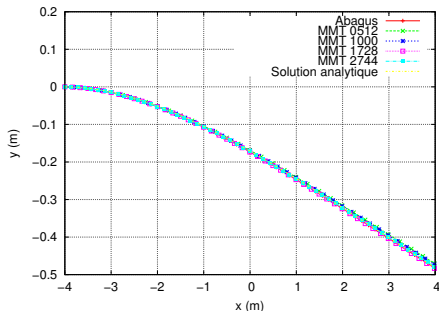


Ce qu'il faut retenir à ce niveau : méthode basée sur la mécanique des milieux continus

- loi élasticité linéaire (Hooke) ou non (StVK)
- contrôle de volume, anisotropie
- force définie sur le sommet (et pas sur l'élément) => gestion plus simple des interactions et modifications topologiques
- pré-calcul des constantes
- parallélisation sur GPU

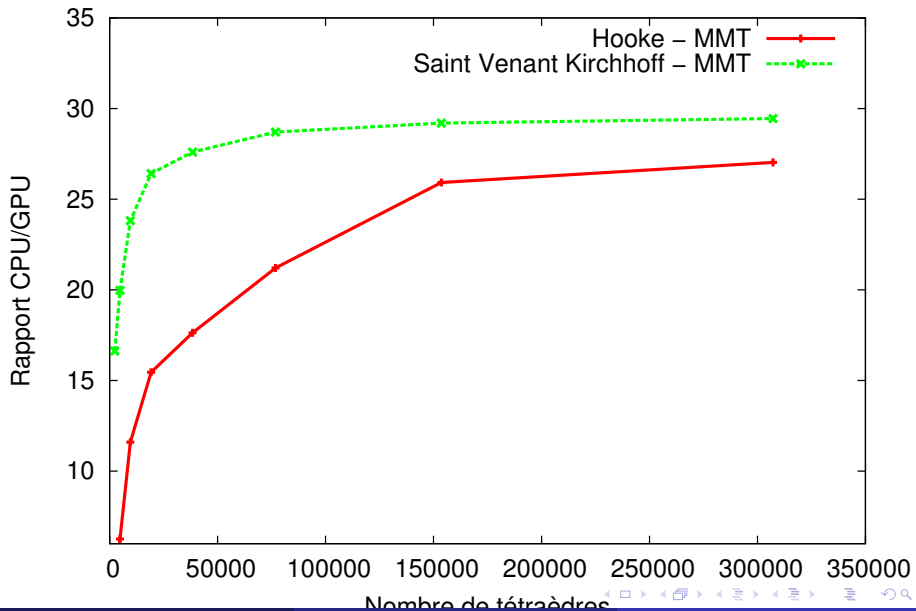
# Convergence de la solution et comparaison

## Poutre encastrée $1 \times 1 \times 8$ , 100MPa, Hexaèdres



env. 0.1% d'erreur

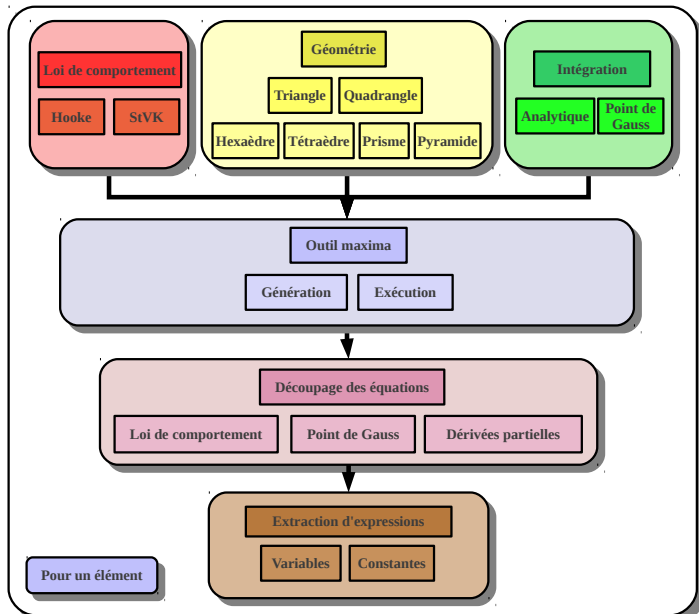
# Parallélisation sur GPU (OpenCL)



Utilisation avantageuse de génération formelle des équations des forces (internes) : [Faure 2012]

- type d'éléments quelconque : il suffit d'avoir les fonctions de forme
  - triangle, quad, tetra, hexa, prisme, pyramide
  - maillages à éléments mixtes
- formulation analytique dépendant de la loi de comportement et type d'élément
  - énergie et force (intégration explicite)
  - dérivée de la force (intégration implicite **uniquement en formel**)

# Le schéma de génération formelle des forces internes



# Principe de la génération formelle :

## Différentes étapes mise en place :

- 1 Énergie dépendant des : positions et déplacements des points, coefficients de Lamé et points de Gauss et leur poids
- 2 Génération de graphes d'expressions commutatives ;
- 3 Facilite la recherche et le regroupement des termes des expressions constantes ;
- 4 Formulation de la force et des dérivées ;

## Difficultés à résoudre :

- taille des équations : fragmentation pour la compilation et ré-assemblage
  - 2 termes de la loi de Hooke, puis par point de Gauss et enfin par coordonnée
- extraction automatique des expressions constantes



Coût en temps et mémoire :

Sur un maillage tétraédrique P1 :

- Loi de Hooke : pas de gain (équation analytique simple)
- Loi de StVK
  - formel : 15x plus rapide que analytique
  - mémoire : 280 réels (cf tableau) / 11 expressions constantes par tétra

=> Selon le type d'éléments : plus la complexité augmente, plus la formalisation est intéressante.

Simple d'ajouter une nouvelle expression

Long à compiler, mais plus rentable à l'exécution !

## Souplesse du modèle discret : Définitions de stratégies d'adaptation du modèle

Différentes **actions** possibles, pendant la boucle de simulation :

- loi de comportement : Hooke  $\Leftrightarrow$  StVK ;
- différentes géométries : triangle, quad, tétras, etc...
- nombre de nœuds d'interpolation ;
- fonctions de formes différentes ;
- raffinement/déraffinement.

=> Calcul ou recalcul des constantes le cas échéant.

[Caillaud 2013]

## Différents critères

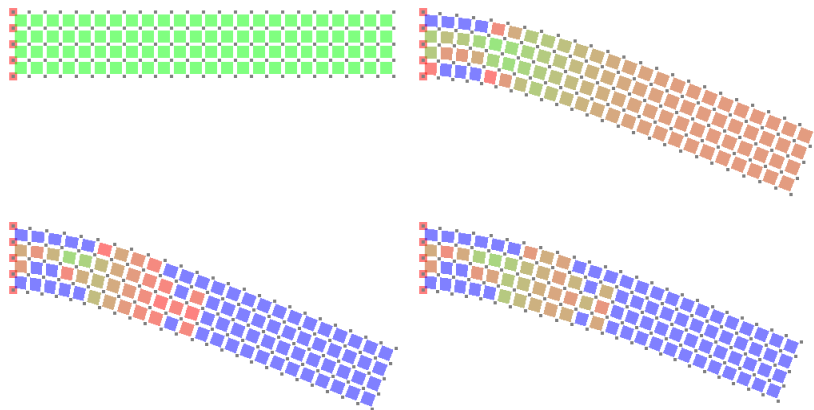
### Géométriques :

- aspect ratio, longueur d'arête, Jacobien, etc.

ou mieux, **physiques** :

- stress de l'élément  $R_S = \frac{1}{V(t)} \int_E \mathbf{F}_E(x, y, z) dx dy dz$
- qui doit être discrétisé, et calcul par interpolation des forces aux points de Gauss  $R_S = \frac{1}{A(t)} \sum_{i=0}^{n_g-1} w_i \sum_{j=0}^{n-1} \|\Lambda_j(g_i) \mathbf{F}_E(P_j)\|$

# Changement de loi dynamique



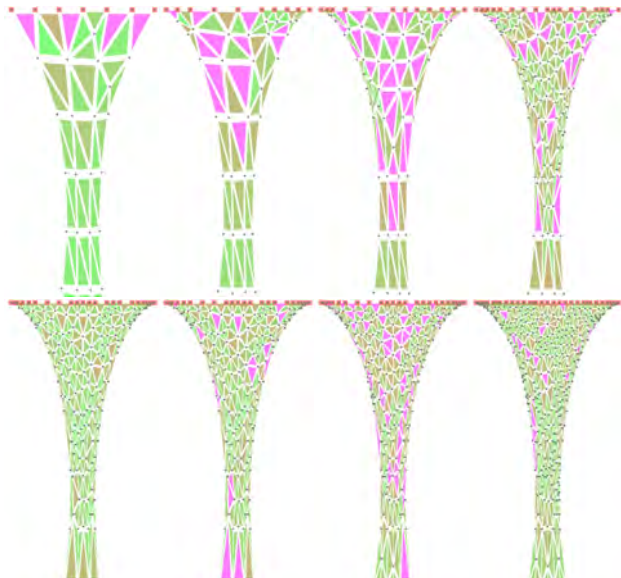
Hooke => 100FPS, 10% erreur

Contrainte faible

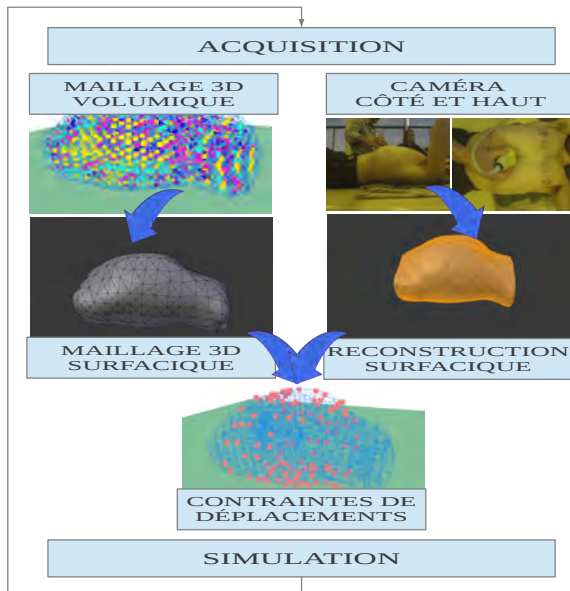
Contrainte forte => intermédiaire 25FPS, 1% erreur

StVK => 1FPS, erreur nulle

# Exemple de raffinement (quasi-statique) sur critère de Stress



# Application en simulation de la déformation du thorax







## Étude en cours :

- Combinaison avec les **cartes combinatoires**
  - assure l'intégrité du modèle lors d'opé. topologiques : découpe, raffinement. . . (avec E. Fléchon et G. Damiand, LIRIS)
- Génération et adaptation dynamique de maillages
  - octree et motifs de subdivision locaux sur **maillages mixtes** (avec C. Lobos, Chili)
  - raffinement/déraffinement local (avec N. Moreau et V. Vidal, LIRIS)
- **validation et tests** de performance complets du modèle !

## À voir ?

- extension à d'**autres formulations** de l'énergie ?
  - Mooney-Rivlin, Yeoh, Neo-Hooke, etc.
- méthode voisine **HEML** [Goulette 2006] ?

-  Florian Caillaud, Xavier Faure, Florence Zara, Fabrice Jaillet, and Jean-Michel Moreau, *Multi-criteria adaptation of physical simulations*, November 2013, Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS 2013), "Work in Progress" session, pp. 1–4.
-  Hervé Delingette, Stéphane Cotin, and Nicholas Ayache, *A Hybrid Elastic Model Allowing Real-Time Cutting Deformations and Force Feedback for Surgery Training and Simulation*, Computer Animation'99, IEEE Computer Society, 1999, pp. 70–81.
-  Xavier Faure, Florence Zara, Fabrice Jaillet, and Jean-Michel Moreau, *An Implicit Tensor-Mass solver on the GPU for soft bodies simulation*, Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS), December 2012, pp. 1–10.
-  François Goulette and Safwan Chendeb, *A framework for fast computation of hyper-elastic materials deformations in real-time simulation of surgery*, Proceedings of Computational Biomechanics for Medecine (2006), 66.
-  Guillaume Picinbono, Herve Delingette, and Nicholas Ayache, *Real-Time Large Displacement Elasticity for Surgery Simulation : Non-linear Tensor-Mass Model*, Proceedings of MICCAI'00 (London, UK), Springer-Verlag, 2000, pp. 643–652.
-  J.M. Schwartz, *Calcul rapide de forces et de déformations mécaniques non-linéaires et visco-élastiques pour la simulation de chirurgie*, PhD, University Laval (2003).