

## Modèle pour la simulation de tissus connectifs

Julien Bosman, Christian Duriez, Stéphane Cotin

► **To cite this version:**

Julien Bosman, Christian Duriez, Stéphane Cotin. Modèle pour la simulation de tissus connectifs. Revue Electronique Francophone d'Informatique Graphique, Association Française d'Informatique Graphique, 2014, pp.13. hal-01095301

**HAL Id: hal-01095301**

**<https://hal.inria.fr/hal-01095301>**

Submitted on 15 Dec 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Modèle pour la simulation de tissus connectifs

Julien Bosman, Christian Duriez, Stéphane Cotin  
INRIA équipe SHACRA / LIFL - Université Lille 1

## Résumé

Dans le cadre de la simulation médicale, les récents travaux ont conduit à de réels progrès en terme de simulation d'organes. Cependant, la modélisation des tissus connectifs est très souvent ignorée malgré le rôle mécanique majeur qu'ils peuvent jouer dans l'interaction entre certains organes. Nous proposons un modèle de simulation de tissus connectifs reposant sur des méthodes sans maillage basées sur des repères. Le réalisme des simulations repose sur une formulation variationnelle des équations de la mécanique des milieux continus, proche de la méthode par éléments finis. Notre approche permet de simuler des objets déformables aux topologies diverses (volumiques, surfaciques et linéiques). Dans le souci d'obtenir une simulation stable de la dynamique de ces corps déformables, nous utilisons une intégration temporelle implicite. Grâce à notre méthode, il est possible de modéliser ces tissus, sans présager de leur géométrie, permettant ainsi le couplage mécanique entre les organes.

**Mots-clés** : Simulation médicale, Méthodes sans maillage, Mécanique, Objets déformables.

## 1 Introduction

La simulation médicale est un domaine de recherche de plus en plus actif et les récents progrès ont permis d'ouvrir de nouveaux champs d'application. Initialement centrés sur la pratique chirurgicale, ces travaux restent nécessaires car les simulateurs commerciaux actuels souffrent parfois d'un manque de réalisme. Cependant, les résultats obtenus en recherche permettent d'améliorer sensiblement la précision et l'efficacité des simulations, tout en offrant la possibilité d'intégrer des données basées sur

des patients réels (généralement issues de l'imagerie médicale) [DPP12], [LPC\*12] ou encore [HNR\*10]. Ces travaux permettent d'envisager de nouvelles applications, notamment la planification pré-opératoire ou encore la simulation comme assistance aux chirurgiens durant l'intervention.

Ces applications ont en commun la nécessité de continuer à fiabiliser la simulation et à améliorer l'interactivité, d'une part en améliorant les modèles biomécaniques, et d'autre part, en réduisant au maximum les temps de calcul. Parmi les travaux récents, nombreux sont ceux qui traitent d'organes spécifiques tels que le foie [MHC\*10], le cerveau [HWM07] ou encore les organes du système pelvien [RRD\*11]. Cependant, ces organes sont grandement influencés par les structures anatomiques environnantes (organes, tendons, ligaments, os, etc). Ces dernières ne peuvent donc pas être négligées lors des simulations, sous peine de ne pas reproduire fidèlement le comportement de chacun des organes.

Actuellement, peu de travaux ont cherché à modéliser les interactions mécaniques entre les organes. Dans [CJA\*10], ces interactions sont modélisées grâce à une loi de contact frottant associée à une méthode de résolution optimisée. Bien que constituant un progrès car permettant de simuler plusieurs organes en interaction, ce modèle n'est pas totalement conforme à la réalité car certains organes ne sont pas en contact direct. En effet, ceux-ci sont entourés et reliés entre eux par des fibres appelées *tissus connectifs* ou *fascias*. On peut prendre pour exemple le foie qui est attaché au diaphragme par le ligament falciforme et à l'estomac par le ligament hépato-gastrique. Or, lorsque ces interactions sont simulées grâce à des contacts frottants, ces organes "glissent" les uns sur les autres alors qu'ils devraient être, au moins partiellement, liés.

Une des principales difficultés posées par les tissus connectifs se trouve dans le fait que certains d'entre eux

ne possèdent pas une topologie uniforme. En effet, ces tissus occupent l'espace présent entre les organes. Ils peuvent ainsi présenter à la fois des parties volumiques, et des parties très fines, quasi surfaciques, au sein d'une même structure anatomique. De manière générale, ces tissus peuvent avoir des géométries très variées (volumiques, surfaciques voire linéiques). Modéliser ces tissus grâce à des maillages volumiques est ainsi problématique. La modélisation des parties très fines nécessite l'utilisation d'éléments de petite taille et soulève des problèmes de performance, dus au grand nombre d'éléments produits, voire de stabilité si ces éléments sont mal formés. De plus, certains de ces tissus sont difficilement visibles par imagerie médicale et sont mal documentés. En raison de ces problèmes, les tissus connectifs sont souvent négligés dans la plupart des simulations. Pourtant, ils jouent un rôle mécanique majeur dans la mesure où ils lient les structures anatomiques entre elles, et influencent les conditions aux limites des modèles des organes.

Dans ce travail, nous cherchons à développer une méthode numérique capable de simuler ces tissus, et notamment la liaison mécanique qu'ils assurent entre les organes afin d'améliorer le réalisme des interactions entre ces derniers. Dans ce but, nous proposons une approche nouvelle, où ces tissus sont modélisés sous la forme d'une interface faisant appel aux techniques de simulation sans maillage, ici basées sur des repères, permettant le couplage entre différents modèles mécaniques simulant des organes. Notre méthode permet également la propagation des conditions aux limites entre des organes en contact indirect.

## 2 Etat de l'art

Couramment employée en simulation mécanique, la simulation basée sur des méthodes sans maillage constitue une alternative aux méthodes de type éléments finis (FEM). La principale différence entre ces deux méthodes intervient au niveau de la discrétisation des objets simulés. Dans une approche FEM, les objets sont discrétisés grâce à un ensemble d'éléments tétraédriques ou hexaédriques constituant un maillage. Les nœuds de ces maillages ont une connectivité structurée par les arêtes de leurs éléments. Dans le cas des méthodes sans maillage, les objets sont discrétisés par un ensemble de points matériels

(souvent appelés particules) que l'on peut assimiler aux nœuds rencontrés dans les FEM. A la différence des méthodes de type éléments finis, ces points ont une connectivité non-structurée. Les méthodes sans maillage sont associées à une description *lagrangienne* du problème. Dans une approche lagrangienne, on observe l'évolution des propriétés de chaque particule (position, déplacement, etc.) au cours du temps, contrairement à l'approche eulérienne où l'on observe cette évolution au cours du temps sur un ensemble de positions fixes dans le domaine (*ex* : un champ de vitesse). Le principal avantage des méthodes sans maillage par rapport aux FEM se situe au niveau de la discrétisation. En effet, l'échantillonnage d'un objet par un ensemble de points est plus simple et plus rapide que la construction d'un maillage volumique. De fait, ces méthodes se révèlent intéressantes dans le cas d'un échantillonnage adaptatif ou dans le cas où il est nécessaire de gérer des changements topologiques sur un objet (découpes, fractures). Cependant, un des inconvénients des méthodes sans maillage est qu'elles sont plus coûteuses en calcul, notamment en raison du parcours des particules voisines et du calcul des fonctions de forme, qui est plus complexe que dans le cas des éléments finis. Il est intéressant de noter que les premiers travaux en mécanique envisageaient les méthodes sans maillage comme une généralisation des méthodes par éléments finis [NTV92]. Pour une introduction et une nomenclature des méthodes sans maillage, le lecteur pourra se référer à [FM04].

Les méthodes sans maillage ont été introduites en informatique graphique par Cani et Desbrun avec [DC96] en popularisant la méthode des *Smoothed Particle Hydrodynamics* (SPH), qui était initialement destinée à la simulation de corps célestes. Le principe de cette méthode repose sur l'interaction réciproque d'un ensemble de particules en utilisant des fonctions à base radiale et à support compact. Les SPH permettent l'animation de modèles très déformables. Dans [BIT09] Becker *et al.* proposent un modèle corotationnel dédié apportant une réponse aux problèmes des non-linéarités géométriques (rotations locales). Dans leur travaux de Paiva *et al.* [PPLT09], et Hosseini *et al.* [HMH07], adaptent l'utilisation des SPH à la simulation de matériaux non-newtoniens, aux comportements plus complexes. Dans [GGB09] Gerszewski *et al.* proposent une approche sans maillage proche des SPH permettant de simuler la plasticité des matériaux. Cependant, les SPH souffrent de certains problèmes pouvant li-

miter leur utilisation. En effet, cette méthode offre une précision limitée. Le traitement des frontières est également difficile en raison de la déficience en particules sur les frontières de l'objet. De plus, les SPH posent le problème du phénomène connu sous le nom d'*instabilité en tension* qui apparaît lorsque l'objet est fortement étiré. Pour ces raisons, les SPH sont difficilement utilisables dans le cadre de la simulation de grandes déformations.

Parallèlement aux SPH, d'autres méthodes d'animation sans maillage ont été développées. Des approches basées sur la mécanique des milieux continus ont été proposées par Keiser *et al.* dans [KNP\*04] et par Horton *et al.* dans [HWJM10]. Dans ces travaux, le champ de déformation (*i.e.* le champ vectoriel représentant la déformation) est approximé par la méthode des *Moving Least Squares* (MLS). Cette méthode, couramment employée en mécanique [BLG94], offre une bonne précision. Cependant, l'approximation MLS n'est pas toujours possible, notamment si la distribution des nœuds est colinéaire ou coplanaire. En effet, avec de telles distributions de points, la matrice des moindres carrées devient singulière rendant impossible le calcul des fonctions de forme.

Des techniques d'animation non basées sur des lois physiques, mais gérant les rotations locales, ont également été développées. Dans [MHTG05], Müller *et al.* proposent la méthode du *Shape Matching* dans laquelle les points de l'objet simulé sont partitionnés en plusieurs groupes pour lesquels le mouvement est décomposé en une rotation optimale et une translation. Le retour à la configuration non déformée est ensuite assuré par un système masse-ressort. Cette idée est reprise dans [MC11], où l'orientation des particules est estimée grâce à un *Shape Matching* généralisé. Cependant, malgré leur rapidité et leur simplicité d'utilisation, ces méthodes ne permettent pas, à proprement parler, une simulation basée sur les lois de la physiques.

Des méthodes de simulation hybrides ont également été proposées, notamment par Kauffmann *et al.* dans [KMBG08] ou par Sifakis *et al.* dans [SSIF07], où un nuage de points est associé au maillage des objets simulés afin de faciliter la gestion des collisions et des changements topologiques (découpés et fractures). Dans [GLB\*06], Guo *et al.* proposent une simulation basée sur une formulation en éléments coque à partir d'un nuage de points.

Des méthodes sans maillage basées sur des nœuds à

six degrés de liberté ont également été proposées. Dans [MKB\*10] Martin *et al.* proposent un modèle de simulation unifiée d'objets linéiques, surfaciques et volumiques grâce à une généralisation des MLS permettant de gérer les distributions de points coplanaires et colinéaires. Ils introduisent également la notion d'*elaston*, une généralisation de la déformation indépendante de la topologie des objets. Cependant, l'utilisation des MLS généralisées est plus coûteuse en terme de calcul, et cette méthode est difficilement utilisable dans le contexte de la simulation temps-réel. Enfin, dans [GBFP11], Gilles *et al.* proposent un modèle, reposant sur la mécanique des milieux continus, basé sur des repères et utilisant des *duals-quaternions*. Cette méthode permet de gérer efficacement de grandes déformations, et plus particulièrement sur des modèles faiblement échantillonnés. Cependant, l'arithmétique sur les duals-quaternions est plus coûteuse en calcul et leur utilisation se justifie principalement par l'utilisation de modèles faiblement échantillonnés. Dans la mesure où nos modèles sont plus densément échantillonnés, nous utilisons une méthode plus simple. De plus, l'approche présentée par Gilles *et al.* repose également sur le précalcul des matrices de masse et de raideur permettant la généralisation du modèle à un système masse-ressort plus efficace en terme de calcul. Cependant, le précalcul des matrices de raideur n'est pas possible dans le cadre de l'utilisation d'une loi de comportement hyperélastique (élasticité non linéaire).

### 3 Contribution

Nos contributions sont multiples. Notre travail offre la possibilité de simuler les tissus connectifs et plus particulièrement le couplage entre différents modèles mécaniques. L'originalité de ces travaux repose sur l'utilisation d'une méthode sans maillage couplée à des modèles de déformation maillés utilisés pour simuler les organes. L'avantage d'une méthode sans maillage se trouve dans une discrétisation plus aisée et plus efficace d'objets présentant à la fois des parties volumiques et des parties très fines. En effet, discrétiser un tel objet par un maillage volumique nécessiterait d'utiliser des éléments de très petite taille pour préserver les parties fines, et augmenterait sensiblement les temps de calcul. De plus, notre travail permet le couplage d'objets de différentes résolutions de

façon simple sans requérir d'opérations supplémentaires : l'utilisation d'une interface modélisée par une méthode sans maillage permet de lier différents objets sans poser de contrainte sur leurs résolutions. Notre approche permet également de simuler des structures aussi bien volumiques, surfaciques que linéiques. Enfin nous adaptons les formulations de la mécanique des milieux continus à des objets discrétisés par des repères à six degrés de liberté afin d'obtenir une simulation réaliste. A notre connaissance, il n'existe pas d'autre travaux concernant la simulation temps-réel de tissus connectifs.

## 4 Simulation

### 4.1 Nomenclature

$\bar{\mathbf{x}}_i, \mathbf{x}_i$	coordonnées du repère $i$ dans la configuration au repos et déformée
$\bar{\mathbf{r}}_i, \mathbf{r}_i$	orientation du repère $i$ dans la configuration au repos et déformée
$\bar{\mathbf{p}}_s, \mathbf{p}_s$	coordonnées globales du point d'intégration $s$ dans la configuration au repos et déformée
$\bar{\mathbf{I}}_s^i, \mathbf{I}_s^i$	coordonnées du point d'intégration $s$ dans le repère $i$ dans la configuration au repos et déformée.
$\bar{\mathbf{R}}_i, \mathbf{R}_i$	matrice de rotation associée à l'orientation du repère $i$ dans la configuration au repos et déformée
$[\mathbf{v}^\wedge]$	matrice antisymétrique associée au produit vectoriel de $\mathbf{v}$
$[\mathbf{M}]^i$	$i^{\text{ème}}$ ligne de la matrice $\mathbf{M}$
$[\mathbf{M}]^{\cdot j}$	$j^{\text{ème}}$ colonne de la matrice $\mathbf{M}$
$\{\mathbf{S}\}$	tenseur en notation de Voigt

### 4.2 Cinématique

Les objets déformables sont discrétisés par un ensemble de repères et de points d'intégration, aussi appelés *points de Gauss* (figure 1). Ces derniers sont disposés entre les repères et permettent d'échantillonner le volume des objets simulés. Les repères  $\mathbf{q}_i$  possèdent six degrés de liberté (3 en translation et 3 en rotation) :

$$\mathbf{q}_i = [x_1 \ x_2 \ x_3 \ r_1 \ r_2 \ r_3]^t = [\mathbf{x}_i \ \mathbf{r}_i]^t \quad (1)$$

où  $\mathbf{x}_i$  représente les coordonnées et  $\mathbf{r}_i$  le vecteur rotation correspondant à l'orientation du repère. Les points d'intégration ne possèdent que trois degrés de liberté (3 en translation). L'approche s'appuyant sur une formulation en lagrangien total, la position globale des points d'intégration est exprimée en fonction de leur position dans chacun des repères dans la configuration non déformée. De cette façon, les points d'intégration sont asservis aux repères environnants. Les coordonnées d'un point d'intégration  $s$  dans le repère  $i$  dans la configuration non déformée sont données par :

$$\bar{\mathbf{I}}_s^i = \bar{\mathbf{R}}_i^t \cdot [\bar{\mathbf{p}}_s - \bar{\mathbf{x}}_i] \quad (2)$$

où  $\bar{\mathbf{R}}_i$  et  $\bar{\mathbf{x}}_i$  sont la matrice de rotation et les coordonnées du repère  $i$  et  $\bar{\mathbf{p}}_s$ , les coordonnées du point d'intégration  $s$  dans la configuration non déformée. Les coordonnées du point d'intégration  $s$  dans la configuration déformée sont données par :

$$\mathbf{p}_s = \sum_i w_i(\bar{\mathbf{p}}_s) (\mathbf{x}_i + \mathbf{R}_i \cdot \bar{\mathbf{I}}_s^i) \quad (3)$$

où  $w_i(\cdot)$  est une fonction de forme associée au point d'intégration  $s$ , et  $\mathbf{x}_i$  et  $\mathbf{R}_i$  les coordonnées et la rotation du repère  $i$  dans la configuration déformée. La fonction de forme  $w_i(\cdot)$  permet l'interpolation des valeurs de déplacements qui ne sont connues qu'aux repères afin d'estimer le déplacement en chaque point d'intégration. Elle peut être à support compact (du type *fonction de Wendland* ou *spline cubique*) ou non (*fonction de Shepard* par ex).

De la même façon, la cinématique des repères est influencée par le mouvement des points d'intégration qui lui sont associés. La variation de l'orientation du repère  $i$  en fonction des déplacements des points d'intégration est donnée par :

$$\frac{d\mathbf{r}_i}{d\mathbf{p}} = \sum_s w_i(\bar{\mathbf{p}}_s) (\mathbf{I}_s^i \wedge d\mathbf{p}_s) \quad (4)$$

où  $\mathbf{I}_s^i$  sont les coordonnées du point d'intégration  $s$  dans le repère  $i$ .

### 4.3 Dynamique

#### 4.3.1 Expression des forces internes

Notre approche se base sur la mécanique des milieux continus (MMC). Les matériaux étudiés sont considérés

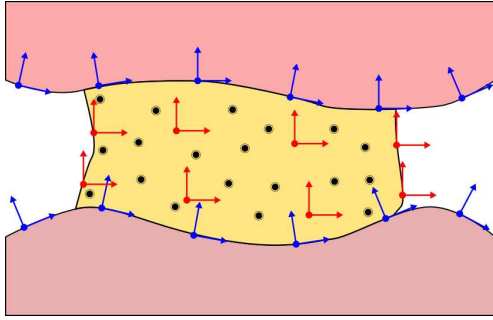


FIGURE 1 – Discretisation des objets. Les tissus connectifs (en jaune) sont discrétisés grâce à des repères (en rouge) et des points d'intégration (en noir). Ils sont attachés à des organes (en rose) dont les surfaces sont elles-mêmes constituées de repères (en bleu).

comme continu, c'est-à-dire à une échelle suffisamment grande pour négliger les espaces inter-moléculaires. La MMC permet de mettre en relation un déplacement (*i.e.* les déformations subies par un point) avec une force de rappel selon le schéma suivant :

$$\underbrace{\mathbf{u}_i}_{\text{déplacements}} \rightarrow \underbrace{\mathbf{F}_s}_{\text{gradient de déformation}} \rightarrow \underbrace{\mathbf{L}_s}_{\text{tenseur de déformation}} \rightarrow \underbrace{\mathbf{S}_s}_{\text{tenseur de contrainte}} \rightarrow \underbrace{\mathbf{f}_i}_{\text{force}}$$

Au cours de la simulation, les forces sont uniquement appliquées sur les repères qui sont les degrés de liberté de nos modèles déformables. Les points d'intégration servent au calcul du gradient de déformation et des tenseurs de déformation et de contrainte qui permettront ensuite le calcul des forces. En effet, le calcul des forces internes à appliquer sur les repères doit se faire par intégration sur le volume de l'objet car ces forces sont volumiques. La force à appliquer sur un repère s'effectue donc en accumulant la contrainte calculée sur les points d'intégration voisins. Les points d'intégration servent ainsi à échantillonner le volume de l'objet entre les repères. Plus l'échantillonnage est dense, plus le calcul des forces est précis.

La méthode décrite dans cet article utilise une formulation en lagrangien total. Dans une formulation en lagrangien total, les déplacements sont évalués par rapport à la configuration non déformée de l'objet, contrairement à la formulation en lagrangien actualisé, où ils sont évalués par

rapport à la configuration de l'objet au pas de temps précédent. Le calcul des forces internes  $\mathbf{f}$  fait appel au principe des *travaux virtuels* :

$$\delta W = \delta \mathbf{q}^t \cdot \mathbf{f} = \delta \mathbf{q}^t \int_{\Omega} \frac{\partial \mathbf{L}^t}{\partial \mathbf{q}} \{\mathbf{S}\} d\Omega \quad (5)$$

où  $\delta W$  représente la variation de travail,  $\mathbf{L} = \frac{1}{2}(\mathbf{F}^t \mathbf{F} - \mathbf{I})$  est le tenseur de déformation de *Green-Lagrange* avec  $\mathbf{F}$ , le gradient de déformation.  $\{\mathbf{S}\}$  est le second tenseur de contrainte de *Piola-Kirchhoff* dans la notation de *Voigt* et  $\Omega$  le domaine. La dérivée du tenseur de Green-Lagrange par rapport aux degrés de liberté des repères est donnée par :

$$\frac{\partial \mathbf{L}}{\partial \mathbf{q}} = \frac{1}{2} \left( \frac{\partial \mathbf{F}^t}{\partial \mathbf{q}} \mathbf{F} + \mathbf{F}^t \frac{\partial \mathbf{F}}{\partial \mathbf{q}} \right) \quad (6)$$

La première étape du calcul des forces internes consiste à calculer le tenseur de déformation en un point d'intégration  $s$ , en d'autres termes, la dérivée spatiale des déplacements  $\mathbf{F}_s = \nabla \mathbf{p}_s$ . Or, les déplacements sont donnés par l'équation (3). On peut remarquer que dans cette équation, seuls les termes  $w_i(\bar{\mathbf{p}}_s)$  et  $\bar{\mathbf{l}}_s^i$  sont concernés par la dérivation. En reprenant l'équation (2), on remarque que :

$$\frac{\partial \bar{\mathbf{l}}_s^i}{\partial \bar{\mathbf{x}}} = \frac{\partial (\bar{\mathbf{R}}_i^t \cdot [\bar{\mathbf{p}}_s - \bar{\mathbf{x}}_i])}{\partial \bar{\mathbf{x}}} = \bar{\mathbf{R}}_i^t \quad (7)$$

En appliquant les règles de dérivation simples à l'équation (3) et en utilisant le résultat de l'équation (7), on obtient le gradient de déformation au point d'intégration  $s$  qui est une matrice  $3 \times 3$  :

$$\mathbf{F}_s = \frac{\partial \mathbf{p}_s}{\partial \bar{\mathbf{x}}} = \sum_i \left[ \frac{\partial w_{is}}{\partial \bar{\mathbf{x}}} (\mathbf{x}_i + \mathbf{R}_i \cdot \bar{\mathbf{l}}_s^i) + w_{is} \mathbf{R}_i \bar{\mathbf{R}}_i^t \right] \quad (8)$$

où  $w_{is} = w_i(\bar{\mathbf{p}}_s)$ . Afin d'obtenir la dérivée du tenseur de *Green-Lagrange* de l'équation (6) il est nécessaire de calculer la dérivée du gradient de déformation par rapport à  $\delta \mathbf{q}$  en chaque point d'intégration. On obtient  $\delta \mathbf{F}_s = \partial \mathbf{F}_s / \partial \mathbf{q}$ , qui est également une matrice  $3 \times 3$ . Ces neuf valeurs s'expriment donc en fonction d'une variation infinitésimale de déplacement et d'orientation des repères  $\delta \mathbf{q}$ , c'est-à-dire en fonction de  $\delta \bar{\mathbf{x}}$  et  $\delta \mathbf{r}$  ( $\delta \mathbf{q} = [\delta \bar{\mathbf{x}} \ \delta \mathbf{r}]^t$ ). Pour chaque coefficient  $i, j$  de  $\delta \mathbf{F}_s$  on a :

$$\delta \mathbf{F}_{ij}^s = \sum_n \frac{\partial w_{ns}}{\partial \bar{\mathbf{x}}_j} \delta \bar{\mathbf{x}}_n^i - \left( \frac{\partial w_{ns}}{\partial \bar{\mathbf{x}}_j} [\mathbf{R}_n \cdot \bar{\mathbf{l}}_s^i]^{i,j} + w_{ns} [\mathbf{R}_n \cdot [\bar{\mathbf{R}}_n^t]^{j,i}]^{i,j} \right) \cdot \delta \mathbf{r}_n \quad (9)$$

où les indices en minuscule dénotent la coordonnée spatiale et  $n$  identifie un repère.

$\mathbf{R}_n \cdot \bar{\mathbf{I}}_s^n$  correspond à la rotation de  $\bar{\mathbf{I}}_s$  par  $\mathbf{R}_n$ ,  $[\mathbf{v}^\wedge]$  la matrice antisymétrique  $3 \times 3$  associée au produit vectoriel,  $[\mathbf{M}]^i$  et  $[\mathbf{M}]^j$  correspondent respectivement à la  $i^{\text{ème}}$  ligne et à la  $j^{\text{ème}}$  colonne de la matrice  $\mathbf{M}$ .

En réécrivant les neuf valeurs de  $\delta \mathbf{F}_s$  sous forme d'un vecteur colonne, l'équation (9) peut être exprimée sous la forme d'un produit matrice-vecteur :

$$\begin{bmatrix} \delta \mathbf{F}_{11}^s \\ \delta \mathbf{F}_{12}^s \\ \vdots \\ \delta \mathbf{F}_{33}^s \end{bmatrix} = \sum_i \mathbf{B}^{i,s} \cdot \delta \mathbf{q}_i \quad (10)$$

avec  $\delta \mathbf{q}_i^t = [\delta \mathbf{x}_i \ \delta \mathbf{r}_i]^t$ . La matrice  $\mathbf{B}$  ( $9 \times 6$ ) est donnée en annexe.

L'équation (10) exprime la dérivée du gradient de déformation. Il faut maintenant exprimer  $\partial \mathbf{L}^{i,s} / \partial \mathbf{q}$ , la dérivée du tenseur de déformation par rapport aux repères qui est également une matrice  $3 \times 3$ . Il est à noter que  $\partial \mathbf{L}^{i,s} / \partial \mathbf{q}$  doit être calculé pour chaque couple repère/point d'intégration en interaction (i.e.  $w_{is} > 0$ ). En reprenant l'équation (6) on remarque que son expression fait intervenir la dérivée du gradient de déformation. On réutilise donc les équations (6) et (10) pour exprimer  $\delta \mathbf{L}^{i,s} = \partial \mathbf{L}^{i,s} / \partial \mathbf{q}$  sous la forme d'un produit matrice-vecteur. En effet, pour  $\delta \mathbf{L}_{ij}$  avec  $i = 1$  et  $j = 2$ , on obtient :

$$\begin{aligned} \delta \mathbf{L}_{12} &= \frac{1}{2} ([\delta \mathbf{F}']^1 \cdot [\mathbf{F}']^2 + [\mathbf{F}']^1 \cdot [\delta \mathbf{F}']^2) \\ &= \frac{1}{2} ([\delta \mathbf{F}_{11} \ \delta \mathbf{F}_{21} \ \delta \mathbf{F}_{31}] \cdot [\mathbf{F}]^2 + [\mathbf{F}]^1 \cdot [\delta \mathbf{F}_{12} \ \delta \mathbf{F}_{22} \ \delta \mathbf{F}_{32}]) \end{aligned} \quad (11)$$

Or, d'après l'équation (10) :

$$[\delta \mathbf{F}_{11} \ \delta \mathbf{F}_{21} \ \delta \mathbf{F}_{31}] = [[\mathbf{B}]^1 \cdot \delta \mathbf{q} \ [\mathbf{B}]^2 \cdot \delta \mathbf{q} \ [\mathbf{B}]^3 \cdot \delta \mathbf{q}] \quad (12)$$

et

$$[\delta \mathbf{F}_{12} \ \delta \mathbf{F}_{22} \ \delta \mathbf{F}_{32}] = [[\mathbf{B}]^4 \cdot \delta \mathbf{q} \ [\mathbf{B}]^5 \cdot \delta \mathbf{q} \ [\mathbf{B}]^6 \cdot \delta \mathbf{q}] \quad (13)$$

En substituant les équations (12) et (13) à l'équation (11) puis en factorisant par  $\delta \mathbf{q}$  on obtient :

$$\begin{aligned} \delta \mathbf{L}_{12} &= \frac{1}{2} [\mathbf{F}_{12}[\mathbf{B}]^1 + \mathbf{F}_{22}[\mathbf{B}]^2 + \mathbf{F}_{32}[\mathbf{B}]^3, \\ &\quad + \mathbf{F}_{11}[\mathbf{B}]^5 + \mathbf{F}_{21}[\mathbf{B}]^5 + \mathbf{F}_{31}[\mathbf{B}]^6] \cdot \delta \mathbf{q} \end{aligned} \quad (14)$$

En écrivant ainsi chacune des six valeurs de  $\delta \mathbf{L}^{i,s}$  (matrice symétrique) avec la notation de Voigt, on obtient :

$$\begin{bmatrix} \delta \mathbf{L}_{11}^{i,s} \\ \delta \mathbf{L}_{22}^{i,s} \\ \vdots \\ 2\delta \mathbf{L}_{12}^{i,s} \end{bmatrix} = \sum_i \mathbf{M}^{i,s} \cdot \delta \mathbf{q}_i \quad (15)$$

où chaque ligne de  $\mathbf{M}^{i,s}$  est de la même forme que l'équation (14). Il est désormais possible d'évaluer les forces internes s'appliquant sur les repères. En effet, en substituant l'équation (15) à l'équation (5) puis en divisant par  $\delta \mathbf{q}$ , on obtient :

$$\mathbf{f} = \int_{\Omega} \mathbf{M}^t \{\mathbf{S}\} d\Omega \quad (16)$$

Cependant, l'équation (16) donne l'expression de la force dans le cas continu. Etant donné que cette intégrale ne peut être résolue analytiquement, hormis pour des cas simples, il est nécessaire de passer à une version discrète. L'intégrale de l'équation (16) doit être évaluée sur la totalité du domaine  $\Omega$  occupé par l'objet. Pour ce faire, sous la forme discrète, on utilise les points d'intégration qui échantillonnent l'objet. On obtient ainsi l'expression discrète de la force en un nœud :

$$\mathbf{f}_i = \sum_s [\mathbf{M}^{i,s}]^t \cdot \{\mathbf{S}_s\} v_s \quad (17)$$

où  $v_s$  est le volume associé au point d'intégration  $s$  et  $\{\mathbf{S}_s\}$  en notation de Voigt est calculé grâce au modèle de *Saint Venant - Kirchhoff*, qui est un modèle hyperélastique généralisant la *loi de Hooke* aux grandes déformations. Cependant, la méthode est également compatible avec des lois de comportement plus complexes. Le second tenseur de Piola-Kirchhoff est donné par :

$$\mathbf{S}_s = \lambda \text{tr}(\mathbf{L}_s) \mathbf{I} + 2\mu \mathbf{L}_s \quad (18)$$

avec  $\lambda$  et  $\mu$  qui sont les *coefficients de Lamé*

#### 4.3.2 Formulation implicite

La formulation précédente (équation 17) est utilisable avec un solveur explicite. En effet, en utilisant le principe fondamental de la dynamique (seconde loi de Newton) :

$$m\mathbf{a} = \sum \mathbf{f}^{int} + \mathbf{f}^{ext} \quad (19)$$

puis en effectuant successivement deux intégrations sur le temps, on peut obtenir les nouvelles positions et les nouvelles vitesses de chaque nœud au pas de temps suivant.

Cependant, bien qu'ils sont rapides, les solveurs explicites présentent certains désavantages, notamment l'instabilité. Les oscillations et l'introduction d'énergie supplémentaire conduisent la simulation à diverger [BW98]. Ces phénomènes peuvent être rencontrés lors d'une simulation comportant des modèles très détaillés ou des matériaux ayant une forte raideur. Une solution couramment employée pour répondre à ce problème consiste à utiliser des pas de temps très courts, avec pour effet d'augmenter les temps de calcul, dans la mesure où le temps simulé s'écoule plus lentement que le temps réel. Pour ces raisons, on utilise un solveur implicite qui permet l'utilisation de pas de temps plus longs tout en gardant une simulation stable – une propriété intéressante dans le cadre de la simulation d'objets déformables. Les solveurs implicites permettent de résoudre un problème non-linéaire de manière itérative selon la méthode de *Newton-Raphson* [FDD\*12]. Pour permettre au solveur de résoudre l'équation du mouvement, il est nécessaire de linéariser la force, c'est-à-dire de dériver l'expression de la force par rapport aux repères, ce qui équivaut à exprimer la variation de la force en fonction d'une variation infinitésimale de la contrainte. D'après [BLM00](§6.4.2) la dérivée des forces internes se décompose en deux parties :

$$\delta \mathbf{f}^{int} = \delta \mathbf{f}^{mat} + \delta \mathbf{f}^{geo} \quad (20)$$

où  $\delta \mathbf{f}^{mat}$  est la raideur tangentielle et  $\delta \mathbf{f}^{geo}$  est la raideur géométrique. Dans notre cas, nous nous limitons à l'utilisation de la première. En effet, les déformations subies par les objets permettent de négliger la raideur géométrique. L'expression de la raideur tangentielle est donnée par :

$$\delta \mathbf{f} = \int_{\Omega} \mathbf{M}^t \{ \delta \mathbf{S} \} d\Omega \quad (21)$$

avec :

$$\{ \delta \mathbf{S} \} = \mathbf{C} \{ \delta \mathbf{L} \} \quad (22)$$

où la matrice  $\mathbf{C}$ , donnée en annexe, est la matrice des coefficients élastiques associés à la loi de Hooke. En substituant l'équation (15) à l'équation (21) on obtient :

$$\delta \mathbf{f} = \int_{\Omega} \mathbf{M}^t \mathbf{C} \mathbf{M} \cdot \delta \mathbf{q} d\Omega \quad (23)$$

De même qu'avec l'équation (16), il est nécessaire de passer à une formulation discrète afin d'intégrer l'équation (23)

$$\delta \mathbf{f}_i = \sum_s \mathbf{M}^{i,s} \cdot \Delta \mathbf{S}^s v_s \quad (24)$$

avec :

$$\Delta \mathbf{S}^s = \sum_i \mathbf{C} \mathbf{M}^{i,s} \cdot \delta \mathbf{q}_i \quad (25)$$

qui exprime la variation de la contrainte en un point d'intégration en fonction d'une variation des DDL des différents nœuds voisins.

## 5 Résultats

### 5.1 Expérimentation

Notre méthode a été implémentée sous la forme d'un composant dans le framework SOFA [Sof] et testée avec différentes structures. L'implémentation CPU est monthreadée et les tests ont été effectués avec un processeur Intel Xeon W3520 cadencé à 2,67 GHz. Le premier test porte sur la simulation d'objets volumiques, ici en modélisant une poutre. La section de la poutre est modélisée avec 9 repères et 36 points d'intégration. Les points d'intégration sont placés entre les repères. Une des extrémités de la poutre est fixée. La figure (2) montre la modélisation de la poutre. La figure (3) montre des résultats de tests effectués avec trois poutres modélisées de façon identique mais avec un module de Young différent. Chacune de ces poutres n'est soumise qu'à son propre poids (masse : 0,3 kg). Le coefficient de Poisson de chacune de ces poutres est de 0,3.

Le second test concerne la simulation d'objets surfaciques. La structure de la figure (4) est modélisée avec 91 repères et 1024 points d'intégration coplanaires. Les bords gauche et droit de la surface sont fixés. La déformation visible sur cette figure résulte d'une simulation où une force constante de 300 N a été appliquée sur les repères, mis en évidence par les marqueurs oranges, perpendiculairement à la surface et orientée vers le bas. Le module de Young de la surface est de 25000 Pa et son coefficient de Poisson est de 0,3. Malgré la disposition coplanaire des repères et des points d'intégration, la simulation est stable.

Le troisième test concerne la simulation d'objets linéiques. La figure (5) montre une structure composée de



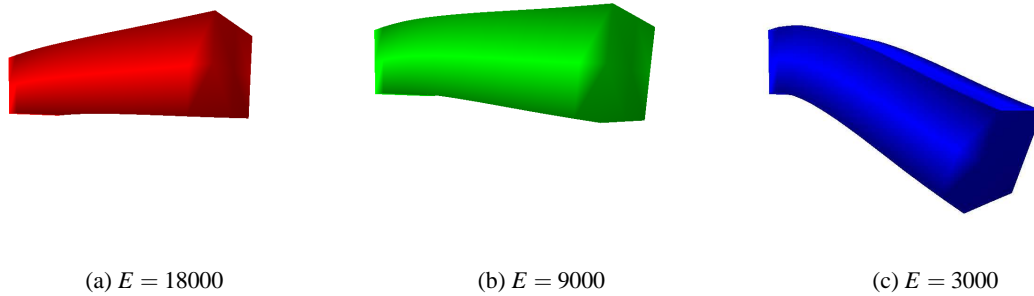


FIGURE 3 – Poutres simulées avec différents modules de Young ( $\nu = 0.3$ ).

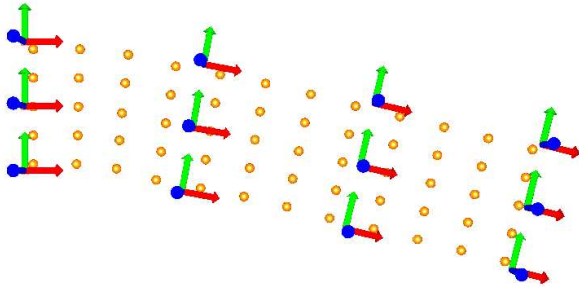


FIGURE 2 – Modélisation de la poutre avec notre méthode.

10 repères et de 50 points d'intégration disposés de façon colinéaire. Le repère de gauche est fixé en rotation et en translation. Initialement, tous les repères ont la même orientation (identique à celle du repère de gauche). La simulation consiste à effectuer une rotation d'un demi-tour sur le repère le plus à droite. On observe alors que tous les repères de la structure subissent une torsion.

## 5.2 Comparaison

Dans le but de vérifier la validité du modèle mécanique, différents tests sur des poutres ont été effectués. Le premier test consiste à simuler une poutre soumise uniquement à son propre poids de  $16 \times 0,5 \times 0,5$  mètres, pesant 31200 kg, dont le module de Young est de 200 GPa et le coefficient de Poisson de 0,3. On étudie ici le comportement de la poutre lorsqu'on fait varier le nombre de degrés

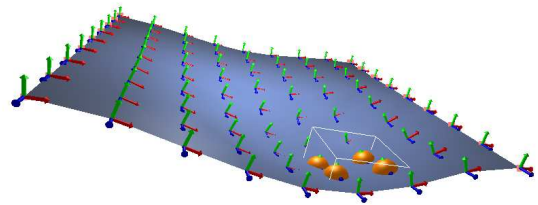


FIGURE 4 – Modélisation d'un objet surfacique avec notre méthode.

de liberté. Ces derniers sont disposés suivant une grille régulière. Chaque point d'intégration possède 8 repères voisins. La fonction de forme utilisée est une interpolation tri-linéaire. La déflexion maximale (à l'extrémité de la poutre) obtenue pour chaque simulation est comparée à la solution analytique de la poutre *cantilever*. Le solveur utilisé pour ce test est un gradient conjugué ayant

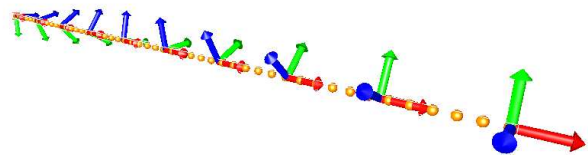


FIGURE 5 – Test de torsion sur une structure linéique.

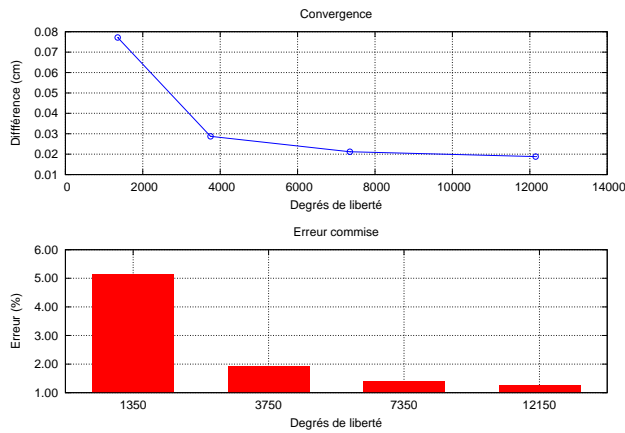


FIGURE 6 – Précision de la méthode sur le cas de la poutre (résolution sur la longueur fixée).

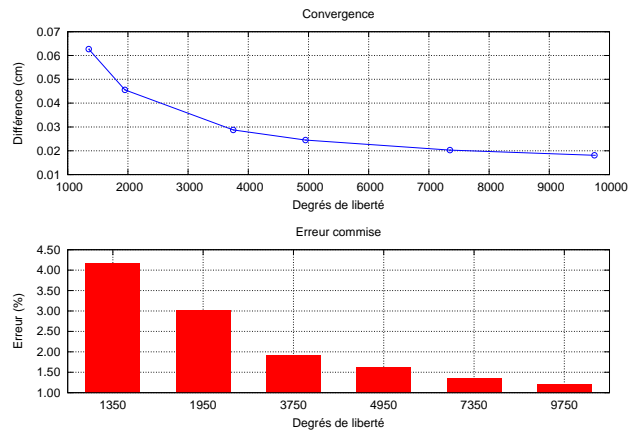


FIGURE 7 – Précision de la méthode sur le cas de la poutre (résolution de la section fixée).

une précision fixée à  $10^{-10}$ . Les résultats de ces tests sont présentés sur les figures (6) et (7). La figure (6) présente l'écart entre la solution analytique et la simulation variant en fonction du nombre de degrés de liberté utilisé, ainsi que le pourcentage d'erreur commise par rapport à la solution analytique. On fait varier le nombre de degrés de liberté de la section de la poutre. Le nombre repère sur la longueur est fixé à 25. De même, la figure (7) présente les mêmes données dans le cas où on fait varier le nombre de repères sur la longueur de la poutre, avec le nombre de repères de la section fixé à  $5 \times 5$  repères.

Le second test de vérification porte sur la comparaison de deux poutres simulées avec notre méthode, à deux poutres simulées avec la méthode des éléments finis (modèle corotationnel) proposée par SOFA. Ces poutres sont uniquement soumises à leur propre poids. Les degrés de liberté de chacune de ces 4 poutres sont disposés de manière identique, en grille régulière. La première poutre simulée avec notre méthode utilise 4 repères voisins par point d'intégration, et une interpolation linéaire comme fonction de forme. La seconde poutre simulée avec notre méthode utilise 8 repères voisins et une interpolation trilineaire. Les poutres simulées grâce à la méthode des éléments finis sont modélisées avec des tétraèdres pour la première, et des hexaèdres pour la seconde. Les dimensions de la poutre sont de  $16 \times 1 \times 1$  mètres, sa masse est de 5 kg, son module de Young est de 40 000 Pa, et

son coefficient de Poisson de 0,3. Le choix d'une poutre très déformable est volontaire et a pour but d'étudier la convergence de la méthode en cas de grandes déformations. Les résultats de ce test sont présentés en figure (8) où on voit la déflexion maximale de la poutre varier en fonction du nombre de degrés de liberté. On remarque sur ce graphique l'apparition d'une limite asymptotique montrant la convergence de chaque méthode. On y voit également que la méthode basée sur les repères nécessite moins de degrés de liberté pour fournir la même solution qu'une méthode par éléments finis utilisant des tétraèdres ou des hexaèdres. On peut également remarquer qu'un faible nombre de degrés de liberté introduit une rigidité artificielle de la structure. Les poutres présentées sur la figure (13), dont les propriétés mécaniques sont les mêmes que celles décrites précédemment, sont modélisées avec  $21 \times 3 \times 3$  repères disposés en grille régulière (à droite), et avec  $64 \times 4 \times 4$  hexaèdres à gauche. La différence entre les déflexions de ces deux poutres est inférieure à 2%.

Le troisième test est tiré du guide de *Validation des Progiciels de Calcul de Structures* (VPCS), publié par l'AFNOR [VPC90]. Le but de ce test est de valider le comportement d'une poutre en traction. La poutre mesure 6 mètres, sa section est circulaire et d'un rayon de 1 mètre. Le test, schématisé en figure (9), consiste à effectuer une traction dans l'axe du cylindre de façon à obtenir un déplacement de 3 mm et à mesurer le déplacement des

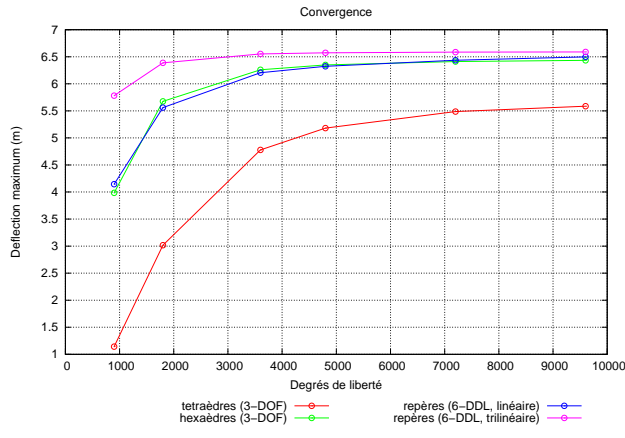


FIGURE 8 – Comparaison de la convergence de la méthode.

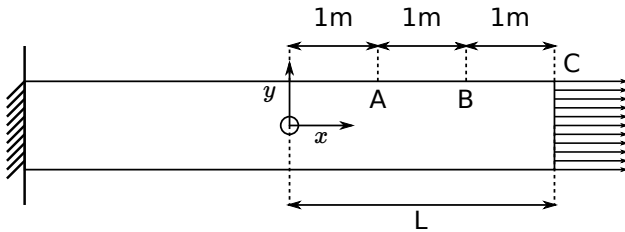


FIGURE 9 – Test en traction (VPCS).

points A, B, C sur l'axe  $y$ . Ce test est proposé pour un coefficient de Poisson de 0,3. Pour reproduire ce test, nous fixons une des faces du cylindre et effectuons la traction sur l'autre face. Le mouvement des repères de la face subissant la traction n'est contraint que sur l'axe  $x$ . La poutre est discrétisée par des repères disposés de façon régulière. Chaque point d'intégration possède 8 voisins et les fonctions de forme utilisées sont des interpolations trilineaires. Les résultats de ce test sont présentés dans la table (1). La différence entre les résultats expérimentaux et les résultats théoriques est inférieure à 0,7%.

### 5.3 Couplage

Le couplage est rendu possible par la capacité de notre modèle à partager ses repères avec d'autres modèles simulés avec des méthodes différentes basées sur l'utilisation de repères. Cette technique permet de modéliser

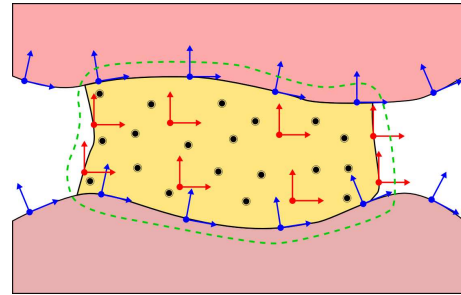


FIGURE 10 – Technique de couplage de deux modèles mécaniques grâce à notre méthode. Les repères des deux surfaces (bleu) à l'intérieur de la région en vert sont partagés avec les repères servant à modéliser les tissus connectifs (rouge).

Test	$u_y(A)$	$u_y(B)$	$u_y(C)$
Valeur VPCS	0,015	0,015	0,015
Valeur obtenue	0,0151	0,0151	0,0151

TABLE 1 – Résultats des tests en traction.

rapidement des tissus connectifs entre deux organes utilisant d'autres modèles : il suffit de définir la zone correspondant aux tissus connectifs en  $y$  plaçant des repères et des points d'intégration puis en définissant quels sont les repères des organes qui doivent être partagés avec les repères des tissus. La figure (10) présente ce principe. Les repères des surfaces (en bleu) et des tissus connectifs (en rouge) à l'intérieur de la région délimitée en vert sont considérés comme faisant partie d'un même objet mécanique. De plus grâce à l'utilisation de méthodes sans maillage, et à leur propriété de connectivité non-structurée, notre méthode permet de coupler simplement et efficacement des modèles de résolutions différentes dans la mesure où un point d'intégration influence les repères présents dans son rayon d'influence.

La figure (11) montre le résultat d'un test de couplage entre notre méthode et d'autres modèles mécaniques, en l'occurrence des éléments coque ou *shells* [CDC10], qui permettent de simuler des organes creux et/ou aux parois très fines. Ce cas test représente typiquement la situation où deux organes (en rose) sont reliés par des tissus connectifs (en jaune). Les tissus connectifs sont modélisés grâce à deux couches de repères. Les nœuds des

deux surfaces sont également des repères. La surface supérieure est fixée sur deux bords opposés et les repères de la surface inférieure sont totalement libres. Les centres des deux surfaces sont “attachés” de part et d’autre de l’objet jaune car ils partagent leurs repères avec lui. La masse des tissus connectifs est de 1 kg, son module de Young est de 6000 Pa et son coefficient de Poisson de 0,3. On peut observer que les conditions aux limites sont transmises d’un modèle à l’autre tandis que les déformations subies par les objets sont cohérentes.

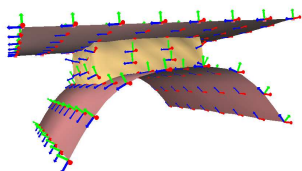


FIGURE 11 – Couplage multi-modèle. L’image montre la modélisation des organes et des tissus connectifs.

Des tests de couplage avec des modèles plus complexes ont été réalisés. La figure (12) montre le résultat d’un modèle d’utérus couplé à un modèle de vessie grâce à des tissus connectifs (en jaune). L’utérus et la vessie sont simulés grâce à des éléments coque. L’échantillonnage des tissus connectifs est réalisé grâce à deux *distribution de Poisson* basée sur la distance inter-particule. La première est utilisée comme position des repères et la seconde, avec une distance inter-particule plus faible, sert de position aux points d’intégration. Les tissus connectifs sont modélisés avec 22 repères et environ 800 points d’intégration.

## 5.4 Performances

### 5.4.1 Implémentation CPU

Le tableau 2 recense les performances obtenues lors de nos tests. Les paramètres pris en compte sont : le nombre de repères, le nombre de points d’intégration et le nombre de repères voisins par point d’intégration. On remarque clairement que les principaux facteurs affectant les performances sont le nombre de repères voisins pour chaque

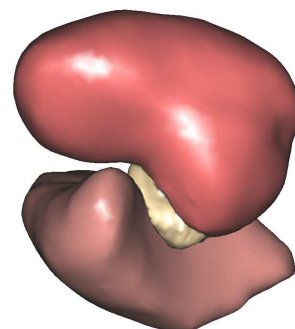


FIGURE 12 – Un modèle d’utérus (en haut) relié à un modèle vessie (en bas)

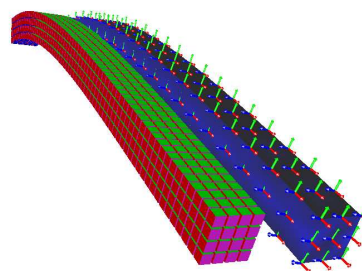


FIGURE 13 – La même poutre modélisée avec notre méthode (à gauche) et avec des hexaèdres (à droite).

point d'intégration ainsi que le nombre de points d'intégration.

Test	Repères	Pts. d'int.	Voisins	Temps (ms).	
Poutre	72	400	4	32,65	
			8	45,42	
			6	55,15	
			10	69,56	
	144	750	4	500	39,18
				864	43,42
				1470	114,3
				2340	160,53
				144	54,49
				288	56,64
				500	58,5
792	69,12				
Couplage	160	400	4	53,22	

TABLE 2 – Performances de l'implémentation CPU.

#### 5.4.2 Implémentation GPU

La méthode présentée a fait l'objet d'une implémentation sur GPU en utilisant CUDA. La principale difficulté posée par l'implémentation de cette méthode est due au fait que cette dernière se déroule en deux étapes. La première consiste à calculer le gradient de déformation et les tenseurs de déformation et de contrainte en chaque point d'intégration puis la dérivée du tenseur de Green-Lagrange pour tout couple repère/point d'intégration en interaction. La seconde étape consiste ensuite à calculer les forces internes sur chaque repère en accumulant la contrainte des points d'intégrations voisins. Cette deuxième étape est problématique dans le cas de l'implémentation GPU car le nombre de repères est moins élevé que le nombre de points d'intégration. De ce fait, allouer un thread à chaque repère pour le calcul des forces internes ne permet pas d'extraire suffisamment de parallélisme, c'est-à-dire qu'il est difficile d'exploiter au maximum les capacités des GPU qui sont capables de gérer l'exécution de milliers de threads en parallèle. De plus, une telle solution contraint à stocker au minimum le gradient de déformation et la dérivée du tenseur de Green-Lagrange en mémoire vidéo, ce qui est très coûteux en terme de performance. On ne peut également pas envisager que les points d'intégration accumulent eux-mêmes

les forces internes sur les repères en raison des problèmes d'écritures concurrentes (*ex* : deux points d'intégration tentant d'écrire simultanément dans le vecteur force d'un repère).

La solution adoptée pour palier à ce problème et augmenter le taux d'occupation de la puce graphique consiste à effectuer la quasi-totalité des calculs sur les points d'intégration. Lors du calcul des forces internes, un thread est alloué à chaque point d'intégration. Chaque point d'intégration produit des contributions partielles de force (une contribution pour chaque repère voisin) qui sont stockées dans un vecteur temporaire. Dans un second temps, on effectue une opération de réduction parallèle : un thread est alloué à chaque repère et accumule les contributions partielles de chacun de ses nœuds voisins grâce à une table d'indirection. Cette opération de réduction parallèle est très simple et donc relativement rapide ce qui permet de limiter l'effet de la faible occupation du GPU et de maintenir des performances plus hautes. La figure (14) schématise le fonctionnement de cet algorithme. Le calcul de la dérivée des forces est similaire. Chaque point d'intégration étant fortement indépendant des autres, le calcul des forces ne nécessite que très peu de points de synchronisation, ce qui limite le nombre de threads inactifs. Enfin, dans l'implémentation GPU, les matrices **B** et **M** des équations (10) et (15) ne sont pas construites explicitement et stockées en mémoire : chaque composante du vecteur force est calculée "à la volée". De cette façon, la consommation de mémoire vidéo et de registres est limitée, ce qui permet au GPU d'ordonnancer l'exécution d'un plus grand nombre de thread et donc d'augmenter les performances.

Les tests comparatifs ont été effectués avec un solveur de type *gradient conjugué* effectuant 60 itérations à chaque pas de temps. Chaque point d'intégration possède 4 repères voisins. La carte graphique utilisée pour ces tests est une Nvidia GTX 570 avec 1280 Mo de mémoire vidéo. Le tableau 3 présente le résultat de ces tests. Les données recueillies sont : le nombre de points d'intégration, la durée nécessaire pour un pas temps sur CPU et sur GPU, ainsi que le facteur d'accélération. On remarque que l'implémentation GPU apporte un gain de performance non négligeable avec un facteur d'accélération allant jusqu'à plus de 52.

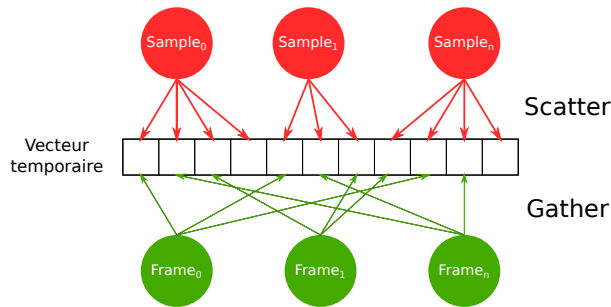


FIGURE 14 – Schéma de l'algorithme GPU

<i>Pts d'int.</i>	<i>Temps (CPU)</i>	<i>Temps (GPU)</i>	<i>Accel.</i>
360	34,84	14,85	2,35
800	78,42	15,25	5,14
2600	251,77	17,11	14,72
6070	595,77	23,39	25,47
12700	1228,8	32,29	38,07
16210	1561,8	39,29	39,75
19990	1990,06	43,12	46,15
26750	2675,47	53,09	50,39
34610	3468,54	66,1	52,47

TABLE 3 – Performances CPU/GPU (temps en ms).

## 6 Conclusion et travaux futurs

La méthode proposée permet de simuler des tissus connectifs aux topologies variées (linéiques, surfaciques ou volumiques). Elle offre la possibilité de coupler différentes structures anatomiques (organes, os, etc.), permettant l'imposition et la propagation des conditions aux limites de ces mêmes structures. Elle permet ainsi de simuler aisément les interactions entre les organes. De plus, la formulation basée sur des repères permet, à précision égale, d'utiliser moins de degrés de liberté, diminuant ainsi la taille des systèmes d'équations, et l'intégration temporelle implicite permet l'utilisation de pas de temps plus grands tout en conservant la stabilité de la simulation. Cependant, il existe certaines limitations. En effet, bien que les repères et les points d'intégrations nous permettent de simuler des structures linéiques et surfaciques, ces cas n'ont pas été validés numériquement. Il serait intéressant de comparer notre méthode à celle des éléments

poutre et des éléments coque. De même, pour ces cas particuliers, il pourrait également être intéressant de donner une formulation de la raideur géométrique. La gestion du couplage pourrait être améliorée. En effet, actuellement, les organes couplés et les tissus qui les unissent sont considérés comme un seul objet mécanique produisant un système à résoudre pouvant être de grande taille. On peut envisager l'utilisation de techniques basées sur la décomposition de domaine qui permettrait de simuler, dans un premier temps, le comportement de chaque organe séparément puis, dans un second temps, de calculer les interactions entre ces organes. Enfin, afin de reproduire plus fidèlement les caractéristiques des tissus connectifs, l'emploi de lois de comportement plus complexes, et la simulation de matériaux hétérogènes sont envisageable.

## Références

- [BIT09] BECKER M., IHMSEN M., TESCHNER M. : Corotated SPH for deformable solids. In *Proc. Eurographics Workshop on Natural Phenomena* (2009), pp. 27–34.
- [BLG94] BELYTSCHKO T., LU Y. Y., GU L. : Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*. Vol. 37, Num. 2 (1994), 229–256.
- [BLM00] BELYTSCHKO T., LIU W. K., MORAN B. : *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.
- [BW98] BARAFF D., WITKIN A. : Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 43–54.
- [CDC10] COMAS O., DURIEZ C., COTIN S. : Shell Model for Reconstruction and Real-Time Simulation of Thin Anatomical Structures. In *MICCAI (2)* (Chine, 2010), pp. 371–379.
- [CJA\*10] COURTECUISSIE H., JUNG H., ALLARD J., DURIEZ C., LEE D. Y., COTIN S. : GPU-based Real-Time Soft Tissue Deformation with Cutting and Haptic Feedback. *Progress in Biophysics and Molecular Biology*. Vol. 103, Num. 2-3 (2010), 159–168.
- [DC96] DESBRUN M., CANI M. : Smoothed particles : A new paradigm for animating highly deformable bodies. *Proceedings of the Eurographics workshop on Computer animation and simulation* (1996), 61–76.
- [DPP12] DERAM A., PAYAN Y., PROMAYON E. : Towards a generic framework for evaluation and comparison of soft tissue modeling. In *Proceedings of the 19th Medicine Meets Virtual Reality Conference* (Newport Beach, USA, 2012), pp. 116–122.
- [FDD\*12] FAURE F., DURIEZ C., DELINGETTE H., ALLARD J., GILLES B., MARCHESSEAU S., TALBOT H., COURTECUISSIE H., BOUSQUET G., PETERLIK I., COTIN S. : SOFA : A Multi-Model Framework for Interactive Physical Simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, Payan Y., (Ed.). Springer, juin 2012.
- [FM04] FRIES T.-P., MATTHIES H.-G. : *Classification and Overview of Meshfree Methods*. Tech. Rep. 2003-3, TU Brunswick, July 2004.
- [GBB09] GERSZEWSKI D., BHATTACHARYA H., BARGTEIL A. W. : A point-based method for animating elastoplastic solids. *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '09* (2009), 133.
- [GBFP11] GILLES B., BOUSQUET G., FAURE F., PAI D. : Frame-based Elastic Models. *ACM Transactions on Graphics*. Vol. 30, Num. 2 (avril 2011).
- [GLB\*06] GUO X., LI X., BAO Y., GU X., QIN H. : Meshless thin-shell simulation based on global conformal parameterization. *IEEE transactions on visualization and computer graphics*. Vol. 12, Num. 3 (2006), 375–85.
- [HMH07] HOSSEINI S. M., MANZARI M. T., HANNANI S. K. : A fully explicit three-step SPH algorithm for simulation of non-Newtonian fluid flow. *International Journal of Numerical Methods for Heat & Fluid Flow*. Vol. 17, Num. 7 (2007), 715–735.
- [HNR\*10] HOSTETTLER A., NICOLAU S., RÉMOND Y., MARESCAUX J., SOLER L. : A real-time predictive simulation of abdominal viscera positions during quiet free breathing. *Progress in Biophysics and Molecular Biology*. Vol. 103, Num. 2-3 (2010), 169 – 184. Special Issue on Biomechanical Modelling of Soft Tissue Motion.
- [HWJM10] HORTON A., WITTEK A., JOLDES G. R., MILLER K. : A meshless Total Lagrangian explicit dynamics algorithm for surgical simulation. *Integration The Vlsi Journal*, Num. March (2010), 977–998.

- [HWM07] HORTON A., WITTEK A., MILLER K. : Subject-specific biomechanical simulation of brain indentation using a meshless method. In *Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention - Volume Part I* (Berlin, Heidelberg, 2007), MICCAI'07, Springer-Verlag, pp. 541–548.
- [KMBG08] KAUFMANN P., MARTIN S., BOTSCH M., GROSS M. : Flexible simulation of deformable models using discontinuous Galerkin FEM. *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2008), 105—115.
- [KNP\*04] KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M. : Point Based Animation of Elastic, Plastic and Melting Objects. *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), 141–151.
- [LPC\*12] LUBOZ V., PROMAYON E., CHAGNON G., ALONSO D., FAVIER C., BARTHOD C., PAYAN Y. : Validation of a light aspiration device for in vivo soft Tissue characterization (LASTIC). In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, Payan Y., (Ed.). 2012, pp. 243–256.
- [MC11] MÜLLER M., CHENTANEZ N. : Solid simulation with oriented particles. *ACM Trans. Graph.* Vol. 30, Num. 4 (juillet 2011), 92 :1–92 :10.
- [MHC\*10] MARCHESSEAU S., HEIMANN T., CHATELIN S., WILLINGER R., DELINGETTE H. : Fast porous visco-hyperelastic soft tissue model for surgery simulation : Application to liver surgery. *Progress in Biophysics and Molecular Biology / Progress in Biophysics & Molecular Biology*. Vol. 103, Num. 2-3 (2010), 185–196.
- [MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M. : Meshless deformations based on shape matching. *ACM Trans. Graph.* Vol. 24, Num. 3 (juillet 2005), 471–478.
- [MKB\*10] MARTIN S., KAUFMANN P., BOTSCH M., GRINSPUN E., GROSS M. : Unified simulation of elastic rods, shells, and solids. *ACM Transactions on Graphics*. Vol. 29, Num. 4 (juillet 2010), 1.
- [NTV92] NAYROLES B., TOUZOT G., VILLON P. : Generalizing the finite element method : Diffuse approximation and diffuse elements. *Computational Mechanics*. Vol. 10, Num. 5 (1992), 307–318.
- [PPLT09] PAIVA A., PETRONETTO F., LEWINER T., TAVARES G. : Particle-based viscoplastic fluid/solid simulation. *Comput. Aided Des.* Vol. 41, Num. 4 (avril 2009), 306–314.
- [RRD\*11] RIVAUX G., RUBOD C., DEDET B., BRIEU M., GABRIEL B., DE LANDSHEERE L., DEVOS P., DELMAS V., COSSON M. : Biomechanical characterisation of uterine ligaments. implications for the pelvic floor. *Pelvi-Perineologie*. Vol. 6, Num. 2 (2011), 67–74.
- [Sof] Simulation open framework architecture. [www.sofa-framework.org](http://www.sofa-framework.org).
- [SSIF07] SIFAKIS E., SHINAR T., IRVING G., FEDKIWI R. : Hybrid simulation of deformable solids. *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), 81–90.
- [VPC90] *Guide de validation des progiciels de calcul de structures*. AFNOR, Paris-La Défense, 1990.



# Appendices

## A Calcul de la matrice B

$$\mathbf{B}^{i,s} = \begin{bmatrix} \frac{\partial w_{is}}{\partial x} \mathbf{I} & \left[ \frac{\partial w_{is}}{\partial x} [-\mathbf{R}_i \cdot \bar{\mathbf{l}}_s^\wedge] - [\mathbf{R}_i \cdot [\bar{\mathbf{R}}_i]^{,x \wedge}] w_{is} \right] \\ \frac{\partial w_{is}}{\partial y} \mathbf{I} & \left[ \frac{\partial w_{is}}{\partial y} [-\mathbf{R}_i \cdot \bar{\mathbf{l}}_s^\wedge] - [\mathbf{R}_i \cdot [\bar{\mathbf{R}}_i]^{,y \wedge}] w_{is} \right] \\ \frac{\partial w_{is}}{\partial z} \mathbf{I} & \left[ \frac{\partial w_{is}}{\partial z} [-\mathbf{R}_i \cdot \bar{\mathbf{l}}_s^\wedge] - [\mathbf{R}_i \cdot [\bar{\mathbf{R}}_i]^{,z \wedge}] w_{is} \right] \end{bmatrix}$$

avec  $w_{is} = w_i(\bar{\mathbf{p}}_s)$  et  $\mathbf{I}$  la matrice identité.

## B Matrice de Hooke

$$C = \frac{E}{(1+\nu)(2-\nu)} \times \begin{bmatrix} 1-\nu & \nu & \nu & & & \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & \mathbf{0} & & \\ & & & & \frac{(1-2\nu)}{2} & \\ & & & & & \frac{(1-2\nu)}{2} \\ & & & & & & \frac{(1-2\nu)}{2} \end{bmatrix}$$

où  $E$  et  $\nu$  sont respectivement le *module de Young* et le *coefficient de Poisson*.