HAL
open science

Modèle déformable par éléments finis - Elasticité linéaire et grands déplacements

Matthieu Nesme

To cite this version:

Matthieu Nesme. Modèle déformable par éléments finis - Elasticité linéaire et grands déplacements. Synthèse d'image et réalité virtuelle [cs.GR]. 2004. inria-00598416

HAL Id: inria-00598416

https://inria.hal.science/inria-00598416

Submitted on 6 Jun 2011

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.

Année Universitaire 2003 / 2004

Master "Mathématiques, Informatique"
Spécialité "Imagerie, Vision, Robotique"

MÉMOIRE DE RECHERCHE

MODÈLE DÉFORMABLE PAR ÉLÉMENTS FINIS
ÉLASTICITÉ LINÉAIRE et GRANDS DÉPLACEMENTS

pour application à la simulation chirurgicale

ENSIMAG - Institut National Polytechnique de Grenoble

Projet GMCAO - Laboratoire TIMC

Projet EVASION - Laboratoire GRAVIR

Matthieu Chabanas tuteurs François Faure

Jury

Augustin Lux
Gilles Debunne
François Faure
Matthieu Chabanas
Pierre Poulin

par Matthieu Nesme

Table des matières

1 Introduction **3**
1.1 Contexte . . . 3
1.2 Motivations . . . 4
1.3 Contributions . . . 5

2 Existant sur les modèles déformables **6**
2.1 Aspects théoriques . . . 6
2.1.1 Cinématique des milieux continus . . . 6
2.1.2 Comportement viscoélastique . . . 8
2.1.3 Globalité . . . 10
2.1.4 Particule . . . 11
2.2 Systèmes masses-ressorts . . . 12
2.2.1 Principe . . . 12
2.2.2 Lois classiques . . . 13
2.2.3 Analyse . . . 13
2.2.4 Conclusion . . . 14
2.3 Méthode des éléments finis . . . 14
2.3.1 Principe . . . 14
2.3.2 Maillage . . . 15
2.3.3 Fonctions d'interpolation . . . 15
2.3.4 Assemblage . . . 16
2.3.5 MEF explicites . . . 16
2.3.6 Conclusion . . . 17
2.4 Diversité de la MEF . . . 17
2.4.1 Méthodes statiques avec tenseur de déformations linéaire . . . 17
2.4.2 Précalculs . . . 18
2.4.3 Méthode des éléments de frontières . . . 19
2.4.4 Systèmes masses-tenseurs . . . 20
2.4.5 Linéarité géométrique et grands déplacements . . . 21
2.4.6 Autres améliorations applicables aux MEF . . . 23
2.4.7 Conclusion . . . 26
2.5 Conclusion . . . 26

3 Ma contribution **28**
3.1 Contexte . . . 28
3.1.1 AnimAL . . . 28
3.1.2 Intégration dynamique . . . 28
3.1.3 Eléments . . . 29
3.1.4 Particules . . . 29
3.1.5 Calcul de la rotation . . . 30
3.1.6 Matrice de rigidité . . . 35
3.1.7 Conclusion . . . 36

4 Résultats **37**
4.1 Précision . . . 37
4.1.1 Premiers résultats . . . 37
4.1.2 Comparaison avec d'autres solutions numériques . . . 39
4.1.3 Comparaisons avec des solutions analytiques . . . 41
4.1.4 Comparaison avec le truth cube . . . 43
4.1.5 Comparaison avec des données réelles de patients dans le cadre de planification de chirurgie maxillo-faciale . . . 45
4.2 Vitesse . . . 47
4.3 Conclusion . . . 49

5 Conclusion et Perspectives **50**
5.1 Conclusion . . . 50
5.2 Perspectives . . . 50
5.2.1 Propagation . . . 50
5.2.2 Matériaux . . . 51
5.2.3 Eléments . . . 51
5.2.4 Tests . . . 51
5.2.5 Compromis temps de calcul / précision . . . 51

A Fonctions d'interpolations **52**
A.1 Triangle linéaire . . . 52
A.2 Tetraèdre linéaire . . . 52
A.3 Elément quelconque . . . 53

B Divers **54**
B.1 Volume du tetraèdre . . . 54
B.2 Maillage d'un hexaèdre . . . 54

C Exemple de calcul **55**
C.1 Matrice de rigidité (en petits déplacements) . . . 55
C.2 Grands déplacements : version géométrique . . . 56
C.3 Grands déplacements : version analytique . . . 57

Chapitre 1

Introduction

1.1 Contexte

Mon stage de DEA se déroule en co-tutelle, dans le laboratoire GRAVIR, au sein de l'équipe EVASION dans les locaux de l'INRIA à Monbonnot, sous la tutelle de François Faure, et, dans l'équipe GMCAO du laboratoire TIMC au CHU de Grenoble, sous la tutelle de Matthieu Chabanas.

Le laboratoire GRAVIR (GRAphisme Vision et Robotique) travaille sur l'infographie, la vision et la robotique. Le projet EVASION (Environnements Virtuels pour l'Animation et la Synthèse d'Images d'Objets Naturels - anciennement iMAGIS) s'intéresse plus particulièrement à l'informatique graphique et la synthèse d'images. Ses principaux thèmes de recherche portent sur la simulation graphique de scènes naturelles végétales, animales et minérales. Les domaines d'application vont des simulateurs médicaux aux jeux vidéo, en passant par la réalité virtuelle et la visualisation scientifique.

L'élaboration de simulateurs sous contraintes de calculs temps réel et avec des objectif de « réalisme » comportemental (texture, retour d'effort) est une problématique centrale pour les équipes de recherche qui travaillent dans le monde de l'informatique graphique. C'est ainsi que l'équipe EVASION a acquis un savoir-faire important en matière de simulateurs dynamiques interactifs. Cette communauté cherche maintenant à étendre ses outils de modélisation vers des méthodes mathématiques plus proches de la théorie qui décrit la mécanique des milieux continus (du « réaliste » vers le « précis »), et ceci avec un souci nouveau de confrontation des modèles avec des données issues du monde réel.

Le laboratoire TIMC (Techniques de l'Imagerie, de la Modélisation et de la Cognition) représente l'interface des applications biologiques et médicales et des mathématiques avec l'informatique. L'objectif du projet GMCAO (Gestes Médicaux Chirurgicaux Assistés par Ordinateur) est clinique : il s'agit d'assister le médecin et le chirurgien dans la réalisation de gestes diagnostiques ou thérapeutiques les plus précis et les moins invasifs possibles.

La surgétique est un domaine transdisciplinaire visant à introduire les derniers outils des technologies informatique et robotique dans la pratique médico-chirurgicale. Dans ce cadre, la planification d'opérations est actuellement en pleine expansion : des simulateurs physiques de tissus biologiques permettent aujourd'hui de prévoir les conséquences de certaines opérations. L'équipe GMCAO a accumulé une expérience importante dans ce domaine. Les praticiens peuvent ainsi planifier, tester et optimiser des stratégies opératoires avant certaines interventions. Ces simulateurs mettent en oeuvre des modèles « précis », mais complexes. La plupart reposent sur des méthodes qui requièrent de nombreux calculs pour déterminer une configuration d'équilibre en réponse aux contraintes appliquées aux systèmes. De ce fait, ces modèles ne peuvent pas être utilisés en mode interactif, quand les conséquences d'un geste doivent être immédiatement perceptibles.

De nos jours, la demande pour des simulateurs interactifs grandit considérablement, pour deux raisons. Premièrement, les cliniciens souhaiteraient disposer de simulateurs rapides pour effectuer des « replanifications » en cours d'opération, dans les cas fréquents où la mise en oeuvre des gestes prévus n'est pas possible (modification de la configuration anatomique, problème technique, etc.), ou en chirurgie en urgence. Deuxièmement, des simulateurs temps-réel permettraient de faciliter, de compléter, voire d'améliorer la formation aux gestes chirurgicaux.

Les équipes EVASION et GMCAO se trouvent donc aujourd'hui face à un même verrou scientifique : être capable de simuler avec précision et réalisme les déformations de structures complexes, tout en conservant une rapidité compatible avec une utilisation interactive. C'est pourquoi ces deux équipes s'allient afin de coupler leur expertise ; EVASION apporte son savoir-faire en terme de rapidité et GMCAO apporte l'apesct précision.

1.2 Motivations

Les applications de la modélisation précise et rapide d'objets déformables sont multiples. En synthèse d'image, elles restent classiquement les films d'animation, mais le plus grand intérêt réside dans les simulateurs intéractifs, tels que les jeux vidéos, où le gain se situerait à la fois au niveau de la précision du mouvement, mais aussi dans la possibilité d'utiliser des objets dont les propriétés sont réelles, utilisant des matériaux connus. Le réglage des paramètres serait alors plus facile.

En simulation chirurgicale, un tel modèle permetterait d'accélérer considerablement les planifications pré-opératoires, les rendant moins contraignantes en utilisation réelle. Ces simulations permettent de planifier avec précision une intervention donnée à l'aide de modèles spécifiquement adaptés à chaque patient. S'il est plus rapide d'effectuer des simulations, on peux imaginer pouvoir en faire plus, et ainsi se rapprocher au mieux de la meilleure. Actuellement, il faut compter plusieurs dizaines de minutes pour aboutir à une prévision d'opération.

Il existe d'autres types de planifications chirurgicales : les planifications per-opératoires, celles qui seraient effectuées au cours de l'intervention. Généralement, avant l'opération, la planification donne juste une idée de la direction à suivre, mais il serait souvent nécessaire de réeffectuer une planification juste avant, voire pendant l'opération. En effet, la plupart des organes auront changé de position et de forme depuis la planification pré-opératoire, ou un imprévu peut modifier les plans, il faut donc être capable de planifier rapidement. Il faut aussi considérer les cas où le patient arrive en urgence, la planification doit pouvoir se faire réactivement.

On pourrait même imaginer des simulations per-opératoires. Elles guideraient le chirurgien pendant qu'il opère, lorsqu'il n'est pas possible de bien visualiser son travail avec les imageries traditionnelles. On pourrait reproduire tous ses gestes dans un simulateur qui donnerait l'aspect de la partie invisible. A l'heure actuelle, de telles simulations sont impossibles car les modèles sont soit trop lents, soit trop imprécis.

Un autre type de simulation chirurgicale, est le simulateur temps réel, permettant, entre autre l'entraînement à des gestes chirurgicaux. Dans le cas de laparoscopie, l'intérêt d'un tel simulateur est évident. Puisque le chirurgien n'a pas de contact direct avec le patient, il permetterait de reproduire fidélement une opération (cf figure 1.1). A long terme, il pourrait limiter l'apprentissage sur animal ou sur pièces anatomiques ; il permetterait de pratiquer des entraînements intensifs sans être limités par les problèmes éthiques liés à l'utilisation de cadavres ou d'animaux. On peut aussi imaginer la modélisation de pathologies rares, pour lesquelles les chirurgiens ne peuvent pas s'entraîner habituellement.

FIG. 1.1 – Une laparoscopie et un simulateur laparoscopique (epidaure-INRIA)

Actuellement, il existe déjà ce dernier type de simulateurs, mais la contrainte forte de rapidité imposée par le temps réel, oblige l'utilisation de modèles très peu précis. Ces simulateurs ne sont pour l'instant pas réalistes fonctionnellement, en effet ils n'utilisent pour la plupart que des modèles géométriques de structures anatomiques, sans tenir compte de leur réalité physique.

1.3 Contributions

Ce document se divise en cinq chapitres. Le présent introduit le contexte et les motivations de notre travail.

Le chapitre suivant expose rapidement les bases physiques nécessaires à la bonne compréhension de la problématique, en introduisant la mécanique des milieux continus, la théorie de l'élasticité et les lois des systèmes dynamiques. Puis il se poursuit par une présentation des modèles déformables existants, en détaillant particulièrement les approches basées sur la méthode des éléments finis.

Les deux chapitres suivants sont consacrés aux travaux que nous avons menés sur les modèles déformables. Nous commençons par présenter une amélioration de méthodes existantes et ses apports en terme de réalité physique et de vitesse de calcul. Puis un chapitre est consacré à tester cette nouvelle approche, tant au point de vue de la précision que de sa rapidité. Pour tester si notre modèle est physiquement réaliste, nous l'avons comparé avec d'autres méthodes numériques issues de l'ingénierie mécanique reconnues pour être précises. Nous l'avons aussi confronté avec des solutions analytiques et avec des données réelles, dans le cadre d'expériences sur un cube de silicone, et dans le cadre de planification de chirurgie maxillo-faciale.

Enfin, le dernier chapitre conclut ces travaux et propose un certain nombre d'orientations pour les développements futurs sur le sujet des simulations d'objets déformables.

Chapitre 2

Existant sur les modèles déformables

On s'intéresse à des corps viscoélastiques qui tendent à reprendre une forme de référence dite « forme au repos ». Pour simuler de tels corps, nous devons : modéliser les déformations (2.1.1), en déduire des actions mécaniques (2.1.2) et traiter les aspects temporels pour des applications dynamiques (2.1.4).

Après quelques rappels théoriques, nous passerons en revue les principaux modèles déformables basés sur la physique, utilisés aussi bien en synthèse d'images, que pour des simulateurs chirurgicaux.

2.1 Aspects théoriques

Pour bien apprécier la modélisation physique, il est important d'avoir quelques pré-requis. Nous allons ici, passer en revue les notions qui semblent importantes et nécessaires (informations issues des cours [5-GAR02, 6-OHA00]).

2.1.1 Cinématique des milieux continus

Nous savons que la matière est discontinue à l'échelle moléculaire, mais à notre échelle macroscopique, elle se présente comme un milieu continu, c'est à dire dont les propriétés physiques varient continuement d'un point à un autre.

On s'intéresse à trouver les forces qui s'exercent sur un objet qui a été déformé, et réciproquement les déformations résultantes aux forces exercées sur l'objet.

2.1.1.1 Déplacement

Sous l'action de forces, un corps non rigide se déforme. On peut définir la position initiale de chaque point à l'intérieur de ce corps par son vecteur position X_0. Lorsque le corps est déformé, chacun de ses points se retrouve à une position différente X. Le déplacement u correspond au vecteur entre l'état déformé et l'état au repos d'une particule x : $u(x) = X - X_0 = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$.

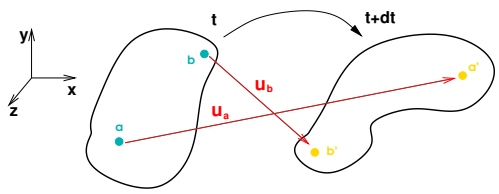
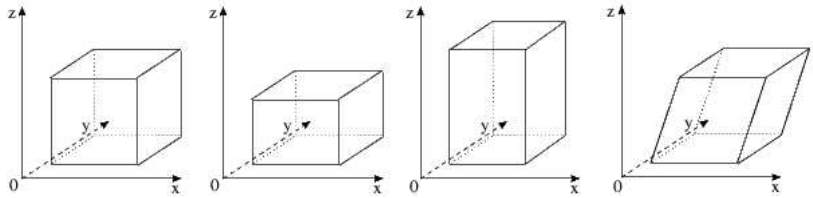


FIG. 2.1 – Déplacements u_a et u_b des particules a et b

2.1.1.2 Déformation

Les déformations s'obtiennent à partir des déplacements grâce à un tenseur de déformations qui fait intervenir la géometrie du milieu. Les déformations représentent les dilatations (élongation, compression) et les cisaillements que subit un objet (figure 2.2).



(a) au repos (b) compression suivant l'axe z (c) étirement suivant l'axe z (d) cisaillement suivant le plan x,z

FIG. 2.2 – Les différentes déformations d'un héxaèdre

Pour être insensible aux translations, on calcule les déformations en considèrant le gradient du déplacement $grad\,\overrightarrow{u}$ car $\overrightarrow{u}(\overrightarrow{M}+\overrightarrow{dM}) = \overrightarrow{u}(\overrightarrow{M}) + grad\,\overrightarrow{u}\,\overrightarrow{dM}$ avec

$$grad\,u = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} & \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} & \frac{\partial u_z}{\partial y} & \frac{\partial u_z}{\partial z} \end{bmatrix} = \underbrace{\frac{1}{2}(grad\,u + grad^T\,u)}_{deformation\,pure} + \underbrace{\frac{1}{2}(grad\,u - grad^T\,u)}_{petite\,rotation}$$

Le tenseur de déformations de Cauchy contient l'état de déformation de l'objet en une particule, il se note

$$[\varepsilon] = \frac{1}{2}(grad\,u + grad^T\,u)$$

Le problème de ce tenseur est qu'il n'est pas invariant en rotations, pour cela le tenseur de déformation $[\varepsilon]$ de Green-Lagrange ajoute des termes quadratiques :

$$[\varepsilon] = \frac{1}{2}([grad(u)] + [grad(u)]^T + [grad(u)]^T[grad(u)])$$

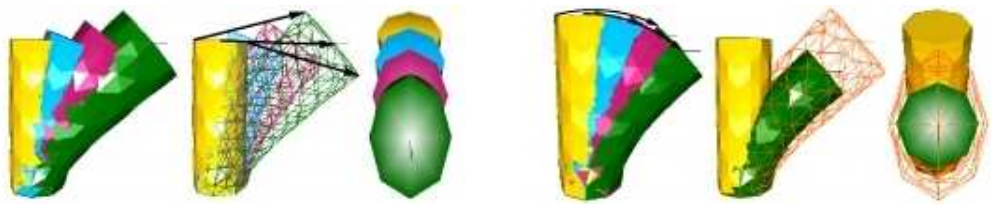
Le gros avantage de Green-Lagrange est d'utiliser à la fois le repère global, mais aussi des repères locaux, garantissants plus de liberté dans les mouvements et plus de précision car il est invariant aux rotations locales, donc parfaitement adapté aux objets qui se déforment beaucoup.

Linéarité géométrique Il est possible de ne considérer que de faibles déformations et par conséquent les termes du second ordre du tenseur de déformations, qui deviennent très petits, sont négligés. D'où le nom linéaire qui indique que la relation entre déplacements et déformations est du premier ordre. Le tenseur des déformations utilisé est celui de Cauchy.

Inconvénients Les résultats restent proches de ceux calculés à partir d'un tenseur de déformations non-linéaire, tant que les déplacements sont « petits ». Lorsque les déplacements augmentent, le terme quadratique grossi et on s'éloigne du vrai comportement, on devient alors de plus en plus imprécis. On est donc obligé de travailler seulement sur des objets qui se déplacent peu.

Un avantage démontré du modèle non-linéaire est d'éliminer certains artefacts inhérents au modèle linéaire pour de forts déplacements, notamment lorsqu'une partie du système subit une rotation. La méthode habituelle pour contourner ce problème est de calculer les déformations par le tenseur de Green-Lagrange non-linéaire, qui est invariant en rotation. L'inconvénient est qu'il comporte un terme quadratique et donc qu'on sort du contexte de la linéarité géométrique, avec pour problème la résolution d'un système non linéaire.

Dans le cas de grands déplacements, le modèle linéaire peut conduire à une distorsion, car les particules ne peuvent se déplacer que le long de trajectoires droites, tandis que le modèle non-linéaire autorise des déplacements en courbe et conserve la forme de l'objet (cf figure 2.3). Visuellement, le modèle linéaire semble faire gonfler l'objet.



(a) Le modèle linéaire conduit à une distorsion de la forme de la base

(b) Le modèle non-linéaire conserve la forme de la base

FIG. 2.3 – Déformation d'un cylindre par application d'une force tangentielle à l'une des bases, l'autre base étant fixée.

2.1.2 Comportement viscoélastique

Les forces s'obtiennent à partir des contraintes. Les contraintes s'obtiennent à partir des déformations grâce à une loi de comportement qui fait intervenir les propriétés du matériau. La loi de comportement traduit les caractéristiques du matériau.

Une contrainte correspond à une force par unité de surface, elle n'a un sens que par rapport à une facette que l'on oriente par sa normale. Grâce au tenseur de contraintes $[\sigma]$, qui reflète l'état de contrainte de l'objet en une particule, il est possible d'extraire des forces f en fonction de la géométrie alentour à la particule, en particulier en fonction de l'aire dS (petite) et de la normale à la surface d'application $\overrightarrow{n}$.

$$f = [\sigma]\,\overrightarrow{n}\,dS$$

Modèle élastique Les objets déformables sont classiquement représentés par une loi élastique, car celle-ci correspond à un grand nombre de matériaux réels pour de faibles déformations. Elle s'inscrit dans les lois plastiques qui permettent de décrire le comportement de tout matériau solide déformable. Une hypothèse simplificatrice essentielle des milieux élastiques est la non-dissipation intrinsèque.

Élastique indique que l'état des déformations ne dépend que de l'état présent des contraintes, ce qui revient à supposer que les forces internes au matériau n'agissent que par contact immédiat et que leur rayon d'action est proche de zéro. Un matériau élastique qui a été déformé sous l'action de certaines forces regagne son état initial une fois que ces forces ont disparues, et il peut restituer toute l'énergie qu'il a absorbé, sa forme ne dépend pas de l'historique de ses déformations (cf

figure 2.4). Une particule qui est déplacée reviendra à sa position initiale en passant par le même « chemin ».

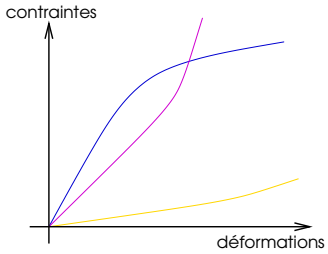


FIG. 2.4 – Des lois de comportement élastique

Linéarité mécanique Il est possible de simplifier l'expression de l'élasticité par l'élasticité linéaire, nous allons aborder quelques éléments de cette théorie.

Dans le cas d'un matériau linéaire, on a à faire à la loi de comportement la plus simple, les forces deviennent proportionnelles aux déformations. Bien sûr les vrais matériaux ne sont pas exactement linéaires. Il est possible de voir l'erreur engendrée par une telle loi : si on étire ou que l'on compresse un matériau à 100%, la même force (au signe près) s'executera ; or on sait que si on pouvait comprimer un objet en un point, les forces exercées seraient infinies.

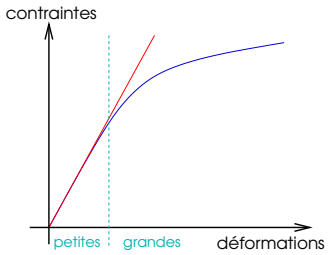


FIG. 2.5 – Approximation linéaire (rouge) d'une loi de comportement élastique sublinéaire (bleue)

On voit apparaitre sur le schéma 2.5, la notion de grandes et de petites déformations. On voit que l'approximation par une droite d'un véritable comportement reste à peu près juste tant que les déformations restent petites. En général, on considère qu'un matériau élastique n'est plus linéaire et qu'on a des grandes déformations au delà de 20%.

Loi de Hooke On introduit à ce niveau l'hypothèse que le matériau est isotrope, ce qui signifie que ses propriétés sont les mêmes dans toutes les directions. Cette condition introduit un certain nombre de liens entre les paramètres, car les relations doivent être les mêmes indépendamment de l'orientation du système de coordonnées. En considérant toutes les symétries possibles, il est alors possible de démontrer que les 36 paramètres caractérisant un matériau élastique se réduisent à seulement deux paramètres indépendants. Ces deux paramètres sont appelés coefficients de Lamé et sont notés λ et μ. La loi de comportement s'écrit alors :

$$[\sigma] = \lambda tr([\varepsilon])I + 2\mu\,[\varepsilon]$$

avec I la matrice identité et $tr(X)$ la trace de X.

La matrice de rigidité du matériau D, qui reflète la loi de comportement et fait passer des

déformations aux contraintes, est du type :

$$D = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}$$

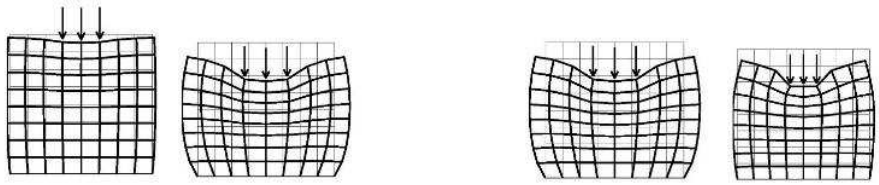
$$\{\sigma\} = D\{\varepsilon\}$$

En écriture matricielle, les déformations$\{\varepsilon(x)\}$ et les contraintes $\{\sigma\}$ s'écrivent alors sous la forme de vecteurs. $\{\varepsilon(x)\} = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{pmatrix}$ avec ε_i la compression/élongation suivant l'axe i et γ_{ij} le cisaillement angulaire entre les axes i et j, pour la particule x.

Une autre écriture des paramètres définissant un matériau introduit deux coefficients plus souvent utilisés en mécanique car plus faciles à interpréter expérimentalement, ce sont le module d'Young E correspondant à la rigidité du matériau, exprimé en Pa, et le coefficient de Poisson ν correspond à sa compressibilité, sans unité (cf figure 2.6).

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \qquad \mu = \frac{E}{2(1+\nu)}$$
$$0 < \nu < 0.5$$

$$E = \frac{(3\lambda+2\mu)\mu}{\lambda+\mu} \qquad \nu = \frac{\lambda}{2(\lambda+\mu)}$$



(a) Plusieurs modules d'Young, plusieurs rigidités

(b) Plusieurs coefficients de Poisson, plusieurs compressibilités

FIG. 2.6 – Influences du module d'Young et du coefficient de Poisson

Le système est inversible pour déduire les déformations des contraintes.

$$[\varepsilon] = \frac{1+\nu}{E}[\sigma] - \frac{\nu}{E}tr([\sigma])I$$

2.1.3 Globalité

On peut raisonner dans les deux sens, c'est à dire calculer des forces à partir de déplacements, ou, calculer des déplacements à partir de forces.

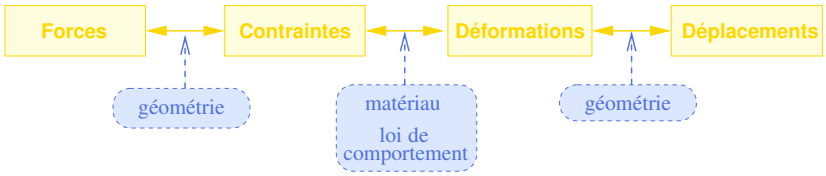


FIG. 2.7 – Relation déplacements/déformations/contraintes/forces en fonction de la géométrie et du matériau

Il est important de noter que l'on peut être linéaire (ou non) à deux endroits : au passage des déplacements aux déformations au niveau du tenseur de déformations, et aussi au passage des déformations au contraintes au niveau de la loi de comportement.

Dans le cas où les deux étapes sont linéaires, les forces sont proportionnelles aux déplacements. On peut ansi résoudre des problèmes de statique par la résolution d'un système matriciel linéaire du type :

$$f = Ku$$

où K est appelée matrice de rigidité.

2.1.4 Particule

Une particule est un élément de matière de forme quelconque autour d'un point. Elle est assez petite pour être considérée mathématiquement comme un infiniment petit, mais elle est assez grande pour contenir un très grand nombre de molécules, elle correspond à une masse ponctuelle.

2.1.4.1 Lois élémentaires

loi d'inertie (première loi de Newton) Elle permet de calculer l'accélération $\ddot{q}_i$ d'une particule i de masse m_i en fonction des forces qui lui sont appliquées :

$$\ddot{q}_i = \frac{\sum_{j \neq i} f_{j \to i}}{m_i}$$

loi d'interaction (seconde loi de Newton) Les forces sont réciproques au signe près :

$$f_{j \to i} = -f_{i \to j}$$

2.1.4.2 Pesanteur

Dans une scène terrestre la pesanteur g est uniforme et applique un poids proportionnel à la masse :

$$f_{\to i} = m_i g$$

2.1.4.3 Viscosité du milieu

La viscosité d'un milieu s'oppose au mouvement des corps qui s'y déplacent. Elle se modélise par un coefficient positif ν qui agit dans la direction opposée à la vitesse $\dot{q}_i$:

$$f_{\to i} = -\nu \dot{q}_i$$

2.1.4.4 Lois temporelles

Dans le cas de simulations interactives, on ne cherche pas explicitement un état d'équilibre, mais plutôt des déformations au cours du temps.

Si on veut visualiser et interagir à tout moment sur le système, il faut intégrer le temps et ainsi gérer le problème en dynamique. Pour cela, on doit résoudre une équation différentielle du second ordre de la forme :

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{f\} \tag{2.1}$$

où $[M]$ est lié à la masse, $[C]$ à l'amortissement et $[K]$ à la rigidité, $\{u\}$ correspond aux déplacements et $\{f\}$ aux forces ; le temps est traité comme un paramètre. $[M]$, $[C]$ et $[K]$ s'obtiennent en regroupant les équations en chaque sommet ; elles peuvent dépendre de la géométrie et donc devoir être recalculées à chaque déformation.

On résout donc un problème à chaque pas de temps, en partant d'un instant initial où les dérivées premières et secondes sont supposées connues : ce sont les conditions initiales.

Le schéma d'intégration est l'algorithme qui permet de calculer l'état à l'instant suivant à partir de l'instant courant en effectuant une intégration du temps, comme exprimé par la formule :

$$q(t+h) = q(t) + \int_t^{t+h} \dot{q}(q,t)dt$$

où $q(t)$ est la position d'une particule au temps t, $\dot{q}(q,t)$ est sa vitesse et h est le pas de temps.

Il existe quantité de schémas d'intégration, qui se distinguent par leur ordre de précision, leur stabilité et leur coût en terme de temps de calcul. Tous sont approximatifs car on ne sait pas calculer l'intégrale de la formule dans le cas général. On peut distinguer deux grandes classes de méthodes :

- Les méthodes explicites : les dérivées premières et secondes déduites de l'état courant sont utilisées pour calculer la valeur suivante du champ inconnu. La stabilité de la solution demande généralement des pas de temps très petits.
  Par exemple, le schéma d'Euler explicite fait l'approximation que la dérivée tout au long du pas de temps est égale à sa valeur au début du pas (la vitesse ne varie pas au cours d'un pas de temps) :
  $$q(t+h) \approx q(t) + h\dot{q}(t)$$
- Les méthodes implicites : les dérivées premières et secondes en fin de pas sont utilisées pour calculer la valeur suivante. La stabilité est meilleure, mais la résolution de chaque pas de temps exige un calcul itératif, avec les problèmes associés à tout calcul itératif : les paramètres algorithmiques et le choix de critères d'arrêt.
  (La méthode d'Euler implicite que nous utiliserons est détaillée au paragraphe 3.1.2)

2.2 Systèmes masses-ressorts

Le premier modèle déformable que nous allons aborder est le système masses-ressorts. Ce modèle est bien connu en informatique graphique pour son efficacité.

2.2.1 Principe

Ce sont des systèmes particulaires, où les maillages volumiques sont remplacés par des éléments linéiques et une particule est attribuée à chaque noeud de ce maillage. On a ainsi peu d'éléments peu complexes, ce qui augmente la rapidité.

Les systèmes masses-ressorts font partie des systèmes d'ordre 2 dans lesquels on sait calculer la dérivée seconde (l'accéleration) du système : $\ddot{q} = f(\dot{q}, q, t)$. De tels systèmes permettent de simuler des lois physiques et ainsi d'obtenir des animations et des formes d'apect naturel. Des lois élementaires communes (Newton) permettent de combiner divers modéles d'action (gravite, ressorts,...).

2.2.2 Lois classiques

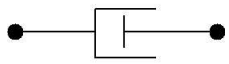
2.2.2.1 Ressort

Un ressort sert à simuler un comportement élastique qui tend à ramener deux particules à une distance donnée l'une de l'autre.

L'action d'un ressort entre les particules i et j est proportionnelle à son allongement (par rapport à sa longueur initiale l_0) et à sa raideur k dans la direction $\overrightarrow{ji}$:

$$f_{j\to i} = k(\frac{\|q_i - q_j\| - l_0}{\|q_i - q_j\|})\frac{q_i - q_j}{\|q_i - q_j\|}$$

2.2.2.2 Amortissement intrinsèque



Un amortisseur sert à réduire le mouvement relatif de deux particules.

L'action d'un amortisseur est proportionnelle à sa viscosité ν et à sa vitesse d'allongement (différence des vitesses projetées sur la direction). L'action sur une particule est donnée par la relation :

$$f_{j\to i} = \nu(\frac{(\dot{q}_i - \dot{q}_j)(q_i - q_j)}{\|q_i - q_j\|})\frac{q_i - q_j}{\|q_i - q_j\|}$$

2.2.2.3 Ressort amorti

Il est possible de combiner des forces en chaque particule en sommant les contributions définies par chacune. Ainsi, le ressort amorti s'obtient en combinant un ressort et un amortisseur en parallèle.

2.2.3 Analyse

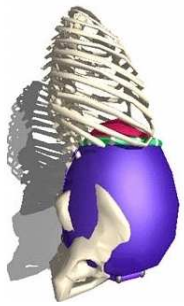
Beaucoup de phénomènes naturels peuvent se représenter par des ressorts amortis, notamment les corps déformables élastiques, mais on imagine facilement les limites d'une approche utilisant des lois physiques aussi simples, en particulier la difficulté de conserver le volume global de l'objet.

Les exemples d'application du modèle masses-ressorts sont nombreux, pour n'en citer que quelques uns : Breene [32-BRE94] ou Volino [33-VOL95] pour animer des vêtements, Cover [29-COV93] ou Lee [30-LEE95] dans le cadre de simulations chirurgicales.

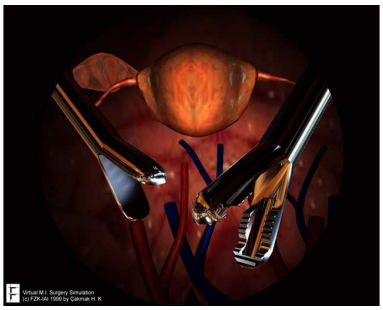
Les travaux d'Emmanuel Promayon [13-PRO97, 14-PRO96] ont montré qu'il est possible d'intégrer des contraintes d'incompressibilité au moyen de lois d'interaction plus perfectionnées. Un de ces modèles utilise une mémoire de forme locale pour définir l'élasticité d'une membrane (figure 2.8 (a)). Cependant seules les surfaces des tissus et organes sont définies, ce qui n'est pas adapté pour la simulation dans un cadre général et en particulier chirurgical. Ce modèle est en train d'être étendu à une représentation volumique. Il devrait être possible de trouver des lois permettant d'approcher des comportements réels à une certaine précision.

Un autre gros problème se situe au niveau du paramétrage. On a accès à la raideur et à l'amortissement des ressorts amortis, mais leurs valeurs peuvent difficilement se calquer au coefficient de Poisson et au module d'Young classiquement expériementé, d'où la difficulté de simuler des matériaux voulus avec des vrais comportements.

Un réseau de masses-ressorts peut servir à représenter une structure déformable. Un habillage géométrique permet alors d'obtenir l'apparence souhaitée.



(a) Modelisation des mouvements du tronc lors de la respiration



(b) Simulation de chirurgie gynécologique par le "Karlsruhe Endoscopic Surgery Trainer"

FIG. 2.8 – Utilisations de modèles masses-ressorts

2.2.4 Conclusion

Grâce à leur simplicité, leur rapidité et leur polyvalence, les modèles masses-ressorts ont un grand succès en synthèse d'images. Pour les mêmes raisons, ils ont été par la suite utilisés dans les simulateurs chirurgicaux.

Bien que les systèmes masses-ressorts soient adaptés à la simulation de déformations élastiques en temps réel, il a été montré que dans leur version simple, ces modèles ne peuvent pas simuler fidèlement des milieux volumiques, notamment, gérer des contraintes d'incompressibilité.

Toutefois, leur ajouter de nouvelles contraintes comme la conservation des volumes semble être une bonne piste pour gagner en précision tout en conservant un modèle rapide, il sera intéressant de voir l'évolution de tels modèles.

Un point positif concerne les changements de topologie assez facilement gérables grâce à la représentation des objets et à leur interaction locale.

Il restera tout de même un problème important, qui est le réglage des paramètres, car on ne sait pas les relier aux données recueillies lors d'expériences, aux caractéristiques réelles des matériaux.

2.3 Méthode des éléments finis

Nous avons vu qu'il est souvent impossible de trouver analytiquement la solution à un problème de la mécanique des milieux continus. On doit alors utiliser des méthodes numériques pour approcher cette solution. La méthode des éléments finis (MEF) est une de ces méthodes, elle est la plus utilisée dans le cas de calcul sur des structures.

Bien que cette méthode nous intéresse uniquement dans le cas de problème de mécanique des solides, elle s'applique aussi pour les fluides et l'électromagnétisme. Plus généralement, elle permet de résoudre des équations différentielles avec conditions aux limites.

2.3.1 Principe

La méthode des éléments finis est un outil de discrétisation, où l'idée est de découper l'objet en un nombre fini d'éléments, dont les sommets sont appelés noeuds. Les propriétés physiques sont interpolées sur chaque élément en fonction de leur valeur aux noeuds. L'équation d'équilibre du système doit être vérifiée sur chacun des éléments, procurant des équations sur les noeuds. Puis un processus d'assemblage prenant en compte les interactions entre éléments voisins mène

à la construction d'un ensemble d'équations. La résolution de ce système d'équations, en tenant compte des conditions imposées aux limites du système, permet d'obtenir une approximation des propriétés physiques du continuum.

On cherche donc à déterminer quelle interpolation locale on attribue à chaque sous domaine pour que le champ global obtenu par juxtaposition de ces champs locaux soit proche de la solution du problème. La qualité de la solution approchée dépend de la division en sous domaines (nombre et dimensions des éléments), du choix de la famille des fonctions d'interpolation dans chaque élément, et des conditions de continuité qu'on impose aux frontières des sous domaines (C^0, C^1, ...).

2.3.2 Maillage

La première étape d'une analyse par éléments finis consiste à définir un maillage du continuum en éléments finis. Les éléments finis peuvent être de différentes formes, il est possible d'utiliser différents types d'éléments dans un même maillage mais cela complique significativement l'écriture du système, dans la majorité des cas un seul type d'éléments est utilisé dans un maillage donné.

mailles linéiques Elles servent à mailler une courbe qui peut représenter un domaine linéique (plongé dans un espace physique à 2 ou 3 dimensions).

mailles surfaciques Elles servent à mailler une surface qui peut représenter un domaine surfacique (plongé dans un espace à 2 ou 3 dimensions).

D'après [5-GAR02], en élasticité et en plasticité, domaines qui nous intéressent, les quadrilatères offrent un meilleur compromis vitesse/précision qu'avec des triangles.

mailles volumiques Les mailles sont des volumes. En élasticité et en plasticité, on obtient de meilleurs résultats avec des hexaèdres qu'avec des tetraèdres.

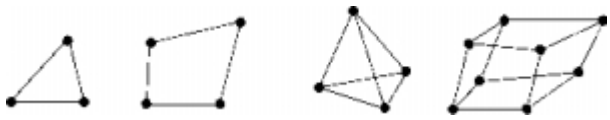


FIG. 2.9 – Mailles surfaciques et volumiques

Plus le maillage est fin, plus le calcul est précis, mais coûteux. On doit un compromis entre la finesse de la représentation géométrique et le coût du calcul. Quand on a une intuition du résultat final on peut décider de mailler grossièrement dans certaines régions et plus finement dans d'autres, en fonction de l'importance des déformations.

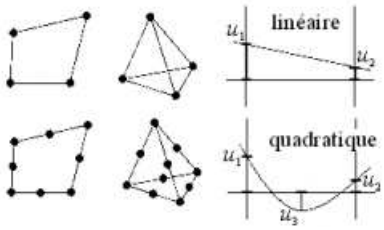
Un avantage supplémentaire est apporté par cette représentation locale des propriétés mécaniques : comme les paramètres mécaniques sont associés à chaque élément d'un maillage, il est possible de construire des modèles hétérogènes, composés de différentes structures imbriquées de propriétés mécaniques différentes.

2.3.3 Fonctions d'interpolation

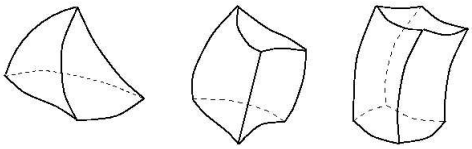
Le principe de la méthode des éléments finis étant d'interpoler les champs physiques en tout point du continuum en fonction de leur valeur aux noeuds, diverses fonctions d'interpolation (ou fonctions de forme) peuvent être choisies. Il est important de bien définir la famille de champs locaux dans les mailles pour s'approcher au mieux de la solution. Il est possible de construire un espace de fonctions d'interpolation directement sur chaque maille réelle, mais généralement, on procède en deux temps : on construit un espace de fonctions d'interpolation sur une maille de référence standard topologiquement équivalente à la maille réelle, puis on le transforme pour qu'il

devienne un espace de fonctions d'interpolation sur les mailles réelles. Ce procédé a l'avantage de faire gagner du temps de calcul.

Parmi les contraintes qu'on impose à la solution approchée cherchée, il y a souvent au moins une continuité simple C^0 à la frontière entre les sous domaines. Pour plus de continuité, il est nécessaire d'ajouter des noeuds. L'ensemble des noeuds ne se limite pas forcément aux sommets des éléments, il est possible de définir des noeuds supplémentaires, par exemple un quadrilatère à 8 noeuds ou un tétraèdre à 10 noeuds.



(a) Courbes d'interpolations en fonction du nombre de noeuds



(b) Plus le degré d'interpolation est élevé, plus l'élement peut se déformer

FIG. 2.10 – Ajouter des noeuds aux éléments augmente les degrés de liberté, qui permettent d'avoir des degrés d'interpolation plus élevé

2.3.4 Assemblage

Il ne suffit pas de prendre en compte le cas d'un seul élément isolé. Dans un maillage d'éléments finis, un noeud donné appartient en général à plusieurs éléments et la déformation de chacun d'entre eux induit une force en ce noeud.

Le calcul des forces se fait élément par élément avec des systèmes locaux. Or des sommets sont communs à plusieurs éléments, il faut donc assembler tous les systèmes locaux en un gros système global contenant les équations de tous les noeuds.

Pour constuire le système global, on va utiliser le fait que l'équation d'équilibre doit être vérifiée sur chacun des éléments, ce qui va se traduire par des équations sur chacun des sommets. Cette équation comporte des dérivés partielles qui s'expriment comme une combinaison des valeurs prises par la déformation sur les autres sommets voisins.

En regroupant toutes les équations de chaque sommet, on obtient un système où les inconnues sont les valeurs de la fonction sur les sommets du maillage :

$$F = g(U)$$

où f est un vecteur contenant les forces pour chaque noeud, u est un vecteur contenant les deplacement de chaque noeuds, $g(u)$ est la fonction transformant les déplacements en forces.

Pour un problème statique, il ne reste plus qu'à résoudre ce système pour obtenir l'approximation de la fonction déformation sur le maillage, dont on pourra déduire les contraintes internes grâce à la loi de comportement du matériau.

2.3.5 MEF explicites

La méthode classique décrite jusqu'à maintenant nécessite l'inversion d'un système global, on appelle de telles méthodes "éléments finis implicites". Au contraire, il existe aussi les "éléments finis explicites" qui s'apparentent plus aux systèmes de particules afin de profiter de leur flexibilité.

Contrairement aux applications traditionnelles de la mécanique par éléments finis, les simulateurs interactifs n'ont pas pour vocation de calculer directement une configuration d'équilibre, ils ne nécessitent que des calculs dynamiques que l'on peut appliquer directement.

Le principe est de calculer les forces qui s'exercent sur chaque sommet pour chaque élément indépendamment. Ensuite, on somme toutes les forces qui s'appliquent à un sommet auquel est associée à une masse. On ne cherche pas à trouver la position d'équilibre du sytème global, mais à calculer les accélérations des sommets.

On gagne énormément en vitesse de calcul, mais plusieurs pas de calcul sont nécessaires pour propager les déformations, chaque sommet ne réagissant qu'à la position de ses voisins à l'instant précédent. Par contre l'état final d'équilibre du système amorti est le même que calculé par la statique (la position peut varier).

Un autre avantage de ce type de résolution est de facilité la combinaison de plusieurs types d'éléments dans le maillage, puisque chaque élément est traité indépendamment.

Un autre point important, pas encore soulevé, et celui des changements de topologie dans l'objet représenté. En effet, il peut être important de simuler des déchirures et des découpes dans le cadre chirurgical. Dans le cas de MEF explicites, cette gestion se trouve très fortement simplifiée par l'absence d'un système global.

2.3.6 Conclusion

Bien que la méthode des éléments finis semble plus compliquée et plus lourde en calcul, c'est une méthode qui a déjà fait ses preuves en terme de précision. C'est le côté précision de cette méthode qui est très intéressant ; c'est le point de vue vitesse de calcul qu'il faut améliorer. Nous aborderons, par la suite, des formulations plus simples et plus rapides.

2.4 Diversité de la MEF

Il existe plusieurs modèles physiques basés sur la méthode des éléments finis. Nous allons passer en revue les plus importants et tenter de voir leurs qualités et défauts.

2.4.1 Méthodes statiques avec tenseur de déformations linéaire

La première formulation plus ou moins rapide de la MEF que nous allons étudier est celle basée sur la théorie de l'élasticité linéaire.

Nous avons vu précedemment l'avantage, en terme de temps de calcul, d'utiliser un tenseur de déformation et une loi de comportement linéaires. La résolution du sytème qui devient matriciel est beaucoup plus facile à résoudre. Par contre, dans ce cas, nous avons aussi vu que plus les déformations deviennent grandes, plus le modèle devient imprécis, avec l'apparition de gonflements lors de rotations.

2.4.1.1 Le système global

Dans le cas de l'elasticité linéaire, le système peut s'écrire sous forme matricielle :

$$F = KU$$

où K est la matrice de rigidité du système, U est le vecteur regroupant chaque composante de déplacement de chaque sommets, et F est le vecteur regroupant chaque composante de force appliquée aux sommets.

En pratique, puisque seuls les sommets les plus proches interviennent dans l'expression des dérivées, chaque équation n'implique qu'un petit nombre de sommets. La matrice K née de leur regroupement sera donc creuse. Il est possible de la mettre sous la forme d'une bande de faible largeur afin de rendre la numérisation optimale. Des méthodes de résolution adaptées peuvent alors être appliquées.

On peut noter que le conditionnement de la matrice (i.e. sa facilité à être inversée) dépend directement des coefficients de raideur du matériau : plus il est rigide, plus l'inverse est difficile à calculer.

2.4.1.2 Le système local

Dans cette partie, nous allons voir la formulation linéaire des tenseurs qui rend l'équation plus facile à implémenter.

Pour chaque élément, on a :

$$f = B^T DBu$$

- B est la matrice de rigidité de l'élément :

$$B = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & ... & \frac{\partial N_n}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & ... & 0 & \frac{\partial N_n}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & ... & 0 & 0 & \frac{\partial N_n}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & ... & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & ... & 0 & \frac{\partial N_n}{\partial z} & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & ... & \frac{\partial N_n}{\partial z} & 0 & \frac{\partial N_n}{\partial x} \end{bmatrix}$$

où N_i est la fonction d'interpolation du i^{ime}noeud et n le nombre de noeuds de l'élément ; les fonctions d'interpolation dépendent du type de l'élément (cf annexe A).

- D est la matrice de rigidité du matériau :

$$D = \varphi(V)\frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$$

où $\varphi(V)$ est une fonction dépendante du volume et du type d'élément.

- $u = \begin{pmatrix} u_{1_x} \\ u_{1_y} \\ u_{1_z} \\ ... \\ u_{n_x} \\ u_{n_y} \\ u_{n_z} \end{pmatrix}$ et $f = \begin{pmatrix} f_{1_x} \\ f_{1_y} \\ f_{1_z} \\ ... \\ f_{n_x} \\ f_{n_y} \\ f_{n_z} \end{pmatrix}$ représentent respectivement les déplacements et les forces de tous les n noeuds d'un élément pour les axes x, y, z.

Bu correspond aux déformations : $\{\varepsilon\} = Bu$
DBu correspond aux contraintes : $\{\sigma\} = DBu = D\{\varepsilon\}$
B^T permet de transformer les contraintes en forces.
$K = B^T DB$ est la matrice de rigidité.

2.4.2 Précalculs

Il est possible d'améliorer l'efficacité des calculs de la MEF en élaticité linéaire grâce à des précalculs.

L'approximation linéaire procure des hypothèses intéressantes :

- La déformation résultante d'une déformation d d'un point est la même que la somme de celles qui seraient produites par les trois déformations d_x, d_y, d_z appliquées séparement dans chacunes des trois direction x, y, z.

– Avec un déplacement double, on obtient une déformation double.
Cotin ([4-COT97]) remarque qu'il est alors possible de stocker pour chaque déplacement dans chaque direction de chaque point séparement, quelle va être la déformation de tout l'objet. (Ces informations sont filtrées en éliminant les déplacements trop petits afin d'économiser de la mémoire).

Un problème est de connaître la déformation lorsqu'on déplace plus d'un point à la fois. Cotin n'applique qu'une partie des déformations créées par chacun des déplacements, en utilisant une matrice d'influence mutuelle des noeuds déplacés, dont l'inverse va determiner les déplacements à appliquer aux différents noeuds.

Bro-Nielsen ([21-BRO98]) et Cotin ([22-COT99]) ont réussi à faire évoluer des maillages de l'ordre de 250 éléments avec une résolution dynamique et 1500 éléments avec une résolution quasi-statique (entre chaque pas de temps est appliqué un déplacement résolu par une méthode statique, figure 2.11).

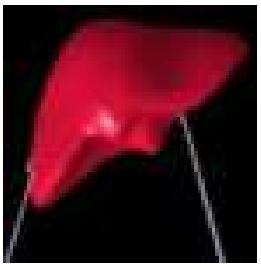


FIG. 2.11 – Déformation d'un modèle de foie par une méthode d'éléments finis quasi-statique

Les temps de précalculs et la place mémoire utilisée sont très important, mais cette méthode a permis des animations temps-réel.

Même s'il est possible de déplacer plusieurs noeuds en même temps, c'est au prix d'une inversion de matrice (de taille proportionnelle au nombre de noeuds).

En plus d'être basée sur l'élasticité linéaire, l'inconvénient majeur de cette méthode est qu'elle est quasi-statique, c'est à dire qu'elle ne donne qu'une suite d'états d'équilibre. Donc concrètement, si on lache un point que l'on a déplacé, quelque soit la distance de déplacement, celui-ci va revenir instantanément à sa position initiale, sans oscillation.

On a à faire à une méthode basée sur la méthode des éléments finis qui permet des animations interactives. Malheureusement, celle-ci est trop imprécise en grands déplacements, de plus le réalisme visuel n'est pas là à cause des retours brusques.

Un autre défaut est l'impossibilité de faire des changements de topologie du maillage. Il n'est pas possible avec cette méthode de simuler des coupures, déchirures ou perforations, car dès qu'il y a changement de topologie, la matrice de raideur du système se trouve modifiée. Il faudrait alors la ré-inverser, ce qui est une opération bien trop lourde pour être réalisée en temps réel.

2.4.3 Méthode des éléments de frontières

Cette méthode, proposée par James et Pai ([24-JAM99]) est basée sur la MEF. Elle utilise l'équation de Navier, qui est obtenue en intégrant l'équation de conservation de l'énergie en combinaison avec les équations constitutives d'un matériau continu, isotrope et élastique linéaire :

$$(Nu)(x) + b(x) = 0$$

où x sont les coordonnées cartésiennes du milieu, N est un opérateur linéaire différentiel du second degré, b est le champ des forces appliquées au système, u est le champ des déplacements.

Cette équation peut être résolue par une méthode des éléments de frontière, qui est le pendant de la méthode des éléments finis pour les équations intégrales. La frontière du système est découpée

en éléments disjoints, à l'intérieur desquels le champ des déplacements est interpolé linéairement en fonction des déplacements des noeuds. On a bien une méthode par éléments finis améliorée en remarquant qu'il n'y aura pas de forces externes appliquées ailleur que sur la surface de l'objet.

Quoique mathématiquement complexe, cette méthode est utilisable en temps réel pour deux raisons. D'une part, seule la surface est discrétisée et le nombre d'équations est seulement proportionnel au nombre de noeuds de surface bien que la structure volumique soit prise en compte par les équations. D'autre part, on peut exploiter la linéarité du système en pré-calculant un ensemble de réponses élémentaires (fonctions de Green), qui peuvent ensuite être combinées en temps réel. Malheureusement, parcequ'elle exploite abondamment les propriétés de linéarité, il n'est a priori pas possible d'adapter cette méthode à des lois mécaniques plus complexes que l'élasticité linéaire.

Monserrat ([25-MON01]) a présenté la même méthode et tenté de l'appliquer à la simulation de déformations mécaniques du foie de porc. Cependant ils ont conclu que le modèle de tissu élastique linéaire n'était réaliste que pour de très faibles déformations et de faibles vitesses, d'où la nécessité de modifier et d'accélérer l'algorithme pour une meilleure modélisation.

De plus, un tel modèle reste très limité dans la gestion des changement de topologie à cause de son intérieur non maillé.

2.4.4 Systèmes masses-tenseurs

Ce modèle présenté par Delingette ([23-DEL99]) se base sur la méthode des éléments finis avec des éléments tetraèdriques linéaires. L'idée est d'extraire chaque calcul pour chaque point, afin de mettre de côté tous les termes utilisés plusieurs fois. Ces termes pourront ainsi n'être calculés qu'une fois au bon moment, puis réutilisés. Ainsi les calculs des tenseurs puis l'assemblage sont cachés. Ce modèle s'inscrit donc dans les MEF explicites.

À partir de la discrétisation du champ des déplacements sur un élément, on peut exprimer l'énergie de déformation de l'élément en fonction des déplacements de ses sommets, puis dériver cette expression pour obtenir la force élastique exercée sur chacun des sommets. En élasticité linéaire, cette force s'exprime sous la forme :

$$f_i = K_{ii}u_i + \sum_{j \neq i} K_{ij}u_j$$

où f_i est la force totale exercée sur le noeud i, K_{ii} représente la somme des contributions du sommet i dans tous les tétraèdres adjacents à i, K_{ij} représente les sommes des contributions des autres sommets j de tous les tétraèdres adjacents à i.

Les différents tenseurs K_{ij} peuvent être pré-calculés en considérant qu'ils ne dépendent que de la géométrie du maillage au repos et des propriétés mécaniques.

Ensuite cette force est réintroduite dans une équation dynamique locale pour calculer le champ des déplacements à l'instant suivant, et l'algorithme est répété de façon itérative :

$$m_i\ddot{u}_i = -d_i\dot{u}_i + f_i$$

où m_i est la masse associée au noeud i, d_i est le coefficient d'atténuation associé au noeud i.

Étant donné que toutes ces opérations sont locales, il est possible de procéder à des changements de topologie en temps réel : il suffit pour cela de mettre à jour les adjacences entre éléments voisins et les tenseurs correspondants à ces adjacences.

L'optimisation des calculs peut être effectués quels que soient les tenseurs utilisés. Il est ainsi possible d'utiliser un masses-tenseurs en élasticité linéaire ou non-linéaire, suivant la modélisation souhaitée.

Afin de prendre en compte les grands déplacements, Picinbono ([9-PIC03]) a présenté une extension du modèle des masses-tenseurs basée sur le modèle d'élasticité de St. Venant-Kirchhoff, qui intègre la non-linéarité géométrique et conserve la linéarité mécanique. Le procédé consiste à discrétiser l'expression généralisée de l'énergie mécanique dérivée du modèle de St Venant-Kirchhoff de la même façon que pour la méthode des masses-tenseurs linéaire, ce qui résulte en une expression plus générale de la force incluant des termes quadratiques et cubiques. Cet algorithme a conduit à un temps de calcul cinq fois supérieur à celui de l'algorithme des masses-tenseurs linéaire, mais est resté suffisamment rapide pour autoriser des simulations en temps réel sur des objets relativement complexes (figure 2.12).

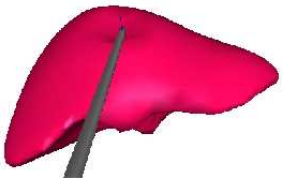


FIG. 2.12 – Palpation d'un foie animé par une méthode de masses-tenseurs

Dans son déroulement, la méthode des masses-tenseurs est assez semblable à celle des masses-ressorts car dans les deux cas le système d'équations dynamiques est résolu itérativement et localement. Cependant les deux modèles sont physiquement très différents car les masses-ressorts sont une représentation discrète et dépendante de la topologie du maillage, tandis que la méthode des masses-tenseurs est basée sur une représentation continue de la matière qui est indépendante de la topologie.

L'algorithme de masses-tenseurs s'inscrit bien dans le cadre des objectifs fixés. Il utilise une représentation continue et volumique de la matière, qui est bien plus proche de la réalité physique qu'une représentation discrète de type masses-ressorts ou qu'un modèle surfacique.

Un autre avantage non négligeable est la possibilité de changements topologiques (coupures, déchirures...), grâce à l'absence de matrice globale.

En fait, l'algorithme apparait comme une bonne optimisation de l'écriture mathématique de la méthode des éléments finis, et peut offrir de bonnes performances en termes de temps de calcul, tout en conservant les avantages des éléments finis. Cette approche est particulièrement intéressante.

Une question intéressante est de savoir si le modèle de St. Venant-Kirchhoff convient aux propriétés de tissus biologiques, ce qui nécessite une comparaison avec des données expérimentales, qui ne semble pas encore réalisée. De plus, dans ce modèle, il y a une simplification qui peut faire perdre de la précision, car toute les matrices de rigidité sont précalculées, or, normalement, celles-ci varient au cours des déformations.

Un autre point négatif est que le temps est intégré suivant un schéma explicite, donc la rigidité est importante, plus il faut réduire les pas de temps donc augmenter les calculs.

2.4.5 Linéarité géométrique et grands déplacements

Il existe des méthodes conservant un tenseur de déformations linéaire mais sans avoir les gonflements provoqués par des grandes déplacements. Pour cela, il faut trouver un moyen de conserver l'invariance en rotation, normalement fournie par le terme quadratique du tenseur de déformations non-linéaire.

Ces méthodes utilisent un modèle explicite, c'est à dire sans assemblage ; elles ont pour but le dynamisme et ont l'avantage de faciliter la gestion des changements de topologie.

L'idée principale est d'éliminer les rotations locales subies au cours de l'animation en se ramenant dans un repère global pour calculer les déplacements. Ensuite, les forces sont calculées classiquement en fonctions des déplacements issus du repère global. Cela revient à faire chaque calcul localement en petits déplacements et ainsi de rester au plus proche de la vraie loi de comportement.

Les forces calculées subissent ensuite la rotation inverse pour être appliquées à l'objet.

Localement :

$$u_i = x_i^{repos} - R_{repos}^{deforme} x_i^{deforme}$$

et

$$f_i = R_{global}^{dforme} K_i u_i$$

avec R_j^i la rotation du repère i au repère j.

Différentes méthodes ont été proposées pour calculer cette rotation.

2.4.5.1 Méthode de Müller

Müller [1-MUL02] travaille sur les sommets d'éléments tetraèdriques à 4 noeuds et cherche les rotations subies par chaque noeud. Pour trouver ces rotations, il utilise un repère orthonormé direct construit à partir des arètes incidentes au sommet traité. La méthode de construction des repères étant toujours la même, il suffit d'extraire la rotation entre le repère du noeud sur l'objet au repos et sur l'objet déformé pour obtenir la rotation subie par le noeud au cours de l'animation (figure 2.14). Il faut donc trouver une rotation par sommet, et faire « tourner » quatre fois (le nombre de points de l'élément) chaque élément (figure 2.13).

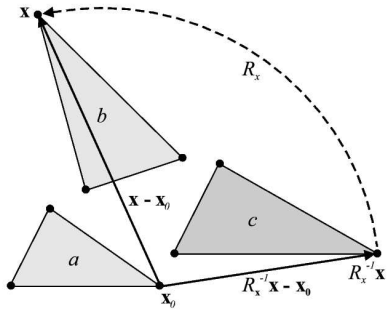


FIG. 2.13 – Le triangle a est l'élément au repos. Le triangle b est le déformé. Le triangle c est le triangle b ramené dans le même repère que a. Les petites déformations sont mesurées entre a et c.

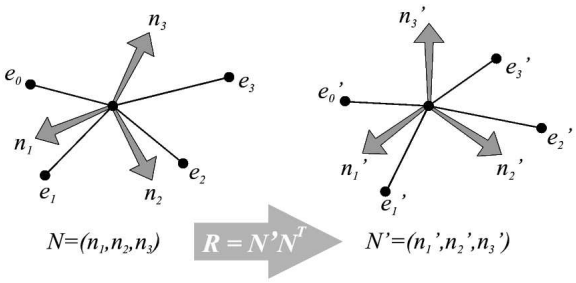


FIG. 2.14 – Pour trouver la rotation qui ramène l'élement au repos dans le repère de l'élement déformé il faut construire les repères orthonormés directs N et N' centrés sur le même sommet en fonction des arêtes incidentes e_i, respectivement sur l'objet au repos et sur l'objet déformé. La rotation du repère N vers N' est alors $R = N'N^T$.

Au premier abort, cette méthode semble bien adaptée à notre problème puiqu'elle utilise un apect continu du milieu, qu'elle est très rapide grâce à son calcul qui reste linéaire et permet d'animer en grands déplacements un très grands nombre d'éléments sans gonflement.

Par contre, on ne sait pas jusqu'à quel point cette méthode respecte la réalité dans ses animations. Surtout que Müller parle de forces fantômes qu'il est obligé de compenser « à la main »... D'après lui [10-MUL04], le fait de travailler noeud par noeud ne garanti pas une somme des forces nulle, c'est pour cette raison qu'il a depuis abandonné cette méthode pour celle de Etzmuß (paragraphe suivant 2.4.5.2) généralisée aux tetraèdres.

2.4.5.2 Méthode de Etzmuß

Etzmuß dans [2-ETZ03] anime seulement des surfaces triangulées pour simuler des vêtements, mais son approche s'extrapole intuitivement aux volumes.

Il ne travaille pas sur les sommets comme Müller, mais sur les éléments. Il cherche donc la rotation subie par chacun des éléments. Il faut trouver une rotation par élément et chaque élément sera « tourné » une seule fois. Pour cela, il utilise la transformation qui amène un élément au repos à son état déformé (cf figure 2.15). Il en extrait la rotation à l'aide des valeurs et des vecteurs propres (cf algorithme 1), que l'on peut voir géométriquement comme des repères qui répartissent au mieux la matière.

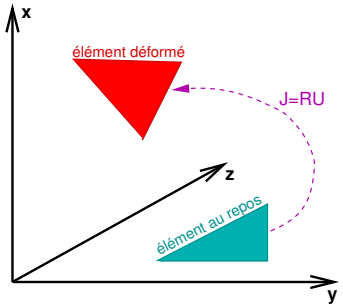


FIG. 2.15 – J transforme l'élément au repos (vert) en l'élément déformé (rouge)

Algorithme 1 Calcul de la rotation par décomposition polaire

Trouver J la matrice de passage de l'élément au repos vers l'élément déformé
Décomposer J en une rotation R et une déformation pure U : $J = RU$
En utilisant la décomposition polaire :

1. Calcul de $U^2 = J^T J$ (car $J^T J = U^T R^T RU = U^T U$)
2. Calcul des valeurs propres λ_1, λ_2 et des vecteurs propres v_1, v_2 de U^2
3. Calcul de $U = \sqrt{\lambda_1} v_1 v_1^T + \sqrt{\lambda_2} v_2 v_2^T$
4. Calcul de $R = JU^{-1}$

Cette approche plus mathématique garantit de trouver les plus petits déplacements possibles, et ainsi d'être au plus proche de la loi de comportement réelle et de garantir la meilleure précision.

En terme de précision la méthode semble idéale. Son problème vient de l'efficacité. Car même si le calcul reste linéaire, la recherche des valeurs et des vecteurs propres est longue. Elle est juste assez rapide sur des éléments surfaciques pour être utilisée dans le cadre d'applications interactives.

2.4.6 Autres améliorations applicables aux MEF

Par dessus ces méthodes issues de la MEF, peuvent venir se gréfer des méthodes permettant un gros gain en temps de calcul ou en précision.

2.4.6.1 Modèles hybrides

Ces modèles mélangent plusieurs approches, afin de décomposer au mieux le problème, et de simuler chaque partie avec la méthode qui lui est le plus adaptée. Il n'est pas forcément évident de faire interagir les différentes couches entre elles.

Avec cette idée, Cotin [8-COT02] a mis en place en modèle hybride (figure 2.16) utilisant à la fois un modèle précalculé (paragraphe 2.4.2) et un modèle masses-tenseurs (paragraphe 2.4.4). L'avantage du modèle précalculé est sa rapidité, donc on peut l'utiliser sur un grand modèle détaillé. Par contre il ne permet pas facilement les grosses manipulations, ni les changements topologiques qui seront alors gérés en certains endroits par le modèle masses-tenseurs.

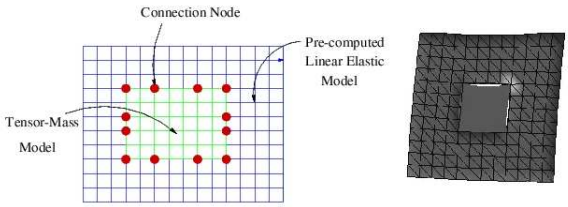


FIG. 2.16 – Approche hybride

2.4.6.2 Multi-résolutions

L'approche multi-résolutions rappelle le modèlehybride dans l'idée de simuler un problème en plusieurs sous problèmes adaptés. Mais ici, la difficulté est que les couches doivent être générées automatiquement suivant l'état du sytème, au cours de l'animation, afin de permettre la meilleure interactivité possible.

Debunne a présenté un modèle multi-résolutions du foie [12-DEB00] (figure 2.17) où les parties en déformations sont gérés par un modèle tetraèdrique plus raffiné, et où le reste est plus grossier, afin de limiter le nombre d'éléments. Il a aussi ajouté plusieurs résolutions temporelles : les parties en forte déformation demandant plus de précision sont gérées à un pas de temps plus petit.

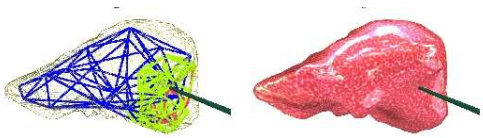


FIG. 2.17 – Approche multi-résolution

2.4.6.3 Surfaces de formes libres (Free Form Deformation)

Il est possible de s'extraire (au moins partiellement) de la géométrie de l'objet pour le mailler, et ainsi de faire des éléments indépendants de la surface. L'étape de maillage est fortement simplifiée et les éléments sont générés avec plus de contrôle, évitant les dégénérescences afin de préserver la meilleure stabilité possible. N'importe quel modèle surfacique peut ansi être aisément animé, alors que les modèles découpés en mailles volumiques sont complexes à fabriquer.

Frisch [19-FRI02] puis Müller [16-MUL04] (figure 2.18) ont utilisé des FFD comme éléments à animer, le modèle surfacique à afficher est alors interpolé.

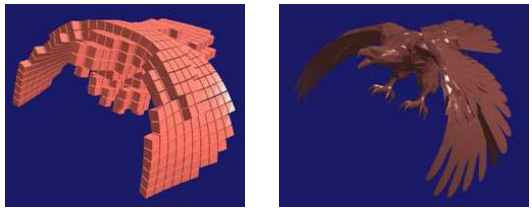


FIG. 2.18 – Maillage en FFD

2.4.6.4 D'autres matériaux

Assimiler les matériaux déformables par une loi de comportement élastique linéaire est très réducteur, nous présentons ici quelques techniques simples pour en améliorer le comportement.

Matériaux plastiques En petites déformations, un matériaux plastique a un comportement élastique. En grandes déformations, celui-ci aura un comportement plastique pas du tout linéaire. Au delà d'un certain seuil de déformation, celui-ci ne va pas revenir à sa position au repos et va rester un peu déformé. A partir de ce nouvel état au repos, il aura de nouveau un comportement élastique avec la même pente que précédemment. Jusqu'à une certaine déformation pour laquelle il y aura déchirure (illustré par la figure 2.19).

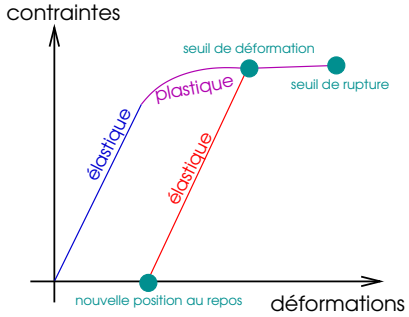


FIG. 2.19 – Une loi de comportement plastique

On peut voir ce comportement comme une succession de comportements élastiques pour lesquels il suffit de donner un seuil.

Matériaux anisotropes Picinbono [7-PIC02] a présenté une méthode traitant le cas d'un matériau non-isotrope. Le cas traité était celui de matériaux dont les propriétés sont différentes pour une seule direction donnée, qui sont appelés transversalement isotropes, c'est par exemple le cas des structures à renfort fibreux, comme les muscles. Elle conduit à introduire deux paires de coefficients de Lamé au lieu d'une : deux coefficients dans la direction longitudinale et deux autres dans le plan transverse. L'expression de la force qui en découle est de la même forme, seul le calcul des matrices de raideur est différent et dépend de quatre paramètres au lieu de deux.

Matériaux non-linéaires Il est possible d'approcher une non-linéarité mécanique en faisant évoluer le coefficient de Poisson et le module d'Young dans certaines directions, en fonction de la déformation dans cette direction. Pour cela, on peut, par exemple, utiliser des fonctions prenant en compte les déformations dans chaque direction, mais le plus simple (et efficace) reste l'utilisation de courbes par morceaux (figure 2.20).

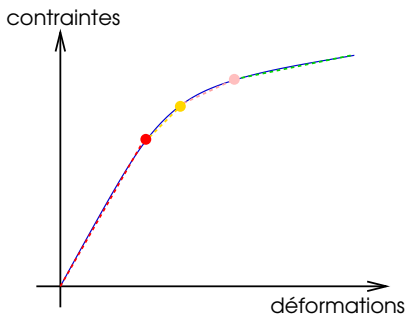


FIG. 2.20 – Loi de comportement non-linéaire approximée par une courbe par morceaux

2.4.6.5 Conclusion

Toutes ces méthodes d'amélioration de temps de calcul ou de précision ne sont pas discriminantes, mais sont complémentaires, on peut très bien imaginer les utiliser ensemble. Par exemple, il serait possible d'utiliser des FFD multi-résolutions sur des matériaux plastiques anisotropes.

2.4.7 Conclusion

Le gros avantage de l'élasticité linéaire est sa facilité de calcul. Mais, ses approximations peuvent être très regrétables dans le cas d'objets déformables pour lesquels il est nécessaire de gérer les grands déplacements.

L'écriture en masses-tenseurs permet de faire de la dynamique et ce de manière très rapide. De plus grâce à Picinbono, il est possible de gérer les grands déplacements en utilisant une non-linéarité géométrique. Cette méthode est très intéressante pour notre problématique, et à d'ailleur été crée dans un but chirurgical.

Parrallélement, des méthodes récentes essayent de gérer ces grands déplacements tout en conservant une linéarité géométrique, malheureusement ces méthodes sont, soit encore trop lentes (Etzmuß), soit la précision ne semble pas au rendez-vous (Müller). Cette voie qui est lancée semble bonne, mais pas encore assez fouillée pour des animations temps-réelles précises.

2.5 Conclusion

On peut voir qu'il existe un grand nombre d'outils mis à disposition, mais que tous découlent des mêmes bases physiques. Tout à déjà été plus ou moins décrit dans le domaine de la mécanique des solides, mais on voit que l'adaptation à l'informatique n'est pas si évidente et pas immédiate, surtout que les objectifs principaux des deux domaines ne sont pas les mêmes.

Il est actuellement possible de manipuler des objets déformables interactivement, même dans le cas de simulations chirurgicales. Par contre, si la manipulation et l'aspect visuel n'est pas à remettre en cause, on peut se poser des questions sur la précision de ces simulations. En effet, la précision de ces modèles rapides a très peu (pas ?) été testée.

On peut aussi se questionner sur la facilité de mise en place et de réglages des paramètres sur les modèles physiques : que le module d'Young et le coefficient de Poisson soient pris en compte est une chose, mais est-ce que leur valeur est bien exactement respectée et correspond à des valeurs expérimentales ?

Regardons maintenant dans quelle mesures ces modèles peuvent répondre à notre problématique.

Les systèmes particulaires montrent des faiblesses pour animer avec précision des modèles volumiques, et même si des nouvelles contraintes basées sur le volume peuvent être ajoutées, il nous semble plus intéressant de se tourner vers les méthodes basées sur les éléments finis. Celles-ci ont montré un comportement plus précis et il commence à en exister des versions rapides.

Les modèles masses-tenseurs en sont une formulation très intéressante, mais ils ne font qu'optimiser les calculs qui eux restent classiques, cette formulation est très bien utilisable sur d'autres calculs. De plus, le principe est basé sur le précalcul de la matrice de rigidité à partir des éléments au repos, cette approche peut créer des imprécisions et peut entraîner des instabilités. Enfin, contrairement aux masses-tenseurs de Delingette ou Picinbonno, on préfére utiliser une intégration du temps suivant un schéma implicite, afin que les temps de calculs soient plus indépendants de la rigidité du matériau.

Les modèles qui nous ont le plus intéressés sont ceux qui conservent la linéarité géométrique tout en gérant les grands déplacements. Le raisonnement apporte réellement quelque chose de nouveau en allant à la base du modèle physique.

Les autres améliorations comme la multi-résolution, ou la formulation en masses-tenseurs sont orthogonales et peuvent toujours être mises en place sur un nouveau modèle.

Les modèles de cette voie déjà existants ne nous conviennent pas totalement. Celui de Müller (paragraphe 2.4.5.1) est très rapide et stable mais il comporte tout de même des forces fantômes. De plus, Müller lui-même n'utilise plus son modèle dans ses articles récents [10-MUL04], mais étend celui de Etzmuß (paragraphe 2.4.5.2) au cas des volumes. La méthode de Etzmuß est pourtant moins rapide, mais son concept de traitement élément par élément est intéressant. Pour accéder au temps réel, Müller précalcule toutes les matrices de rigidité des élements. Une telle approche ne nuit pas à la qualité visuelle des résultats, par contre elle change forcément les résultats en termes de précision.

Cette approche nous paraît compatible avec les contraintes d'interactivité propres à notre projet. Nous désirons apporter des améliorations à cette méthode, telle qu'une gestion plus efficace du déplacement rigide et des lois de comportement plus réalistes.

Chapitre 3

Ma contribution

Mon travail a consisté dans un premier temps à mettre en place un modèle déformable rapide basé sur l'idée des MEF explicites, avec un souci de précision et de respect des paramètres (travail chez EVASION). Nous avons donc choisi de garder l'aspect du traitement par élément qui semble intéressant en terme de précision, sans précalculer les matrices de rigidité, toujours dans un souci de précision. Pour gagner en vitesse de calcul, les nouveautés vont se situer au niveau du calcul de la rotation et au niveau du calcul de la matrice de rigidité de l'élément au cours de l'animation. La formulation utilisée pour calculer les forces en fonction des déplacements est celle de la MEF avec tenseurs linéaires développée paragraphe 2.4.1.2.

Dans la deuxième partie de mon stage, j'ai testé la précision de ce modèle, puis je l'ai intégré dans une application de simulation chirugicale (travail chez GMCAO).

3.1 Contexte

3.1.1 AnimAL

Mon travail s'intégre dans la bibliothèque AnimAL (Animation Algorithms Library) développée par François Faure.

Cette librairie a pour but de mutualiser les travaux menés dans l'équipe EVASION et de les rendre aisément réutilisables par différentes personnes dans différents contextes. Elle consiste en une collection de classes et algorithmes dédiés à la simulation dynamique et la visualisation graphique de phénomènes complexes. Elle rassemble des classes de base (vecteurs, quaternions, ...), des méthodes d'intégration numérique (Euler, Runge-Kutta, Euler implicite...), des méthodes expérimentales pour l'animation par modèles physiques, des outils divers... Ecrite en C++, elle fait largement appel aux types génériques (template) et à la programmation objet. Elle se veut une boite à outils où chacun puise les éléments nécessaires plutot qu'une plate-forme monolithique.

Il a fallu ajouter des méthodes facilement manipulables par quelqu'un qui utilise la bibliothèque, tout en ecrivant du code réutilisable et en laissant la possibilité de l'étendre.

Mon travail se situe donc au niveau du calcul des contraintes en fonctions des déplacements.

3.1.2 Intégration dynamique

L'intégration du temps ne faisait pas explicitement partie de mon travail, mais son importance est capitale, car elle intervient aussi bien au niveau de la vitesse que de la précision du modèle et aussi au niveau de sa stabilité. La méthode qui a été retenue comme référence pour mon travail est celle d'Euler implicite. D'après la littérature, il semble que les intégrations basées sur cette méthode offrent un bon compromis stabilité / vitesse / précision.

Notre intégration dynamique se formule de la manière suivante :

$$\begin{cases} x(t+h) = x(t) + h\dot{x}(t+h) = x(t) + \Delta x \\ \dot{x}(t+h) = \dot{x}(t) + h\ddot{x}(t+h) = \dot{x}(t) + \Delta\dot{x} \end{cases}$$

où x est la position d'une particule, $\dot{x}$ est sa vitesse, $\ddot{x}$ est son accélération et h est le pas de temps.

On cherche à determiner $\Delta\dot{x}$, en utilisant les forces $f(x+\Delta x, \dot{x}+\Delta\dot{x})$ au pas de temps $t+h$:

$$\Delta\dot{x} = h\ddot{x}(t+\Delta t) = h\frac{1}{m}f(x+\Delta x, \dot{x}+\Delta\dot{x}) = h\frac{1}{m}(f(x,\dot{x}) + \frac{\partial f(x,\dot{x})}{\partial x}h(\dot{x}+\Delta\dot{x}) + \frac{\partial f(x,\dot{x})}{\partial \dot{x}}\Delta\dot{x})$$

En factorisant par $\Delta\dot{x}$, on obtient :

$$(I - \frac{1}{m}h\frac{\partial f}{\partial \dot{x}} - \frac{1}{m}h^2\frac{\partial f}{\partial x})\Delta\dot{x} = \frac{1}{m}h(f(x) + \frac{\partial f}{\partial x}h\dot{x})$$

En 1D, on a

$$\Delta\dot{x} = \frac{\frac{1}{m}h(f(x) + \frac{\partial f}{\partial x}h\dot{x})}{(I - \frac{1}{m}h\frac{\partial f}{\partial \dot{x}} - \frac{1}{m}h^2\frac{\partial f}{\partial x})}$$

En plusieurs dimensions, il faut résoudre un système. Il n'est pas nécessaire d'inverser la matrice, car on n'a pas besoin de résoudre tous les sytèmes, mais seulement un seul.

Une simplification très courante permet de réduire considérablement les calculs. Il s'agit de considérer $\frac{\partial f}{\partial \dot{x}} = \alpha\frac{\partial f}{\partial x}$, α correspondant au facteur d'amortissement. Analytiquement, cela revient à faire varier (très faiblement) le pas de temps.

On a alors :

$$(I - \frac{1}{m}hk\frac{\partial f}{\partial x} - \frac{1}{m}h^2\frac{\partial f}{\partial x})\Delta\dot{x} = \frac{1}{m}h(f(x) + \frac{\partial f}{\partial x}h\dot{x})$$

$$(I - \frac{1}{m}h(h+k)\frac{\partial f}{\partial x})\Delta\dot{x} = \frac{1}{m}h(f(x) + \frac{\partial f}{\partial x}h\dot{x})$$

Ce système est résolvable par la méthode des gradients conjugués, et pour cela, il faut pouvoir calculer la variation de la force à partir de la variation de la position. Une partie de mon travail a donc consisté à implémenter $\frac{\partial f}{\partial x}\Delta\dot{x}$. Vu que l'on développe une méthode explicite, cette matrice n'est pas globalement construite, mais pour chaque noeud, on somme les contributions de chaque élément.

3.1.3 Eléments

Pour commencer, nous allons utiliser des éléments tetraédriques comme c'est le cas dans quasiment toutes les méthodes étudiées précédemment. Le tétraèdre étant l'élément de référence de la 3D, il est plus facile à utiliser car il représente directement une interpolation trilinéaire.

Dans un premier temps, nous utiliserons des tetraèdres linéaires à 4 noeuds. Dans ce cas, $\varphi(V) = \frac{1}{36V^2}$, en effet, pour un matériau donnée, plus l'élément est petit au repos, plus celui-ci est rigide.

Les fonctions d'interpolation de cet élément sont données en annexe A.2 et le calcul de son volume en annexe B.1.

3.1.4 Particules

Il a été intéressant d'ajouter un paramètre pour définir un matériau : la densité, exprimée sans unité. Elle permet de placer intuitivement les masses ponctuelles liées à chaque noeud. En acceptant $d = \frac{m}{V}$, il suffit de calculer le volume V de chaque élément, puis sa masse $m = dV$ et de la répartir uniformément sur chaque noeud $m_{noeud} = \sum_{\forall elements} \frac{m_{element}}{nb\,noeuds}$.

3.1.5 Calcul de la rotation

Nous avons choisi de traiter des élements finis explicites élément par élément. Pour gérer les grands déplacements, il est nécessaire de trouver la rotation subie par chaque élément.

Il n'existe pas une seule rotation par élément, ce qui est important, c'est d'en trouver une qui permet de mesurer des petits déplacements. La rotation donnée par la décomposition polaire dans la méthode de Etzmuß est celle qui permet de trouver les plus petits déplacements, donc les plus petites déformations, celles qui approcheront au mieux la vraie loi de comportement.

Nous présentons une autre rotation. Notre rotation ne donne pas les plus petits déplacements, mais donnera quand même des petits déplacements. Nous verrons dans la partie suivante si cette approximation est regrettable en terme de précision.

Nous avons deux manières différentes de trouver cette rotation, une façon géométrique et une façon plus analytique. Dans les deux cas, l'odre d'énumération des points des tetraèdres est important et doit toujours être le même, chaque sommet est donc numéroté pour un élément donné.

Tout d'abord, nous allons étudier les deux méthodes d'extraction de la rotation. Un exemple de calcul pour chacune des version est présenté annexe C.

3.1.5.1 Version géométrique

Cette version peut faire penser à celle de Müller (paragraphe 2.4.5.1), car elle utilise des repères orthonormés pour calculer la rotation. Mais le principe est différent, car on travaille sur les éléments, ce qui va nous garantir une somme des forces nulles, contrairement à Müller qui travaille sur les sommets.

La première idée consiste à créer un repère pour chaque élément. Pour cela, on utilise un sommet du tetraèdre comme le centre du repère et une arète comme le premier axe du repère. Les autres axes sont déduits par construction d'un repère orthonormé direct. Bien sûr, pour un élément, on doit toujours utiliser le même sommet-centre et la même arète-abscisse.

Grâce à ce repère, on connait la rotation qui ramène les coordonnées globales des sommets, aux coordonnées locales de l'élément.

En connaissant les coordonnées locales de l'élément au repos et celles de l'élément déformé, on peut en déduire un déplacement dans le même état de rotation, à partir duquel on peut calculer les déformations, puis les forces en chacun des sommets, exercées par la déformation de l'élément.

En considérant un tetraèdre formé des points de positions a, b, c et d définies dans le repère global, et une rotation $R^{(local)}_{(global)}$ autour du point de position a amenant un point défini dans le repère $local$de l'élément vers le repère $global$,

$$R^{(local)}_{(global)} = \left[\ (1) \quad (2) \quad (3)\ \right]$$

$$(1) = \frac{b-a}{\|b-a\|} \qquad (3) = \frac{ab \wedge ac}{\|ab \wedge ac\|} \qquad (2) = \frac{ab \wedge (3)}{\|ab \wedge (3)\|}$$

Pour ramener les points connus en coordonnées globales dans les coordonnées locales grâce à la rotation :

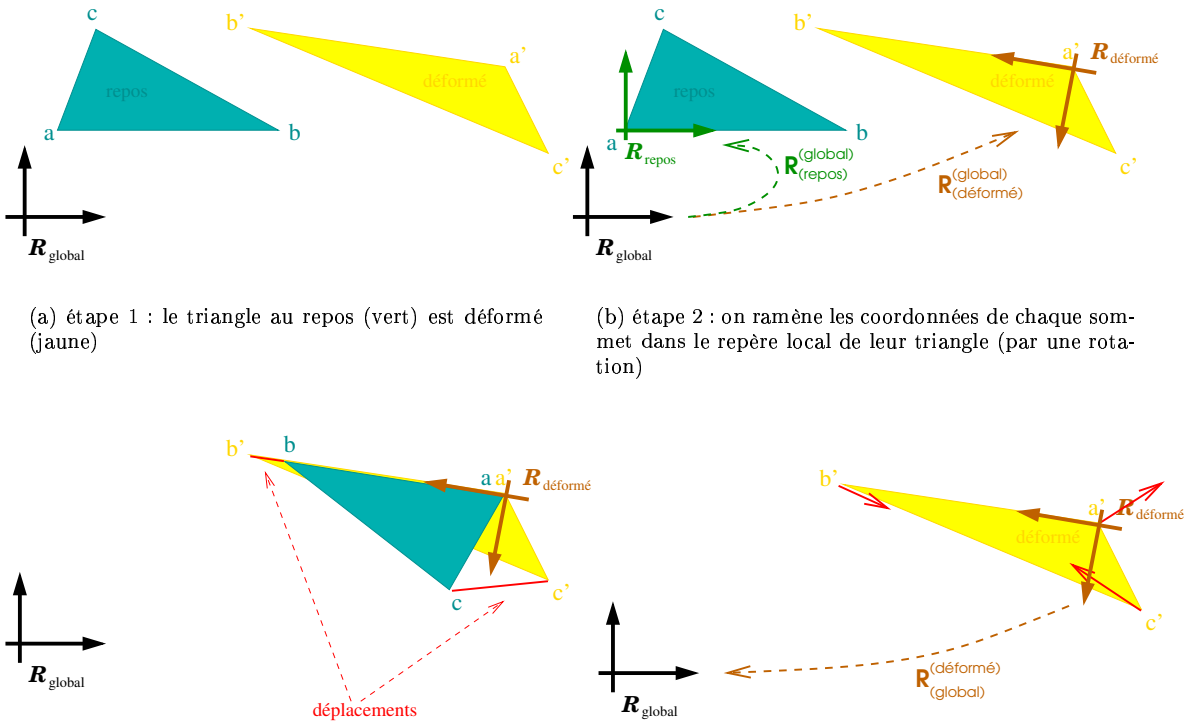
$$x^{(local)} = R^{(global)}_{(local)} x^{(global)} \quad avec \quad R^{(global)}_{(local)} = \left(R^{(local)}_{(global)}\right)^T \tag{3.1}$$

où $x^{(j)}_i$ est la position du noeud x à l'état i exprimée dans le repère j.

Notre calcul va donc devenir de la forme :

$$\begin{aligned} f &= RR^{(deforme)}_{(global)} B^T D B (X^{(repos)}_{repos} - X^{(deforme)}_{deforme}) \\ &= RR^{(deforme)}_{(global)} B^T D B (RR^{(global)}_{(repos)} X^{(global)}_{repos} - RR^{(global)}_{(deforme)} X^{(global)}_{deforme}) \end{aligned} \tag{3.2}$$

où $RR^{(j)}_{(i)}$ est une matrice contenant en diagonale la rotation $R^{(j)}_{(i)}$ du repère j vers le repère i et $X^{(j)}_i$ est un vecteur contenant les positions des sommets du tetraèdre à l'état i dans des



(a) étape 1 : le triangle au repos (vert) est déformé (jaune)

(b) étape 2 : on ramène les coordonnées de chaque sommet dans le repère local de leur triangle (par une rotation)

(c) étape 3 : avec ces coordonnées locales, on peut mesurer des petits déplacements. On calcule la nouvelle matrice de rigidité de l'élément et les forces dans ce repère local.

(d) étape 4 : les forces calculées dans le repère déformé sont ramenées dans le repère global

FIG. 3.1 – Version géométrique du calcul des petits déplacements

coordonnées du repère j.

$$RR_{(i)}^{(j)} = \begin{bmatrix} R_{(i)}^{(j)} & 0 & 0 & 0 \\ 0 & R_{(i)}^{(j)} & 0 & 0 \\ 0 & 0 & R_{(i)}^{(j)} & 0 \\ 0 & 0 & 0 & R_{(i)}^{(j)} \end{bmatrix}$$

$$X_i^{(j)} = \begin{bmatrix} a_i^{(j)} \\ b_i^{(j)} \\ c_i^{(j)} \\ d_i^{(j)} \end{bmatrix}$$

La rotation $R_{(repos)}^{(global)}$ peut être pré-calculée, et on peut, avant de commencer l'animation, déjà avoir les coordonnées locales des sommets de l'élément au repos $X_{repos}^{(repos)}$.

3.1.5.2 Version analytique

Une autre manière de trouver la même rotation, est de considérer la matrice de passage (à une translation près) de l'élément déformé vers l'élément au repos comme le fait Etzmuß. On peut en extraire à la fois une transformation rigide et une autre transformation. On va faire en sorte

que la transformation rigide corresponde à la rotation recherchée, et l'autre transformation à ses déformations.

Dans notre méthode nous allons adopter une analyse similaire, mais en ne passant pas directement de l'élement déformé vers l'élement au repos, mais en passant par un élément unitaire. Celui-ci va nous permettre d'effectuer des étapes en précalcul et va nous permettre d'avoir directement la rotation qui nous intéresse ($R^{(deforme)}_{(global)}$).

Pour trouver la déformation de l'élément déformé vers l'élément au repos, on passe par un élément unitaire. Pour un tetraèdre quelconque, la matrice A de passage vers le tetraèdre unitaire (représenté figure 3.2) est :

$$A = \left[\begin{array}{ccc} (b-a) & (c-a) & (d-a) \end{array} \right]$$

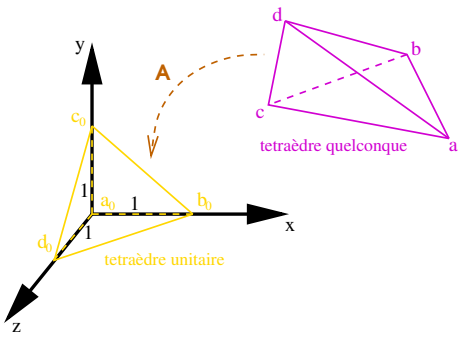
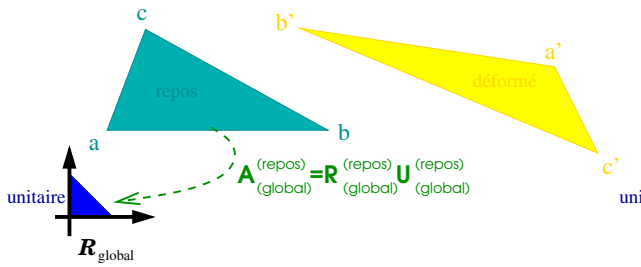
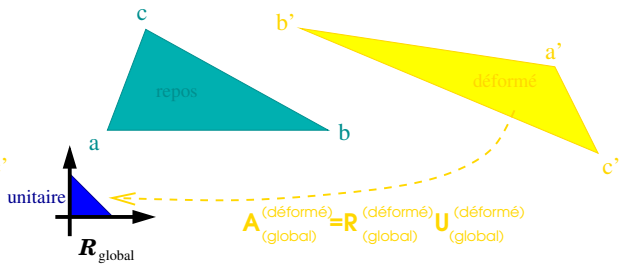


FIG. 3.2 – Transformation d'un tetraèdre quelconque vers le tetraèdre unitaire

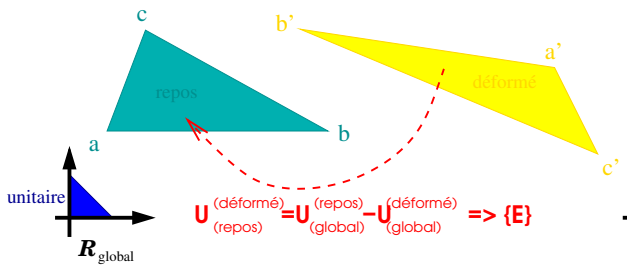
On décompose la transformation A en une rotation R et une déformation pure U, $A = RU$, en suivant les étapes présentées figure 3.3.



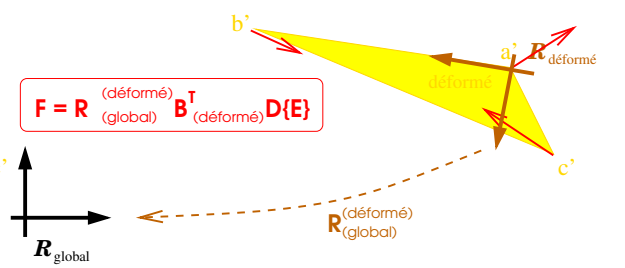
(a) étape 1 : on précalcule la déformation de l'élément au repos vers l'élément unitaire



(b) étape 2 : on calcul la rotation et la déformation de l'élément déformé vers l'élément unitaire



(c) étape 3 : on obtient la déformation entre l'élément déformé et l'élément au repos en passant par l'élément unitaire



(d) étape 3 : on calcul la matrice B et les forces dans le repère local de l'élément déformé. Il faut ensuite ramener les forces dans le repère global.

FIG. 3.3 – Version analytique du calcul des petits déplacements

Un avantage de la version analytique est d'avoir directement les déformations, sans passer par les déplacements, la matrice de rotation sert seulement à ramener les forces, calculées dans le repère local, vers le repère global.

Notre formule revient donc à :

$$f = R^{(deforme)}_{(global)} B^T D \{\varepsilon\}$$

Par contre, un inconvénient de la version analytique est que pour pouvoir calculer la tranformation J, puis la rotation R, il faut que le tétraèdre ne soit pas dégénéré, c'est à dire, que les 4 sommets ne soient pas dans le même plan (qu'il ne soit pas applati). Sinon la déformation U est une matrice singulière et le calcul de R est impossible (comme en MEF classique qui nécessite l'inversion de la matrice).

Il faut aussi noter que si cette transfomation existe en cas général pour des tetraèdres, elle n'existe pas pour des éléments à plus de noeuds.

Pour extraire la rotation et la déformation de cette matrice de passage, Etmuß utilise une décomposition polaire, lourde en calcul. Nous proposons d'utiliser la factorisation de Cholesky beaucoup plus rapide.

Algorithme 2 Version analytique utilisant la décomposition de Cholesky

EXTRACTIONRU (**DONNÉE :** *etat* , **RÉSULTATS :** $R^{(etat)}_{(global)}$, $\varepsilon^{(etat)}_{(global)}$)

| A est la matrice de passage de l'élément dans un état vers l'élément unitaire
| Décomposer A en une rotation R et une déformation pure U : $A = RU$
| En utilisant la décomposition de Cholesky :
| 1. Notons que $A^TA = U^TR^TRU = U^TU$
| 2. Pour A carrée définie positive, par Cholesky :
| $U^TU = A^TA = LL^T$ où L est triangulaire inférieure
| 3. En fixant U triangulaire supérieure :

$$U = L^T = \begin{bmatrix} 1+\varepsilon_x & \gamma_{xy} & \gamma_{xz} \\ 0 & 1+\varepsilon_y & \gamma_{yz} \\ 0 & 0 & 1+\varepsilon_z \end{bmatrix} \Rightarrow \varepsilon = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{pmatrix}$$

| 4. $R = AU^{-1}$

Preuve :
Vérifions que R est bien une rotation, commençons par regarder si $RR^T = R^TR = I$:
• $RR^T = AU^{-1}(AU^{-1})^T = AU^{-1}U^{-T}A^T = A(U^TU)^{-1}A^T = A(A^TA)^{-1}A^T = AA^{-1}A^{-T}A^T = I$
• $R^TR = (AU^{-1})^TAU^{-1} = U^{-T}A^TAU^{-1} = U^{-T}U^TUU^{-1} = I$
R est soit une rotation soit une symétrie.
Supposons que R est une symétrie, alors U contient des termes négatifs. Or U, triangulaire, est construit par décomposition de Cholesky donc tous ses termes diagonaux sont positifs $\Rightarrow$ contradiction. Donc R est une rotation.

PROGRAMME PRINCIPAL

Précalul :
| Calculer la matrice de rigidité du matériau : $D = fonction(E, \mu, V)$
| ExtractionRU(repos, $R^{(repos)}_{(global)}$, $\varepsilon^{(repos)}_{(global)}$)

A chaque pas de temps :
| ExtractionRU(déformé, $R^{(deforme)}_{(global)}$, $\varepsilon^{(deforme)}_{(global)}$)
| Ramener les coordonnées des points déformés dans leur repère local :
| $$x_{(deforme)} = R^{(global)}_{(deforme)} x_{(global)}$$
| Calculer la matrice de rigidité de l'élément dans le repère local :
| $$B_{(deforme)} = fonction(x_{(deforme)})$$
| Calculer les forces dans le repère global :
| $$F = R^{(deforme)}_{(global)} B^T D(\varepsilon^{(repos)}_{(global)} - \varepsilon^{(deforme)}_{(global)})$$

3.1.6 Matrice de rigidité

Dans cette partie, nous allons montrer la nécessité de recalculer la matrice de rigidité.

3.1.6.1 Description

La rigidité K dépend à la fois du matériau D et de l'élément B ($K = B^T DB$). La matrice de rigidité du matériau D, ne dépendant que du matériau et de la géométrie au repos, peut être précalculée. La rigidité de l'élément B dépend de sa forme, son calcul revient à calculer les fonctions d'interpolations comme on a vu précedemment (paragraphe 2.4.1.2). Lors de déformations, cette interpolation varie, il est donc important de la recalculer à chaque pas de temps pour chaque élément.

Classiquement, le calcul de cette matrice prend un peu de temps, mais grâce à des simplifications apportées par notre rotation et aussi par l'application d'une translation, il est possible de l'accélérer. C'est à cause des cette lourdeur des calcul qu'actuellement, aucune méthode temps-réel (masses-tenseurs non-linéaires, méthode de Müller/Etzmuß) ne peut se permettre de recalculer ces matrices de rigidités, et qu'elles sont précalculées sur l'élément au repos.

3.1.6.2 Moments fantômes

Nous montrons ici que l'utilisation d'une matrice précalculée introduit des moments fantômes.

Cette simplification, si elle n'entraîne pas de forces fantômes, peut entraîner des moments fantômes. En effet, même si la somme des forces est nulle sur un élément, il se peut que celui-ci subisse un couple, qui dépend de la direction des forces par rapport son orientation, le système n'est alors plus physiquement correct. Dans le cas où les forces sont calculées avec une ancienne matrice de rigidité de l'élément, c'est comme si les forces sont calculées sur l'élément au repos. Donc les forces fournies donnent un moment nul sur l'élément au repos, mais ne vont pas donner le même moment sur le triangle déformé (figure 3.4).

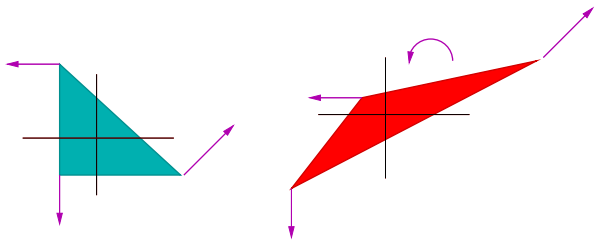


FIG. 3.4 – Les mêmes forces appliquées sur des éléments différents ne donnent pas les mêmes moments. Sur le triangle vert, le moment est nul alors que le rouge va tourner.

Müller est conscient du problème, mais ne se soucie pas vraiment de ces moments, puisque son but n'est pas de faire le modèle le plus précis possible. Il précise qu'il n'a aucun souci jusqu'à 25% de déformation.

Grâce au calcul de la matrice de rigidité à chaque déformation, notre modèle va forcément gagner en précision. En tout cas, les moments étants nuls, ils garantissent une certaine réalité physique. De plus, même si pour des grandes déformations les résultats vont être imprécis, notre modèle ne va avoir aucun moment fantôme, il est alors possible d'étirer un élément de plusieurs fois sa taille initiale sans instabilité !

3.1.6.3 Méthode de calcul

Les fonctions d'interpolation sont invariantes aux translations, mais dans le souci de simplifier leurs calculs, on va quand même éliminer les déplacements en translation subis par les éléments.

On aura ainsi pour centre des repères locaux les premiers sommets des éléments. Le formule 3.1 devient :

$$x_{etat}^{(local)} = R_{(local)}^{(global)}(x_{etat}^{(global)} - a_{etat}^{(global)})$$

Le calcul des forces 3.2 devient :

$$f = RR_{(global)}^{(deforme)} B^T DB(RR_{(repos)}^{(global)}(X_{repos}^{(global)} - A_{repos}^{(global)}) - (RR_{(deforme)}^{(global)}(X_{deforme}^{(global)} - A_{deforme}^{(global)})))$$

où le vecteur $A_i^{(j)}$ contient une succession de la position du premier sommet a dans l'état i exprimée dans le repère j, afin d'appliquer une translation de a vers l'origine à tous les sommets.

On sait alors que les coordonnées dans le repère local du premier point a sont nulles.

De plus, par construction de nos rotations, on sait que le deuxième sommet b se trouve sur l'axe $\overrightarrow{Ox}$, que le troisième point c est dans le plan $(\overrightarrow{Ox}, \overrightarrow{Oy})$, et que le dernier point d est libre dans les trois directions x, y, et z. Grâce à ses nombreux termes nuls, le calcul de la matrice de rigidité du tétraèdre (cf annexe A.2) se trouve simplifié. Il est alors possible de la recalculer à chaque pas de temps pour un coût faible. Les calculs simplifiés des facteurs des fonctions de forme du tetraèdre du type $N_i = a_i + b_i x + c_i y + d_i z$ sont les suivant :

$$\begin{array}{cccc} b_1 = -c_y d_z & b_2 = -c_y d_z & b_3 = 0 & b_4 = 0 \\ c_1 = (c_x d_z) - (b_x d_z) & c_2 = c_x d_z & c_3 = d_z b_x & c_4 = 0 \\ d_1 = c_y d_x - c_x d_y + b_x d_y - b_x c_y & d_2 = c_y d_x - c_x d_y & d_3 = -d_y b_x & d_4 = -b_x c_y \end{array}$$

3.1.7 Conclusion

La nouvelle méthode présentée est une méthode explicite. Elle permet de calculer des forces pour des grands déplacements tout en conservant un tenseur de déformations linéaire.

La grosse nouveauté se situe au niveau de la prise en compte des rotations locales. Nous avons présenté deux méthodes plus rapides pour ce faire : une version géométrique et une analytique.

L'approche analytique est peut être un tout petit peu plus rapide, mais elle a un gros inconvénient, celui de devoir calculer l'inverse de la matrice de déformation. Dans certains cas de déformation, ce n'est pas possible, la méthode n'est donc plus en mesure de fournir les forces. De plus cette version ne fonctionne, à première vue, que sur des tetraèdres linéaires à quatre noeuds. On préférera donc la méthode géométrique qui fonctionne quel que soit l'état du système et quel que soit le type d'éléments.

Un autre gros avantage de notre méthode, est le calcul très rapide de la matrice de rigidité à chaque pas de temps, afin d'assurer un maximum de précision ; cette matrice était toujours approximée en précalcul sur l'élément au repos dans les autres méthodes rapides déjà existantes.

De plus, notre méthode est très stable. La stabilité d'un modèle dépend en grand partie de l'état d'équilibre des éléments, pour lequel ils ne doivent ni subir de couple, ni de force. La mise à jour de la matrice de rigidité assure une somme des moments nulle, quelque soit l'état de déformation. La somme des forces nulle est assurée par le calcul linéaire classique des forces, appliqué en même temps sur tous les sommets d'un élément.

Le modèle est stable, la contrainte de vitesse est bien respectée, et celle de précision est prise en compte. Nous désirons maintenant évaluer notre méthode dans un contexte prédictif, alors qu'elle n'a jusqu'à présent été appliquée qu'à du divertissement.

Chapitre 4

Résultats

Dans le chapitre précédent, nous avons présenté une nouvelle méthode de calcul des forces. Dans certains domaines, comme la simulation chirurgicale, la précision est capitale, nous allons donc tout d'abord tester la précision de notre modèle avec des validations qualitatives et quantitatives, en le comparant avec d'autres méthodes numériques, puis avec une solution analytique dans le cas de la poutre encastrée et enfin avec des résultats expérimentaux. Ensuite, nous allons voir jusqu'à quel point il peut être intéractif en testant sa vitesse.

4.1 Précision

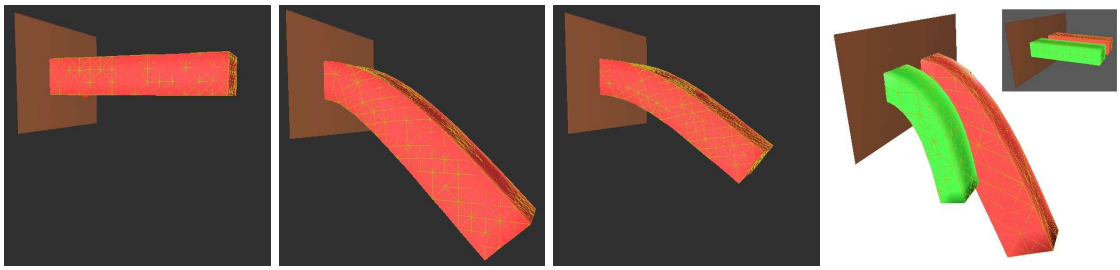
Il est très difficile de bien tester la précision de seulement le modèle mécanique, car divers facteurs importants interviennent, comme le maillage, l'intégration dynamique ou encore le réglage des paramètres. Nous essayerons autant que possible d'effectuer les comparaisons avec les mêmes facteurs, dans les mêmes conditions.

4.1.1 Premiers résultats

4.1.1.1 Grands déplacements

Notre modèle doit être capable de calculer efficacement les forces à l'aide d'un tenseur de déformations linéaire, sans qu'il n'y ait les artefacts induits par le linéaire, que sont les « gonflements » dans le cas de grands déplacements et particulièrement de rotations.

Nous avons testé des méthodes sur un cas d'école très classique, qui s'appelle problème de la poutre encastrée (cf figure 4.1).



(a) au départ (b) sans gestion des grands déplacements (c) avec gestion des grands déplacements (d) gestion des grands déplacements : en vert = avec, en rouge = sans

FIG. 4.1 – Une poutre (même maillage, mêmes paramètres) est fixée à un mur, elle est lachée de l'état horizontal et est seulement soumise à son poids

Nous avons donc comparé visuellement nos résultats avec ceux d'une méthode linéaire classique. On voit bien que sans la gestion des grands déplacements, la barre gonfle énormément, et n'a pas du tout une déformation naturelle, alors qu'avec notre méthode, les déformations sont visuellement correctes.

Ensuite, pour aller plus loin que l'impression visuelle, il est important de vérifier que le volume ne varie pas trop quelques soient les déformations, même si dans un corps élastique, il n'y a pas une conservation exacte du volume. Il peut être intéressant de faire une moyenne de tous les volumes à chaque pas de temps de l'état de départ, jusqu'à l'état d'équilibre, pour vérifier que lors de l'intégration dynamique, on reste aussi correct.

SOMME DES VOLUMES DES ÉLÉMENTS FORMANT LA POUTRE	**au repos**	**à l'état d'équilibre**	**moyenne**
sans gestion des grands déplacements	360	1022	794
avec gestion des grands déplacements	360	362	361

On voit bien que les gonflements disparaissent avec notre méthode.

4.1.1.2 Grandes déformations

Même si notre modèle dans sa version actuelle n'est pas conçu pour gérer les grandes déformations (mail seulement les grands déplacements), il peut être intéressant de voir jusqu'à quel niveau de déformation, on reste linéaire. Pour cela, on va tester la force exercée en un point d'un triangle unitaire pour un déplacement connu comme indiqué sur la figure 4.2.

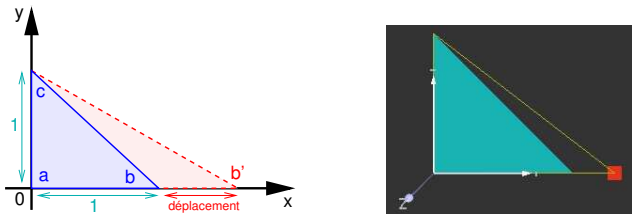
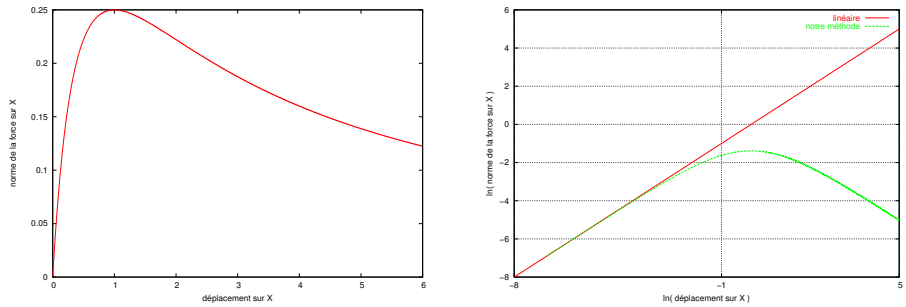


FIG. 4.2 – Le triangle de test. On bouge le point b sur l'axe des abscisses.



(a) en échelles linéaires (b) en échelles logarithmiques

FIG. 4.3 – Comportements de la force exercée au point b en fonction de son déplacement

On voit sur la courbe 4.3 (a) qu'a partir d'une extension de 100% (X=1 en partant avec un triangle de longueur 1), le système donne des résultats non intuitifs, puisque la force diminue alors que la déformation augmente. Mais rien de dramatique, car il existe peu (pas ?) de matière telle quelle. Il ne faut pas oublier non plus que les grandes déformations ne sont pas gérées. Au delà, il faudrait sûrement générer des cassures, ou des déchirures. De plus, notons que les lois de comportement en grandes déformations sont encore mal connues, surtout dans le cas tridimensionnel.

Le courbe 4.3 (b) montre que notre modèle reste linéaire et est valide pour des déplacements jusqu'à $\ln u = -1$, soit $u = 0.37$, donc environ 37% d'extensibilité. Ces résultats sont très valables, puisque de toute façon, au delà (même avant ça), une vraie loi de comportement n'est plus linéaire.

Grâce au calcul de la matrice de rigidité à chaque pas de temps, le modèle a une très grande stabilité, même en très grandes déformations (même si dans ce cas, les résultats fournis sont imprécis).

FIG. 4.4 – De très grandes déformations (repos en bleu, déformé en rouge)

4.1.2 Comparaison avec d'autres solutions numériques

Bien que la méthode se comporte visuellement correctement dans le cas de grands déplacements, il nous faut vérifier à quel niveau les résultats se rapprochent de la réalité.

Nous avons tout d'abord voulu comparer notre méthode avec d'autres méthodes issues de l'ingénierie mécanique. La nouveauté apportée se situant au niveau des grands déplacements, on désire la comparer avec des méthodes classiques non-linéaires reconnues pour bien gérer les grands déplacements, et habituellement utilisées dans de tels cas. Le logiciel de comparaison choisi est Ansys, logiciel de référence des éléments finis. La méthode de comparaison est une méthode statique utilisant le tenseur de déformations non-linéaire de Green-Lagrange et une loi de comportement élastique linéaire. La seule différence avec notre méthode est donc bien au niveau du tenseur de déformation.

Pour celà, on reste toujours sur le cas d'une poutre encastrée. Pour pouvoir comparer les méthodes, on mesure les différences de flèches (déplacements) en certains points, en l'occurence tous les noeuds, avec une distance Euclidienne.

Bien sûr pour pouvoir comparer les méthodes, il est nécessaire d'utiliser les mêmes maillages et le même pararamétrage. Pour cela, on a maillé, en tetraèdres, un parallélépipède grâce au mailleur d'Ansys, qui sera donc parfaitement adapté à Ansys.

écart moyen	0.86 mm
écart maximum	1.55 mm
pourcentage d'erreur moyenne	18 %

L'écart moyen (en distances euclidiennes) obtenu entre les positions d'équilibre fournies par Ansys et celles fournies par notre méthode à l'état d'équilibre est de 18% en moyenne, soit 0.86mm pour une poutre de 150 * 40 * 30 mm^3, un module d'Young de 2000 Pa et un coefficient de Poisson de 0.4.

L'écart peut sembler grand en pourcentage, mais en fait il est assez faible, car là où les déplacements sont très petits, la marge d'erreur prend une place très importante dans le pourcentage. Nos résultats semblent donc cohérents. Surtout que pour la simulation, si le même maillage était utilisé, dans Ansys, il y a seuleument des tétraèdres à 10 noeuds et donc interpolés par une fonction quadratique. Ils réagissent donc mieux aux grandes déformations. Dans notre méthode, seuls les tétraèdres 4 noeuds linéaires étant implémentés, ce qui implique une très légère différence lorsque les déformations augmentent.

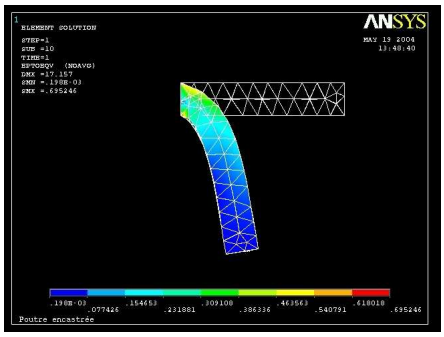


FIG. 4.5 – Résultats sous ansys : la poutre à l'état au repos, et à l'état d'équilibre
(L'échelle de couleur indique le pourcentage de déformation, dans cette simulation, certains éléments sont à plus de 40% de déformation, ce qui est beaucoup)

La conlusion de ce test est que le fait d'éliminer les déplacements rigides avant d'utiliser un calcul linéaire en petits déplacements semble correspondre à des résultats utilisant un calcul non-linéaire.

De plus, plusieurs tests avec différents paramétrages de matériau semblent indiquer que le réglage de nos paramètres correspond à celui connu et utilisé. C'est un point très intéressant, car on sait qu'il est possible d'utiliser des modules d'Young et des coefficients de Poisson réels, offerts par la mécanique des matériaux.

4.1.3 Comparaisons avec des solutions analytiques

4.1.3.1 Problème de la poutre encastrée

Le problème étudié concerne une poutre attachée d'un côté et soumise à sa gravité. Cet exemple consiste à considérer une poutre comme un corps solide élastique uniforme et de l'analyser en supposant un état de contraintes planes.

La poutre étudiée est représenté sur le schéma 4.6. l est la longueur de la poutre, H sa hauteur et T son épaisseur, ρ est sa masse volumique, g la gravité.

Ici on reste dans un cas 1D, donc tous les points comparés se trouveront sur la ligne moyenne, mais l'extension au 2D ou au 3D est intuitive.

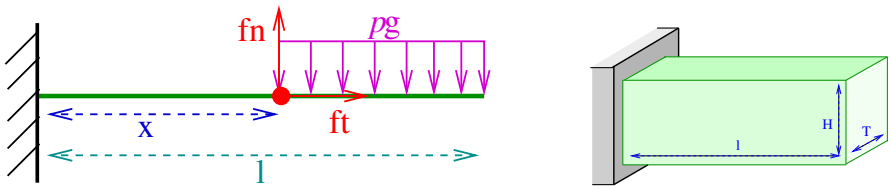


FIG. 4.6 – La poutre encastrée analysée

Pour calculer la flèche d'un point de coordonnée x, on regarde ce qu'il se passe à l'extrémité de la poutre à partir de x. Au point x, il s'exerce une force décomposable en une composante tangentielle $f_t$et une composante normale f_n.

A l'équilibre, la somme des forces est nulle :

$$f_n + f_t + \rho g(l - x) = 0$$

or $f_t = 0$ car aucune force ne vient comprimer, ni allonger la poutre, donc

$$f_n = -\rho g(l - x)$$

On en déduit le moment exercé :

$$M = -\rho g(l - x)\frac{(l - x)}{2} = -\rho g\frac{(l - x)^2}{2}$$

car la gravité s'appliquant uniformément sur tout le volume, on peut admettre qu'elle correspond à une force s'appliquant au milieu du volume.

L'équation différentielle de la théorie des poutres est :

$$EI\frac{\partial^2 u}{\partial x^2} = -M(x)$$

où E est le module d'Young, soit la rigidité du matériau, et I est le moment d'inertie de la poutre, avec $I = \frac{TH^3}{12}$ dans le cas d'une poutre rectangulaire, fonction de la surface de coupe.

On en déduit la pente :

$$\frac{\partial^2 u}{\partial x^2} = \frac{-1}{EI}(\rho g\frac{(l - x)^2}{2})$$

En intégrant, on obtient la pente :

$$\frac{\partial u}{\partial x} = \frac{-1}{EI}(-\rho g\frac{(l - x)^3}{6}) + a$$

En intégrant, on obtient la flèche :

$$u = \frac{-1}{EI}(\rho g \frac{(l-x)^4}{24}) + ax + b$$

La poutre étant fixée en $x = 0$, on a les conditions cinématiques suivantes

$$\frac{\partial^2 u}{\partial x^2}(0) = 0$$

et

$$\frac{\partial u}{\partial x}(0) = 0$$

d'où on déduit a et b :

$$a = \frac{1}{EI}(-\rho g \frac{l^3}{6})$$

$$b = \frac{1}{EI}(\rho g \frac{l^4}{24})$$

4.1.3.2 Résultats

J'ai comparé mon approche avec des résultats analytiques sur une poutre de longueur 40, largeur 4 et hauteur 2, avec un matériau modélisé par un module d'Young à 200000 et une masse volumique de 1.

x	**0**	**5**	**10**	**15**	**20**	**25**	**30**	**35**	**40**		
flèche analytique	0	0.17	0.62	1.28	2.08	2.98	3.93	4.91	5.89		
flèche numérique	0	0.12	0.44	0.92	1.5	2.14	2.82	3.52	4.22		**moyenne**
différence	0	0.05	0.18	0.36	0.59	0.84	1.11	1.39	1.67		0.69

Vu la taille de la poutre, les résultats sont encourageants. Il faut tout de même savoir que la comparaison directe des valeurs est un peu faussée, entre autre parce que la théorie des poutres ne prend pas en compte l'imcompressibilité du matériau (coefficient de Poisson), dans mon test je l'ai mis à 0.3.

Ce qui est surtout intéressant, c'est de comparer la courbe formée par la flexion de la poutre. On sait qu'analytiquement, on a un polynôme de degré 4, on peut regarder ce qu'il en est pour notre méthode avec les résultats précédents. Pour cela, on va comparer les résultats en échelles logaritmiques pour différentes valeurs du coefficient de poisson.

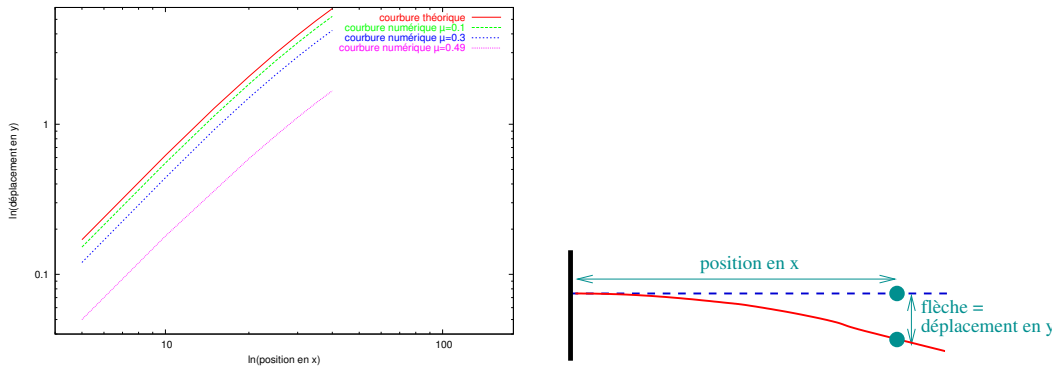


FIG. 4.7 – Comparaison des flèches par rapport aux abscisses en échelles logarithmiques

On voit très bien sur le graphe 4.7 que toutes les courbes sont quasi-parallèles, donc leurs coefficients directeurs sont très proches, donc elles suivent une loi de même degré. Ce résultat est très valorisant, il prouve que notre méthode suit un comportement théorique valable, et ce quelque soit le paramétrage.

Plus le coefficient de Poisson est petit, plus on remarque que la courbure est proche de la théorie, ce qui rejoint parfaitement le fait que l'imcompressibilité du matériau n'est pas prise en compte dans cette approche analytique.

4.1.4 Comparaison avec le truth cube

4.1.4.1 Le truth cube

Le truth cube est une étude menée à l'université de Harvard [27-KER03]. Elle permet de tester la précision de méthodes numériques dans le cas de modèles déformables. Elle a été crée dans l'idée de tester les méthodes utilisées dans le cadre de simulations chirurgicales, donc ce test nous intéresse particulièrement.

Le truth cube est un cube de silicone de 8*8*8 cm^3 dont les propriétés du matériau sont connues (module d'Young = 14.9 kPa et coefficient de Poisson = 0.5). A l'intérieur de ce cube sont disposées des petites billes de métal à intervalles réguliers (cf figure 4.8).

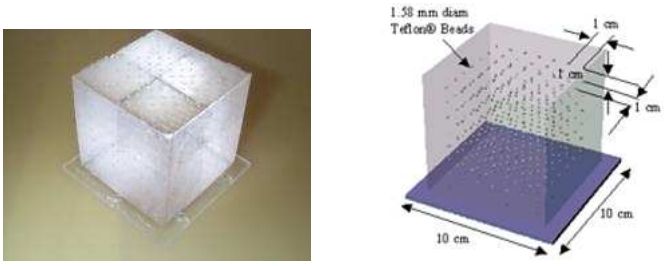


FIG. 4.8 – Le truth cube et ses caractéristiques

4.1.4.2 L'expérience

Des contraintes connues (intensité et direction) sont appliquées, à l'aide d'un dispositif contrôlé en force, sur ce cube. Les déformations sont enregistrées sous imagerie scanner, pour permettre de mesurer le déplacement de chaque bille à l'état stable après l'application de diverses forces (cf figure 4.10).

L'expérience la plus facile à simuler est celle consistant à effectuer une pression sur toute la partie supérieure du cube, comme indiqué dans la figure 4.9.

Les résultats sont pris en compte à trois niveaux de compression : 5%, 12,5% et 18.25%. On approche donc les grandes déformations.

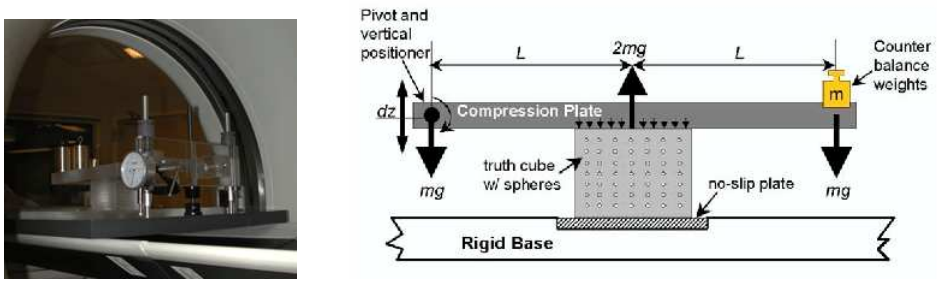


FIG. 4.9 – L'expérience de compression

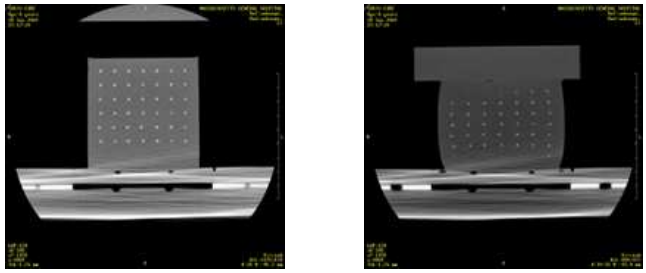


FIG. 4.10 – Visualisation en images scanner de l'expérience de compression pour récupérer la position 3D des billes (au repos et 5% de compression)

4.1.4.3 La simulation

Cette expérience va être simulée avec notre méthode, afin de comparer nos résultats avec les données expérimentales.

Pour cela, on va faire quelques hypothèses. On s'accorde à dire que les petites billes ne changent pas assez les propriétés du matériau pour quelles soient prises en compte lors de la simulation. De plus, le matériau ne peut pas être modélisé par un coefficient de Poisson à 0.5 (analytiquement, cela crée une division par 0), on va donc utiliser 0.499 (cela ne sert à rien de prendre un chiffre plus grand qui créera des imprécisions numériques).

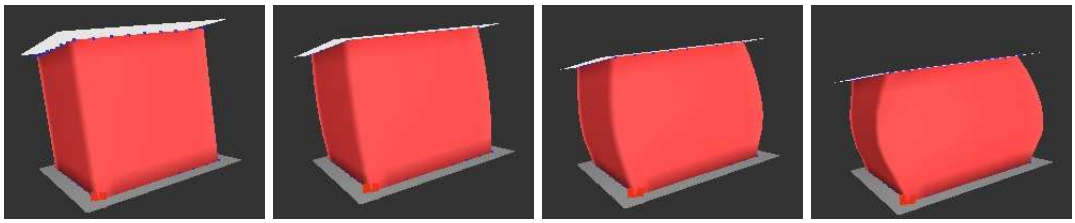
Il a fallu construire un maillage à partir des positions des billes repos. Entre chaque bille, le cube à été divisé en petits cubes maillés en 10 tetraèdres afin de conserver une symétrie (cf annexe B.2).

4.1.4.4 Les résultats

En sus des résultats fournis par les images scanner, sont donnés des déplacements calculés à partir d'une autre méthode numérique d'éléments finis classiques, tenseur de déformations non-linéaire et loi de comportement linéaire élastique (logiciel : ABAQUS v5.8). Cette simulation étant appliquée avec les mêmes hypothèses (billes négligées, mêmes valeurs du matériau) et un maillage hexaèdrique linéaire (hexaèdres à 8 noeuds) comparable à celui de notre simulation, ces résultats peuvent être pris en compte valablement dans la comparaison. C'est très intéressant, car dans cette méthode, c'est un tenseur des déformations non-linéaire qui est utilisé, avec une linéarité mécanique et des éléments linéaires.

On pourra donc comparer notre méthode avec une autre méthode qui est exactement celle qu'on essaye d'approcher linéairement, et le tout avec des données réelles.

Les résultats visuels de notre méthode sont donnés figure 4.11 pour chacun des trois niveaux de compression.



(a) au repos (b) 5% de compression (c) 12.5% de compression (d) 18.25% de compression

FIG. 4.11 – Simulation de l'expérience de compression avec notre méthode

Les comparaisons fournies, entre la réalité et leur MEF non-linéaire, ne prennent seulement en compte que les déplacements des billes d'une tranche médian. Nous allons donc effectuer cette même comparaison (resultats table 4.1), puis on prendra aussi en compte les déplacements calculés de la totalité des billes par rapport aux déplacements réels (resultats table 4.2).

taux de compression	**18.25%**	
réalité vs MEF non-linéaire	2.0 mm	2.5% d'erreur
réalité vs notre méthode	1.6 mm	2% d'erreur

TAB. 4.1 – Moyennes des longueurs euclidiennes entre les déplacements rééls et les déplacements calculés des billes pour une tranche médian seulement

taux de compression	**5%**	**12.5%**	**18.25%**
réalité vs notre méthode	0.85 mm	1.23 mm	1.7 mm
pourcentage d'erreur	1.06%	1.54%	2.13%

TAB. 4.2 – Moyennes des longueurs euclidiennes entre les déplacements réels et les déplacements calculés par notre méthode pour la totalité des billes

Les résultats sur cette expérience sont vraiment bons. On peux même voir sur la table 4.1 qu'ils semblent meilleurs que des résultats fournis par un tenseur de déformations non-linéaire. Il faut sûrement prendre ces résultats avec une précaution, avec l'idée que cette expérience doit remplir les conditions optimales pour le bon fonctionnement de notre modèle. En tout cas, on peut juger que notre modèle donne de bons résultats, proches de la réalité et pas du tout aberrants.

Il faut tout de même noter que la nouveauté du modèle qui est de gérer les grands déplacements, ne semble pas testée à fond dans cette expérience, en effet, les éléments ne semblent pas subir pas d'énormes rotations.

4.1.5 Comparaison avec des données réelles de patients dans le cadre de planification de chirurgie maxillo-faciale

Après avoir testé le modèle sur des cas « simples », il va être intéressant de voir comment il se comporte sur un modèle complexe. La difficulté ici est d'avoir des données réelles. Hors c'est possible dans le cadre d'opérations chirurgicales, où on peut prendre l'état du patient grâce à l'imagerie scanner, avant opération puis après opération. Le tout est de pouvoir reproduire fidélement l'opération. C'est d'autant plus vrai dans le cadre de chirurgie maxillo-faciale, où l'état

du patient après opération reste à peu près stable. C'est sur ce type d'intervention qu'on va mettre en oeuvre notre modèle.

L'opération chirurgicale que l'on a simulée consiste en un repositionnement de la mandibule : concrètement l'os de la machoire est reculé. Cette opération et la simulation qui en découle sont détaillées dans [28-CHA03].

Ce qui est intéressant, ce n'est pas de simuler cette opération avec notre méthode, car seul le résulat final est important, donc une résolution statique est plus adaptée ; mais l'intérêt réside dans le test de notre méthode, un test « grandeur nature ».

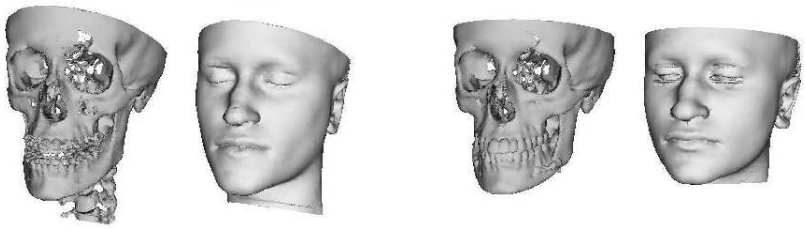
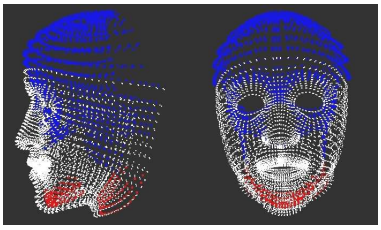
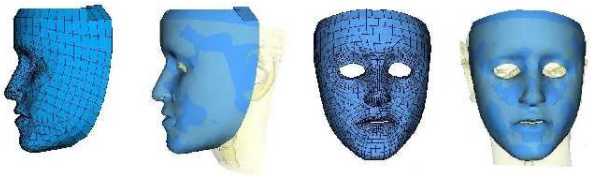


FIG. 4.12 – Les données réelles d'une patiente : état squelettique et cutané, à gauche pré-opératoire, à droite post-opératoire

Nous avons à notre disposition un maillage du visage du patient avant l'opération. Celui-ci à la base maillé en éléments hexaèdriques à été redivisé en tetraèdres (cf annexe B.2), on se retrouve au final avec un maillage de 47730 éléments, ce qui est très important.



(a) Les points bleu sont bloqués et représentent les os. Les points blanc sont la peau qui va subir ces conditions. Les points rouge représentent le recul (certains ne sont là que pour recaler la position du cou avec les données post-opératoires).



(b) Le maillage en éléments hexaèdriques, superposé aux données réelles

FIG. 4.13 – Le maillage pré-opératoire

Pour pouvoir simuler l'opération, certains noeuds dans la partie interne du maillage sont bloqués pour représenter l'attache de la peau aux os. Au contraire, afin de simuler le reculement mandibulaire, certains noeuds ont un déplacement imposé, calqué sur les mesures faites pendant la vraie opération. On simule l'opération à partir de ces conditions. On peut ensuite comparer les résultats calculés à la fois avec un modèle du patient après l'opération et avec d'autres méthodes numériques statiques actuellement utilisées pour effectuer cette planification.

La plus grosse difficulté a été de savoir quand considérer que l'animation été fini. En effet, puisqu'on ne résoud pas un problème en statique mais en dynamique, on ne sait pas avec exactitude quand la position d'équilibre est atteinte. On a donc pris l'état du système à plusieurs échéances

de plus en plus longues, en les comparant avec le modèle post-op, en regardant si elles évoluaient encore.

Après une nuit de calcul et plusieurs dizaines d'étapes, on se rend bien compte que la méthode tend vers les résultats post-op, mais très lentement. On voit peut être là une limitation des MEF explicites sur des très gros maillages, car la propagation est très lente. Il est possible d'accélérer un peu la propagation en augmentant considérablement le nombre d'itérations du gradient conjugué résolvant l'intégration du temps (de l'ordre de 300), on doit alors atteindre plus rapidement la solution, mais les calculs à chaque pas de temps sont beaucoup plus longs, la simulation perd donc tout côté intéractif. Dans tous les cas, cela reste trop lent. Pour un tel type de planification, il pourrait être intéressant de mettre en place notre approche dans une méthode statique.

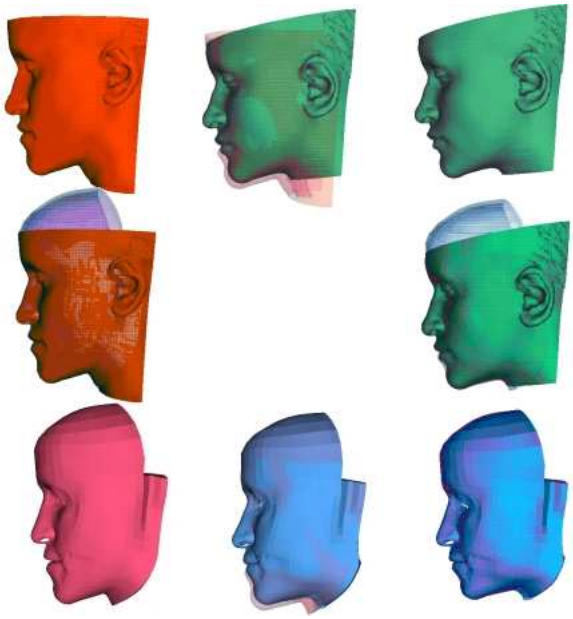


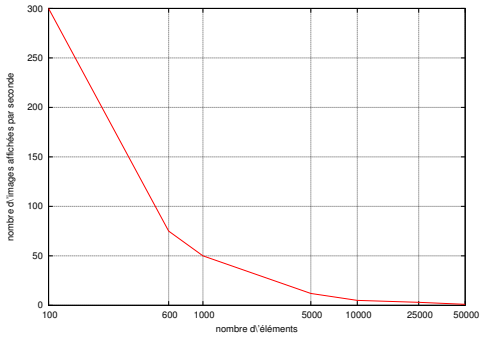
FIG. 4.14 – Les résultats de la simulation. A gauche, le pré-opératoire (rouge : données, rose : modèle), à droite, le post-opératoire (vert : données, bleu : simulation). Au milieux, on trouve une superposition des visages.

Visuellement on voit bien qu'il y a un reculement de la machoire, et que les résultats ressemblent à ceux des données post-opératoires, même si il y a des petits défauts au niveau des lèvres et du cou. Malheureusement par manque de temps, les résultats n'ont pas encore été testé quantitativement. Le test quantitatif fera parti des travaux prochains.

4.2 Vitesse

Les chiffres qui sont donnés ici même s'ils sont prometteurs résultent d'un code optimisable, on peut donc espérer les revoir à la hausse. Les tests ont été effectués sur un pc avec processeur pentium 4 2.4GHz, 512 Mo de RAM et carte vidéo GeForce 4 MX. Les fréquences fournies correspondent aux fréquences d'affichage à l'écran et contienne donc aussi les calculs nécessaires à l'affichage.

nb noeuds	44	132	198	756	1386	3636	6666
nb tetraèdres	100	600	1000	5000	10000	25000	50000
Hz	300	75	50	12	5	3	1



TAB. 4.3 – Vitesse de calcul en fonction du nombre d'éléments (pas de temps = 40 ms)

On peut se situer par rapport au niveau des autres méthodes de référence. La méthode de référence des simulateurs chirurgicaux est le système masses-tenseurs de Picinbono (paragraphe 2.4.4), et la référence de l'informatique graphique est celle de Müller avec son adaptation aux volumes de la décomposition polaire de Etzmuß (2.4.5.2).

MÉTHODE	**masses-tenseurs non-linéaires**	**décomposition polaire**	**notre méthode**		**masses-tenseurs linéaires**	**méthode linéaire** sans recalcul rigidité	
MACHINE	P3 500MHz	P4 1.8GHz	P4 2.4GHz		P3 500MHz	P4 2.4GHz	
NOMBRE D'ÉLÉMENTS	2000	1000	1000	2000	10000	1000	10000
FRÉQUENCE D'AFFICHAGE	25 Hz	30 Hz	50 Hz	40 Hz	25 Hz	53 Hz	6 Hz

TAB. 4.5 – Comparaison avec les méthodes de référence et les méthodes linéaires

On peut voir dans la première partie de la table 4.5 que les chiffres de notre méthodes sont du même ordre de grandeur que les masses-tenseurs non linéaires de Picinbono [9-PIC03] qui semblent plus rapides que la méthode de Etmuß/Müller [2-ETZ03, 10-MUL04] (par rapport à la puissance des machines), notre méthode tourne donc à la même vitesse que les méthodes de référence, alors qu'elle effectue des calculs en plus, pour mettre à jour la matrice de rigidité. Le contrat de rapidité est donc rempli.

Grâce à la deuxième partie du tableau, on peut voir que notre méthode a exactement les mêmes temps de calcul qu'une méthode avec un tenseur de déformations linéaire et sans la mise à jour de la matrice de rigidité. Nos ajouts (traitement des déplacements rigides et calcul de la rigidité) semblent donc quasiment gratuits. Quand on voit l'accélération que peut procurer la formulation en masses-tenseurs [23-DEL99] par rapport à une méthode linéaire calculée classiquement, on peut imaginer les résultats que cette formulation donnerai sur notre méthode : on aurait un modèle très rapide et visuellement réaliste donc parfaitement adaptée à l'informatique graphique. Il faut tout de même poser un bémol sur les vitesses données pour les modèles masses-tenseurs, car ceux-ci utilisent une intégration explicite du temps, et avec des petites rigidités ils peuvent être très

rapides, mais dès que la rigidité augmente, il est nécessaire d'augmenter le pas de temps et ces résultats seront revus à la baisse.

4.3 Conclusion

Bien qu'on utilise un tenseur de déformation linéaire, la méthode élimine les gonflements provoqués par des grands déplacements. Elle conserve bien les volumes. De plus, elle est très stable quelque soit l'état de déformation du système.

En terme de précision, notre modèle suit un comportement physiquement réaliste. Il respecte la théorie de la poutre encastrée, concorde avec des méthodes numériques issues de l'ingénierie mécanique et simule bien l'expérience du truth cube. On peut donc considérer notre méthode comme précise.

Malheureusement, dans le cas de simulations très complexes, où le maillage est composé d'un très grands nombres d'éléments et où les déplacements contraints sont grands et surtout appliqués rapidement (en peu de pas de temps), la méthode montre des faiblesses. En effet, les déplacements ont du mal à se propager, effet indésirable des méthodes explicites. C'est donc l'aspect dynamique qui est moins performant.

Au niveau de la vitesse, les résultats sont très satisfaisant, en effet notre méthode est aussi rapide qu'une méthode utilisant un tenseur de déformations linéaire sans gestion des grands déplacements. Grâce à son intégration implicite du temps, elle ne perd pas de vitesse pour des grandes rigidités. Pour la même vitesse, on a donc une méthode physiquement plus réaliste que les meilleures méthodes actuelles.

Chapitre 5

Conclusion et Perspectives

5.1 Conclusion

Durant ce stage, nous avons réalisé un modèle d'objets défomables stable, à la fois rapide et physiquement réaliste.

Grâce à son utilisation des éléments finis et sa gestion des grands déplacements, il suit une certaine précision. Ses calculs explicites et linéaires lui permettent d'être rapide et de faire des animations interactives pour un nombre relativement important d'éléments (plusieurs centaines).

La nouveauté se situe au niveau du calcul des déformations. Le raisonnement développé est d'extraire les déformations à partir de grands déplacements tout en conservant un tenseur de déformations linéaire. Pour cela, il faut éliminer les déplacements rigides et ne garder que les déformations pures. Deux versions ont été mises au point dans ce but. Une version analytique basée sur une décomposition de Cholesky, afin d'extraire la rotation et la déformation à partir de la matrice de transformation de l'état déformé à l'état au repos. Et une version géométrique où les rotations locales sont calculées à partir de repères locaux en chaque élément. C'est la deuxième version qui s'est révelée être la plus efficace, car elle marche dans toutes les conditions, quelque soit l'état des éléments, de plus elle peut évoluer vers d'autres éléments non tetraèdriques.

En plus d'accélérer les calculs, ce principe permet de mettre à jour à moindre coût les matrices de rigidité, assurant une grande précision et une grande stabilité. Alors que dans toutes les autres méthodes rapides, cette matrice de rigidité est précalulée sur l'élément au repos ; cette approximation créant des couples fantômes donc une certaine irréalité physique.

Autour de ce calcul, seules des loi basiques ont été utilisées (loi de comportement linéaire, éléments tetraèdriques linéaires), il est fortement possible d'améliorer la précision du modèle en utilisant d'autres lois plus complexes.

Malgré ces aspects basiques, les résultats en terme de précision sont bons et on peut espérer beaucoup de cette méthode. Au niveau de la vitesse, là encore les résultats sont satisfaisants, et les temps de calculs sont encore améliorables.

5.2 Perspectives

5.2.1 Propagation

Nous avons que dans le cas de simulation très complexe, utilisant un maillage comportant énormément d'éléments et des grosses déformations appliquées rapidement, le fait d'utiliser une méthode d'éléments finis explicite crée des problèmes de propagation des déplacements. Il faudrait trouver un moyen de mieux propager, afin de pourvoir gérer plus de cas de figure et ne pas se limiter à des modèles qui se déforment lentement.

5.2.2 Matériaux

Il pourrait être bien d'adapter la loi de comportement pour ne pas se restreindre qu'au purement linéaire. Comme vu précédemment, il est facile de la faire évoluer les paramètres du matériau en fonction des déformations pour avoir un matériau non-linéaire et ainsi mieux gérer la précision en grandes déformations.

On pourrait aussi gérer des matériau anisotropes, plastiques...

Au niveau du comportement, un point très important pour gérer tous les cas de figure, serait de gérer les déchirures après un certain seuil de déformation. Dans l'élan, on pourrait aussi gérer les découpes. Bien sûr ces cas de figure sont très complexes, car ils modifient la topologie du maillage.

5.2.3 Eléments

On [6-OHA00] sait que dans le cas de solides, les éléments hexaèdriques donnent de meilleurs résultats ou sont moins chers en temps de calcul : on peut se permettre de mettre moins de plus gros éléments pour la même précision ou de mailler plus finement pour les mêmes temps de calculs. De plus il est plus facile de respecter les symétries de maillage qu'avec des tetraèdres. Actuellement, on gère un hexaèdre par 10 tetraèdres, donc il est sûr que l'on gagnerait beaucoup à utiliser directement des hexaèdres.

Les approches utilisant des maillages indépendant de l'objet, comme les Free Form Deformation semblent intéressantes, car mailler un modèle surfacique est une étape délicate et difficile et pourtant très importante. Il serait alors possible de séparer les mailles, utilisées pour la physique du modèle surfacique, utilisé pour l'affichage. L'étape très délicate du maillage pourrait ainsi être évitée, et les éléments finis pourraient être fabriqués avec des formes parfaitement utilisables non dégénérées.

5.2.4 Tests

Maintenant que le modèle a été testé en statique, il serait intéressant de tester l'aspect dynamique, et de voir si d'une position d'équilibre à une autre, les étapes de déformations intermédiaires sont précises. Mais cet aspect semble très difficile à tester. Je ne sais pas s'il existe des solutions théoriques dynamiques. Il faudrait aussi des résultats expérimentaux tels que le truth cube, pas seulement axés sur différents états d'équilibre, mais aussi sur tout le cheminement pour les atteindre.

5.2.5 Compromis temps de calcul / précision

Le compromis entre vitesse et précision est important si on veux faire une méthode générique qui peut être utilisable à la fois en synthèse d'images, où le plus important est la vitesse, et en simulation chirurgicale, où le plus important est la précision.

On peut définir une fois pour toute dans quelle catégorie se place la simulation, afin de profiter au mieux des qualités désirées du modèle. Par exemple dans le cas de simulations physiquement moins réalistes, on peut imaginer une formulation masses-tenseurs de notre méthode, avec un précalcul des matrices de rigidité, afin d'assurer les meilleurs temps de calcul possibles.

Au contraire, ce compromis peut être géré à la volée, au cours de l'animation. Pour cela, je pense en particulier à la multi-resolutions du maillage ou du temps. Ce principe est intéressant car il permet de limiter les calculs en perdant un minimum de précision. L'idée est de détailler le maillage là où les déformations sont les plus grandes et de le rendre grossier là où elles sont le plus faible. La puissance de calcul est ainsi centrée là où elle est utile. Plus on voudra une simulation physiquement réaliste, plus on choisira le seuil de changement de résolutions tel que les éléments soient le plus petits possibles. Au contraire, pour un jeu vidéo, avec énormément d'éléments, on placera le seuil tel que les éléments soient le plus gros possible.

On peut aussi imaginer une approche multi-résolution couplée avec des FFD, soit une FFD multi-résolutions, qui n'existe pas encore clairement à l'heure actuelle. On peut facilement imaginer les résultats que pourrait donner cette découpe adaptative animée avec notre modèle déformable.

Annexe A

Fonctions d'interpolations

Les coordonnées du point i sont x_i, y_i et z_i. Les fonctions d'interpolation sont de la forme : $N_i = a_i + b_i x + c_i y + d_i z$.

A.1 Triangle linéaire

$$N_1 = \frac{(y_2 - y_3)x + (x_3 - x_2)y + x_2 y_3 - x_3 y_2}{-x_2 y_1 + x_3 y_1 + x_1 y_2 - x_3 y_2 - x_1 y_3 + x_2 y_3} \quad (en\, 2D)$$

Les coefficients des autres fonctions d'interpolations sont les mêmes par rotation des indices.

A.2 Tetraèdre linéaire

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix}$$

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix}$$

$$a_1 = \begin{vmatrix} x_2 & x_3 & x_4 \\ y_2 & y_3 & y_4 \\ z_2 & z_3 & z_4 \end{vmatrix}$$

$$b_1 = - \begin{vmatrix} 1 & 1 & 1 \\ y_2 & y_3 & y_4 \\ z_2 & z_3 & z_4 \end{vmatrix}$$

$$c_1 = \begin{vmatrix} 1 & 1 & 1 \\ x_2 & x_3 & x_4 \\ z_2 & z_3 & z_4 \end{vmatrix}$$

$$d_1 = -\begin{vmatrix} 1 & 1 & 1 \\ x_2 & x_3 & x_4 \\ y_2 & y_3 & y_4 \end{vmatrix}$$

Les coefficients des autres fonctions d'interpolations sont les mêmes par un changement cyclique d'indices (attention les signes changent aussi).

La matrice de rigidité de l'élément B devient donc :

$$B = \begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & b_4 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & c_4 & 0 \\ 0 & 0 & d_1 & 0 & 0 & d_2 & 0 & 0 & d_3 & 0 & 0 & d_4 \\ c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & c_4 & b_4 & 0 \\ 0 & d_1 & c_1 & 0 & d_2 & c_2 & 0 & d_3 & c_3 & 0 & d_4 & c_4 \\ d_1 & 0 & b_1 & d_2 & 0 & b_2 & d_3 & 0 & b_3 & d_4 & 0 & b_4 \end{bmatrix}$$

A.3 Elément quelconque

En posant $N^{(i)} = a_1^{(i)} + a_2^{(i)} B_1 + ... + a_n^{(i)} B_n$ avec $N^{(i)}$la fonction d'interpolation au point i, les $a_j^{(i)}$ ses coefficients pour la base canonique de polynômes (B_1, ..., B_n), si la matrice des $a_j^{(i)}$ est régulière, on a :

$$\begin{bmatrix} a_1^{(1)} & ... & a_1^{(n)} \\ ... & ... & ... \\ a_n^{(1)} & ... & a_n^{(n)} \end{bmatrix} = \begin{bmatrix} B_1^{(1)} & ... & B_1^{(n)} \\ ... & ... & ... \\ B_n^{(1)} & ... & B_n^{(n)} \end{bmatrix}^{-1}$$

rappels sur les bases de polynômes

L'espace des polynômes de degré d est un espace vectoriel dont la dimension dépend du degré des polynômes et du nombre de variables. Cet espace possède une base canonique constituée de tous les monômes de degré non négatif inférieur ou égal à d. Par exemple :

- La base canonique des polynômes à une variable x_1 de degré 3 est constituée des monômes $\{1, x_1, x_1^2, x_1^3\}$.
- La base canonique des polynômes à deux variables x_1 et x_2 de degré 2 est constituée des monômes $\{1, x_1, x_2, x_1^2, x_2^2, x_1x_2\}$.
- La base canonique des polynômes à trois variables x_1, x_2 et x_3 de degré 2 est constituée des monômes $\{1, x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_2x_3, x_1x_3\}$.

Le tableau A.1 donne les dimensions des espaces de polynômes pour des degrés de 1 à 5 pour les polynômes à 1, 2 ou 3 variables.

degré	**1 variable**	**2 variables**	**3 variables**
1	2	3	4
2	3	6	10
3	4	10	20
4	5	15	35
5	6	21	56

TAB. A.1 – Dimensions des espaces de polynômes

On peut voir qu'il n'est pas possible de faire n'importe quel type d'élément avec n'importe quel nombre de noeuds.

Annexe B

Divers

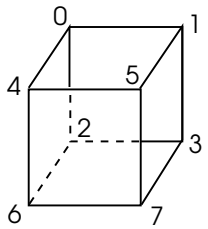
B.1 Volume du tetraèdre

Différents calculs du volume d'un tetraèdre (a, b, c, d) :

$$
\begin{aligned}
volume(tetraedre) &= \frac{1}{6}\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_a & x_b & x_c & x_d \\ y_a & y_b & y_c & y_d \\ z_a & z_b & z_c & z_d \end{vmatrix} \\
&= \frac{1}{6}\begin{vmatrix} x_a - x_b & y_a - y_b & z_a - z_b \\ x_a - x_c & y_a - y_c & z_a - z_c \\ x_a - x_d & y_a - y_d & z_a - z_d \end{vmatrix} \\
&= \frac{1}{6}\left|(\overrightarrow{ab} \wedge \overrightarrow{ac}) \cdot \overrightarrow{ad}\right|
\end{aligned}
$$

B.2 Maillage d'un hexaèdre

Dans notre implémentation, ne sont géré, pour l'instant, que les tetraèdres. Hors quasiment tout les maillages utilisés lors des tests étaient en éléments hexaèdriques. Pour animer au mieux un hexaèdre avec des tetraèdres, la meilleure méthode trouvée est de le découper en 10 tetraèdres. En effet, il existe 2 méthodes pour découper un hexaèdre en 5 tetraèdres. Hors ces 2 découpes ne sont pas symétriques et on obtient donc pas les mêmes résultats avec un hexaedre ou 5 tetraèdres. Alors qu'en superposant les 2 découpes, soit 10 tetraèdres, le maillage est symétrique, et on obtient au moins les mêmes positions d'équilibre. Il faut alors diviser la rigidité des tetraèdres par 2 pour retomber sur la même rigidité que l'hexaèdre.



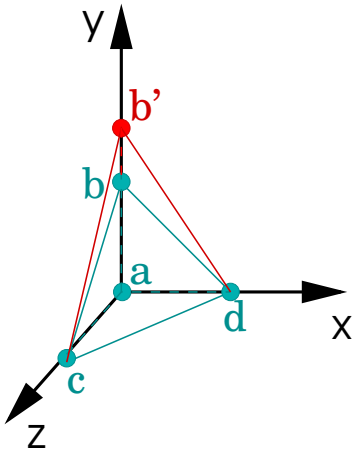
(1,4,5,7)		(5,0,1,3)
(0,1,4,2)		(4,5,0,6)
(1,3,2,7)		(5,7,6,3)
(2,4,6,7)		(6,0,2,3)
(1,2,4,7)		(5,6,0,3)

FIG. B.1 – Les deux découpes en cinq tetraèdres d'un hexaèdre

Annexe C

Exemple de calcul

Prenons l'exemple du tetraèdre défini figure C.1 et développons les calculs.



a=a'=(0,0,0)
b=(0,1,0) b'=(0,1.1,0)
c=c'=(0,0,1)
d=d'=(1,0,0)

E=1000
μ=0.49

FIG. C.1 – Le point b est déplacé de 0.1 suivant l'axe y

C.1 Matrice de rigidité (en petits déplacements)

– Volume au repos :

$$V = \frac{1}{6}\left|(\overrightarrow{ab} \wedge \overrightarrow{ac}) \cdot \overrightarrow{ad}\right| = \frac{1}{6}$$

– Rigidité du matériau :

$$\begin{aligned} D &= \frac{1}{36V^2}\frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \\ &= \begin{bmatrix} 17114.1 & 16443 & 16443 & 0 & 0 & 0 \\ 16443 & 17114.1 & 16443 & 0 & 0 & 0 \\ 16443 & 16443 & 17114.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 335.57 & 0 & 0 \\ 0 & 0 & 0 & 0 & 335.57 & 0 \\ 0 & 0 & 0 & 0 & 0 & 335.57 \end{bmatrix} \end{aligned}$$

– Fonctions de forme

$$b_1 = -c'_y d'_z = -1 \qquad b_3 = 0$$
$$c_1 = (c'_x d'_z) - ('b_x d'_z) = -1.1 \qquad c_3 = d'_z b'_x = 1.1$$
$$d_1 = c'_y d'_x - c'_x d'_y + b'_x d'_y - b'_x c'_y = -1.1 \qquad d_3 = -d'_y b'_x = 0$$

$$b_2 = -c'_y d'_z = 1 \qquad b_4 = 0$$
$$c_2 = c'_x d'_z = 0 \qquad c_4 = 0$$
$$d_2 = c'_y d'_x - c'_x d'_y = 0 \qquad d_4 = -b'_x c'_y = 1.1$$

– Rigidité de l'élément

$$B = \begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & b_4 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & c_4 & 0 \\ 0 & 0 & d_1 & 0 & 0 & d_2 & 0 & 0 & d_3 & 0 & 0 & d_4 \\ c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & c_4 & b_4 & 0 \\ 0 & d_1 & c_1 & 0 & d_2 & c_2 & 0 & d_3 & c_3 & 0 & d_4 & c_4 \\ d_1 & 0 & b_1 & d_2 & 0 & b_2 & d_3 & 0 & b_3 & d_4 & 0 & b_4 \end{bmatrix}$$
$$= \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.1 & 0 & 0 & 0 & 0 & 0 & 1.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.1 \\ -1.1 & -1 & 0 & 0 & 1 & 0 & 1.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.1 & -1.1 & 0 & 0 & 0 & 0 & 0 & 1.1 & 0 & 1.1 & 0 \\ -1.1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 1.1 & 0 & 0 \end{bmatrix}$$

C.2 Grands déplacements : version géométrique

$$f = RR^{(deforme)}_{(global)} B^T DB(RR^{(global)}_{(repos)}(X^{(global)}_{repos} - A^{(global)}_{repos}) - (RR^{(global)}_{(deforme)}(X^{(global)}_{deforme} - A^{(global)}_{deforme})))$$

– Rotations :

$$R^{(global)}_{(repos)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$R^{(global)}_{(deforme)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

– Coordonnées en repères locaux :

$$X^{(repos)}_{repos} = \begin{bmatrix} R^{(global)}_{(repos)} & 0 & 0 & 0 \\ 0 & R^{(global)}_{(repos)} & 0 & 0 \\ 0 & 0 & R^{(global)}_{(repos)} & 0 \\ 0 & 0 & 0 & R^{(global)}_{(repos)} \end{bmatrix} \begin{pmatrix} a_x - a_x \\ a_y - a_y \\ a_z - a_z \\ b_x - a_x \\ b_y - a_y \\ b_z - a_z \\ c_x - a_x \\ c_y - a_y \\ c_z - a_z \\ d_x - a_x \\ d_y - a_y \\ d_z - a_z \end{pmatrix}$$

$$X_{deforme}^{(deforme)} = \begin{bmatrix} R_{(deforme)}^{(global)} & 0 & 0 & 0 \\ 0 & R_{(deforme)}^{(global)} & 0 & 0 \\ 0 & 0 & R_{(deforme)}^{(global)} & 0 \\ 0 & 0 & 0 & R_{(deforme)}^{(global)} \end{bmatrix} \begin{pmatrix} a'_x - a'_x \\ a'_y - a'_y \\ a'_z - a'_z \\ b'_x - a'_x \\ b'_y - a'_y \\ b'_z - a'_z \\ c'_x - a'_x \\ c'_y - a'_y \\ c'_z - a'_z \\ d'_x - a'_x \\ d'_y - a'_y \\ d'_z - a'_z \end{pmatrix}$$

– Petits déplacements :

$$U = X_{repos}^{(repos)} - X_{deforme}^{(deforme)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -0.1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

– Forces :

$$F = RR_{(global)}^{(deforme)} B^T DBU = \begin{pmatrix} 1711.41 \\ 1808.73 \\ 1808.73 \\ -1711.41 \\ 0 \\ 0 \\ 0 \\ -1808.73 \\ 0 \\ 0 \\ 0 \\ -1808.73 \end{pmatrix}$$

C.3 Grands déplacements : version analytique

$$f = RR_{(global)}^{(deforme)} B^T DB\varepsilon$$

$$A_{(global)}^{(repos)} = \begin{bmatrix} b_x - a_x & c_x - a_x & d_x - a_x \\ b_y - a_y & c_y - a_y & d_y - a_y \\ b_z - a_z & c_z - a_z & d_z - a_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$
\begin{aligned}
&= R^{(repos)}_{(global)} U^{(repos)}_{(global)} \\
&= R^{(repos)}_{(global)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

$$
\varepsilon^{(repos)}_{(global)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

$$
\begin{aligned}
A^{(deforme)}_{(global)} &= \begin{bmatrix} b'_x - a'_x & c'_x - a'_x & d'_x - a'_x \\ b'_y - a'_y & c'_y - a'_y & d'_y - a'_y \\ b'_z - a'_z & c'_z - a'_z & d'_z - a'_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1.1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
&= R^{(deforme)}_{(global)} U^{(deforme)}_{(global)} \\
&= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1.1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

$$
\varepsilon^{(deforme)}_{(global)} = \begin{pmatrix} 0.1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

$$
\varepsilon^{(deforme)}_{(repos)} = \begin{pmatrix} -0.1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}
$$

$$
F = RR^{(deforme)}_{(global)} B^T DBU = \begin{pmatrix} 1711.41 \\ 1808.73 \\ 1808.73 \\ -1711.41 \\ 0 \\ 0 \\ 0 \\ -1808.73 \\ 0 \\ 0 \\ 0 \\ -1808.73 \end{pmatrix}
$$

Bibliographie

[1-MUL02] M. Müller, *Stable Real-Time Deformation*, SCA 2002

[2-ETZ03] O. Etzmuß, M. Keckeisen and W. Straßer, *A Fast Finite Element Solution for Cloth Modelling*, PG 2003

[3-PIC01] G. Picinbono, *Modèles géométriques et physiques pour la simulation d'interventions chirurgicales*, thèse 2001

[4-COT97] S. Cotin, *Modèles anatomiques déformables en temps réel*, thèse 1997

[5-GAR02] J. Garrigues, *Initiation à la méthode des éléments finis*, cours 2000

[6-OHA00] J. Ohayon, *Mécanique des milieux continus et méthodes des éléments finis*, cours 2000

[7-PIC02] G. Picinbonno, J.-C. Lombardo, H. Delingette et N. Ayache, *Improving realism of a surgery simulator : linear anisotropic elasticity, complex interaction and force extrapolation*, JVCA 2002

[8-COT02] S. Cotin, H. Delingette et N. Ayache, *A hybrid model for real-time cutting, deformation, and force feedback for surgery training and simulation*, Visual Computer 2002

[9-PIC03] G. Picinbonno, H. Delingette et N. Ayache, *Non-liear anisotropic elasticity for real-time surgery simulation*, Graphical Models 2003

[10-MUL04] M. Müller and M. Gross, *Interactive Virtual Materials*, GI 2004

[11-FAU01] F. Faure, *Systèmes dynamiques*, cours 2001

[12-DEB00] G. Debunne, *Animation multirésolution d'objets déformables en temps-réel. Application à la simulation chirurgicale*, thèse 2000

[13-PRO97] E. Promayon, P. Baconnier et C. Puech, *Physically-based model for simulating the human trunk respiration movements*, CVRMed 1997

[14-PRO96] E. Promayon, P. Baconnier et C. Puech, *Physically-Based Deformations Constrained in Displacements and Volume*, Comput. Graph. Forum 1996

[15-CHA02] M. Chabanas, *Modélisation des tissus mous de la face pour la chirurgie orthognatique assistée par ordinateur*, thèse 2002

[16-MUL04] M. Müller, M. Teschner and M. Gross, *Physically-based simulation of objects represented by surface meshes*, CGI 2004

[17-SED86] T. W. Sederberg and S.R. Parry, *Free-Form Deformation of solid geometric models*, ACM 1986

[18-FAT97] P. Faloutsos, M. van de Panne and D. Terzopoulos, *Dynamic Free-Form Deformations for animation synthesis*, IEEE Transactions on visualization and computer graphics 1997

[19-FRI02] N. Frisch and T. Ertl, *Deformation of finite element meshes using directly manipulated Free-Form Deformation*, ACM 2002

[20-CAP02] S. Capell, S. Green, B. Curless, T. Duchamp and Z. Popovic, *A Multiresolution Framework for Dynamic Deformations*, SCA 2002

[21-BRO98] M. Bro-Nielsen, *Finite element modeling in surgery simulation*, Proceedings of the IEEE 1998

[22-COT99] S. Cotin, H. Delingette, N. Ayache, *Real-time elastic deformations of soft tissues for surgery simulation*, IEEE Transactions on Visualization and Computer Graphics 1999

[23-DEL99] H. Delingette, S. Cotin, N. Ayache, *Efficient linear elastic models of soft tissues for real-time surgery simulation*, MMVR 1999

[24-JAM99] D.L. James, D.K. Pai, *Accurate real-time deformable objects*, SIGGRAPH 1999

[25-MON01] C. Monserrat, U. Meier, M. Alcañiz, F. Chinesta, M.C. Juan, *A new approach for the real-time simulation of tissue deformations in surgery simulation*, Computer Methods and Programs in Biomedicine 2001

[26-SCH03] J.M. Schwartz, *Calcul rapide de forces et de déformations mécaniques non-linéaires et visco-élastiques pour la simulation de chirurgie*, thèse 2003

[27-KER03] A. E. Kerdok, S. M. Cotin, M. P. Ottensmeyer, A. M. Galea, R. D. Howe and S. L. Dawson, *Truth Cube : Establishing Physical Standards for Real Time Soft Tissue Simulation*, Medical Image Analysis 2003

[28-CHA03] M. Chabanas, V. Luboz, Y. Payan, *Patient specific finite element model of the face soft tissues for computer-assisted maxillofacial surgery*, Medical Image Analysis 2003

[29-COV93] S.A. Cover, N.F. Ezquerra, J.F. O'Brien, R. Rowe, T. Gadacz and E. Palm, *Interactively deformable models for surgery simulation*, IEEE Computer Graphics and Applications 1993

[30-LEE95] Y. Lee, D. Terzopoulos and K. Waters, *Realistic modeling for facial animation*, Proc. SIGGRAPH 1995

[32-BRE94] D.E. Breen, D.H. House and M.J. Wozny, Predicting the drape of women cloth using interacting particles, Proc. SIGGRAPH 1994

[33-VOL95] P. Volino, M. Courchesne et N. Magnenat Thalmann, *Versatile and efficient techniques for simulating cloth and other deformable objects*, Proc. annual conference on Computer graphics and interactive techniques 1995

Un grand nombre des images de ce document sont tirées de ces références.