

---

---

# Milieu mécanique déformable multirésolution pour la simulation interactive

---

---

Matthieu Nesme

## Thèse

présentée pour l'obtention du titre de  
**Docteur de l'Université Joseph Fourier**

spécialité  
**Imagerie, Vision, Robotique**

préparée au sein des laboratoires  
**TIMC-IMAG (GMCAO)** UMR CNRS 5525  
**LJK (EVASION)** UMR CNRS 5224 / INRIA

dans le cadre de l'Ecole Doctorale  
Mathématiques, Sciences et Technologies de l'Information, Informatique

### Composition du jury :

Denis Favier	Professeur	UJF	Grenoble	Président
Hervé Delingette	Directeur de Recherche	INRIA	Nice	Rapporteur
Mathieu Desbrun	Associate Professor	Caltech	Los Angeles	Rapporteur
Stéphane Cotin	Chargé de Recherche	INRIA	Lille	Examineur
Yohan PAYAN	Chargé de Recherche	CNRS	Grenoble	Directeur
François FAURE	Maître de Conférences	UJF	Grenoble	Co-directeur



# Avant-propos

---

Cette thèse a été financée à l'aide d'une bourse ministérielle, ainsi que d'un monitorat du CIES de Grenoble effectué à l'Université Pierre Mendès-France sous la tutelle de Nathalie Denos. Ma quatrième année a été assurée par un poste de demi-ATER de l'Université Joseph Fourier. J'ai aussi bénéficié d'une bourse exploradoc de la région Rhône-Alpes afin d'effectuer un séjour de huit mois au Laboratoire d'Informatique Graphique de l'Université de Montréal (LIGUM) sous la responsabilité de Pierre Poulin.

Cette thèse s'est déroulée dans deux équipes en parallèle, équipe GMCAO du laboratoire TIMC-IMAG au CHU de Grenoble et équipe EVASION du laboratoire Jean Kuntzmann à l'INRIA Rhône-Alpes.

L'objectif majeur de l'équipe Gestes Médico-Chirurgicaux Assistés par ordinateur (GMCAO) 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. Cet objectif nécessite l'exploitation de données médicales multi-modales et de connaissances à priori, la fusion de ces données, la planification optimale de l'intervention par rapport à un critère donné (par exemple : position optimale d'une prothèse, délivrance optimale d'une dose de radiation pour traiter une tumeur) et la mise en oeuvre de systèmes de guidage permettant de réaliser la stratégie planifiée de façon la plus précise et la plus sûre possible.

Les travaux de recherche de l'équipe Environnements Virtuels pour l'Animation et la Synthèse d'Images d'Objets Naturels (EVASION) sont dédiés à la modélisation, à l'animation, et à la visualisation d'objets et de phénomènes naturels. Pour cela, deux grands axes de recherche sont privilégiés. D'une part le développement d'outils fondamentaux destinés à la spécification de scènes et d'objets naturels, à la mise au point de modèles alternatifs pour la forme, le mouvement et l'apparence ainsi qu'à la conception d'algorithmes reposant sur un niveau de détail adaptatif pour gérer au mieux la complexité. D'autre part la validation de ces outils sur des scènes naturelles spécifiques, qui vont du monde minéral (océan, ruisseaux, lave, avalanches, nuages) au monde animal (simulation d'organes, visages, corps et chevelure d'un personnage, mouvements d'animaux), en passant par les scènes végétales (morphogénèse de plantes, prairies, arbres).

Parallèlement à mon travail de thèse concernant l'animation physique d'objets déformables, durant cette période, j'ai aussi étudié la gestion des collisions [Faure 07] et la triangulation de surfaces implicites animées [Nesme 06a, Bouthors 07]. Dans ce document n'apparaissent que les détails relatifs aux travaux concernant strictement mon sujet de thèse qui ont donné lieu à plusieurs publications [Nesme 05b, Nesme 06c, Nesme 06b].

J'ai eu l'honneur de co-encadrer deux étudiants durant leur stage de deuxième année ingénieur ENSIMAG. Maxime Tournier a fait un stage purement ingénierie sur une librairie d'octree adaptée à ce travail de thèse durant ma première année. Damien Rumiano a eu une démarche de recherche sur les modifications topologiques de surfaces implicites animées durant ma troisième année.

Des vidéos illustrent ce document. Elles sont accessibles sur <http://evasion.imag.fr/Membres/Matthieu.Nesme/these/> ou directement en cliquant sur les icônes de marge dans la version électronique de ce manuscrit.



# Remerciements

---

En premier lieu, je remercie Jocelyne Troccaz et Marie-Paule Cani de m'avoir accueilli dans leurs équipes, ainsi que Pierre Poulin pour m'avoir permis de découvrir une autre façon de travailler outre-Atlantique.

Merci aux membres de mon jury et principalement aux rapporteurs pour leur lecture approfondie et leur quantité de remarques pertinentes qui sont appréciées à leur juste valeur.

Un énorme merci à mes directeurs de thèse pour toute leur aide, leur grande disponibilité, leur sympathie, leurs discussions scientifiques ou autour d'une bière, et quantité d'autres choses. Yohan, pour ton encadrement toujours judicieux et ton esprit d'ouverture vers divers domaines scientifiques ; et François pour ta grande disponibilité, ta façon d'imager les théories les plus théoriques, d'avoir répondu à toutes mes questions sans relâche et de m'avoir appris une quantité inlistable de trucs, le tout sans pression et toujours avec plaisir.

Un sincère remerciement à Anne Pierson, pour sa façon de s'investir pour l'équipe et de toujours offrir le meilleur envisageable.

A tous mes collègues des deux équipes aux ambiances bien différentes et pourtant toutes deux extrêmement appréciables et confortables. A leur bonne humeur et leur qualité et diversité scientifiques. Aux gens qui ont partagé mon bureau pour avoir supporté mon insatiable envie d'écoute musicale. Je ne citerai qu'en particulier Antoche avec qui partager un bureau à évoluer jusqu'à partager une bonne amitié et même à s'encorder pour ne plus à avoir à se séparer.

Une pensée aux excellents stagiaires que j'ai eu l'honneur d'encadrer, et qui m'ont fait découvrir une autre facette excitante du métier de chercheur, ainsi qu'aux élèves à qui j'ai enseigné et dont les yeux pétillants d'apprendre (si si ça arrive quelquefois) apportent un plaisir rarement égalé.

A tous mes potes et surtout celles et ceux avec qui j'ai partagé un appart et qui ont dû supporter mes humeurs variables au rythme des deadlines.

A mes parents et ma frangine qui m'ont toujours poussé et soutenu dans mes propres choix, et plus généralement à ma famille, merci d'être venu si nombreux à la soutenance et pour le pot dont vous m'avez honoré.

A ma tendre Mido, je te souhaite autant de plaisir dans ton éventuelle future thèse.

Le plus important pour la fin : vive la tectonique des plaques pour nous avoir fait de si jolies montagnes toutes proches !



# Sommaire

---

<b>Introduction</b>	<b>1</b>
<b>Contexte scientifique</b>	<b>9</b>
<b>1 Etat de l'art</b> . . . . .	<b>11</b>
<b>2 Notions de mécanique</b> . . . . .	<b>29</b>
<b>Milieu mécanique multirésolution</b>	<b>57</b>
<b>3 Animation d'objets par grilles de déformations</b> . . . . .	<b>63</b>
<b>4 Mécanique des hexaèdres englobants</b> . . . . .	<b>75</b>
<b>5 Propagation dynamique</b> . . . . .	<b>91</b>
<b>6 Validations du modèle</b> . . . . .	<b>111</b>
<b>Conclusion</b>	<b>127</b>
<b>Bibliographie</b>	<b>133</b>
<b>Annexes</b>	<b>149</b>
<b>Tables détaillées</b>	<b>165</b>
<b>Table des figures</b> . . . . .	<b>167</b>
<b>Table des matières</b> . . . . .	<b>171</b>





# Notations

---

## Écriture :

$a$	scalaire
$\mathbf{a}$	vecteur
$\mathbf{A}$	matrice
$a_{ij}$	entrée d'une matrice ou d'un vecteur avec ses indices
$[\alpha] = \begin{bmatrix} \alpha_{xx} & \alpha_{xy} & \alpha_{xz} \\ \alpha_{yx} & \alpha_{yy} & \alpha_{yz} \\ \alpha_{zx} & \alpha_{zy} & \alpha_{zz} \end{bmatrix}$	tenseur sous forme matricielle
$\{\alpha\} = \begin{bmatrix} \alpha_{xx} \\ \alpha_{yy} \\ \alpha_{zz} \\ \alpha_{xy} \\ \alpha_{yz} \\ \alpha_{xz} \end{bmatrix}$	tenseur symétrique sous forme vectorielle

## Opérateurs :

$\mathbf{a} \cdot \mathbf{b}$	produit scalaire
$\mathbf{a} \wedge \mathbf{b}$	produit vectoriel
$\mathbf{a}^T$	transposée
$tr(\mathbf{A})$	trace
$det(\mathbf{A})$	déterminant
$\nabla \mathbf{a} = \begin{bmatrix} \frac{\partial a_x}{\partial x} & \frac{\partial a_x}{\partial y} & \frac{\partial a_x}{\partial z} \\ \frac{\partial a_y}{\partial x} & \frac{\partial a_y}{\partial y} & \frac{\partial a_y}{\partial z} \\ \frac{\partial a_z}{\partial x} & \frac{\partial a_z}{\partial y} & \frac{\partial a_z}{\partial z} \end{bmatrix}$	gradient

## Symboles :

$\mathbf{x}$	positions
$\mathbf{u}, \dot{\mathbf{u}}, \ddot{\mathbf{u}}$	déplacements, vitesses, accélérations
$\varepsilon$	déformations
$\sigma$	contraintes
$\mathbf{f}$	forces
$t$	temps
$h$	pas de temps
$E$	module d'Young
$\nu$	coefficient de Poisson
$\mathbf{B}$	matrice déplacements-déformations
$\mathbf{D}$	matrice déformations-contraintes
$\mathbf{K} = \mathbf{B}^T \mathbf{D} \mathbf{B}$	matrice de rigidité
$\mathbf{M}$	matrice de masse
$\mathbf{C}$	matrice d'amortissement
$\mathbf{I}$	matrice identité (de taille adaptée)



---

# Introduction



# 1 Motivation

La simulation d'objets déformables est devenue incontournable en informatique graphique. Elle sert pour un grand nombre d'applications, des jeux vidéo aux simulations chirurgicales. Pour être au plus proche de la réalité, il faut adopter des calculs de déformations basés sur des lois physiques issues de la mécanique des solides déformables. Les simulations physiques permettent de synthétiser avec précision un grand nombre de phénomènes physiques. Elles permettent aussi l'intégration de différents types de matériaux réels connus. Elles sont indispensables dès que l'on veut une grande précision, comme dans le cas de simulations chirurgicales ou de films d'animations réalistes. Malheureusement, il est difficile de les adopter dans un contexte temps-réel car les modèles sous-jacents sont d'une grande complexité calculatoire, et leur précision numérique dépend de beaucoup de paramètres pas toujours maîtrisables, tel que la taille et la qualité de la discrétisation ou la raideur des objets. Lorsque la rapidité est de mise, par exemple dès que l'on veut interagir avec un objet, des simplifications dans les calculs des déformations sont obligatoires, généralement au détriment de la précision du comportement.

La première difficulté évoquée concerne la discrétisation. Les modèles mécaniques nécessitent une représentation volumique de l'objet qui est souvent difficile à obtenir avec des critères garantissant une bonne rapidité et une bonne stabilité numérique. Cette phase est classiquement une tétraédrisation d'une surface "propre" (manifold, sans trous...) Dans certains cas, celle-ci peut produire des éléments de diverses tailles et formes donnant un système mal conditionné rendant la résolution lente et instable. Ceci limite donc fortement les objets animables facilement en temps-réel. Il serait très utile de disposer d'une méthode simple permettant à un infographiste de fabriquer un maillage mécanique pour n'importe quel objet qu'il vient de modéliser, ou d'automatiquement construire un maillage mécanique spécialisé à un patient, dans le cas de simulations chirurgicales, à partir des données médicales dont on dispose (de type scanner ou IRM).

Une autre difficulté concerne les problèmes numériques. Si le système est mal conditionné (discrétisation irrégulière, raideurs trop grandes, cf. annexe A), les calculs peuvent avoir du mal à converger, voire diverger. Garantir des éléments réguliers et résoudre les calculs rapidement et avec robustesse est crucial. Simuler interactivement des *matériaux mal conditionnés* (i.e. générant des systèmes mal conditionnés, cf. annexe A) en temps-réel est un problème ouvert.

La majorité des applications nécessitent non seulement une simulation physique de qualité, mais également un affichage détaillé des objets et un modèle de collision. La qualité de l'affichage des objets déformés est une phase importante, aussi bien en rendu réaliste qu'en visualisation scientifique. Afin de bien gérer séparément ces aspects, nous pensons qu'il est important de découpler les résolutions des maillages mécanique, d'affichage et de contact. En effet, si, comme classiquement, le maillage mécanique est utilisé pour le rendu, puisqu'en temps réel on ne peut animer que peu d'éléments grossiers, le rendu sera pauvre. A l'inverse, pour un rendu détaillé, la mécanique sera très lente. Il est nécessaire d'être en mesure d'animer des objets visuellement détaillés rapidement.

Les travaux présentés dans ce manuscrit s'inscrivent dans la modélisation de tissus mous au sens le plus large possible, sans cadre applicatif particulier, mais avec le souci d'offrir une méthode simple et efficace, facilement utilisable en pratique. Même si notre motivation profonde se place dans le contexte des simulations chirurgicales, aucun simulateur d'une intervention précise n'a été mis en place au cours de cette thèse. Notre conviction est qu'actuellement, les modèles interactifs sont trop difficilement exploitables pour réaliser facilement des simulateurs pertinents, et qu'il faut d'abord améliorer l'informatisation des modèles mécaniques sous-jacents pour profiter d'applications efficaces.

## 2 Contexte

Modéliser des corps déformables relève de différents domaines scientifiques, principalement de la mécanique, de l'informatique et des méthodes numériques.

Pour modéliser des corps mous, il faut étudier les modèles physiques issus de la mécanique des milieux continus (théorie physique réaliste qui prédit entre autre le comportement de solides déformables). Ces modèles sont complexes et très peu adaptés à être programmés efficacement dans un ordinateur. Ils ont été initialement implémentés par les numériciens, qui dans un premier temps se sont intéressés en particulier à la précision sans grande contrainte de temps de calcul. Aujourd'hui ils tentent d'exploiter au mieux les machines pour rendre leurs algorithmes plus rapides afin de permettre des simulations de plus grande échelle ou d'une encore plus grande complexité. Orthogonalement, en synthèse d'animations, très tôt on a essayé de faire des modèles très rapides, avec lesquels on pouvait "jouer" interactivement, mais ces modèles se basent souvent sur des lois empiriques sans grande précision. Plus récemment, les chercheurs en informatique graphique essaient d'avoir des modèles plus proches de la physique dans une quête perpétuelle de réalisme. Un autre avantage d'utiliser des modèles issus de la mécanique pour ce domaine, serait la facilité de simuler des matériaux réels connus, sans avoir à adopter de manière ad hoc des paramètres à la main, pour obtenir le comportement désiré. On voit donc que si le cheminement est complètement opposé, les deux communautés se retrouvent aujourd'hui face à un même verrou scientifique : faire du rapide et précis, et retombent sur les deux contraintes fondamentales radicalement opposées...

L'équipe GMCAO du laboratoire TIMC a accumulé une expérience importante dans le domaine de la modélisation biomécanique de tissus mous pour l'assistance à la planification du geste chirurgical, avec des modèles de la face [Chabanas 03], de l'orbite [Luboz 04], et des voies aériennes supérieures [Gerard 03, Chouly 03]. Les praticiens peuvent ainsi tester et optimiser des stratégies opératoires avant l'intervention. Ces simulateurs mettent en œuvre des modèles "précis", mais lents. De ce fait, ces modèles ne peuvent pas être utilisés en mode interactif, dans lequel 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 œuvre in situ des gestes prévus n'est pas possible (modification de la configuration anatomique, problème technique, *etc*), ou en chirurgie en urgence. Deuxièmement, des simulateurs interactifs (voire temps réel) permettraient de faciliter, de compléter, voire d'améliorer la formation aux gestes chirurgicaux, et à plus long terme de limiter l'apprentissage sur animal ou sur pièces anatomiques. Des détails concernant les différents types de simulateurs envisagés dans le domaine médical sont proposés en annexe B.

L'élaboration de simulateurs sous contraintes de calculs temps réels et avec des objectifs de "réalisme" (ou plausibilité) comportemental 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 du laboratoire LJK a acquis un savoir-faire important en matière de simulateurs dynamiques interactifs [Debunne 01, Raghupathi 03]. Comme beaucoup de scientifiques membres de cette communauté de la synthèse d'animations, l'équipe EVASION 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.

C'est pourquoi l'équipe GMCAO plus orientée gros modèles numériques précis, mais lents et l'équipe EVASION connaissant bien la problématique de temps réel et les techniques d'accélération des calculs adaptés à l'informatique, décident de travailler ensemble.

Elles ne sont pas les seules, plusieurs projets les réunissent à des équipes diverses pour résoudre ces problèmes. Le projet IMAG MIDAS groupe aussi l'équipe AABC du GIPSA-lab, qui vise à mettre en place des modèles biomécaniques de la langue [Gerard 03, Gerard 04]. Pour cela un nombre extrêmement important de simulations dynamiques doit être effectué, ce qui pose de manière cruciale la question du temps de calcul, et justifie des efforts importants pour tester et adapter des algorithmes rapides. Le projet européen ODYSSEUS, visant à reproduire une cavité abdominale, intègre plusieurs équipes de l'INRIA, ainsi que des partenaires privés européens.

L'implémentation de cette thèse s'est effectuée principalement autour de deux plates-formes d'animation physique. La première, [AnimAL], était un travail collaboratif des membres de l'équipe EVASION. La seconde, [SOFA], plus récente et beaucoup plus ambitieuse, réunit les talents de diverses équipes de l'INRIA et du CIMIT, est en pleine effervescence et acquiert de plus en plus d'utilisateurs dans le monde académique et privé. Nos méthodes d'animation d'objets déformables par éléments finis y sont une brique de base.

### 3 Problématique

Concrètement, actuellement on sait assez bien faire, soit des simulations lentes et plutôt précises, soit des simulations rapides, qui commencent à être physiquement réalistes mais qui ne sont pas validées en terme de précision.

Les simulateurs précis, tels que les planificateurs médicaux actuels sont basés sur des outils classiques de calcul de structures issus de l'ingénierie et adaptés à des calculs statiques (déterminer un équilibre de forces) ou quasi-statiques (représenter le mouvement comme une suite d'états d'équilibre) mais très peu dynamiques (suivre l'évolution des déformations au cours du temps). Ils sont lents, avec des temps de calcul pour chaque image allant de plusieurs secondes (petits déplacements) à plusieurs minutes (grands déplacements), voire plusieurs heures (dans le cas de calculs dynamiques [Gerard 04]). Ils ne se prêtent donc pas à des applications interactives. Les méthodes numériques perfectionnées développées pour les éléments finis permettraient en principe d'obtenir une bonne précision sur les matériaux complexes qui composent le corps (peau, graisse, muscles,...). Toutefois ces méthodes sont très sensibles aux données (lois de comportement, conditions aux limites) et les résultats peuvent être très différents pour des données assez proches [Chabanas 02]. Or ces données sont justement mal connues. Les caractéristiques mécaniques sont très complexes et variables d'un individu à l'autre. La géométrie des patients n'est pas forcément connue avec une grande précision. Les gestes médicaux simulés sont très variables aussi en fonction du chirurgien, du patient et de toutes autres conditions extérieures. Pour ces raisons, il nous paraît difficile et aléatoire d'essayer d'adapter les simulateurs actuels aux contraintes d'interactivité et de précision. Notre conviction est qu'il est plus prometteur de se baser sur des méthodes numériquement performantes en leur intégrant quand et là où c'est nécessaire des améliorations sur la précision.

Dans cette optique, divers travaux issus de l'informatique graphique [Cotin 96, Delingette 99, Picinbono 00b, Müller 02], mais aussi récemment de mécaniciens [Miller 07] proposent des algorithmes efficaces pour résoudre les équations de la mécanique afin de permettre des simulations interactives.

Notre travail poursuit cette lignée. Des progrès restent à réaliser afin de faciliter leur utilisation dans des applications réelles. A notre avis, cela passe en grande partie par la construction simple d'un maillage efficace pour le temps réel, car cette étape reste problématique et il n'est pas évident de fabriquer de nouveaux exemples, et encore moins de créer automatiquement des modèles dépendants-patient pour des simulations interactives. Leur stabilité est aussi à améliorer, afin de les rendre très robustes face aux fortes contraintes qui peuvent être imposées par un utilisateur. La plupart des modèles interactifs exploitent abondamment la rapidité des intégrations explicites, alors que les formulations implicites semblent mieux adaptées aux simulations interactives grâce à leur grande stabilité et leur compromis rapidité/précision. Le temps réel nécessite une résolution extrêmement rapide des équations de la dynamique, et seules des solutions très approchées peuvent être évaluées (avec de grands pas de temps par exemple) en faisant perdre quelque peu la précision offerte par leur formulation basée sur les équations mécaniques. Ces approximations confèrent souvent un comportement "gélatineux" aux objets simulés qui mettent du temps à répondre aux contraintes ; il semble donc indispensable d'améliorer ce comportement dynamique en propageant plus rapidement les déformations. Enfin, afin de réaliser des simulations réalistes agréables à l'œil, il faut être capable d'animer des objets visuellement détaillés à une résolution mécanique grossière compatible avec le temps réel.

## 4 Contributions

Afin de faciliter l'utilisation de modèles physiques et d'améliorer leur comportement en temps réel, nous nous sommes concentré à obtenir facilement un maillage mécanique régulier en lui appliquant au mieux une mécanique rapide, stable et précise, même dans les cas difficiles, tels que pour des matériaux mal conditionnés.

■ **Maillage** Nous proposons d'utiliser une voxelisation de l'objet à animer comme maillage mécanique. Une telle représentation est automatique et offre un critère intuitif sur la résolution. De plus, elle fournit un maillage très régulier, très bien adapté numériquement pour des calculs rapides et stables. Une autre originalité est qu'elle peut être construite à partir de diverses données d'entrée, aussi bien des surfaces triangulées comme en fabrique un modéleur, que des données voxelisées issues de scanner médicaux. On montre qu'il est ainsi possible de faciliter considérablement l'animation de n'importe quel modèle quelque soit sa complexité et sa qualité, ou la prise en compte de modèles spécifiques à un patient.

■ **Visualisation** Dès que l'on veut représenter des objets réalistes comme dans les jeux vidéo ou en environnements virtuels, il est crucial d'avoir une belle qualité visuelle. Pour cela il faut être capable de découpler les résolutions servant au rendu de celles servant à la mécanique. A l'aide de la voxelisation, il est aisé d'animer avec peu d'éléments un maillage très détaillé. De plus, puisqu'il est possible d'animer facilement des données volumiques, nous proposons un rendu volumique efficace, qui se base sur le même maillage que la mécanique afin de limiter les coûts en calcul et en mémoire.

■ **Simulation mécanique** Une animation rapide et stable est proposée en appliquant des éléments finis corotationnels sur les voxels. De plus, puisqu'il est facile d'avoir des voxelisations (modèles mécaniques) à différentes résolutions, il est facile d'adopter une approche multirésolution. Pour cela, nous proposons une gestion simple et justifiée physiquement des nœuds en T. Il semble en effet indispensable de bien maîtriser le niveau de détails pour toutes méthodes interactives précises.



■ **Dynamique** Actuellement il est difficile de simuler des matériaux mal conditionnés en temps-réel, car ils impliquent des problèmes numériques. Nous étudions comment améliorer ces problèmes à l'aide d'approches hiérarchiques et multigrid afin d'améliorer la convergence des résolutions numériques et ainsi la propagation des déformations dans le milieu, dans le cas d'une intégration implicite qui nous semble la mieux adaptée pour calculer l'évolution du système dans le temps, grâce à sa stabilité et son compromis précision/vitesse.

■ **Validations** Une originalité de notre projet est d'apporter des éléments de comparaison, en confrontant notre approche avec d'autres méthodes numériques et des données réelles, afin d'éprouver sa validité. Nous proposons de comparer rigoureusement notre proposition à des méthodes de référence en terme de précision et d'interactivité.

Les problématiques concernant la gestion des collisions et les couplages avec d'autres méthodes sont assez orthogonales, donc on ne s'y attardera pas dans cette thèse. Par contre même si on ne s'intéresse pas explicitement à simulation de la découpe et de la déchirure des objets, ces traitements étant très dépendants de la méthode d'animation, on a veillé à ce que la mise en place de telles méthodes déjà existantes soit facilitée.

## 5 Organisation du document

Ce document commence par une revue du contexte scientifique avec un **état de l'art** sur les modèles déformables en informatique (chapitre 1), principalement centré sur les modèles interactifs. Nous analysons le pas entre les modèles proposés et ceux vraiment utilisés dans les applications existantes. Puis, pour bien être en mesure d'apprécier les méthodes basées sur la **mécanique des milieux continus**, nous faisons un bref rappel des notions essentielles (chapitre 2), pour donner les bases et l'intuition de son fonctionnement aux non-initiés afin qu'ils soient en mesure de suivre cette thèse. Nous détaillons ensuite les approches numériques basées sur cette mécanique.

Puis avec ces bases, nous sommes en mesure d'expliquer nos contributions. La première consiste en la fabrication d'une **représentation volumique** nécessaire à la résolution des équations de la mécanique (chapitre 3). Celle-ci est automatique, rapide, et fonctionne pour des représentations variées d'objets (surfacique ou volumique par exemple issue d'un scanner médical). Elle permet ainsi d'aisément et rapidement traiter des cas particuliers. Elle fournit des éléments réguliers à différentes résolutions adaptés à une approche temps-réel et multirésolution. Son originalité réside dans le fait qu'elle ne représente pas parfaitement l'objet et certains éléments ne sont que partiellement remplis.

Concernant la **mécanique**, notre apport réside principalement dans la prise en compte de cette structure volumique non-conventionnelle. Nous proposons un calcul des déformations rapide et stable avec un souci de qualité dans la précision des résultats (chapitre 4).

Puis nous proposons deux pistes pour améliorer la **propagation** des déformations dans un mouvement dynamique, avec l'objectif de bien retranscrire la raideur des matériaux (chapitre 5). Ces approches permettent de mieux simuler interactivement des matériaux mal conditionnés et de bien retranscrire les interactions.

Enfin nous comparons notre modèle à divers résultats théoriques, à des données réelles ainsi qu'à des méthodes numériques certifiées pour en étudier sa précision, dans le but de tester sa **validité** (chapitre 6), avant de conclure.



---

# **Contexte scientifique**



chapitre **1**  
Etat de l'art

---

---

<b>1</b>	<b>Introduction</b>	<b>12</b>
<b>2</b>	<b>Critères de qualité</b>	<b>12</b>
<b>3</b>	<b>Modèles non-physiques</b>	<b>14</b>
<b>4</b>	<b>Modèles physiques</b>	<b>14</b>
4.1	Champs de forces	15
4.2	Intégration du temps	18
4.3	Améliorations	20
4.4	Déchirure / découpe	23
4.5	Gestion des collisions	24
<b>5</b>	<b>Applications</b>	<b>25</b>
5.1	Informatique graphique	26
5.2	Simulateurs chirurgicaux	26
5.3	Plates-formes d'animation physique	27
<b>6</b>	<b>Discussion</b>	<b>27</b>

---

## 1 Introduction

L'animation physique en informatique est un domaine très large qui concerne un grand nombre de variétés d'objets et de comportements. Parmi les grandes familles de modèles physiques, on compte les corps rigides, les solides articulés, les solides déformables, les fluides (gaz et liquides). Chacune peut avoir affaire à des lois très particulières reflétant des besoins très variés et synthétisées par des algorithmes très différents. Leur unique point commun est d'implémenter les lois de la dynamique énoncées par Newton afin de créer un système dit *générateur*, dont l'état final résulte de lois appliquées à un état de départ pendant un court laps de temps sans nécessairement d'intervention externe durant l'application des lois. Les lois physiques choisies peuvent aller des plus simples aux plus complexes. Dans l'encore plus vaste domaine de l'animation en informatique, l'animation physique se discerne clairement des approches purement cinématiques telles que l'animation par squelettes suivant des positions clés, principalement utilisée pour l'animation de personnages.

Parallèlement à l'animation de ces corps, divers domaines sont à prendre en considération. Par exemple, pour faire interagir plusieurs objets entre eux et avec leur environnement, il faut être en mesure de gérer les contacts, qui passent par l'étude de la gestion des collisions. Celle-ci consiste en la détection d'une intersection de matières, qu'il faut être capable de modéliser pour enfin y réagir.

La physique ne sert pas qu'à créer des animations visuellement réalistes. Elle ajoute une nouvelle dimension d'interactivité en réalité virtuelle en offrant plus de sensations réalistes pour une meilleure immersion. En plus du visuel, des retours haptiques [Batter 71, Brown 94] ou sonores [Cadoz 99, van den Doel 01, Tsingos 07] peuvent aussi être modélisés en suivant des lois de la physique.

Tous ces domaines sont très complémentaires et doivent tous être étudiés et réunis dans l'idée de faire un "simulateur ultime". Chaque domaine est très pointu, mais peut être étudié assez indépendamment du fait de la diversité des lois mises en œuvre. On constate ainsi de nombreux grands thèmes de recherche très spécialisés en animation physique.

Dans cette thèse on ne s'intéresse qu'aux déplacements et déformations des solides déformables, et principalement pour leur rendu visuel, les parties haptique et sonore n'ayant pas été abordées.

Dans ce chapitre nous allons passer en revue les différentes méthodes proposées pour simuler des corps déformables.

## 2 Critères de qualité

L'animation physique peut avoir des objectifs très différents, allant du très simple pour faire un objet "plausible" aux simulations du réel avec une erreur maximale garantie. Afin de faire un état de l'art critique quand à la cohérence des méthodes existantes pour notre problématique de réalisme interactif, on doit tout d'abord mettre en avant nos propres critères de qualité.

Ces critères vont essentiellement se baser sur les propriétés qui nous semblent essentielles : la précision, la vitesse, la robustesse, la liberté du rendu visuel, la facilité de prendre en compte des modèles spécifiques issus de données médicales. De nombreux autres critères secondaires sont bien évidemment à prendre en compte, tels que la facilité d'implémentation, la possibilité d'adapter un algorithme de coupure ou de gérer facilement les collisions.

En terme de **précision** dans l'animation, on va différencier des grandes familles. Tout d'abord les modèles qui permettent de représenter un objet dont le mouvement est physiquement plausible, sans rigueur au niveau de la précision, tel que cela peut être le cas dans les jeux. Sans chercher la rigueur physique, il est possible à partir de lois simples de faire émerger des phénomènes complexes qui donnent illusion de la réalité [Luciani 84]. Dans notre contexte, on va commencer à être satisfait si

le modèle respecte une certaine plausibilité physique, sans garantir que c'est exactement le comportement qui serait arrivé, mais qui offre un comportement qui aurait pu arriver. Dans une ultime mesure, on va considérer les modèles très rigoureux, rimant avec très complexes et coûteux qui tentent de se rapprocher le plus possible de la réalité.

Un point très important à considérer, est la possibilité de **paramétrer les matériaux** en intégrant des mesures expérimentales issues de la mécanique afin de facilement modéliser des matériaux connus, et de pouvoir modéliser plusieurs valeurs de ces matériaux, étape nécessaire à la réalisation d'un modèle spécifique à un patient.

La **vitesse** de calcul est bien sûr un élément crucial du modèle désiré. Il doit être suffisamment rapide pour proposer des animations interactives. Offrir un compromis entre précision et vitesse est intéressant afin de proposer la meilleure solution approchée dans un laps de temps limité.

Il faut aussi considérer la **robustesse** du modèle, c'est-à-dire, sa résistance à ne pas diverger dans les configurations extrêmes ou numériquement mal-conditionnées. Il faut même noter que des configurations non-physiques telles que des inversions de matière ou d'aplanissement total peuvent arriver, d'autant plus dans le cadre d'applications interactives, dans lesquelles un utilisateur peut appliquer des contraintes irréalistes. Il est important que la simulation reste stable, continue à fonctionner et tente de fournir la solution la plus adaptée pour pouvoir continuer la manipulation.

L'interaction finale avec l'utilisateur passant par des activités sensorielles, il est important que celles-ci soient le plus détaillées possible. Dans notre cas, on pense principalement à la **qualité du rendu**. En effet, un simulateur ne pourra pas se contenter d'une mécanique irréprochable, il devra être en plus visuellement réaliste. Cela signifie généralement d'avoir une résolution d'affichage détaillée sur laquelle pourront être appliquées toutes les techniques de rendus réalistes mises au point en synthèse d'images. Il semble donc important que l'objet animé soit capable de s'afficher en utilisant une résolution différente de celle utilisée par la mécanique, qui soit plus adaptée à l'affichage.

Principalement dans le cadre des simulations chirurgicales, il est très important d'être capable d'animer facilement n'importe quel morphologie afin de faire des applications **spécifiques au patient**. Cela peut-être aussi très pratique en graphisme pour ne pas compliquer la tâche de l'animateur qui est avant tout un artiste et qui n'a pas forcément à comprendre ce qui va être bien adapté du point de vue de la mécanique. On va préférer aussi les méthodes permettant de travailler avec différents types de données ; dans le médical on pense aux images scanner qui sont les principales sources de données sur la morphologie d'un patient.

Un dernier critère nous semble important d'être souligné, même s'il n'a pas été étudié explicitement : il s'agit de la facilité d'adapter une méthode de **découpe**. Celle-ci rimant généralement avec la refonte du maillage sous-jacent, il est important de penser à comment va se comporter le modèle mécanique : va-t-il encore pouvoir fonctionner ? Va-t-il devoir refaire énormément de calculs coûteux pour se mettre à jour ?

Dans la suite, on va lister des méthodes respectant au moins un des critères ci-dessus, en regardant les avantages que l'on peut en tirer, et ce qu'il faudrait y ajouter pour qu'elles respectent d'autres de nos critères.

### 3 Modèles non-physiques

Les premiers modèles permettant de déformer des objets se basaient uniquement sur leur géométrie sans prendre en compte leur aspect physique. Les principales idées consistent à avoir des surfaces interpolées sur des points de contrôles mobiles, l'objet pouvant être défini procéduralement comme une surface paramétrique. Les Free Form Deformation [Sederberg 86] permettent de donner des points de contrôle à une surface déjà existante.

Ces approches ont été initiées par [Barr 84], maintenant encore, les méthodes de déformation purement géométriques restent un domaine de recherche actif [Alexa 05], la principale difficulté étant de trouver et d'intégrer des contraintes rendant la déformation naturelle, telle que la conservation du volume [Hirota 99] ou des détails [Alexa 06, Botsch 06].

Avec de telles déformations, il est possible de créer une animation en faisant évoluer la position des points de contrôle dans le temps selon une trajectoire définie par l'utilisateur.

Au premier abord, ces modèles non-physiques nous intéressent peu dans notre étude, mais ce travail est la base sur laquelle sont venues s'inclure les lois mécaniques. Grossièrement, on pourrait considérer un modèle physique comme reposant sur un tel modèle géométrique sur lequel des lois physiques sont appliquées aux points de contrôles.

### 4 Modèles physiques

Afin de rendre les déformations plus réalistes, plutôt que d'utiliser des lois de déformations empiriques, des auteurs se sont tournés vers les hypothèses des mécaniciens qui tentent depuis des dizaines d'années d'évaluer la déformation de solides. Ils ont ainsi donné naissance aux premiers modèles purement mécaniques.

Terzopoulos est le pionnier dans cette approche. Dès 1987, il a proposé le premier modèle déformable mécanique en synthèse d'images [Terzopoulos 87]. Pour cela, il a pris telles quelles les méthodes complexes de résolutions issues de la mécanique avec pour objectif de fabriquer des images. En terme de qualité de déformations, les résultats étaient déjà très bons puisque les équations de la mécanique sont respectées. Par contre, vu la puissance limitée des machines de l'époque, l'ambition n'était pas du tout tournée vers le temps réel, et du coup, peu de grosses optimisations algorithmiques ont été mises en œuvre. L'approche ressemblait plus à celle des numériciens en mathématiques appliquées avec en plus le souci de fabriquer des images.

Depuis ce travail préliminaire, une multitude d'approches a été proposée pour déformer des objets avec différents objectifs, que peuvent être la rapidité, la diversité des comportements, la robustesse...

A cette étape, on se doit de souligner le fait qu'il faut faire attention à ne pas réinventer la roue, car déjà beaucoup de choses ont été proposées par les mécaniciens, qui travaillent sur cette thématique depuis très longtemps. Par exemple la 'warping stiffness' de Müller [Müller 02] se rapproche étrangement des méthodes dites corotationnelles en mécanique [Felippa 00], ou encore l'approche multirésolution de CHARMS [Grinspun 02] s'apparente aux éléments finis dits hiérarchiques en mécanique [Zienkiewicz 82]. De notre avis, il semble très ambitieux de faire mieux que les mécaniciens pour prédire ces mouvements, nous pensons qu'il est préférable de comprendre les lois déjà proposées, de les analyser en tentant de voir comment elles peuvent s'intégrer dans un programme. Certaines seront peut-être plus facile à implémenter, tiendront moins de place en mémoire, seront très rapides à résoudre en utilisant du parallélisme, *etc.* Notre travail consiste donc en partie à jouer avec l'aspect numérique et algorithmique afin d'intégrer au mieux ces lois dans l'ordinateur.



Fréquemment on peut voir la catégorie des modèles physiques découpée en modèles continus et modèles discrets (*i.e.* milieu continu *vs* masses-ressorts). D'un point de vue mécanique, un modèle est dit discret lorsque les équations peuvent être écrites indépendamment pour chaque particule discrétisant l'objet, alors que dans les modèles continus le comportement va dépendre de l'intégralité de la matière et les états peuvent être évalués à n'importe quel point de l'objet. Mais d'un point de vue algorithmique, la différence n'est pas si claire, car même avec des modèles issus de la mécanique des milieux continus, une pratique commune est de "lumper" les masses, c'est-à-dire de concentrer la masse sur les particules afin de pouvoir gérer chaque particule indépendamment. Avec une telle vision, les éléments finis deviennent proches des systèmes masses-ressorts qui sont pourtant fréquemment présentés comme les deux grandes méthodes radicalement différentes.

De plus, dans tous les cas, pour être modélisé, l'objet doit être discrétisé. Les degrés de liberté sur lesquels vont être appliquées les équations sont couramment appelées particules. L'approche la plus simple est de considérer que ces particules représentent elles-mêmes l'objet. Elles peuvent aussi être des points de contrôles d'une surface paramétrique. Par exemple, très rapidement, est née l'idée de mettre de la physique sur les points de contrôles d'une FFD (Free Form Deformation) [Faloutsos 97]. Cette idée continue d'être exploitée, par exemple dans [Capell 02b] mais aussi dans cette thèse. Les particules peuvent aussi représenter des squelettes définissant une surface implicite (par exemple par metaballs) : solides déformables [Cani 93], solides inélastiques et fluides avec le cas des SPH (Smoothed Particle Hydrodynamics) [Desbrun 95, Müller 03, Clavet 05].

Qu'il soit discret ou continu, on peut considérer un modèle physique comme un ensemble de degrés de liberté (ddl) qui vont interagir en obéissant à des forces internes. Des chargements externes peuvent être appliqués sur ce modèle, principalement la gravité, des champs de forces, des contraintes (telles que fixer un ddl, contraindre sa position sur une ligne ou limiter sa vitesse). Un modèle physique a aussi une loi d'évolution. Elle peut être statique, pour retrouver un état d'équilibre annulant les forces sans prendre en compte ni les masses ni le temps. C'est ce qui est utilisé quand seule la configuration finale est intéressante comme en ingénierie mécanique et dans les planificateurs chirurgicaux. Mais dès lors que l'on cherche à avoir un modèle interactif avec une évolution dans le temps, il faut se tourner vers des méthodes dynamiques qui prennent en compte les accélérations et les vitesses.

On appelle système de particules, une approche utilisant une résolution dynamique qui va suivre l'évolution dans le temps de degrés de liberté en respectant les équations de la dynamique introduites par Newton. Ce sont ces méthodes qui se rapprochent le plus de la mécanique classique, dite lagrangienne et c'est donc vers celles-ci qu'on va se tourner. Numériquement cela se résume à être capable de calculer deux choses distinctes : d'un côté des **champs de forces internes** en fonction de contraintes extérieures, et de l'autre un nouvel état après un court laps de temps à l'aide d'une méthode d'**intégration du temps**. C'est pourquoi dans cette section nous nous appliquons à lister les méthodes numériques déjà proposées en informatique graphique dans ces deux domaines.

## 4.1 Champs de forces

La force agissant sur une particule dépend de la déformation de l'objet dans un voisinage plus ou moins grand autour de cette particule. Elle peut dépendre de toutes les particules de l'objet (Lennard-Jones par exemple), ou de simplement quelques particules voisines. Ce voisinage peut être prédéfini par un maillage (comme pour les masses-ressorts ou les éléments finis) ou bien être évalué et varier au cours du temps (méthodes dites *meshless*).

■ **Lennard-Jones** c'est une des première loi physique informatisée de façons plus ou moins approximées [Miller 89, Terzopoulos 91, Luciani 91]. Elle est plutôt associée aux comportements de type fluide. Dans ce cas, chaque particule est indépendante mais influence et est influencée par toutes les autres. De par cette globalité des interactions, les forces sont assez lentes à calculer. Des optimisations sont possibles pour ne pas calculer des influences insignifiantes en prenant par exemple un voisinage gaussien significatif. Il n'est pas facile de représenter des solides élastiques avec cette méthode, plutôt adaptée aux objets "pâteux" du type fluides visqueux. D'un point de vue macroscopique, il est de même très difficile d'associer des paramètres physiques réels aux matériaux

■ **Masses-ressorts** Il s'agit de la loi la plus simple faisant intervenir un maillage, limitant l'influence d'une particule à son voisinage direct [Breen 92, Cover 93]. Cette force est beaucoup plus rapide à calculer. Plusieurs lois de ressorts sont possibles entre les particules voisines. On peut retenir qu'un ressort est un élément fini 1D plongé dans un univers 3D donc très peu adapté pour représenter des objets volumiques et même des surfaces. On ne sait pas lui assigner des propriétés tridimensionnelles mécaniques mesurées. Des méthodes ont permis de les rendre très stables, par exemple en limitant leur élongation [Provot 95]. Très rapides et faciles à implémenter, ils sont donc très bien adaptés aux simulations en synthèse d'images de type jeux vidéo mais sont peu adaptés aux simulations précises de type médical. [Bourguignon 00] donne une solution permettant de gérer l'isotropie dans des objets volumiques simulés par des ressorts, tout en précisant qu'il est impossible de retrouver les paramètres mécaniques pour les ressorts correspondant à ceux issus de la mécanique des milieux continus dans un contexte général. Tout de même, pour certains cas particuliers, [Lloyd 07] montre que les ressorts peuvent être équivalents aux éléments finis. Notons aussi qu'il est possible d'améliorer le comportement volumique de cette représentation 1D. Des méthodes récentes permettent de mieux prendre en compte les volumes grâce à des "mémoires de forme" [Promayon 96] ou même en considérant l'objet rempli d'un fluide parfait dans lequel les variations de volumes fournissent des pressions transformées en forces à la surface [Sundaraj 02, Matyka 03].

■ **Méthodes des éléments finis** Pour une meilleure précision, on préfère les modèles suivant les lois issues de la mécanique des milieux continus. Les équations étant plus complexes, ils sont généralement plus longs à calculer, mais ils sont bien plus précis, et surtout leur précision est évaluable, car ils permettent de prendre en compte des propriétés mesurées de matériaux réels. Diverses méthodes numériques permettent de calculer ces lois, les plus célèbres sont la méthode des éléments finis [Cotin 96, Delingette 99, Picinbono 00b, Müller 02] et la méthode de différences finies [Terzopoulos 87, Debunne 01].

Les grandes avancées concernant l'animation d'objets déformables avec les contraintes d'interactivité, de précision et de qualité de rendu peuvent être rappelées chronologiquement :

- [Cotin 96] propose un simulateur quasi-statique interactif en évaluant de petites déformations par des combinaisons linéaires de déformations unitaires précalculées.
- [Delingette 99] fournit une écriture en "masses-tenseurs" qui optimise l'intégration dynamique explicite, et simplifie les mises à jours dans le cas de changements topologiques.
- [Picinbono 00b] étend cette représentation à un calcul non-linéaire des déformations afin de gérer les déplacements en rotation.
- [Debunne 01] présente un modèle multirésolution.
- [Müller 02] donne un calcul corotationnel des déformations plus rapide et simplifiant l'utilisation d'intégrations implicites plus stables.

■ **Meshless** Il s'agit d'une méthode très élégante basée sur la mécanique des milieux continus qui permet de calculer les déformations d'une particule en redéfinissant son voisinage et ses dérivées spatiales à chaque pas de temps, sans ainsi avoir besoin d'un maillage volumique. Elle est en ce sens aussi précise, plus générale que la méthode des éléments finis, mais aussi plus complexe à mettre en place. Elle est très bien adaptée aux coupures et aux milieux fluides-élastiques. Par contre, elle est relativement longue à calculer, et n'est pas encore adaptée à des simulations interactives [Müller 04b, Pauly 05].

■ **Coordonnées réduites par décomposition modale** Dans cette approche [Hauser 03, Barbič 05], l'idée est d'exprimer les déformations de l'objet par un jeu de coordonnées réduites obtenues par analyse modale. Il est en effet possible en prétraitement d'étudier tous les modes vibratoires de l'objet et de ne conserver que les composantes principales. On obtient ainsi des jeux de déplacements de base qui donnent des déformations globales de l'objet. L'idée de ne considérer qu'un ensemble de déplacements globaux avait été initiée par [Pentland 89, Witkin 90]. En interpolant ces déplacements, on peut obtenir toute une palette complète de déformations. L'avantage est qu'avec ce jeu très réduit de coordonnées, il est très rapide d'obtenir une déformation plausible pour des mouvements globaux, quelle que soit la complexité du modèle de départ (quels que soit le nombre de degrés de liberté initial). L'inconvénient est qu'il est difficile d'obtenir des déformations très locales, les déformations sont réduites à la palette de déplacements globaux dont on dispose. Cela rend cette approche très profitable en synthèse d'animations, car un résultat très réaliste peut être obtenu rapidement. Malheureusement, cette approche est peu adaptée aux simulations chirurgicales pour lesquelles les manipulations locales sont très importantes.

■ **Shape-matching** Il s'agit du dernier modèle proposé en graphique interactif en date [Müller 05]. L'idée est découper l'objet en sous-groupes (grossiers hexaèdres englobants et interpolants). Pour chaque groupe, une transformation géométrique entre la position au repos et la position déformée est évaluée. Cette transformation peut être un recalage rigide, ou un recalage élastique plus coûteux mais plus précis. Des déformations sont déduites dans des repères comparables à partir des positions des points de l'objet au repos et déformés après le recalage. Des forces sont ainsi créées sur les points pour tenter de les ramener vers leur position non déformée. Une version superposant ces groupes permet encore plus de souplesse dans la déformation [Rivers 07]. Ces modèles sont visuellement très impressionnants par rapport à leur coût de calcul, même s'ils n'incluent pas de force de restauration de volume. Malheureusement, ils peuvent difficilement être considérés lorsque l'on désire une grande précision car la qualité de leur déformation et le paramétrage des matériaux sont complètement empiriques. Ils semblent être la proposition la plus adaptée aux jeux vidéo, de par leur souplesse, leur rapidité et leur stabilité.

Parmi tous ces moyens de calcul des forces internes d'un objet déformable, le seul moyen vraiment précis et qui cherche à refléter la réalité en permettant la prise en compte d'un paramétrage de vrais matériaux est la mécanique des milieux continus (à travers la décomposition en éléments finis et la méthode meshless). C'est pourquoi nous allons nous tourner vers elle, et nous la détaillerons par la suite dans le chapitre 2.

## 4.2 Intégration du temps

Dans notre cas, on ne cherche pas explicitement un état d'équilibre répondant aux contraintes appliquées, mais plutôt l'évolution des déformations au cours du temps.

Un système de particules correspond à un problème du second ordre de la forme :

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}_{ext} \quad (1.1)$$

où  $\mathbf{u}$  correspond aux déplacements et  $\mathbf{f}_{ext}$  aux forces externes appliquées sur toutes les particules.  $\mathbf{M}$  est lié à la masse,  $\mathbf{C}$  à l'amortissement et  $\mathbf{K}$  à la rigidité, elles traduisent les interactions entre les particules. Le temps est traité comme un paramètre. Les dérivées première  $\dot{\mathbf{u}}$  et seconde  $\ddot{\mathbf{u}}$  des déplacements représentent respectivement vitesses et accélérations. Les matrices  $\mathbf{M}$ ,  $\mathbf{C}$  et  $\mathbf{K}$  s'obtiennent en regroupant les équations en chaque sommet ; elles peuvent dépendre de la géométrie et donc devoir être recalculées à chaque déformation. Le calcul des forces internes  $\mathbf{K}\mathbf{u}$  dépend de la méthode utilisée, présentée en 4.1.

Plus généralement les forces internes peuvent être définies non-linéairement  $\mathbf{f}_{int}(\mathbf{u})$  ; la résolution du système fait toutefois introduire les variations infinitésimales de forces représentées par les matrices  $\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}$  et  $\mathbf{C} = \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{u}}}$ .

On résout donc un problème à chaque pas de temps, en partant d'un instant initial  $t_0$  où les déplacements  $\mathbf{u}(t_0)$  et les dérivées premières  $\dot{\mathbf{u}}(t_0)$  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 :

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

où  $\mathbf{q}(t) = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix}$  représente l'état (position et vitesse) d'une particule au temps  $t$  et  $h$  le pas de temps.

Il existe quantité de schémas d'intégration, qui se distinguent par leur ordre de précision, leur convergence, 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. La convergence signifie que plus le pas de temps  $h$  se rapproche de 0, plus la solution analytique se rapproche de la solution réelle. On peut distinguer deux grandes classes de méthodes :

■ **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.

**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) :

$$\mathbf{q}(t+h) \approx \mathbf{q}(t) + h\dot{\mathbf{q}}(\mathbf{q}, t)$$

■ **méthodes implicites** les dérivées premières et secondes en fin de pas sont utilisées pour calculer la valeur suivante. La stabilité est meilleure, mais chaque pas de temps exige la résolution d'un système, couramment lié à un calcul itératif, avec les problèmes associés à tout calcul itératif : les paramètres algorithmiques et le choix de critères d'arrêt.

**Exemple :** le schéma d'Euler semi-implicite décrit dans [Baraff 98] (avec l'amortissement de Rayleigh, combinaison linéaire de la raideur et la masse  $\mathbf{C} = \alpha\mathbf{K} + \beta\mathbf{M}$ ) :

$$\begin{cases} ((1-h\beta)\mathbf{M} - h(h+\alpha)\mathbf{K})\Delta\dot{\mathbf{u}} = h(\mathbf{f}(t) + h\mathbf{K}\dot{\mathbf{u}}) \\ \dot{\mathbf{u}}(t+h) = \dot{\mathbf{u}}(t) + \Delta\dot{\mathbf{u}} \\ \mathbf{u}(t+h) = \mathbf{u}(t) + h\dot{\mathbf{u}}(t+h) \end{cases}$$

Ainsi, les méthodes explicites avancent directement vers l'inconnu d'après les états actuels pour fournir une solution assez rapidement. Par contre la stabilité n'est pas garantie, et les erreurs s'accumulant peuvent faire diverger le système. C'est pourquoi il est nécessaire d'utiliser des petits pas de temps car plus l'intervalle d'intégration est petit, moins l'erreur sera grande. Les pas de temps doivent par ailleurs être inversement proportionnels à la racine carrée du matériau simulé, plus le matériau est raide, plus le pas de temps doit être petit [Belytschko 76, Miller 05] (la condition de Courant impose que  $h < d_{min} \sqrt{\frac{\rho}{\lambda+2\mu}}$  avec  $d_{min}$  la plus petite distance entre deux particules voisines,  $\rho$  la masse volumique du matériau, et  $\lambda + 2\mu$  relatif à la raideur du matériau cf. chapitre 2 section 2.5).

Ces méthodes sont donc peu adaptées aux matériaux raides dans le cadre de simulations rapides. De l'autre côté, les méthodes implicites regardent en avant pour avancer, en choisissant un état d'arrivée à l'équilibre. Elles sont ainsi inconditionnellement stables mais aussi plus lentes à calculer car elles demandent la résolution d'un système d'équations. Dans ce cas, la taille du pas de temps est moins un problème et de grands pas de temps et des matériaux raides peuvent être simulés. Par contre, dans ce cas, le système est plus difficile à résoudre, et obtenir une bonne solution approchée coûte cher, les méthodes de résolution convergeant lentement. Tout de même, un avantage des méthodes implicites est la possibilité de choisir la qualité de la solution approchée, tout en étant toujours stable. Plus la solution est approximative, plus le calcul est rapide et la simulation interactive est possible, mais plus le comportement des matériaux est ramolli en ne répondant plus assez rapidement aux contraintes. De plus, l'erreur due à l'approximation des solutions induit une certaine viscosité.

Il n'existe aucune recette miracle pour définir quel est le meilleur intégrateur, chacun ayant ses caractéristiques, le mieux est de les comparer concrètement pour en déduire le plus adapté en fonction des besoins. Pour donner des exemples, dans le cadre de simulation de crash tests automobile, ce sont des intégrateurs explicites qui sont utilisés [Kirkpatrick 01]. Vu la rigidité des matériaux, de très petits pas de temps sont utilisés. Quand aucune contrainte de temps de calcul est fixée, il semblerait qu'avec de très petits pas de temps, ce soient les méthodes explicites qui offrent les solutions les plus proches d'un comportement réel (RungeKutta2/MidPoint offrant la meilleure rapidité pour une précision donnée). Dans la communauté graphique, l'intégration implicite a été initialement utilisée [Terzopoulos 87]. Mais à cause de sa grande complexité due à une résolution très lente, les méthodes explicites lui ont été préférées pour le temps réel, se limitant à des matériaux mous [Delingette 99, Debunne 01]. Depuis que des méthodes de résolutions rapides ont été proposées, l'intégration implicite a retrouvée une place privilégiée dans l'animation interactive grâce à sa stabilité qui ne nécessite pas d'adapter le pas de temps [Baraff 98, Desbrun 99, Hauth 01, Volino 05].

Un grand nombre de méthodes d'intégration du temps sont détaillées et comparées dans [Eberhardt 00, Volino 01]. On peut se rendre compte que dans tous les cas, rien n'est bien adapté pour simuler précisément et rapidement des matériaux mal conditionnés (tels que raides ou quasi-incompressibles). Dans le cas explicite il faut faire énormément de petits calculs (car petits pas de temps) ; du côté implicite il faut calculer une coûteuse solution approchée correcte (demandant beaucoup d'itérations pour converger).

### 4.3 Améliorations

Dans le cas de simulations interactives, il est très important d'assurer une complexité minimale par rapport au problème.

La complexité d'un modèle physique dépend des éléments suivants :

- **la taille du système** : autrement dit le nombre de degrés de liberté
- **les lois d'interactions** : la construction du système traduisant les interactions entre les degrés de liberté peut être plus ou moins complexe (par exemple des ressorts sont plus faciles à traduire que des éléments finis), et le système en résultant peut être plus ou moins délicat à résoudre (linéaire ou non, conditionnement...)
- **les algorithmes de résolution** : une fois le système en place, l'algorithme calculant sa solution a une place très importante dans la complexité qui dépend de la convergence de la résolution. Certains algorithmes sont plus rapides, mais ne correspondent qu'à certains systèmes. Par exemple, concernant les systèmes linéaires symétriques, Gauss-Seidel ne fonctionne bien que sur des matrices à diagonale dominante. Si la matrice est définie positive, le gradient conjugué sera plus performant.

Par conséquent, il est possible de mettre en œuvre des procédés, plus algorithmiques, permettant d'accélérer les calculs relativement orthogonalement aux formulations des calculs des forces et d'intégration du temps. Même si elles sont très importantes, voire indispensables, du fait que ces méthodes peuvent relativement être appliquées à n'importe quel type de calcul, elles apparaissent à nos yeux comme des améliorations. Une scène bien modélisée aura des degrés de liberté et des interactions de nature adaptée (modèles hybrides), un nombre de degrés de liberté raisonnable et bien placé (multi-résolution, adaptativité), des méthodes de résolution à la meilleure convergence possible (méthodes hiérarchique, multigrid, parallélisme).

#### 4.3.1 Modèles hybrides

Il est évidemment possible de mélanger plusieurs modèles mécaniques afin d'animer différents objets ensemble, mais ce mélange est possible pour un même objet. L'objectif principal est d'économiser en coût de calcul en utilisant la méthode adaptée la moins coûteuse. En interne, plusieurs modèles sont alors utilisés. Les couches peuvent être choisies au départ suivant des connaissances a priori : si on sait que telle partie est un os, on l'animerait comme un solide, et la peau comme du déformable. Mais elles peuvent aussi être adaptatives et ainsi varier en fonction de critères. [Cotin 00] propose de simuler un objet déformable comme un corps rigide tant qu'il n'est pas déformé pour économiser des calculs, puis de résoudre ses petites déformations par la méthode des éléments finis linéaires et enfin d'utiliser un calcul non-linéaire des déformations lorsqu'elles sont importantes.

### 4.3.2 Multirésolution

La multirésolution est une approche classique en synthèse d'images, et en informatique en général, permettant de concentrer les calculs uniquement là où ils sont le plus nécessaires ou de manipuler à différentes échelles en raffinant les zones d'intérêt.

Nous proposons une classification, utilisant les termes relatifs à la multirésolution trouvés dans la littérature. Un schéma récapitulatif est donné en Figure 1.1.

Pour nous le terme **multirésolution** est le plus général et qualifie toutes méthodes utilisant plusieurs résolutions. Que la résolution soit fixe ou varie au cours de l'animation.

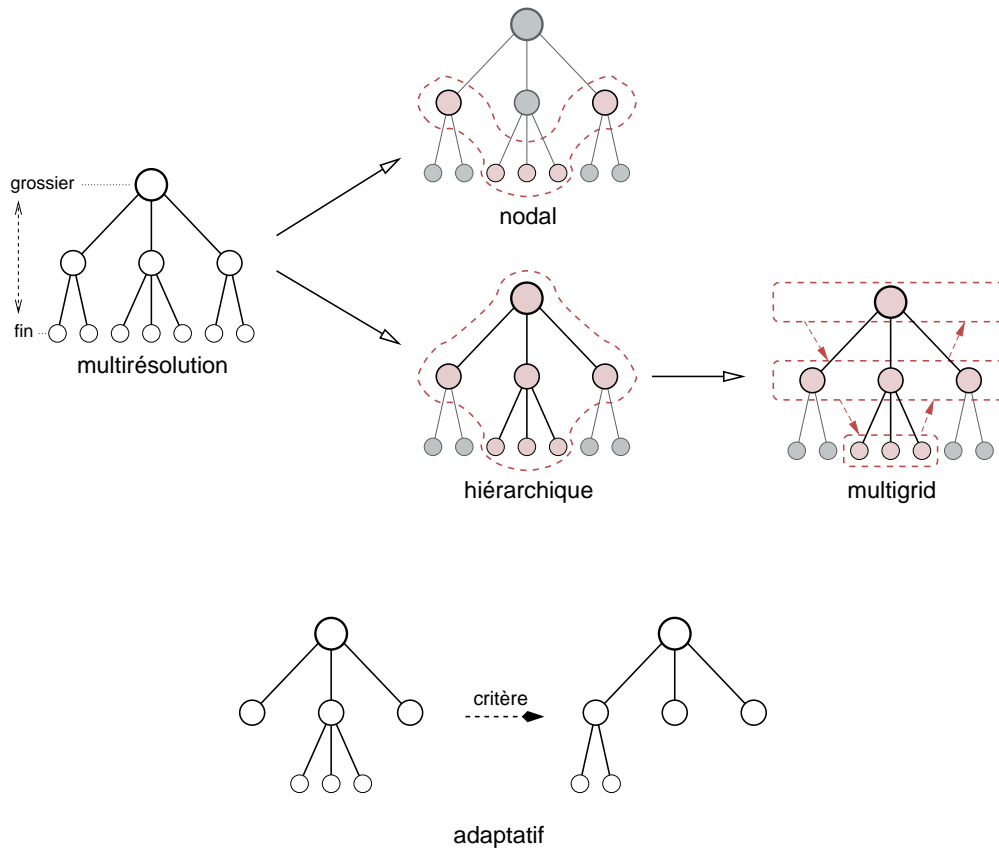


FIG. 1.1 – Diverses déclinaisons de la multirésolution

Une méthode **adaptative** est une méthode multirésolution pour laquelle les niveaux de résolutions utilisés varient au cours de l'animation, s'adaptant en fonction de critères. Ces critères peuvent être variés. Le plus classique en animation d'objets déformables est celui qui raffine les parties où les contraintes ou les courbures sont fortes [Debunne 01]. [Wu 01] compare plusieurs critères basés sur les courbures, les états internes de déformations ou de contraintes. Dans le cas particulier où le critère concerne les ressources du système, la méthode est appelée **temps critique** et essaie de raffiner tant que les ressources permettent de respecter une limite de temps. On retrouve classiquement ce type de critère en détection de collisions [Dingliana 00, Klein 03, Mendoza 05], mais aussi en animation de corps déformables [Dequidt 05]. Il faut noter que pour le temps réel, l'adaptatif demande une structure très appropriée, permettant un changement de résolution rapide, sans latence.

La notion d'approche **hiérarchique**, à la différence de **nodale**, signifie que plusieurs niveaux de résolutions sont utilisés en parallèle pour résoudre le problème. Dans l'approche nodale plus classique, au contraire, seul le niveau le plus fin est considéré. Le hiérarchique traite donc le problème à un niveau nodal et global simultanément, ce qui permet de converger plus rapidement vers la solution, en gérant à la fois des déformations globales et locales.

La résolution **multigrid** est une approche hiérarchique particulière pour laquelle plusieurs niveaux de résolutions sont utilisés en suivant des règles spécifiques où l'idée générale est d'utiliser les niveaux plus grossiers comme correcteurs du niveau fin. Plus de détails concernant le multigrid seront présentés dans le chapitre 5 section 3

Les modèles **multi-couches** réfèrent généralement aux modèles hybrides utilisant différents modèles en internes, comme présenté en 4.3.1.

Dans la littérature, on retrouve plusieurs travaux multirésolution proches du domaine de l'animation physique de corps déformables, les plus notables sont décrits ci-après.

[Gain 01] a proposé des interpolations hiérarchiques pour les FFD ; même si ces travaux ne touchent pas directement l'animation adaptative, le schéma de raffinement peut sûrement être exploité. [Chenney 97] est le premier à animer des objets à différentes échelles. Il propose de continuer à faire vivre très grossièrement un objet qui n'est plus dans le champs de vision, afin que s'il y revienne, son état ait quand même évolué de façon cohérente tout en limitant les coûts.

En animation physique d'objets déformables, [Hutchinson 96, Howlett 98] ont raffiné des surfaces 2D animées par des systèmes masses-ressorts. [Ganovelli 99, Ganovelli 00b] a proposé une subdivision de systèmes masses-ressorts en trois dimensions, à partir d'arbres octaux puis de subdivisions tétraédriques, tout en essayant de préserver la cohérence des propriétés mécaniques à tous les niveaux.

La technique a été portée pour les éléments finis tétraédriques, en utilisant plusieurs maillages volumiques précalculés représentant l'objet à plusieurs résolutions [Wu 01]. [Debunne 01] étend cette méthode en ajoutant le traitement de points fantômes interpolés, qui permettent d'utiliser des maillages tétraédriques non-emboîtés. Il ajoute aussi un pas de temps adaptatif. Son critère d'adaptation se déroule au cours du temps de façon géométrique, le raffinement s'effectuant là où les courbures des déplacements sont les plus fortes. Ce même critère d'adaptativité a été repris dans [Dequidt 05] pour un octree d'éléments finis englobant l'objet déformé. Dans ce travail, ont en plus été ajoutés des critères concernant les ressources système, afin de permettre un taux de calcul minimal pour garantir l'interactivité. L'approximation des solutions données pour chaque pas de temps est donc variable et dépend de la difficulté à résoudre le système.

Une décomposition en ondelettes des fonctions d'interpolation d'éléments finis a été présentée dans [Grinspun 02], à la manière des éléments finis hiérarchiques. Elle propose une approche très générique de la multirésolution, en limitant les problèmes soulevés aux jointures de maillages de différentes tailles.

Afin d'exploiter au mieux les différents niveaux de résolution, des approches multigrid ont été proposées sur des maillages tétraédriques non-emboîtés. [Georgii 06] propose cette approche à la fois pour un calcul linéaire, corotationnel et non-linéaire des déformations. [Otaudy 07] propose une méthode multigrid adaptative basée sur [Brandt 77] pour laquelle il intègre une méthode de détection de collisions, afin de vraiment exploiter la représentation hiérarchique, et propose ainsi un modèle d'animation multirésolution assez complet.

Dans tous ces travaux, les principaux problèmes liés à la multirésolution soulevés sont :

- la cohérence des propriétés mécaniques à chaque niveau. Les raideurs, masses... doivent simuler un comportement semblable (à une précision près) à tous les niveaux.



- la présence de nœuds en T aux jointures entre différents niveaux de résolution, car ces nœuds ne sont pas des ddl libres et subissent des contraintes d'interactions particulières.
- les transitions d'un niveau à l'autre dans le cas de méthodes adaptatives. Il est en effet important de trouver un critère qui lissera les changements de résolutions, et qui fera intervenir des opérations de passage peu coûteuses pour éviter les ralentissements.

Sont plus rarement soulevés les problèmes liés à la fabrication des maillages, étape pourtant déjà bien compliquée pour obtenir un seul niveau de résolution. Dans [Grinspun 02, Georgii 06], les niveaux fins sont obtenus en raffinant les niveaux grossiers, ce qui fait perdre à notre sens le réel intérêt d'une méthode multirésolution pour le temps réel. L'idée étant plutôt de simplifier certains calculs, donc de rendre certaines parties du système plus grossières, plutôt que d'encore plus complexifier certaines zones. Ce problème de maillage est pour nous un des facteurs très limitants des méthodes multirésolutions en animation actuellement, même si [Capell 02b, Dequidt 05] ont ouvert la voie vers des maillages à différentes résolutions plus automatiques. [Capell 02b] représente les objets comme étant des surfaces de subdivision interpolées dans des "éléments fictifs". Dans cette approche, les objets ne sont pas maillés directement, mais englobés dans un continuum plus facilement maillable, ce qui permet de facilement construire des maillages à différentes échelles, et il propose ainsi une approche multirésolution intéressante, dont les travaux qu'on propose s'inspirent fortement.

### 4.3.3 Parallélisme

L'évolution actuelle des architectures matérielles s'oriente de plus en plus vers le parallélisme. Celui-ci peut être mis en jeu de façons variées, comme les multiprocesseurs ou le multi-cores, les clusters de PC, ou les processeurs vectoriels tels que les récents GPU programmables. Pour être capable de simuler de grandes scènes complexes contenant plusieurs objets détaillés, il est indispensable de profiter de ces moyens de calcul. Profiter pleinement de ces architectures est à la fois très spécifique à l'architecture elle-même, mais aussi à l'algorithme à paralléliser. D'énormes gains ont pu ainsi être démontrés en animation physique, que ce soit sur un simulateur en général [Zara 03, Zara 04], sur un modèle déformable seul [Georgii 05] ou en détection de collisions [Myszkowski 95, Shade 98, Heidelberger 04, Govindaraju 07]. Plusieurs solutions de parallélisme sont étudiées dans la plate-forme d'animation SOFA [SOFA]. Les études préliminaires montrent des résultats impressionnants en utilisant le GPGPU avec un gain allant jusqu'à  $\times 10$  pour les objets déformables. Même si nous sommes convaincus de l'importance de développer des méthodes profitant du parallélisme, nous nous sommes pas directement intéressés à ce domaine, car cela reste un travail qui nous semble complémentaire, et surtout qui vient après le développement d'un modèle. Par contre nous avons toujours gardé en tête les propriétés efficaces sur GPU pour que nos méthodes soient un maximum portables et efficaces (en théorie) sur GPU.

## 4.4 Déchirure / découpe

Bien que l'on n'ait pas explicitement abordé ce thème dans le cours de cette thèse, une fonctionnalité indispensable à un bon simulateur chirurgical concerne la découpe et la déchirure. Découpe et déchirure sont assez similaires à synthétiser. La découpe intervenant suivant un critère donné par l'utilisateur (par exemple la collision avec un objet coupant), la déchirure apparaissant suivant un critère plus automatique, souvent dépendant des contraintes internes. Celles-ci se modélisent principalement par une modification topologique de la structure représentant l'objet animé. Le modèle mécanique s'appuyant fortement sur cette représentation, les méthodes de découpe et les modèles mécaniques sont donc très fortement interconnectés. Le modèle mécanique doit très rapidement se réadapter à la

nouvelle topologie à animer. En plus des problèmes concernant cette mise à jour rapide du modèle mécanique, une problématique complexe concerne l'affichage des zones découpées, car pour les objets généralement représentés par leur surface externe, il faut pouvoir créer et afficher les surfaces internes nouvellement visibles.

L'approche la plus simple pour faire de la découpe consiste à éliminer des éléments [Cotin 97], la mécanique et l'affichage pouvant trivialement s'adapter. Malheureusement, la perte de matière ainsi engendrée peut rendre la simulation peu réaliste, surtout dans le cas d'une simulation interactive utilisant des gros éléments. Il est possible de vraiment découper des éléments [Bielser 04]. [Mor 00] fournit par exemple une liste de motifs de découpe de triangles et de tétraèdres. Dans ce cas, les modifications ne sont pas une tâche aisée, et de très nombreux petits, voire mal conditionnés, éléments vont être créés, pouvant rendre la simulation lente et instable. Il est possible de remailler après la phase de subdivision afin d'éviter les éléments générés [Ganovelli 00a, Ganovelli 01], mais cette tâche reste complexe et coûteuse. Plutôt que de subdiviser les éléments découpés, il est possible de recréer un maillage aux bords respectant les passages de la coupure [O'Brien 99, O'Brien 02], mais encore une fois, ceci est réalisé hors du contexte temps réel.

Une autre approche consiste à modifier les fonctions d'interpolation d'un élément découpé, de telle façon à ce qu'une particule à l'intérieur de l'élément découpé ne dépende plus que de certains nœuds, selon sa position, d'un côté ou de l'autre de la découpe [Sukumar 00]. C'est une approche très élégante permettant des résultats d'une bonne qualité sans gros ralentissement ni instabilité [Vigneron 04, Jerabkova 07].

Le méthode qui nous paraît la plus intéressante et prometteuse s'appelle les "virtual nodes" [Molino 04, Müller 04c, Sifakis 07a]. L'idée consiste à dédoubler virtuellement des nœuds, qui deviennent indépendants, et permettent ainsi à deux éléments contigus de se séparer. Il est même possible de dédoubler des éléments, qui seront ainsi superposés dans l'espace et entraîneront avec eux, chacun une part de la matière. Une limitation semble être le dédoublement de masse, rendant l'objet plus lourd à chaque découpe, même si l'impact est très faible pour un gros objet ne subissant que quelques petites coupures. Nous envisageons volontiers de rendre compatible cette méthode à notre modèle, car elle semble y être très bien adaptée. Pour éviter les problèmes de remaillage de la surface au niveau de la coupure, nous proposerions bien d'exploiter les données volumiques, pour lesquelles la représentation de l'intérieur de l'objet existe déjà.

Les méthodes "meshless" [Müller 04b], qui n'utilisent pas de représentation volumique de l'objet, permettent plus directement des coupures d'une grande qualité. L'idée est de faire varier les critères de voisinage d'une particule, marquant ainsi la discontinuité de la matière. Malheureusement de telles méthodes ne sont encore pas adaptées au temps réel [Pauly 05].

On peut noter qu'il existe aussi les simulations d'insertion d'aiguilles ; un très récent état de l'art peut se trouver dans [Abolhassani 07]. Mais il nous semble que ces simulations peuvent être vues de façon plus éloignées, plus indépendantes du modèle mécanique. Elles se synthétisent plutôt comme une condition aux limites particulière, sa communication avec le modèle ne se traduisant qu'au travers de forces ou de déplacements (même si sa réalisation est plus ou moins complexe suivant le modèle mécanique).

## 4.5 Gestion des collisions

Lorsque l'on désire faire interagir différents objets entre eux, ou un outil virtuel sur un objet, voire carrément un objet se repliant sur lui-même, il faut être en mesure de répondre aux possibles pénétra-

tions entre les objets. Cette étape est appelée la gestion des collisions qui peut se décomposer en trois étapes : la détection des intersections, la modélisation des pénétrations (point de contact, normale au contact, profondeur de pénétration, instant du contact...) et la réaction à la collision. Ces trois étapes sont très complexes, spécialement concernant les objets déformables, car ceux-ci évoluant constamment au cours du temps, il est très compliqué de faire des précalculs puisque tout doit être remis à jour à chaque mouvement. De plus, les objets déformables peuvent s'auto-collisionner, compliquant grandement la tâche de détection.

C'est un domaine de recherche très actif, beaucoup de solutions très diverses ont été proposées ; un état de l'art peut être trouvé dans [Teschner 05]. Malheureusement, elles sont fréquemment spécifiques, et aucune méthode vraiment universelle n'a encore été proposée. Par exemple, certaines méthodes vont être seulement efficaces entre objets rigides [Cohen 95, Gottschalk 96, Guendelman 03], d'autres entre corps déformables et fluides [Guendelman 05], mais aucune ne va s'adapter efficacement sur tous types de modèles, obligeant leurs utilisations en parallèle pour un simulateur animant différents types d'objets (donc difficile à mettre en œuvre). On peut quand même noter l'effort d'unification dans [Faure 07], malgré le manque de performance. Beaucoup de méthodes efficaces sont trop dépendantes des modèles mécaniques sous-jacents et ne sont donc pas exploitables sur d'autres modèles [James 04a]. La détection reste encore un sujet ouvert, et pour notre avis, nous pensons qu'une bonne méthode de détection de collisions exploitable, ne doit pas être dépendante du modèle mécanique (ou au moins facilement adaptable à tous types de modèles), c'est pourquoi nous n'avons étudié cette problématique que parallèlement à cette thèse.

Au niveau de la réaction, on peut distinguer deux grandes familles : le traitement par forces de pénalité ou le traitement par contraintes dures (sur les positions et/ou les vitesses). La grande difficulté à ce niveau est de proposer une méthode proche de la physique bien que la pénétration de matière n'existe pas dans la réalité. Notons que les méthodes dites "continues" tentent de détecter le moment et l'endroit exacts du contact, au milieu d'un pas de temps, afin de ne pas avoir à gérer de pénétration de matière [Bridson 03a]. A un niveau plus global, il faut aussi être capable de répondre à des collisions sans créer d'énergie ou créer d'autres intersections ailleurs, phénomène complexe à obtenir en temps réel.

Ce qui nous semble important à ce niveau, et qui est très rarement soulevé est l'importance d'être en mesure de choisir la résolution à laquelle vont être gérées les collisions. L'approche classique consiste à considérer en dernière phase précise, le modèle surfacique utilisé pour l'affichage. Or celui-ci peut être très détaillé et donc énormément alourdir la recherche de collisions, pour laquelle un modèle plus grossier pourrait faire l'affaire. Dans ce cas, il faut être cependant en mesure de relier différents maillages et de faire passer des contraintes de l'un à l'autre tout en respectant la physique, comme proposé dans [Allard 07].

## 5 Applications

Comme on peut le voir, il existe de nombreuses propositions de modèles mécaniques depuis un certain nombre d'années dans le monde académique. Dans ce paragraphe nous allons passer en revue les méthodes utilisées dans les logiciels et applications proposés actuellement dans différents domaines.

## 5.1 Informatique graphique

Nos problématiques sont proches de celles actuelles des chercheurs de l'animation en graphisme. Il est donc intéressant de voir ce qui est utilisé actuellement dans ce domaine. On voit deux écoles très partagées. D'un côté, celle qui fait des modèles très complexes, pour obtenir de très belles animations pour des effets spéciaux ou des films d'animation. De l'autre côté, le monde des jeux vidéo s'essaie au contraire à fournir les modèles les plus simples possibles, mais très rapides et très stables, souvent issus de modèles robustes connus depuis longtemps [Müller 06, PhysX, Havok, Bullet] Dans les deux cas, avant la réalité physique, ce qu'on va rechercher concerne la "plausibilité" dans l'idée de faire de belles images, le tout avec le plus grand contrôle possible offert à l'utilisateur qui peut vraiment agir sur le déroulement. Dans de grands logiciels d'animation tels que Maya [Maya] et Blender [Blender], on peut remarquer que les outils concernant l'animation géométrique sont très évolués alors que le module de simulations physiques est peu complet. Il est peu utilisé surtout à cause de sa complexité à être manié par des artistes, non avertis à la physique, qui ont de réelles difficultés à obtenir la scène désirée avec du réalisme.

L'animation physique en est encore à ses balbutiements concernant une utilisation concrète, même dans le graphisme. Seuls quelques jeux vidéos récents utilisent des lois simples, ou quelques scènes de films laissent agir la physique pour un plus grand réalisme, mais elles restent remarquables et se réalisent très étroitement avec des chercheurs.

## 5.2 Simulateurs chirurgicaux

Les simulations chirurgicales requièrent beaucoup plus de contraintes, telles que l'intégration de propriétés mécaniques issues du monde réel suivant les composants modélisés (muscles, graisse, veines, tumeurs...).

Ces contraintes ont essentiellement été intégrées dans des planificateurs non-interactifs (tels que ceux modélisant le visage [Waters 90, Chabanas 03]) alors que les simulateurs interactifs restent encore assez éloignés de la simulation du réel.

La plupart des méthodes d'animations interactives développées dans le monde académique pour le monde médical n'aboutissent à aucune application concrète (comme c'est le cas dans notre travail). Leur exemple de base est un "foie volant" qui n'a rien de représentatif de la réalité.

Parallèlement, quelques simulateurs chirurgicaux ont fait leur apparition sur le marché. Ceux-ci ont généralement une belle représentation virtuelle de plusieurs organes interagissants et manipulables par divers outils. Notons par exemple :

- VIRGY [Baur 98]
- LASSO [Szekely 00]
- Karlsruhe Endoscopic Surgery Trainer [Kuhnappel 00]
- SPIC [Meseure 03]

Leur finition est plus ou moins avancée au niveau de la qualité du rendu, des textures, les détails des modèles, la présence de différents types d'outils pour interagir. D'un point de vue simulation, en général, ils sont peu génériques, et il n'est pas facile de faire varier les méthodes de calcul (tel que choisir la détection de collisions ou le calcul de forces). Mais surtout, tous se basent sur un calcul simpliste des forces à base de ressorts 1D, en mettant en avant la rapidité et la robustesse de ce modèle. Si ces simulateurs permettent une première étape, celle de pouvoir jouer avec les outils de laparoscopie pour se faire la main, leur mécanique trop simpliste ne permet pas de reproduire la réalité et de se passer des autres modes d'entraînements.

### 5.3 Plates-formes d'animation physique

Un domaine particulièrement intéressant concerne les plates-formes généralistes, orientées vers l'animation temps-réel. Même celles qui sont spécifiquement orientées vers le jeu vidéo [Bullet, PhysX, Havok], ont des atouts intéressants telle que leur implémentation efficace dans une recherche de rapidité et le besoin d'une grande stabilité. Elles peuvent apporter un grand savoir faire, surtout au niveau de la gestion des collisions, des variétés de modèles en interaction et de la mise en réseau. Des plates-formes plus spécifiques aux simulations médicales existent aussi [SOFA, SPRING, GiPSi, VRASS, CHAI 3D]. Elles récupèrent les avantages cités précédemment tout en y ajoutant des lois plus complexes, plus représentatives de la physique. Et surtout leur intérêt principal est l'idée de disposer d'un choix dans les méthodes de calculs des différentes étapes (forces, temps, collisions...). Il est alors plus simple de faire évoluer une étape particulière d'une simulation déjà existante sans toucher les autres étapes ; c'est à notre sens le moyen le plus efficace d'avancer.

## 6 Discussion

De nombreux modèles existent déjà en mécanique, et les plus adaptés à la simulation informatique ont été implémentés avec toutes sortes de solutions élégantes pour optimiser les temps de calcul, la robustesse... Alors que parallèlement, dans les applications concrètes, tels que les jeux vidéo ou les simulateurs chirurgicaux, seules les méthodes les plus simples sont employées. Comme principales limitations qui font que certaines méthodes ne sont pas utilisées, on peut retenir leur difficulté à être implémentées efficacement, leur difficulté à être mises en place et paramétrées par un utilisateur non averti et leur comportement face à l'instabilité. Inversement, les méthodes simples employées le sont principalement grâce à leur fiabilité et leur bonne maîtrise par l'utilisateur. Il nous semble donc crucial de respecter les critères qui semblent être prioritaires, que sont la simplicité, la rapidité et la stabilité, afin que les méthodes mises en œuvre puissent être vraiment exploitées. On cherche donc à tirer profit des méthodes précises basées sur la méthode des éléments finis déjà proposées en les intégrant dans un système simple à mettre en place et à utiliser, rapide et robuste.

Alors qu'on pense que la gestion des collisions peut être traitée comme un problème annexe auquel nous ne nous intéresserons pas dans ce document, la gestion des découpes, quant à elle, est plus proche du modèle mécanique puisqu'elle intervient sur les mêmes représentations de l'objet simulé. C'est pourquoi nous la prendrons en compte dans nos réflexions, même si son sujet n'a pas été concrètement abordé.

Regardons plus précisément dans quelles conditions on veut utiliser un modèle déformable : on dispose d'un modèle, qui peut être représenté sous des formes variées (typiquement une surface triangulée ou des données volumiques). Premièrement, il faut fabriquer un maillage volumique sur lequel calculer la mécanique, puis y appliquer une méthode d'animation avec un bon choix de paramètres de déformations. Pour un utilisateur, il est difficile de définir quels devraient être le maillage volumique mécanique le plus adapté, (surtout au temps réel), la méthode mécanique, le schéma d'intégration (avec quel pas de temps), les matériaux à utiliser. Nous voulons proposer une méthode où le maillage mécanique est construit automatiquement avec un paramètre intuitif pour choisir sa résolution, sur lequel seront appliquées des lois issues de la mécanique implémentées de façon simple, robuste et rapide et pour lequel les paramètres sont issus de la physique classique.



chapitre **2**

---

# Notions de mécanique

---

<b>1</b>	<b>Introduction</b>	<b>30</b>
<b>2</b>	<b>Notions théoriques</b>	<b>30</b>
2.1	Mécanique des milieux continus	30
2.2	Déplacements	31
2.3	Déformations	32
2.4	Contraintes	35
2.5	Lois de comportement des matériaux	36
2.6	Modèles répandus	40
<b>3</b>	<b>Notions numériques</b>	<b>41</b>
3.1	Méthode des éléments finis	42
3.2	Implémentations	44
3.3	Validation	53
<b>4</b>	<b>Conclusion</b>	<b>54</b>

---

## 1 Introduction

Depuis très longtemps, dans le domaine de la mécanique, on essaie de simuler le comportement d'objets déformables soumis à des contraintes extérieures. Les lois mises en œuvre essaient de se rapprocher au maximum de la réalité. Il semble donc plus que pertinent qu'il faille se tourner vers ces lois pour notre problématique. Afin de 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. Ces informations sont issues des cours [Ohayon 00, Garrigues 00] et sont présentées avec un œil de non-mécanicien pour les non-mécaniciens. Puis nous verrons différentes méthodes numériques proposées en informatique pour résoudre ces équations théoriques, suivant différents critères de vitesse, de stabilité...

## 2 Notions théoriques

### 2.1 Mécanique 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 continûment d'un point à un autre. Cet espace est représenté mathématiquement par l'espace euclidien de dimension 3.

On appelle ce modèle mathématique, permettant de simplifier les propriétés physiques à cette échelle humaine, la Mécanique des Milieux Continus (MMC). La MMC permet de définir en n'importe quel point matériel (ou particule) de l'objet des champs continus. Dans le cas d'objets déformables, ces grandeurs physiques sont par exemple les champs de déformations et de contraintes. Pour donner un autre exemple, sont aussi attachées des températures en thermodynamique.

Les valeurs de ces champs continus sont représentées en chaque particule sous la forme de tenseurs (de déformation, de contrainte), représentant une description locale de l'état (de déformation, de contrainte) en cette particule. L'ordre d'un tenseur est le nombre d'indices matriciels nécessaires pour décrire une telle quantité. Par exemple les quantités scalaires, telles que la masse ou la température sont des tenseurs d'ordre 0. La force, le déplacement et autres quantités vectorielles sont des tenseurs d'ordre 1. Les déformations et contraintes sont des tenseurs d'ordre 2, et sont représentés en trois

dimensions par des matrices  $3 \times 3$   $[\alpha] = \begin{bmatrix} \alpha_{xx} & \alpha_{xy} & \alpha_{xz} \\ \alpha_{yx} & \alpha_{yy} & \alpha_{yz} \\ \alpha_{zx} & \alpha_{zy} & \alpha_{zz} \end{bmatrix}$ . L'écriture sous forme vectorielle des

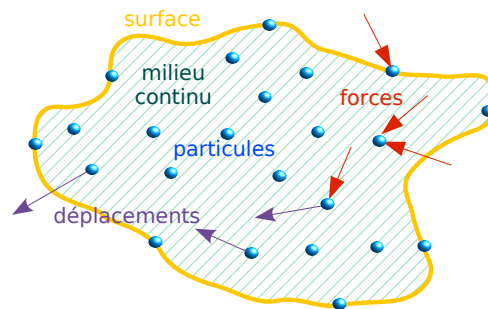
tenseurs symétriques est souvent usitée :  $\{\alpha\} = \begin{bmatrix} \alpha_{xx} \\ \alpha_{yy} \\ \alpha_{zz} \\ \alpha_{xy} \\ \alpha_{yz} \\ \alpha_{xz} \end{bmatrix}$ .

Deux représentations du continuum sont envisageables : lagrangienne et eulérienne. Ces descriptions identifient les particules par leur position 3D à un instant donné. La description lagrangienne identifie les particules par leurs positions à un instant arbitraire, généralement nommé configuration de référence. De façon plus imagée, on suit l'évolution des mêmes particules au cours du temps. Contrairement à la description d'Euler qui identifie une particule par sa position actuelle. Concrètement, le champ d'observation est fixe et on regarde les variations des particules passant par ce point



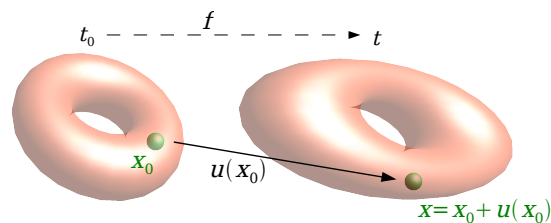
d'observation au cours du temps. Ces deux descriptions, qui ont chacune leur utilité, sont équivalentes et on peut passer de l'une à l'autre. On préfère souvent la représentation eulérienne pour les fluides où on s'intéresse peu à la position individuelle d'une particule mais plus à l'écoulement en un certain point d'observation. Pour les solides déformables, on préfère la description de Lagrange car la configuration de référence est facilement identifiable comme l'état au repos et la frontière du domaine est plus clairement définie. C'est pourquoi dans la suite du document on s'applique simplement à décrire la MMC en utilisant la description lagrangienne.

Un objet, ou milieu continu, est un ensemble matériel délimité par une surface à partir de laquelle sont définis l'intérieur et l'extérieur. Des contraintes extérieures peuvent être appliquées à l'objet. Celles-ci peuvent être distantes, comme la gravité, ou de contact, quand elles sont appliquées à la surface. 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.



**FIG. 2.1** – Un milieu continu, sa surface, des particules sur lesquelles s'appliquent des forces externes et des déplacements.

## 2.2 Déplacements



**FIG. 2.2** – Déplacement  $u$  d'un point de position  $x$ .

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  $\mathbf{x}_0$ . Lorsque le corps est déformé, chacun de ses points se retrouve à une position différente  $\mathbf{x}$ . Le déplacement  $\mathbf{u}$  correspond au vecteur

entre l'état déformé et l'état au repos d'une particule  $\mathbf{x}$  :  $\mathbf{u}(\mathbf{x}) = \mathbf{x} - \mathbf{x}_0 = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$ .

### 2.3 Déformations

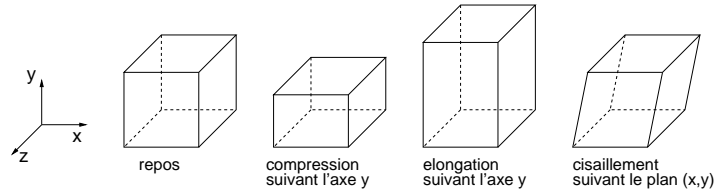


FIG. 2.3 – Différentes déformations d'un hexaèdre

La déformation n'a de sens qu'entre deux positions (entre deux instants) du mouvement de l'objet.

Le tenseur de déformation représente une description locale de l'état de déformation en une particule du corps déformé, *i.e.* les dilatations (élongation, compression) et les cisaillements que subit l'objet dans l'entourage infinitésimal de la particule (Figure 2.3). Il s'écrit généralement (pour des problèmes tridimensionnels) sous la forme d'une matrice  $3 \times 3$

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} \varepsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \varepsilon_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \varepsilon_z \end{bmatrix}$$

ou sous la forme condensée d'un vecteur à six entrées dans le cas d'un tenseur symétrique comme pour les problèmes d'élasticité

$$\{\boldsymbol{\varepsilon}\} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix}$$

$\varepsilon_i$  représente l'étirement (compression/élongation) suivant l'axe  $i$  et  $\gamma_{ij}$  le cisaillement angulaire entre les axes  $i$  et  $j$ .

Théoriquement, seules les déformations 'pures' doivent être considérées, c'est-à-dire qu'un déplacement rigide n'introduit pas de déformation. Une mesure de la valeur des déformations est obtenue à partir des déplacements en faisant intervenir la géométrie du milieu. Cette mesure a été donnée en considérant les changements de longueurs d'un segment entre son état au repos et l'état déformé (Figure 2.4).

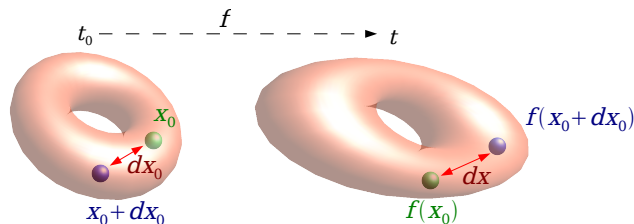


FIG. 2.4 – Déformation : variation de longueurs dans un corps déformé.

Considérons deux particules voisines de position  $x$  et  $x + dx$  d'un objet  $\Omega$  à deux instants  $t_0$  et  $t$  (Figure 2.4). On appelle  $f$  la transformation de  $\Omega$  entre  $t_0$  et  $t$ .

$$f : x_0 \in \Omega_0 \longrightarrow x_t = f(x_0) \in \Omega_t$$

Avec  $f$  différentiable, une approximation du premier ordre d'une série de Taylor donne :

$$dx_t = \nabla f . dx_0$$

La variation des carrés des longueurs est

$$\begin{aligned} \|dx_t\|^2 - \|dx_0\|^2 &= dx_t^T . dx_t - dx_0^T . dx_0 \\ &= (\nabla f . dx_0)^T . \nabla f . dx_0 - dx_0^T . dx_0 \\ &= dx_0^T . \nabla f^T . \nabla f . dx_0 - dx_0^T . dx_0 \\ &= dx_0^T . (\nabla f^T . \nabla f - id) \end{aligned}$$

Ainsi à été introduit le tenseur de Green-Lagrange, choisi pour représenter l'état de déformation :

$$\epsilon_G = \frac{1}{2} (\nabla f^T . \nabla f - id)$$

Son expression peut être donnée en fonction des déplacements car  $\nabla f = id + \nabla u$ .

**Preuve :**

$$\begin{aligned} dx_t &= d(x_0 + u(x_0)) \\ &= dx_0 + d(u(x_0)) \\ &= dx_0 + \nabla u . dx_0 \\ &= \underbrace{(id + \nabla u)}_{\nabla f} . dx_0 \end{aligned}$$

Le tenseur de Green-Lagrange peut donc s'écrire :

$$\begin{aligned} \epsilon_G &= \frac{1}{2} ((id + \nabla u)^T . (id + \nabla u) - id) \\ &= \frac{1}{2} (\nabla u + \nabla u^T + \nabla u^T . \nabla u) \end{aligned} \tag{2.1}$$

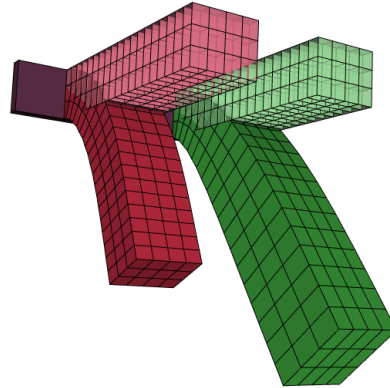
Sa linéarisation

$$\epsilon_C = \frac{1}{2} (\nabla u + \nabla u^T) \tag{2.2}$$

est appelée tenseur de déformation de Cauchy. Lorsque les déplacements sont peu importants, le terme quadratique  $\nabla u^T . \nabla u$  est en effet négligeable. Ce tenseur est alors souvent employé dans un souci d'efficacité des calculs sur des matériaux très durs tels que l'acier, qui se déforment peu. Le problème de ce tenseur est qu'il n'est pas invariant en rotation, provoquant des gonflements dûs aux approximations, comme illustré à la Figure 2.5. Il n'est donc pas approprié aux matériaux mous pouvant se déformer beaucoup et/ou aux objets subissant de grands déplacements rotationnels dans l'espace.



vidéo 1

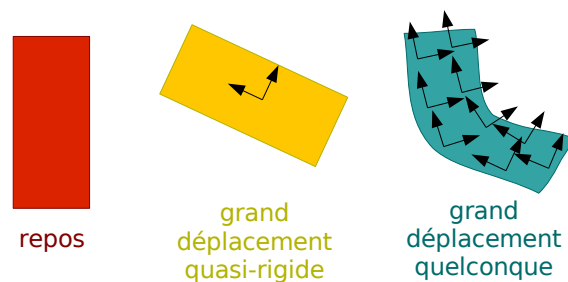


**FIG. 2.5** – Comparaison des tenseurs de déformations linéaire et non-linéaire.

Deux poutres encastrées similaires soumises à la gravité ont les mêmes propriétés mécaniques. La position au repos est en transparent, la position à l'équilibre est opaque. A gauche, en rouge, les rotations sont prises en compte, le comportement est plausible. A droite, en vert, le calcul des déformations est linéaire, la poutre gonfle.

**Remarque :** Le tenseur de Green-Lagrange ne donne pas les déformations linéaires décrites par Hooke  $\varepsilon = \frac{l-l_0}{l_0}$  où pour un segment  $l_0$  est la longueur initiale et  $l$  la longueur déformée. Considérons un exemple unidimensionnel où  $l_0 = 1$ , pour une déformation de 10%,  $l = 1.1$ . Le tenseur de Green-Lagrange donne une déformation  $\varepsilon_G = \frac{1}{2}(0.1 + 0.1 + 0.1^2) = 0.105$  soit 10.5%.

Une méthode, dite corotationnelle, permet de traiter les grands déplacements avec rotations tout en conservant un calcul linéaire des déformations. L'idée générale est d'extraire à la main la transformation rigide pour obtenir des petits déplacements sur lesquels le tenseur de Cauchy est suffisant. Pour cela, il faut être capable de fixer un repère local à la particule concernée afin d'analyser son déplacement rigide (Figure 2.6). Si le matériau est assez rigide, entraînant de petites déformations, un seul repère pour tout l'objet peut être suffisant, ramenant dans le cas d'utilisation normale du tenseur de Cauchy.



**FIG. 2.6** – Approche corotationnelle

Déduire la transformation rigide à l'aide de repères locaux. Dans le cas d'un déplacement quasi-rigide, un seul repère peut suffire, alors que dans le cas de grands déplacements, un repère par particule doit être envisagé.

## 2.4 Contraintes

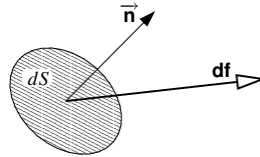
Les contraintes modélisent les efforts intérieurs par un champ tensoriel. Dans le cas général, un élément de matière situé au cœur d'une pièce est soumis à des contraintes dans diverses directions. On représente cet état de contrainte par un tenseur  $3 \times 3$  appelé tenseur des contraintes.

L'état de contrainte d'un point traduit une description des forces internes que subit ce point. Alors que c'est une mesure utile pour l'étude de la résistance d'un objet, dans notre problématique elle ne va nous servir que d'un calcul intermédiaire pour en déduire des forces internes. Plus de détails peuvent être trouvés dans [Salençon 05, Basar 00].

Pour interpréter la notion de contrainte, l'objet déformable est décomposé en volumes infinitésimaux. La force élémentaire  $d\mathbf{f}$  en un point peut être définie suivant chaque élément de surface défini par une normale  $\vec{n}$  et une surface  $dS$  (figure 2.7).

Telle que définie par Cauchy, la contrainte est caractérisée dans la configuration actuelle (déformée) ce qui permet de la spécifier comme un tenseur symétrique  $[\sigma]$  par point de l'objet.

$$d\mathbf{f} = [\sigma] \vec{n} dS$$



**FIG. 2.7** – Force élémentaire

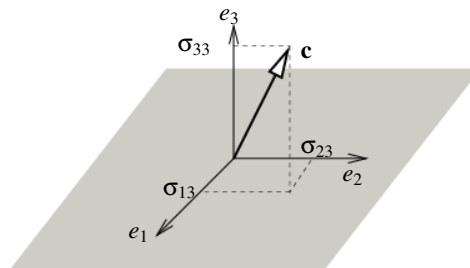
pour une facette de direction  $\vec{n}$  et de surface  $dS$ .

En normalisant le vecteur de force  $d\mathbf{f}$  par la surface  $dS$ , on obtient le vecteur contrainte de Cauchy :

$$\mathbf{c} = [\sigma] \vec{n}$$

qui représente une force par unité de surface.

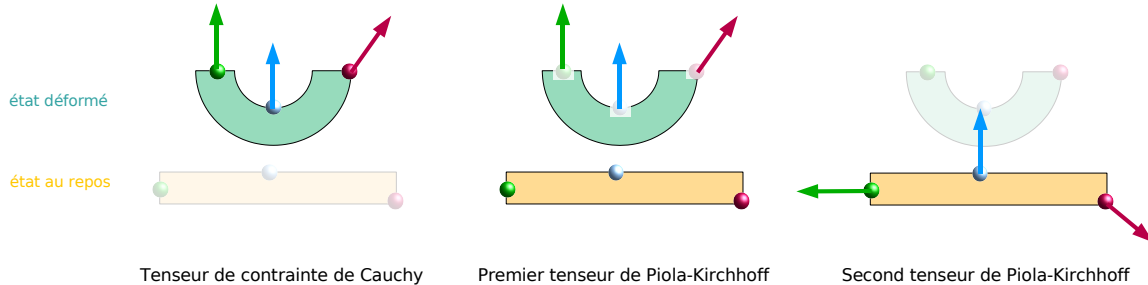
Si une base orthonormée est choisie pour représenter  $[\sigma]$  et  $\vec{n}$ , les termes diagonaux du tenseur de contrainte représentent la contrainte normale  $c_n$  et les termes non-diagonaux représentent la contrainte tangentielle  $c_t$  dans le plan de  $dS$  (figure 2.8).



**FIG. 2.8** – Contrainte

Contraintes normale  $c_n$  et tangentielle  $c_t$ .

Des descriptions des efforts intérieurs sur la configuration initiale de référence ont aussi été proposées, tels que les tenseurs de Piola-Kirchhoff (figure 2.9).



**FIG. 2.9** – Divers tenseurs de contraintes en fonction des configurations considérées

*Tenseur de Cauchy : forces et configuration de référence dans l'état déformé. Premier Tenseur de Piola-Kirchhoff : forces dans l'état déformé et configuration de référence dans l'état au repos. Second tenseur de Piola-Kirchhoff : forces et configuration de référence dans l'état au repos.*

## 2.5 Lois de comportement des matériaux

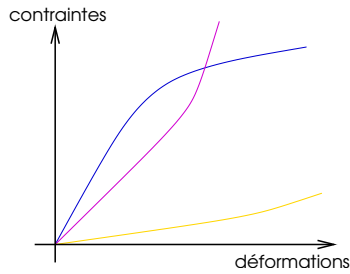
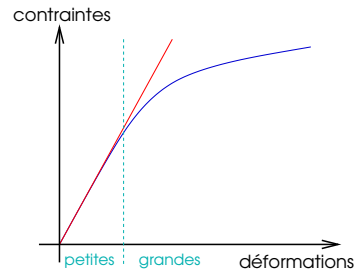
La réponse d'un objet soumis à des sollicitations extérieures dépend de la nature physique du milieu le constituant. Les solides déformables peuvent par exemple être plus ou moins souples. L'étude qui vise à caractériser des matériaux réels, s'appelle la rhéologie.

Dans cette section, on va introduire brièvement quelques notions de rhéologie, des informations bien plus complètes peuvent être trouvées dans le nombreux ouvrages, tel que [Fung 93].

On appelle loi de comportement, une relation qui exprime les contraintes actuelles en fonction des changements d'états subis. Elle s'exprime par une expression sur le tenseur de déformation. Ce lien souvent très complexe, est généralement exprimé à partir d'un petit nombre de paramètres mécaniques dont les valeurs sont déduites de l'expérimentation. Ces lois mécaniques simplifiées sont suggérées pour coller au mieux à des essais expérimentaux des matériaux. Expérimentalement, le comportement d'un matériau peut être étudié en observant sa déformation lorsqu'on lui applique une contrainte, et sa réaction lorsque cette contrainte cesse d'être appliquée. Un résumé généralement considéré comme suffisant de l'histoire d'une particule est l'ensemble de ses variables d'états actuelles.

**Elastique** 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. 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 restitue toute l'énergie qu'il a absorbé, sa forme ne dépend pas de l'historique de ses déformations (cf figure 2.10). Une particule qui est déplacée reviendra à sa position initiale en passant par le même « chemin ».


**FIG. 2.10** – Des lois de comportement élastique

**FIG. 2.11** – Approximation linéaire (rouge) d'une loi de comportement élastique sublinéaire (bleue)

Le comportement peut être plus ou moins complexe en correspondant à différentes lois mécaniques intégrant des paramètres variés comme les différentes lois hyperélastiques [Bathe 82].

**Elastique linéaire** 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. La formulation est du type  $\{\sigma\} = \mathbf{D}\{\varepsilon\}$  avec  $\mathbf{D}$  une matrice  $6 \times 6$ .

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 comprime un matériau à 100%, la même force (au signe près) s'appliquera ; or on sait que si on pouvait comprimer un objet en un seul point, les forces exercées seraient infinies.

On voit apparaître sur le schéma 2.11, 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 10%.

**Loi de Hooke** En plus d'être élastique linéaire, 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  $\lambda$  et  $\mu$ . La loi de comportement s'écrit alors :

$$[\sigma] = \lambda \text{tr}([\varepsilon])\mathbf{I} + 2\mu[\varepsilon]$$

La matrice de rigidité du matériau  $\mathbf{D}$ , qui reflète la loi de comportement et fait passer des déformations aux contraintes, est du type :  $\{\sigma\} = \mathbf{D}\{\varepsilon\}$  avec

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

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  $\nu$  correspondant à sa compressibilité, sans unité (cf figure 2.12).

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

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

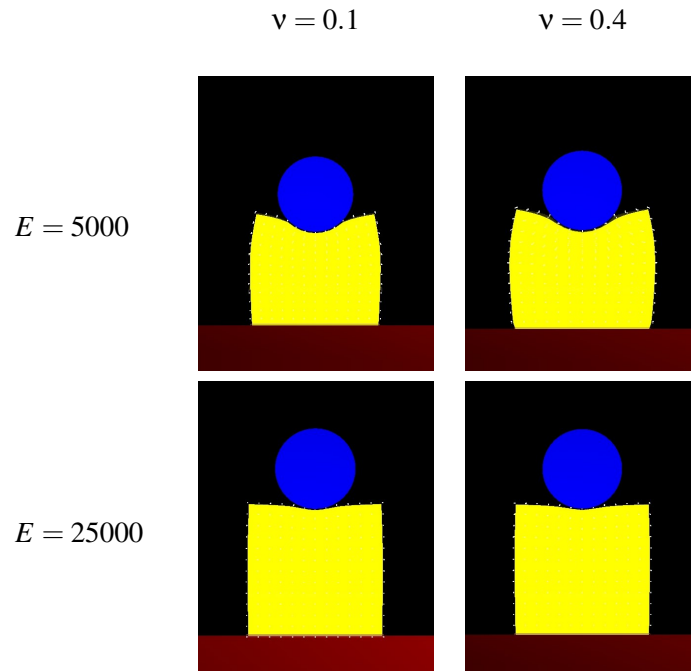


FIG. 2.12 – Influences du module d'Young et du coefficient de Poisson

Le système est inversible pour déduire les déformations des contraintes.

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



Cette simplicité à la fois au niveau des calculs que de l'expérimentation fait que la loi de Hooke se retrouve dans la grande majorité des simulateurs interactifs actuels.

**Visco-élasticité** Pour un matériau visco-élastique, la réponse à une contrainte dépend non seulement de l'intensité de celle-ci, mais également de la vitesse à laquelle elle est appliquée. La relation entre contraintes et déformations n'est alors pas unique, mais est une fonction dépendante du temps. Une caractéristique est que le système revient à son état au repos quand la contrainte est relâchée (car il est élastique), en suivant un chemin différent de la phase de chargement. Cet écart correspond à une perte d'énergie (sous la forme d'un dégagement de chaleur). Le fait que le comportement soit différent entre la phase de chargement et de relâchement se nomme l'*hystérésis*.

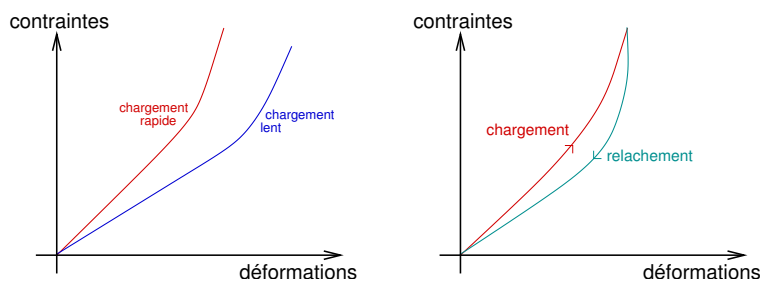


FIG. 2.13 – Matériau viscoélastique

**Plasticité** Lorsque l'on dépasse le domaine élastique, un matériau devient plastique. C'est à dire, qu'après une déformation trop forte, il revient à une nouvelle position d'équilibre, différente de la configuration initiale, par un trajet similaire à son comportement élastique. La striction est la phase où le matériau revient à une position d'équilibre par un chemin non-parallèle au comportement élastique. Une contrainte trop forte peut entraîner une rupture (déchirure) du matériau.

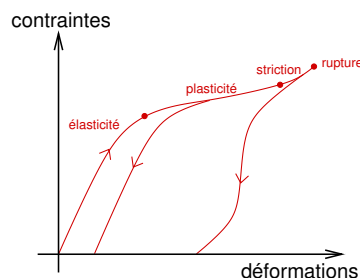
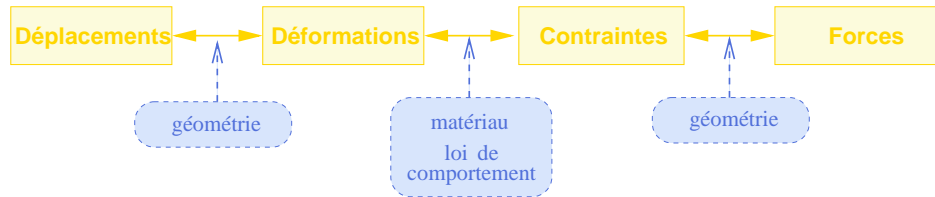


FIG. 2.14 – Matériau plastique

**Autres lois** Bien d'autres comportements existent, en particulier pour des composites mi-fluides mi-structure (poroélasticité, comportement pâteux...).

tel-00293531, version 1 - 4 Jul 2008

## 2.6 Modèles répandus



**FIG. 2.15** – Relation déplacements/déformations/contraintes/forces en fonction de la géométrie et du matériau

En analysant le modèle global de la mécanique des milieux continus (illustré Figure 2.15), on peut voir qu'on a principalement le choix des lois entre déplacements et déformations, et entre déformations et contraintes. Attention quand on parle donc d'éléments finis linéaires ou non-linéaires, cette linéarité (ou non) peut se retrouver à deux endroits et demande une précision. Les choix les plus répandus de part leur simplicité sont présentés ci-après.

**Pur linéaire** La méthode la plus simple à évaluer consiste à prendre deux lois linéaires, correspondant au tenseur de déformation de Cauchy avec la loi de Hooke. Les forces sont alors proportionnelles aux déplacements. On peut ainsi résoudre des problèmes de statique par la résolution d'un système matriciel linéaire du type :

$$f = Ku$$

où  $K$  est appelée matrice de rigidité. Bien sûr ce modèle est très limitatif, mais il est très simple à calculer donc très rapide. Dans le cas d'objets très raides qui se déforment peu, son approximation n'est pas si mauvaise et il intervient souvent en étude de structures.

Grâce à cette simplicité de calculs, les premiers modèles interactifs suivaient cette loi [Bro-Nielsen 96, Cotin 97].

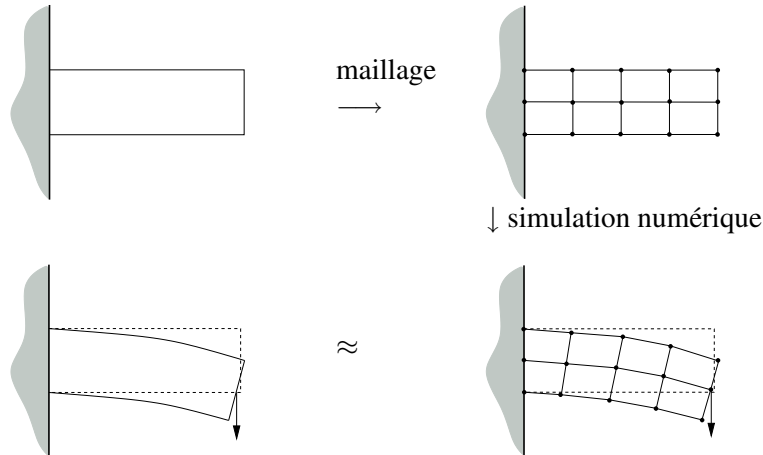
C'est vers ce modèle qu'on se réfère quand on parle d'*éléments finis linéaires*.

**S' Venant-Kirchoff** Dans ce modèle, les grands déplacements sont pris en compte pour une plus grande liberté dans les mouvements de l'objet. Pour cela, le tenseur de déformations de Green-Lagrange (eq. 2.1) est employé. La loi de Hooke est utilisée pour le matériau, qui reste souvent valide dans le cas de petites déformations. Ce modèle est beaucoup employé dans le cas de simulations interactives et/ou médicales pour lesquelles les objets peuvent beaucoup se déplacer mais en considérant que la gestion des petites déformations est suffisante. Il constitue le plus simple modèle non-linéaire et donne un compromis vitesse/précision très réputé.

Grâce à sa gestion des grands déplacements, cette loi donne plus ce qu'on attend d'une simulation interactive. Elle a été utilisée dans la deuxième "fournée" de simulateurs interactifs [Picinbono 00b, Debunne 01].

### 3 Notions numériques

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.



**FIG. 2.16** – Une solution exacte (gauche) approximée par une simulation numérique sur une représentation discrète (droite).

Il existe deux grandes méthodes pour résoudre numériquement une équation aux dérivées partielles et en particulier trouver une solution à celle de la mécanique des milieux continus : la méthode des différences finies (MDF) et la méthode des éléments finis (MEF).

Pour minimiser la fonction d'énergie, on a besoin de connaître ses dérivées spatiales, qu'on peut approximer par différences finies :

$$\frac{\partial f}{\partial x} = \frac{f(x+h) - f(x-h)}{2h}$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

Cette évaluation est évidente pour des grilles régulières mais devient plus complexe pour des maillages non-structurés, pour lesquels la MEF est plus appropriée.

La MDF est bien adaptée aux domaines discrétisés suivant une grille régulière pour lesquelles les dérivées partielles sont faciles à évaluer. Des extensions ont été proposées pour gérer aussi les maillages non réguliers [Meyer 02], mais leur utilité n'est pas claire, du moment qu'elles mènent normalement aux mêmes équations que la MEF linéaire. Dans le cas non-linéaire, la similitude entre les deux méthodes n'est pas prouvée, et rien n'assure que la méthode des différences finies (par exemple [Debunne 01]) converge vers la bonne solution. L'unique avantage que nous voyons à cette méthode est de faciliter l'évaluation des matrices, qui s'effectue en pré-traitement. Même si, comme nous le verrons par la suite, nous utilisons un maillage régulier, nous nous sommes contenté d'étudier la méthode des éléments finis que nous détaillons ci-dessous.

La MEF est la méthode la plus utilisée dans le cas de calculs de structures et d'analyses numériques. Bien que cette méthode nous intéresse uniquement dans le cas de problèmes 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.

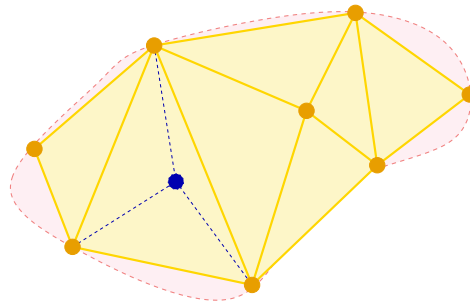
### 3.1 Méthode des éléments finis

Cette méthode dont la théorie présentée dans [Zienkiewicz 67] est très bien illustrée d'un point de vue plus pratique et technique dans [Bathe 82] dans lequel les nombreux exemples numériques détaillés nous ont été d'une grande utilité.

#### 3.1.1 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 nœuds. Les propriétés physiques sont interpolées sur chaque élément en fonction de leur valeur aux nœuds. L'équation d'équilibre du système doit être vérifiée sur chacun des éléments, procurant des équations sur les nœuds. Les interactions entre éléments voisins mènent à la construction d'un ensemble d'équations. Ce système d'équations peut être obtenu par un processus d'assemblage. Sa résolution, 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 ( $C^0$ ,  $C^1$ , ...).



**FIG. 2.17** – Principe de la Méthode des Elements Finis

*Un milieu continu (rose) est discrétisé et approximé par des éléments (ici des triangles jaunes). Les équations de la mécanique des milieux continus peuvent être résolues aux nœuds du maillage (points jaunes). Les valeurs aux nœuds peuvent être interpolées à l'intérieur d'un élément (point bleu).*

Par définition, les propriétés mécaniques sont uniformes à l'intérieur d'un même élément.

Logiquement, plus le maillage est petit, plus la solution numérique se rapproche de la solution théorique. Mais en pratique, il y a une limite, due à des problèmes numériques.

La qualité est souvent opposée avec la rapidité, donc il n'existe pas de solution miracle, unique à tous les problèmes ; il est important d'utiliser les lois les plus simples, suffisantes au cas par cas.

#### 3.1.2 Mailles

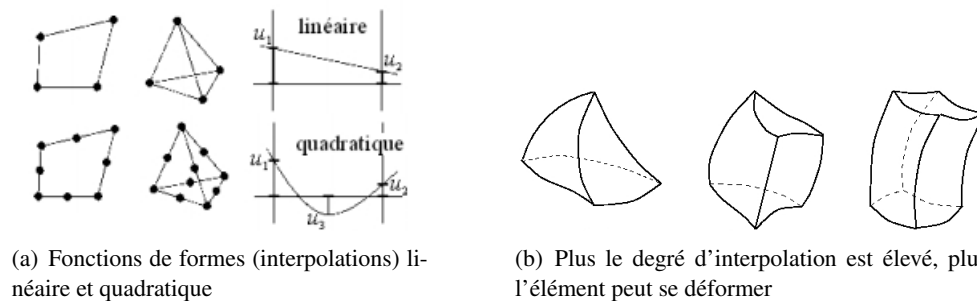
La première étape d'une analyse par éléments finis consiste à définir un maillage volumique du continuum. Les éléments finis peuvent être de différentes formes (tétraèdres, hexaèdres, prismes...), il est possible d'utiliser différents types d'éléments dans un même maillage.

Le maillage est très important dans la qualité de la solution. Dans ce paragraphe, on donne quelques éléments pour en donner la conviction.

Plus le maillage est fin, plus le calcul est précis, mais coûteux. On doit trouver 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 certaines régions en fonction de l'importance des déformations. Sinon une approche multirésolution adaptative peut être une bonne alternative, en adaptant automatiquement la résolution du maillage au cours de la simulation.

Le principe de la méthode des éléments finis étant d'interpoler les champs physiques en tous points du continuum en fonction de leur valeur aux nœuds, diverses fonctions d'interpolation (ou fonctions de forme) peuvent être choisies. On veut au moins une continuité simple  $C^0$  à la frontière entre les éléments. Pour plus de continuité, il est nécessaire d'ajouter des nœuds. L'ensemble des nœuds ne se limite pas forcément aux sommets des éléments, il est possible de définir des nœuds supplémentaires, par exemple un quadrilatère à 8 nœuds ou un tétraèdre à 10 nœuds, comme présenté dans la figure 2.18.

Il est en théorie 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 régulière (exemple : carré, cube) topologiquement équivalente à la maille réelle, puis on considère les jacobiens des transformations qui le ramènent vers les mailles réelles. Ce procédé a l'avantage de simplifier grandement les calculs.



**FIG. 2.18** – Ajouter des nœuds aux éléments permettent d'avoir des degrés d'interpolation plus élevés

La qualité du maillage est directement reliée au conditionnement du système implicite correspondant. Meilleure est sa qualité, meilleur sera le conditionnement du système. Le maillage a donc une grande importance concernant la vitesse des calculs, non seulement au niveau du nombre et du type d'élément, mais aussi au niveau de la qualité de ceux-ci. Plusieurs facteurs interviennent pour mesurer la qualité d'un maillage [Liu 94], on retiendra principalement, la taille relative des éléments (aspect ratio) et la qualité des angles des éléments (twist ratio, warping factor, ...).

### 3.1.3 Calcul des forces

Une fois l'objet discrétisé en éléments, on va être capable de résoudre la mécanique des milieux continus sur ces volumes, donnant des équations aux nœuds.

En ne prenant que les contributions d'un seul élément  $i$  :

$$f_i = B_i^T D_i B_i u_i$$

Dans les cas non-linéaires (des déformations ou des contraintes),  $B$  et  $D$  peuvent dépendre de  $u$ .

Dans le cas purement linéaire,  $\{\varepsilon\} = B u$  correspond aux déformations de cet élément (élongations dans les trois directions et cisaillements dans les trois plans),  $\{\sigma\} = D \{\varepsilon\} = D B u$  à son état de

contrainte et  $\mathbf{f} = \mathbf{B}^T \{\boldsymbol{\sigma}\}$  aux forces internes appliquées à ses nœuds.  $\mathbf{B}$  est la matrice déplacements-déformations, sa construction est donnée en annexe D pour des tétraèdres linéaires.  $\mathbf{D}$  est la matrice déformations-contraintes, elle retranscrit la loi de comportement des matériaux comme présenté dans le chapitre 2 à la section 2.5.  $\mathbf{K} = \mathbf{B}^T \mathbf{D} \mathbf{B}$  est appelée matrice de raideur. Elle est symétrique et retranscrit directement les influences de tous les couples de nœuds. Elle est très creuse, car tous les nœuds ne sont pas connectés.

### 3.2 Implémentations

Plusieurs types de matrices déplacements-déformations et déformations-contraintes sont envisageables, suivant la qualité désirée des déformations et des matériaux, engendrant différents problèmes d'implémentation. Bien sûr, il y a plusieurs façons de calculer numériquement ces matrices, que nous présentons ici. Chacune des avancées est classée comme étant :

- une amélioration de la **vitesse**
- une gestion des **grands déplacements**
- une prise en compte des **changements de topologie**
- un nouveau **matériau**
- des **éléments quadratiques**
- une amélioration de la **stabilité**

#### 3.2.1 Vitesse

Il est possible d'améliorer l'efficacité des calculs de la MEF en élasticité linéaire (2.6). L'approximation linéaire permet des simplifications intéressantes :

- la déformation résultante d'une déformation d'un point est la même que la somme de celles qui seraient produites par les trois déformations appliquées séparément dans chacune des trois directions  $x, y, z$ .
- les déformations sont proportionnelles aux déplacements.

**Précalculs** [Cotin 96] remarque qu'il est alors possible de stocker pour chaque déplacement dans chaque direction de chaque point séparément, 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). [Bro-Nielsen 96] propose de préinverser le système par morceaux.

Un problème est alors 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 nœuds déplacés, dont l'inverse va déterminer les déplacements à appliquer aux différents nœuds. Même s'il est possible de déplacer plusieurs nœuds en même temps, c'est au prix d'une inversion de matrice (de taille proportionnelle au nombre de nœuds).

Les temps de précalculs et la place mémoire utilisée sont importants, mais grâce aux remarques mathématiques induites par la linéarité, la méthode des éléments finis permet des animations interactives. Malheureusement, cette approche est trop imprécise en grands déplacements (notamment du fait des phénomènes de "gonflements"). Une limitation 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 un changement de topologie, la matrice de raideur du système se trouve modifiée. Il faudrait alors la reconstruire et la ré-inverser, ce qui est une opération bien trop lourde pour être réalisée en temps réel.

**Éléments de frontières** Dans [James 99], seule la frontière du système est découpée en éléments disjoints, à l'intérieur desquels le champ des déplacements est interpolé linéairement. Il ne peut y avoir de forces externes appliquées ailleurs que sur la surface de l'objet élastique. Une bonne présentation de cette méthode peut être trouvée dans [Hunter 05].

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 nœuds de surface bien que la structure volumique soit prise en compte par les équations. D'autre part, la linéarité du système peut être exploitée pour pré-calculer un ensemble de réponses élémentaires, qui peuvent ensuite être combinées en temps réel, à la manière des approches précédentes.

Malheureusement, parce qu'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 isotrope. De plus, un tel modèle reste très limité dans la gestion des changements de topologie à cause de son intérieur volumique non maillé.

Il s'agit quand même d'une bonne alternative aux maillage volumique tant que la topologie du maillage n'est pas modifiée. Elle peut bien s'adapter à des approches "dépendantes-patient" (patient-specific) qui ne requièrent pas de coupures.

### 3.2.2 Grands déplacements

Puisque les premières méthodes interactives exploitaient fortement les propriétés de linéarité, elles ne permettaient que de petits déplacements en rotation sous peine de gonflements. Ce facteur très limitant a poussé à implémenter des méthodes gérant les grands déplacements pour plus de libertés dans le mouvement.

**Green-Lagrange** La première solution a consisté à utiliser le tenseur non-linéaire de déformation de Green-Lagrange (équation (2.1)). Il a été intégré par [Volino 95] pour des triangles, où les contraintes peuvent être directement exprimées en fonction des longueurs d'arêtes et de leurs angles, puis par [Pincibono 00b, Debonne 01] sur des tétraèdres dans un système masses-tenseurs. Les déplacements en rotation étaient alors possibles pour un meilleur réalisme, mais au prix d'une plus grande complexité du calcul de déformations. C'est pourquoi ces méthodes ont été proposées avec des intégrations explicites, car l'intégration implicite impliquant la résolution d'un système non-linéaire serait très inefficace. Il est donc difficile de profiter de la robustesse de l'intégration implicite avec des déformations non-linéaires.

**Corotationnel** L'idée des approches corotationnelles est d'utiliser un calcul linéaire des déformations (équation (2.2)) et de gérer indépendamment l'invariance en rotation.

L'idée principale est d'éliminer les rotations locales subies au cours de l'animation en se ramenant dans un repère global non tourné 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 tout en restant au plus proche de la loi de comportement. Les forces calculées subissent ensuite la rotation inverse pour être appliquées à l'objet dans le repère tourné.

Localement :  $u_i = x_i^{repos} - R_{repos}^{deforme} x_i^{deforme}$  et  $f_i = R_{global}^{deforme} K_i u_i$  avec  $R_j^i$  la rotation du repère  $i$  au repère  $j$ .

Différentes méthodes ont été proposées pour calculer cette rotation.

■ **Un repère par objet** Une première proposition a été très tôt intuitée dans [Terzopoulos 88b] où un repère est fixé à l'objet animé, à la manière d'un corps rigide. Dans ce cas, seule la rotation globale à l'objet peut être éliminée, ce qui est performant quand l'objet bouge beaucoup dans l'espace mais se déforme peu localement comme illustré dans la figure 2.19.

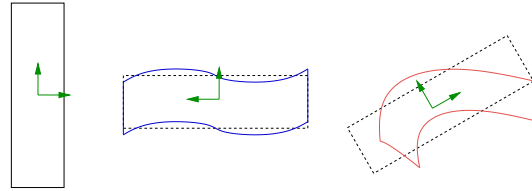


FIG. 2.19 – Une seule rotation pour tout l'objet

*gauche : forme au repos. milieu : suffisant pour des déformations locales petites. droite : ne permet pas de gérer les grandes rotations locales.*

■ **Un repère par nœud** [Müller 02] propose de trouver les rotations subies par chaque nœud des éléments. Pour trouver ces rotations, il utilise un repère orthonormé direct construit à partir des arêtes incidentes au sommet traité. La méthode de construction des repères étant toujours la même, il suffit d'extraire la rotation entre le repère du nœud sur l'objet au repos et sur l'objet déformé pour obtenir la rotation subie par le nœud au cours de l'animation (figure 2.20). Tous les éléments incidents à un nœud doivent être tournés de sa rotation afin de déterminer le petit déplacement et la déformation de ce nœud dans tous ses éléments. Chaque tétraèdre doit donc être tourné quatre fois (le nombre de sommets d'un élément) (figure 2.21). En tout, il faut donc appliquer  $\text{nb\_éléments} \times \text{nb\_nœuds}$  rotations.

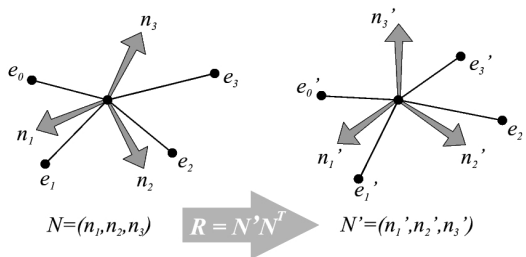


FIG. 2.20 – Repère au sommet

*Pour trouver la rotation qui ramène l'élément au repos dans le repère de l'élément 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  $R = N'N^T$ . [Müller 02]*

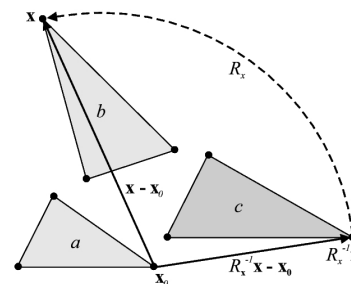


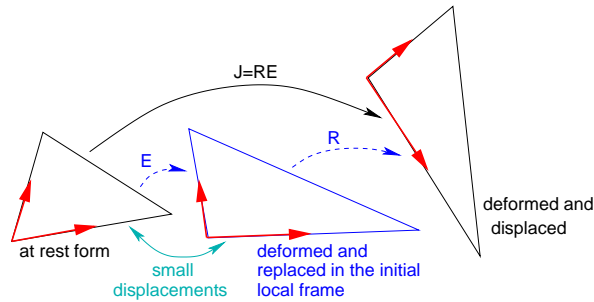
FIG. 2.21 – Elimination de la rotation par sommet

*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  $c$ . [Müller 02]*

Si cette méthode gère les grands déplacements, par contre, le fait de traiter les rotations locales par sommet ne respecte pas la physique. Müller parle de forces fantômes qu'il est obligé de compenser. D'après lui, le fait de travailler nœud par nœud ne garanti pas une somme des forces nulle, c'est pour cette raison qu'il a depuis abandonné cette méthode pour un repère local par élément (paragraphe suivant).



■ **Un repère par élément** Puisque traiter une rotation locale par sommet n'est pas entièrement satisfaisante, il suffit de la traiter par élément. Il faut trouver une rotation par élément. Toutes les forces issues d'un élément peuvent ainsi être calculées après une seule rotation de celui-ci comme illustré en figure 2.22. Deux approches (dont la comparaison est illustrée en figure 2.23) permettent d'évaluer la rotation d'un élément.



**FIG. 2.22** – Elimination de la rotation par élément

#### *Par décomposition polaire*

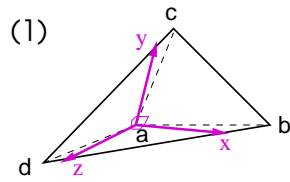
[Etzmuß 03b] a proposé de trouver cette rotation sur des triangles, puis [Müller 04a] a extrapolé cette approche aux volumes sur des tétraèdres. L'idée est d'extraire les vecteurs et valeurs propres de la transformation géométrique entre l'élément au repos et dans son état déformé. On peut voir géométriquement ces vecteurs comme des repères qui répartissent au mieux la matière de l'élément autour d'eux.

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 pour garantir la meilleure précision.

Si 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 un peu coûteuse.

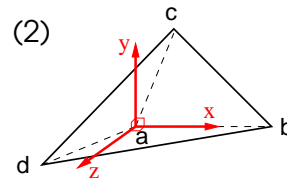
#### *Par décomposition QR*

A cause de ce coût de calcul, nous avons proposé de trouver une rotation plus simplement [Nesme 05a]. L'idée est de caler un repère orthonormal local par élément en prenant un sommet comme origine, la première arête comme premier axe, en mettant le deuxième axe dans un plan de l'élément et en déduisant le troisième axe par construction. Cette méthode est donc dépendante de l'ordre d'énumération des sommets, et ne donne pas les plus petits déplacements. Par contre elle donne toujours des petites déplacements et permet donc l'utilisation du tenseur de Cauchy. Son avantage est d'être très rapide à calculer, et de rester stable face aux configurations non physiques telles que les inversions d'éléments, quand la décomposition polaire devient numériquement incalculable.



*Décomposition polaire*

Repère unique reflétant au mieux la disposition de la matière, au plus près des arêtes, utilisant les valeurs et vecteurs propres.



*Décomposition QR*

Le premier axe est la première arête  $AB$ , le second axe est orthogonal au premier et dans le plan  $(AB, AC)$ , et le dernier axe est obtenu par construction d'un repère orthonormé.

**FIG. 2.23** – Différents repères locaux

### 3.2.3 Changements de topologie

Le système mécanique se base sur une représentation volumique de l'objet. Lors de changements de topologie de cette représentation, il doit se remettre à jour. Afin de traiter interactivement les découpes et les déchirures, cette mise à jour doit être efficace.

**Système masses-tenseurs** [Delingette 99, Picinbono 00b] ont proposé une manière adaptée et élégante pour mieux gérer les problèmes de mise à jour face aux changements topologiques. Le principe est de résoudre les équations très localement de manière à ne devoir remettre à jour que ce qui est nécessaire.

La première idée pour rendre locaux les calculs est de "lumper" la matrice de masse afin de concentrer la masse sur les nœuds plutôt que sur les éléments (mass-lumping). Les nœuds deviennent ainsi des particules à l'idée d'un système masse-ressort, et l'équation de la dynamique peut être formulée indépendamment sur chaque nœud du maillage. Les intégrateurs explicites simples et rapides demandant une inversion de la masse peuvent ainsi être aisément implémentés (puisque le lumping donne une matrice diagonale, inversable trivialement). Afin de rendre aussi local la matrice de raideur, les auteurs partent du principe que celle-ci représente les interactions entre des couples de nœud et décident de stocker ces entrées sur les couples correspondant (*i.e.* les arêtes ou les nœuds du maillage) pour chaque élément. Dans ce contexte, 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 nœud  $i$ ,  $K_{ii}$  représente la somme des contributions du sommet  $i$  dans tous les tétraèdres adjacents à  $i$  (chacune de ces contribution étant stocké dans le nœud),  $K_{ij}$  représente les sommes des contributions des autres sommets  $j$  de tous les tétraèdres adjacents à  $i$  (ces contribution étant stockées dans les arêtes).

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. Ainsi si un élément apparaît ou disparaît, tout le reste des entrées déjà calculées restent valides. L'assemblage n'est pas explicitement réalisé, et la matrice globale n'est donc pas directement connue. Mais ce n'est pas un problème pour utiliser une intégration dynamique, facilement réalisable grâce à la répartition de la masse sur chaque nœud.

Le seul désavantage notable apparaît au niveau de la formulation de la masse lumpée sur les nœuds, même si ce procédé est classique en mécanique pour simplifier des calculs très complexes, sa justification physique n'est pas toujours claire. Il se justifie principalement par sa facilité à l'utilisation d'une intégration explicite du temps, mais dans le cas d'une intégration implicite où l'on n'a pas besoin d'inverser la matrice de masse, son utilité est réduite.

**Élément par élément** D'autres approches [Müller 04a, Nesme 05a] n'utilisent pas d'assemblage, dans l'idée de traiter localement les problèmes afin d'avoir une remise à jour amoindrie par un changement de topologie. Les matrices locales étant idéalement stockées au niveau des éléments, les mises à jour suite à un changement de topologie peuvent être très localisées. De plus nous avons montré que dans le cas d'une intégration implicite, l'utilisation d'un assemblage n'est pas valable lors des premières itérations, qui sont les seules à pouvoir être calculées dans un contexte interactif [Nesme 05a].

### 3.2.4 Stabilité

Une difficulté inhérente à la modélisation physique d'objets déformables concerne la stabilité, qui est d'autant plus importante dans un contexte interactif où on est honoré de la présence d'un utilisateur peu contrôlable. Même si ses mouvements peuvent être limités, il peut toujours introduire des phénomènes complexes à résoudre avec des contraintes sévères, un bon modèle se doit de rester stable, quelle que soit la situation. Des problèmes de stabilité sont aussi liés à des mauvais conditionnements numériques. Il est par exemple plus difficile de résoudre une simulation pour un matériau peu-compressible (coefficient de Poisson élevé) car la matrice de raideur correspondante est moins bien conditionnée, car la matrice déformations-contraintes  $\mathbf{D}$  voit ses entrées se rapprocher d'autant plus de l'infini lorsque le taux de Poisson approche de 0.5. Donc pour deux problèmes quasi-similaires (même objet, même maillage, mêmes équations), faire varier le taux de Poisson peut avoir une grosse influence sur la stabilité et la vitesse de résolution. Pour l'instant peu de travaux parlent de ces problèmes, et la vitesse de calcul est généralement donnée en fonction du nombre de particules ou d'éléments, alors que d'autres paramètres influent énormément. Ces vitesses sont donc généralement fournies pour des matériaux efficaces à simuler, *i.e.* avec une raideur faible et une faible incompressibilité.

**Configurations non-physiques** Dans certaines configurations, où les contraintes sont fortes, des éléments peuvent complètement s'aplatir ou carrément subir une inversion (se "retourner comme des chaussettes"). Face à cette situation non prévue dans les équations, le modèle peut diverger, ou alors l'élément peut trouver un état d'équilibre dans l'état inversé. Ces résultats dépendent de la méthode de résolution numérique et du matériau simulé. Ces conséquences ne peuvent malheureusement pas être ignorées pour un simulateur interactif. Ce cas non-physique n'est pas censé arriver, mais il n'est pourtant pas rare. Ce n'est pas parce qu'un élément s'inverse qu'il faut que la simulation avec laquelle on interagit se stoppe.

[Irving 04] propose une élégante méthode pour traiter ces inversions lors d'un calcul corotationnel des déformations sur des tétraèdres. Il considère que l'inversion est celle du nœud qui a la plus petite déformation, mais cette approche n'est pas destinée pour le temps réel. Dans [Nesme 05a] les inversions sont directement prises en comptes, sans gestion spécifique, mais de façon moins soignée, car pour un élément donné ce sera toujours le même nœud qui sera considéré comme inversé. Par contre, l'inversion est traitée sans calcul supplémentaire, ce qui est un fort avantage pour l'interactivité.

Puisque les phénomènes d'aplatissement et d'inversion sont non-physiques, on ne peut pas vraiment considérer de réalisme, au mieux une plausibilité. L'important et l'essentiel dans ce genre de

situations est de rester stable, est de revenir dans une configuration physiquement possible. Pour cela, les deux approches décrites ci-dessus sont satisfaisantes.

**Incompressibilité** La stabilité liée à l'incompressibilité dépend essentiellement des lois de comportement, de la forme de éléments employés et de la résolution numérique.

Sous une grosse compression, certaines lois de comportement donnent des forces infinies (plus proches de la réalité, mais moins stables), alors que d'autres tendent au contraire à s'annuler (peu réalistes). Ce problème est très vrai en médical, car un grand nombre d'organes sont gorgés d'eau, les rendant quasi-incompressibles. [Irving 07] propose une méthode pour gérer l'incompressibilité sur des tétraèdres linéaires à l'aide de contraintes sur les volumes.

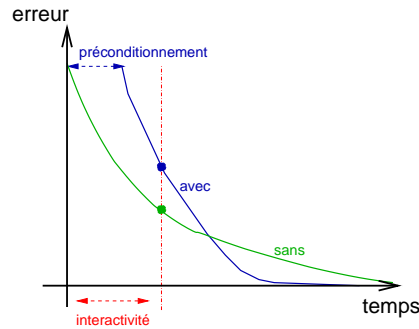
Un problème bien connu en mécanique se nomme le *locking effect* [Hughes 87, Joldes 06]. Il intervient sur les tétraèdres linéaires quasi-incompressibles, qui peuvent devenir trop raides par rapport à la matière qu'ils sont censés représenter, voire se bloquer. Même si des contraintes de volume sur le dual permettent de mieux traiter ce problème [Joldes 06], les hexaèdres ou les tétraèdres non-linéaires sont généralement préférés dans les simulations en mécanique. Pourtant les tétraèdres linéaires sont majoritairement utilisés dans les modèles interactifs, même si dans de nombreux cas l'incompressibilité est de mise, comme pour la majorité des organes en simulation chirurgicale.

De plus, numériquement, quand le coefficient de Poisson est élevé, la matrice de raideur est moins bien conditionnée, offrant un système implicite plus difficile à résoudre. Dans ce cas la solution interactive sera plus approchée et donc la simulation moins stable.

**Matériaux raides** Un autre problème peu soulevé concerne la restitution de la raideur en mode interactif. En effet, plus un matériau est raide, plus une déformation devrait être transférée à tout l'objet rapidement. Or les schémas d'intégration actuels (explicites ou implicites) ne permettent pas une propagation rapide des déformations en temps réel. Dans le temps imparti, il n'est possible de calculer qu'une solution aux équations de la dynamique trop approchée (trop grands pas de temps par exemple). Concrètement, quand l'objet est manipulé interactivement, il semble "gélatineux" et réagir trop lentement aux sollicitations.

Peu de travaux ont été proposés pour améliorer ce phénomène dans le cadre d'intégrations explicites pour lesquelles il faut réduire le pas de temps en conséquence. A notre connaissance, la seule proposition est un espèce de cycle en V de multigrid effectuant des intégrations explicites à différentes échelles [Wu 04] afin que chaque intégration d'une résolution fine démarre sur un meilleur état initial, approché par l'intégration d'un niveau grossier. Malheureusement nous trouvons les justifications numériques de cette approche multirésolution peu convaincantes puisqu'il ne s'agit pas d'une méthode multigrid améliorant la convergence de la résolution d'un système linéaire (de plus l'algorithme présenté dans cet article ne semble pas correspondre au discours associé).

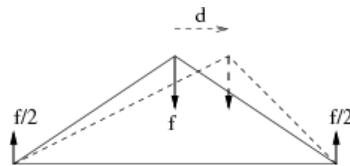
En implicite, mieux propager revient à avoir une meilleure convergence dans les premiers instants de la résolution du système associé. On essaie en effet d'avoir la meilleure solution approchée possible dans un certain laps de temps. Pour cela, plusieurs préconditionneurs ont été proposés. S'ils permettent de converger plus rapidement vers la solution "exacte", il ne semble pas qu'ils apportent un énorme avantage dans le cadre d'un calcul interactif. En effet, dans un cadre interactif, le temps imparti pour approcher la solution est très court, et seules les premières itérations de la résolution du système peuvent être calculées. Or le préconditionnement consume une partie de ce temps au détriment de la résolution à proprement parlée. Au bout du court temps imparti, la solution ne sera pas forcément meilleure que sans préconditionnement, comme dans l'exemple schématisé dans la figure 2.24.



**FIG. 2.24** – Schématisation des convergences avec et sans préconditionnement

*Dans un contexte interactif, le préconditionnement de la résolution de l'intégration implicite n'est pas forcément valable car même s'il permet de converger plus rapidement vers la solution "exacte", il ne donne pas forcément une meilleure solution en temps réel.*

**Moments fantômes** En élasticité linéaire, la variation de la matrice "déplacements-déformations"  $\mathbf{B}$  est souvent négligée car l'objet n'est pas censé subir de grands déplacements. Du coup les méthodes basées sur les déformation linéaires de Cauchy considèrent que la matrice de raideur est constante [Müller 02]. Or si  $\mathbf{B}$  n'est pas mise à jour, la loi d'Euler n'est pas nécessairement satisfaite. Une rotation pure  $\omega$  génère des déplacements  $\Delta u_j = \omega \times u_j$  mais aucune déformation. Cela implique  $\Delta \varepsilon_i = \Sigma_j B_{ij}^T \omega \times u_j = 0$ , ainsi  $\Sigma_j u_j \times B_{ij} = 0$ . Le couple résultant d'une contrainte arbitraire  $\sigma$  est alors  $\Sigma_j u_j \times f_j = \Sigma_j u_j \times \Sigma_i B_{ij} \sigma_i = \Sigma_i \sigma_i \Sigma_j u_j \times B_{ij} = 0$ . Le couple résultant des forces élastiques est nul par construction de  $\mathbf{B}$ . Cependant, ceci n'est plus vrai lorsque  $\mathbf{B}$  est obsolète due à des changements de formes, car les  $u_j$  originaux sont remplacés par d'autres dans la relation précédente. Il est donc nécessaire de recalculer  $\mathbf{B}$  pour chaque élément afin d'éviter des moments créés artificiellement. Un exemple de couple fantôme est donnée en figure 2.25.



**FIG. 2.25** – Couple fantôme.

*L'application d'un déplacement (non montré) sur une forme initiale (lignes pleines) résulte dans jeu de forces au couple nul (flèches pleines). Appliqué sur la forme modifiée (pointillés), ce même déplacement provoque un couple fantôme  $d \times f$ . D'où la nécessité de prendre en compte les formes déformées à chaque pas de temps à travers la matrice déplacements-déformations  $\mathbf{B}$  afin d'appliquer les forces correspondantes à l'état déformé.*

De ce fait, même en corotationnel, le système est non-linéaire car  $\mathbf{B}$  dépend des déplacements. Mais le considérer comme linéaire avec  $\mathbf{B}$  invariant durant un pas de temps est une bonne approximation qui permet de ne pas créer de couple fantôme dans l'exemple figure 2.26.

Remettre à jour  $\mathbf{B}$  a un coût non négligeable. Un avantage de la méthode corotationnelle par décomposition QR [Nesme 05a] est de pouvoir calculer la matrice  $\mathbf{B}$  pour un coût très faible (formules données en annexe D).



vidéo 2

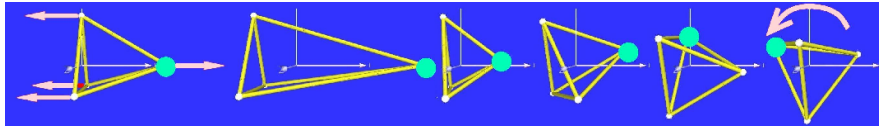


FIG. 2.26 – Conséquence d'un moment fantôme

*Des forces au couple nul sont appliquées à un tétraèdre équilatéral. Si la matrice déplacements-déformations  $\mathbf{B}$  n'est pas mise à jour, le tétraèdre se met à tourner sous l'effet d'un couple généré artificiellement.*

### 3.2.5 Matériaux

La plupart des modèles mécaniques présentés sont implémentés uniquement avec des matériaux élastiques linéaires isotropes (suivant la loi de Hooke), mais certains auteurs ont proposé des méthodes efficaces pour traiter des matériaux plus complexes.

Par exemple, la visco-élasticité est classiquement simulée en utilisant l'amortissement de Rayleigh comme étant une combinaison linéaire de la raideur et de la masse  $\mathbf{C} = \alpha\mathbf{K} + \beta\mathbf{M}$ .

[Schwartz 01] propose une formulation simple d'une élasticité non-linéaire pour laquelle la raideur augmente sous des grandes déformations en compression.

[Hauth 04b] propose de quantifier et simuler efficacement l'hystérésis et donc la visco-élasticité d'un matériau élastique à l'aide du modèle Q. Le facteur Q est un terme mesurable qui décrit la partie visqueuse d'un matériau visco-élastique. Il est infiniment grand pour un matériau purement élastique, et 0 pour un matériau purement visqueux. Pour la plupart des matériaux mous le facteur Q est constant, réduisant l'amortissement à une fonction des propriétés élastiques.

La plasticité peut être représentée en ajoutant une force aux particules qui ont subi une déformation ayant dépassée un seuil [Terzopoulos 88a, O'Brien 02, Müller 04a].

[Picinbono 00a] propose de gérer l'isotropie transverse en attribuant des comportements différents au matériau pour des directions données pour mieux simuler des organes humains, tels que les ligaments ou les yeux.

En proposant une formulation permettant de représenter les déformations uniquement en fonction des longueurs des arêtes des éléments, [Goulette 07] peut facilement dériver n'importe quelle fonction d'énergie et ainsi traiter efficacement plusieurs types de matériaux non-linéaires hyperélastiques.

**Identifier des vraies lois** En mécanique, depuis longtemps, on tente d'extraire des paramètres mécaniques aux matériaux, afin de mieux quantifier théoriquement ou numériquement leur comportement. Pour identifier ces paramètres, il est généralement nécessaire de maltraiter (tirer, casser, percer...) un objet à la forme et la quantité connue. Ces tests sont bien sûr plus difficilement réalisables en biomécanique. Même si certaines études poussées existent sur les parties relativement rigides, tel que les os, les tendons, les muscles. L'étude des tissus mous (peau, organe, cerveau...) est beaucoup plus difficile, car leur comportement est très changeant d'un patient à l'autre, même dans certaines zones d'un même patient, et, bien sûr, la réactivité est très différente sur un corps mort que sur des éléments in-vivo [Miller 00, Ottensmeyer 04, Gerard 05, Hendriks 06]. Or, ce sont les paramètres in-vivo qui sont les plus intéressants. Pour cela de nouvelles techniques de quantification sont apparues, utilisant l'ingénierie inverse, c'est-à-dire, en utilisant une simulation pour déduire les paramètres en fonction d'une expérience dont les conditions aux limites et les résultats sont connus. Il s'agit principalement de l'analyse de la déformation avec un pompage par pipette [Theret 88, Alexopoulos 05, Bou-

dou 06, Gerard 05, Hendriks 06] ou d'analyse de modes vibratoires créés par ultrasons [Ophir 91, Levinson 95, Dutt 96, Catheline 99, Sandrin 02].

On peut espérer des données rhéologiques relativement précises d'un patient dans les prochaines années, à la manière de données scanner, où chaque voxel portera des propriétés mécaniques propres. Il faudra alors intégrer ces données dans les modèles mécaniques numériques.

### 3.2.6 Éléments quadratiques

[Mezger 06] propose d'utiliser des tétraèdres quadratiques pour simuler interactivement des corps déformables, c'est-à-dire dont l'interpolation interne est non-linéaire (*cf.* les différents types d'éléments en section 3.1.2). Les arêtes des éléments ne sont donc plus des lignes, mais des courbes. Les calculs sont alors plus longs, puisque les éléments peuvent mieux se déformer, mais moins d'éléments sont nécessaires pour obtenir des résultats identiques à ceux obtenus avec des petits éléments linéaires. Une approche corotationnelle a été adaptée pour ces éléments, et il semblerait que leur rapport temps de calcul/précision soit plus efficace que celui des tétraèdres linéaires. De plus, théoriquement, cette approche devrait éviter les problèmes liés au locking effect de tétraèdres linéaires pour les matériaux incompressibles.

## 3.3 Validation

La validation d'un modèle numérique est l'étape où est testé en pratique le comportement. Elle est très importante dès qu'on se soucie de la précision. Malheureusement, elle est souvent laissée de côté, surtout pour les modèles développés dans un contexte ludique.

[Debunne 00b] a étudié la dynamique fréquentielle par une analyse modale, de chacun des niveaux de résolution de son approche multirésolution, afin de voir quel était le tenseur de déformations le plus adapté à simuler un même comportement à différentes échelles. Ceci ne consiste pas vraiment en une validation vers la réalité, mais déjà en une prise de conscience qu'il ne suffit pas que les choses bougent, mais qu'elles bougent de façon cohérente.

Pour aller plus loin, il faut comparer les résultats numériques avec des résultats théoriques, avec d'autres résultats numériques dont la solution est certifiée, voire carrément avec des données réelles.

Dans ce sens, [Kerdok 03] fournit des données de déplacements d'un cube de silicone soumis à des contraintes connues.

Si ceci est faisable pour analyser des positions à des états d'équilibre contraints, c'est encore très compliqué de le faire pour évaluer la dynamique (quel chemin est emprunté pour arriver à ces états d'équilibre). Les schémas d'intégrations semblent pourtant relativement empiriques. Rien n'est encore proposé pour cela ; il faut trouver des expériences et des critères de comparaison, et sûrement se tourner vers les mécaniciens qui ont déjà dû classer les matériaux par leur comportement dynamique. Il faut savoir que pour les calculs de hautes précisions, par exemple en simulation de crash-test automobile, seuls les intégrateurs explicites sont utilisés [Kirkpatrick 01]. Ces énormes simulations sont très lentes, puisque étant donnée la raideur des matériaux (métaux) les pas de temps sont très petits. Il pourrait alors être intéressant d'analyser les trajectoires des autres modèles par rapport à ceux-ci.

Nous pensons, tout comme [Miller 06], que dans un premier temps il n'est pas utile de comparer les méthodes numériques à des données réelles (par exemple issues de vraies opérations). Se basant sur la même théorie, les résultats en terme de précision ne seront pas meilleurs que ceux des méthodes numériques éprouvées (type [Ansys]). Il nous paraît déjà une bonne étape de se comparer à ces méthodes en terme de précision, et d'essayer de les approcher au mieux. De plus, les comparaisons avec

des données réelles intègrent une énorme problématique de modélisation [Marchal 06a], plutôt que d'un test valable d'un modèle numérique.

## 4 Conclusion

La Méthode des Éléments Finis est largement prédominante dans l'animation de corps déformables. Son fondement sur la mécanique des milieux continus lui assure une réalité physique et une intégration possible de vrais matériaux. Diverses implémentations ont été proposées, avec chacune leurs spécificités et leurs simplifications, que ce soit au niveau de la rapidité, de la stabilité, de la précision ou de la gestion des changements de topologie. Nous partons avec l'idée de prendre tout ce qui est intéressant dans ces approches afin de mettre au point un modèle reprenant un maximum de ces caractéristiques, tout en se basant sur de solides fondements mécaniques. La gestion de différents types de matériaux ne nous intéresse pas encore ; nous pensons qu'il faut d'abord valider un modèle rapide et stable et qui pourra gérer facilement les changements de topologies.







---

# Milieu mécanique multirésolution



## Cheminement

De gros efforts ont déjà été réalisés pour implémenter efficacement la méthode des éléments finis afin d'utiliser la théorie de la mécanique des milieux continus dans des simulations interactives d'objets déformables. Malheureusement leur efficacité est pleinement effective dans des cas assez rares en utilisation concrète. Il faut par exemple des maillages avec de très bons conditionnements et cette modalité est très dure à respecter sur des objets complexes, spécialement contenant des parties fines et étroites. En effet, la plupart des modèles physiques ont des variations d'échelles que les méthodes de résolutions numériques devraient prendre en compte pour optimiser les calculs. C'est pourquoi nous sommes persuadés que la multirésolution adaptative est une notion essentielle en animation physique interactive. Cette idée n'est pas nouvelle, elle est largement décrite en mécanique où l'on ne désire fréquemment une grande précision qu'à certains endroits. Elle est beaucoup exploitée en informatique graphique pour toutes sortes d'applications interactives, mais elle n'a pas été massivement étudiée en animation physique interactive ; la difficulté majeure résidant dans le fait de respecter les lois physiques malgré l'utilisation de différentes couches. Certaines méthodes adaptatives acceptent de grosses concessions pour passer certains calculs au niveau grossier entraînant une perte de la validation du sens physique des calculs [Debunne 01, Dequidt 05]. Un autre manquement actuel que nous avons souligné concerne la propagation dynamique des matériaux mal conditionnés. Une simulation hiérarchique sera mieux adaptée, car le problème pourra être traité à la fois dans les hautes et dans les basses fréquences.

Adapter localement la résolution induit des complications :

- **maillages** : il faut être capable de créer des maillages à différentes résolutions tout en garantissant leur bon conditionnement. Le tout avec des opérateurs qui permettent de passer facilement et rapidement d'un niveau à l'autre (pour enlever/ajouter des degrés de liberté), sans perdre l'avantage de la multirésolution en gaspillant du temps dans les transitions entre niveaux.
- **résolution efficace** : il faut être capable de résoudre un système à différentes échelles, tout en garantissant une bonne convergence, aussi bien au niveau grossier qu'au niveau fin où la complexité explose.
- **respect de la théorie** : il faut rester le plus rigoureux concernant les lois physiques et ne pas s'éloigner de la théorie mécanique malgré l'approche adaptative.

Actuellement, les problèmes sont bien séparés entre la génération de maillages, les calculs de forces et l'intégration du temps. L'approche générale regroupe en effet les étapes suivantes : acquisition des données, création du maillage mécanique, puis calcul de la mécanique. Chaque étape prenant ce qu'on lui donne en entrée pour ressortir ce qui lui semble le mieux par rapport au besoin de l'étape suivante. Nous pensons cependant qu'il est possible de faire mieux en traitant le problème dans son ensemble.

Afin de répondre au problème des maillages, dans ces travaux, nous faisons le choix des grilles régulières. Nous faisons donc un sacrifice au niveau des bords des objets qui ne sont plus décrits avec précision. Mais ce choix permet la gestion facile du bon conditionnement du maillage mécanique, qui est un point crucial des calculs interactifs. De plus, construire une hiérarchie de grilles régulières est trivial, et une représentation multirésolution peut aisément prendre forme. D'autre part, les niveaux de la hiérarchie peuvent facilement être emboîtés sous la forme d'un arbre octal afin d'optimiser les transitions entre niveaux en ayant des structures de données bien connues, pratiques à créer et à manipuler.

Pour répondre à un fort besoin théorique, une approche est proposée pour limiter les problèmes liés aux grilles régulières concernant les bords du milieu. En effet, les éléments frontières d'une grille régulière ne représentent pas directement l'objet, et il faut être capable de prendre en compte la répartition non régulière de la matière à l'intérieur pour simuler le bon comportement de ces éléments frontières et pourvoir traiter les conditions aux limites avec finesse.

Une fois cette brique de base bien en place, nous avons étudié diverses solutions pour améliorer le dynamisme d'une animation, dans l'idée de bien rendre la raideur des matériaux interactivement. Gardons en tête que notre but n'est pas d'avoir la méthode qui donne la solution exacte dans le laps de temps le plus court, mais d'obtenir la meilleure solution approchée possible dans un temps donné.

Pour finir, une grosse partie de notre travail a concerné la mise en situation de notre modèle pour tester sa validité. En effet, suivre les lois théoriques est une première étape nécessaire, mais il faut ensuite s'assurer que notre interprétation fournit des résultats cohérents. Cette étape est d'autant plus importante dans le cas de simulations médicales où la précision est une priorité nécessaire. Très peu de validations mécaniques ont malheureusement été effectuées sur les modèles interactifs déjà présentés dans la littérature.

Pour résumer, l'utilisation d'hexaèdres en animation nous paraît intéressante car elle nous permet de proposer un modèle répondant à plusieurs besoins difficiles à mettre en œuvre, à savoir :

- la génération automatique et rapide d'une représentation hiérarchique avec un bon conditionnement,
- une mécanique adaptative simple, rapide et stable qui suit les lois de la mécanique des milieux continus,
- un compromis précision-temps de calcul,
- la simulation interactive de matériaux mal conditionnés.

Nous verrons même qu'elle est un bon support pour un rendu volumique efficace.







chapitre **3**

---

# Animation d'objets par grilles de déformations

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>64</b>
<b>2</b>	<b>Principe</b> . . . . .	<b>65</b>
<b>3</b>	<b>Interpolation de la surface</b> . . . . .	<b>67</b>
<b>4</b>	<b>Conditions aux limites sur la surface</b> . . . . .	<b>68</b>
	4.1 Contraintes "dures" . . . . .	68
	4.2 Forces externes . . . . .	68
<b>5</b>	<b>Multirésolution</b> . . . . .	<b>69</b>
<b>6</b>	<b>Résultats</b> . . . . .	<b>70</b>
<b>7</b>	<b>Limitations</b> . . . . .	<b>71</b>
<b>8</b>	<b>Conclusion</b> . . . . .	<b>72</b>

---

## 1 Introduction

Alors que résoudre les équations de la mécanique sous-entend d'avoir une représentation volumique maillée de l'objet, il est difficile de construire un tel maillage bien adapté pour le temps réel. La manière la plus intuitive, et donc la plus répandue, pour mailler volumiquement un objet représenté généralement par une surface triangulée est de le tétraédriciser. Les tétraèdres peuvent cependant présenter des problèmes de comportement dans certaines configurations ("locking effect" par exemple), c'est pourquoi les hexaèdres leurs sont parfois préférés en mécanique. Malheureusement la construction automatique de maillages hexaédriques est encore plus délicate, et aucune méthode robuste n'existe à notre connaissance.

A nos yeux, les critères importants concernant la construction d'un maillage volumique pour résoudre la méthode des éléments finis et adapté au temps réel sont :

- l'automatisation de la construction
- la rapidité de construction
- le choix de la résolution (taille/nombre d'éléments)
- les différentes résolutions emboîtées (pour la multirésolution)
- la régularité (les maillages réguliers sont bien adaptés aux résolutions numériques interactives)
- les données d'entrée variées (surfaces, même imparfaites pouvant être non-manifold ou comporter des trous, volumes)

Il existe divers algorithmes pour la génération de maillages tétraédriques basés sur Delaunay 3D [Watson 81], la projection des frontières [Loehner 88, Phai 82], et les arbres octaux [Yerry 84]. Toutes ces approches se concentrent principalement sur la qualité des éléments générés, en considérant leur forme, leurs angles, leurs volumes et divers autres mesures importantes pour le bon conditionnement du système résultant [Liu 94]. Un état de l'art détaillé sur le génération de maillages peut être trouvé dans [Bern 00].

Traiter les modèles surfaciques n'est pas suffisant, car il existe aussi des représentations volumiques des objets, qui correspondent à une discrétisation spatiale sous forme de voxels. Et particulièrement dans le monde médical, où les acquisitions de données réelles exploitables se font grâce à des scanners ou IRM. A notre connaissance, aucune méthode actuelle ne permet de directement animer de telles données. L'approche classique consiste à déduire une représentation surfacique à partir des voxels (avec un algorithme de type Marching Cubes [Lorensen 87]), puis de refabriquer un maillage volumique à partir de cette surface. Ces deux étapes restent complexes et relativement longues, et engendrent chacune des approximations. De plus, partir d'une représentation volumique, puis passer par une représentation surfacique pour revenir à une représentation volumique peut sembler quelque peu incohérent. Surtout lorsque les données de départ sont la représentation virtuelle la plus précise dont on dispose.

En général, la frontière du maillage volumique correspond exactement au maillage surfacique donné en entrée (suite aux Marching Cubes par exemple). Si on veut ce maillage très raffiné pour un bel affichage, on va se retrouver avec énormément de petits tétraèdres qui seront impossibles à animer interactivement. Pour accélérer les calculs le nombre d'éléments doit être réduit et la surface grossière en résultant offre un rendu médiocre. Certains maillieurs s'affranchissent de conserver exactement la surface initiale [TetGen] afin de garantir des maillages volumiquement bien conditionnés, mais l'objet est alors encore plus approximé. Dans tous les cas, on maîtrise assez mal le nombre et la taille des

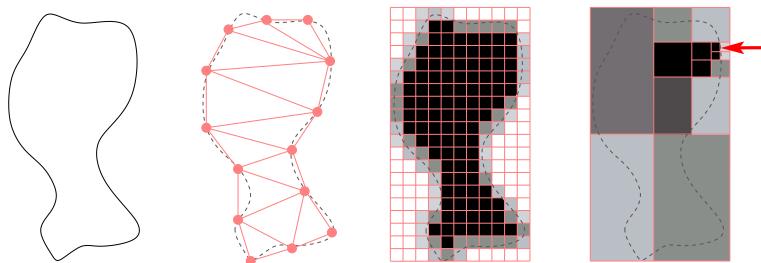
éléments obtenus, et il est très difficile de représenter un objet à diverses résolutions afin de mettre en place une méthode mécanique multirésolution et encore moins de façon automatique. De plus cette opération peut être relativement coûteuse en temps de calcul, ce qui peut poser problème dans les cas où on l'on a besoin d'un modèle spécifique rapidement.

En parallèle, en informatique graphique, des maillages non-conventionnels en mécanique mais bien adaptés aux simulations temps réel ont été proposés. Leur proposition est d'englober la surface de l'objet dans des éléments grossiers constituant un maillage mécanique régulier. [Müller 04a] utilise des tétraèdres englobants qui rendent une description hiérarchique difficile car le passage d'un niveau à l'autre, pour ces maillages non emboîtés [Debunne 01, Georgii 06, Otaduy 07], doit se faire par des interpolations qui compliquent quelque peu. [Müller 04c, James 04b, Dequidt 05] préfèrent l'utilisation d'une voxelisation grossière. La surface de l'objet est alors interpolée dans des éléments l'englobant, à la manière des Free Form Deformation (FFD) [Sederberg 86]. Les gros avantages sont la simplicité de la construction et la régularité du maillage obtenu. Par contre, la rigueur mécanique n'étant pas primordiale pour cette communauté, les conditions aux limites ne sont pas appliquées physiquement car les nœuds des éléments à cheval autour de l'objet ne sont pas sur la surface de l'objet.

Malgré tout, c'est cette représentation qui nous semble le mieux remplir les critères d'un maillage mécanique adapté au temps réel. Plus précisément, nous proposons d'utiliser une voxelisation comme maillage mécanique enveloppant l'objet à animer. Afin d'être au plus proche des lois mécaniques, nous montrons comment appliquer des conditions aux limites justifiables physiquement directement à la surface de l'objet. Enfin nous montrons comment une hiérarchie de maillages à différentes résolutions peut facilement être construite afin de pouvoir utiliser une technique d'animation multirésolution.

## 2 Principe

Une voxelisation est utilisée comme représentation volumique, sur laquelle appliquer la mécanique. La comparaison d'un tel maillage avec un maillage volumique classique est schématisée en deux dimensions sur la figure 3.1. Alors que classiquement les frontières du maillage mécanique correspondent à la surface de l'objet animé, avec la voxelisation, des éléments se retrouvent à cheval sur la surface qui est interpolée à l'intérieur.

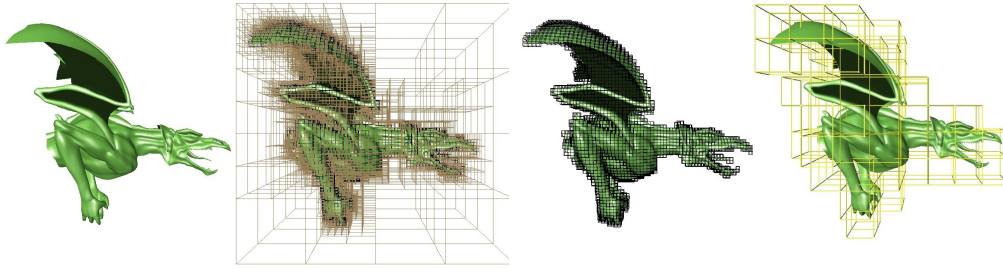


**FIG. 3.1** – Des maillages mécaniques en deux dimensions

*de gauche à droite : l'objet ; un maillage classique à base de triangles ; une résolution de notre maillage voxelisé avec le taux de remplissage ; différentes résolutions suivant un critère multirésolution*

Pour construire des voxelisations d'un objet, il existe une grande variété de schémas, plus ou moins rapides, traitant différentes topologies, voire même basés sur le GPU [Dong 04, Eisemann 06]. Tous peuvent répondre au besoin.

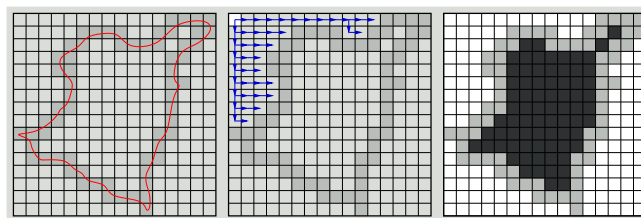
La manière la plus intuitive pour voxeliser un objet représenté par une surface polygonale est de décomposer sa boîte englobante en octree. On part d'un seul élément racine englobant tout l'objet, qu'on subdivise en huit éléments réguliers. On renouvelle récursivement l'opération de division sur chaque élément, et ceci tant qu'un élément contient de la matière ou si le niveau maximal de divisions a été atteint. Un exemple est donné en figure 3.2 à partir de ce principe de subdivision. Il est alors très facile de proposer une représentation hiérarchique de l'objet sous la forme d'un arbre octal qui permettra d'utiliser des méthodes mécaniques multirésolution.



**FIG. 3.2** – Maillage d'un dragon.

*de gauche à droite : la forme de départ. L'arbre octal en entier. Les éléments les plus fins. Une résolution intermédiaire.*

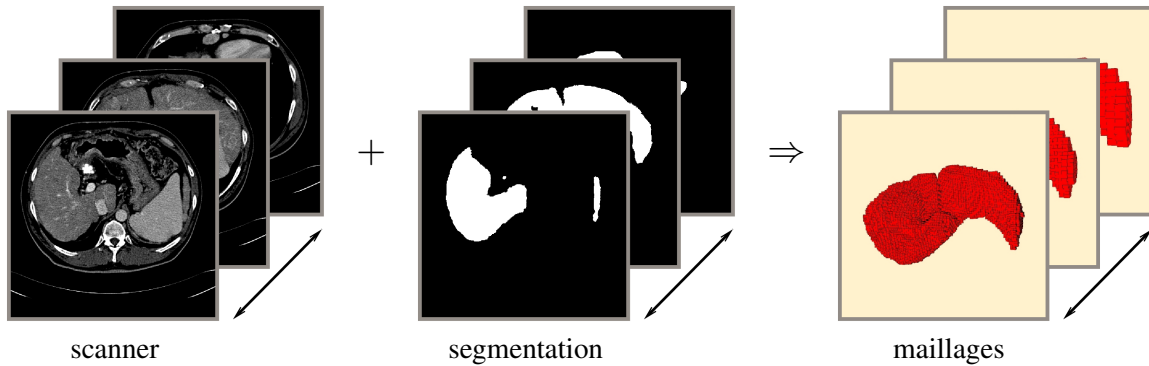
Un point important est de distinguer les éléments vides (extérieurs à l'objet), les éléments pleins (à l'intérieur de l'objet) et les éléments frontières qui contiennent la surface (non entièrement pleins) afin de leur réserver un comportement mécanique adapté. En effet, les éléments vides sont inutiles et donc ignorés, tandis que les éléments frontières vont devoir se comporter différemment des éléments pleins. Pour ce faire, on propose une méthode qui commence par le niveau le plus fin en suivant l'algorithme décrit en figure 3.3. Enfin, ces distinctions de type d'éléments sont remontées vers les résolutions plus grossières.



**FIG. 3.3** – Détection des éléments vides/pleins/frontières

*Tout d'abord, il faut détecter les éléments frontières (gris), i.e. contenant des points de la surface ou intersectant des triangles (rouge). Alors, les autres éléments aux bords de la boîte englobante sont vides (blancs) et se propagent à leur voisinage (flèches bleues). Les éléments restants sont pleins (noirs).*

Fabriquer les maillages mécaniques proposés à partir d'une représentation volumique de type scanner est trivial, puisque l'on dispose d'une voxelisation fine qu'il suffit de regrouper pour obtenir des niveaux plus grossiers, comme illustré en figure 3.4. Si on avait assez de puissance de calcul, on pourrait animer chaque voxel par un élément fini hexaédrique ; mais étant donné leur trop grand nombre, on est amené à les regrouper en hexaèdres plus gros. Notons que l'étape intermédiaire de segmentation est toujours nécessaire, quelle que soit l'approche de maillage utilisée. Il faut en effet pouvoir associer chaque voxel aux différentes régions anatomiques constituant le corps. C'est pourquoi on suppose cette étape de segmentation comme traitée en amont.



**FIG. 3.4** – Plusieurs résolutions de voxelisations à partir de données volumiques segmentées issues d'un scanner médical (ici pour un foie)

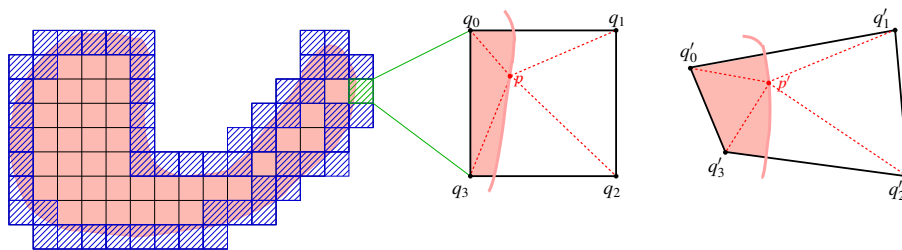
*Données volumiques segmentées (source : IRCAD)*

### 3 Interpolation de la surface

Lorsque les éléments se déforment, l'objet doit suivre ces déformations. Pour cela, à la manière d'une FFD, l'objet est attaché aux éléments englobants : les points de sa surface  $\mathbf{u}(p)$  sont tri-linéairement interpolés en utilisant les coordonnées barycentriques par rapport aux huit nœuds  $\mathbf{u}(q)$  comme illustré en 2D dans la figure 3.5.

$$\mathbf{u}(p) = \mathbf{H}\mathbf{u}(q) \quad (3.1)$$

où la matrice d'interpolation  $\mathbf{H}$  ( $3 \times 24$ ) décrit les influences des sommets de l'élément sur un point de l'objet donné.



**FIG. 3.5** – Interpolation

*Un objet à animer est interpolé à l'intérieur d'éléments l'englobant. Lorsque les éléments se déforment, l'objet suit les déformations.*

Nous nous sommes contentés d'une interpolation linéaire mais des interpolations plus complexes sont envisageables. On pense en particulier qu'il serait intéressant de profiter des polynômes de Bernstein rencontrés dans les FFD afin d'avoir des transitions plus lisses entre des éléments déformés, même si visuellement aucun artefact n'apparaît avec une interpolation tri-linéaire.

Il faut remarquer que c'est la seule étape ajoutée aux méthodes de types éléments finis classiques. Une fois les éléments déformés, il faut calculer les nouvelles positions de l'objet par interpolation, étape qui dépend du nombre de points d'échantillonnage de la surface d'affichage.

## 4 Conditions aux limites sur la surface

Une des grosses différences provoquées par le maillage hexaédrique englobant par rapport aux approches classiques concerne la manière d'appliquer des conditions aux limites. Dans les approches classiques, elles sont appliquées aux nœuds qui se situent sur la surface de l'objet. Mais dans notre approche, les nœuds des hexaèdres qui se situent à cheval sur la surface ne permettent pas d'appliquer des conditions aux limites directement sur cette surface. C'est pourquoi il faut être capable de retranscrire un déplacement ou une force imposé à la surface de l'objet aux nœuds de l'élément l'englobant. C'est une étape indispensable pour le réalisme de la manipulation et des interactions entre objets.

Dans le cas de données volumiques, où il n'y a pas de surface, il faut créer une surface virtuelle pour définir où appliquer les conditions aux limites. Il est possible d'utiliser un algorithme du type marching cubes [Lorenson 87] ou plus simplement de considérer tous les barycentres des voxels frontières au niveau le plus fin (à la résolution des données d'entrée).

### 4.1 Contraintes "dures"

Pour gérer une contrainte de position ou de déplacement sur un point de la surface, il faut être en mesure de retranscrire cette contrainte aux niveaux des sommets de l'hexaèdre le contenant. Concrètement, quand un déplacement est imposé à des points de la surface, il faut déduire les déplacements des nœuds des éléments les englobants tels qu'en les utilisant pour calculer les interpolations aux nœuds contraints, on retrouve les déplacements imposés.

Pour cela on utilise l'inverse de l'interpolation pour déduire comment doivent varier les nœuds des hexaèdres pour que le point interpolé respecte sa contrainte. C'est la même idée que la manipulation directe de Free Form Deformation [Hsu 92] qui cherche les plus petites variations des nœuds au sens des moindres carrés.

Une contrainte en position s'écrit  $\mathbf{u}(p) = \mathbf{c} \Leftrightarrow \mathbf{H}\mathbf{u}(q) = \mathbf{c}$  où  $\mathbf{c}$  est la valeur de la contrainte et  $\mathbf{H}$  est la matrice d'interpolation du point  $p$ . Quand plusieurs contraintes sont appliquées, un système est construit, contenant tous les points  $p$  contraints et tous les nœuds  $q$  alors influencés. Ce système peut être résolu en utilisant une pseudo-inverse de la matrice assemblée  $\mathbf{H}$  :

$$\mathbf{u}(q) = \mathbf{H}^+ \mathbf{c}$$

$$\text{avec } \begin{cases} \mathbf{H}^+ = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T & , \dim(\mathbf{p}) \leq \dim(\mathbf{q}) \\ \mathbf{H}^+ = \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1} & , \dim(\mathbf{p}) > \dim(\mathbf{q}) \end{cases} .$$

Lorsque le problème est sur-contraint, c'est-à-dire lorsqu'il y a plus de contraintes que de nœuds, cette solution donne le meilleur compromis (au sens des moindres carrés). Dans le cas particulier où chaque nœuds est influencé par au plus une contrainte (un seul point de surface contraint par bloc d'éléments),  $\mathbf{H}^T \mathbf{H}$  est un scalaire, et la solution du système est triviale.

### 4.2 Forces externes

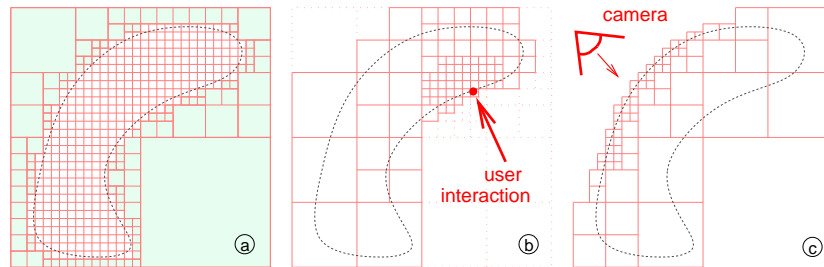
Une force  $\mathbf{f}(p)$  appliquée au point  $p$  de la surface doit être répartie sur les nœuds de l'élément l'englobant. Le principe des puissances virtuelles implique

$$\mathbf{f}(q) = \mathbf{H}^T \mathbf{f}(p)$$

**Preuve :** Les travaux virtuels d'un point de la surface  $\mathbf{p}$  et des nœuds de l'élément l'englobant  $\mathbf{q}$  doivent être égaux, donc :  $\mathbf{f}(\mathbf{q})^T \mathbf{v}(\mathbf{q}) = \mathbf{f}(\mathbf{p})^T \mathbf{v}(\mathbf{p})$  où  $\mathbf{f}$  est la force et  $\mathbf{v}$  une vitesse virtuelle. Or  $\mathbf{v}(\mathbf{p}) = \mathbf{H}\mathbf{v}(\mathbf{q})$  donc  $\mathbf{f}(\mathbf{q})^T \mathbf{v}(\mathbf{q}) = \mathbf{f}(\mathbf{p})^T \mathbf{H}\mathbf{v}(\mathbf{q})$ . Ceci étant vrai pour toute vitesse virtuelle  $\mathbf{v}(\mathbf{q})$ , on peut simplifier par  $\mathbf{v}(\mathbf{q})$ , d'où  $\mathbf{f}(\mathbf{q}) = \mathbf{H}^T \mathbf{f}(\mathbf{p})$ .

## 5 Multirésolution

L'utilisation d'un arbre octal pour représenter une hiérarchie de maillages permet des algorithmes simples de transition entre les résolutions. Il est alors possible d'exploiter au maximum une représentation multirésolution en jouant sur les niveaux de résolution. On peut avoir un modèle adaptatif non seulement pour la mécanique, mais aussi d'autres modèles en parallèle, par exemple une résolution pour l'affichage, une pour la collision... Tous les éléments existant déjà en mémoire, il suffit de créer des arborescences virtuelles en ajoutant des étiquettes correspondant aux types d'arborescences aux éléments. Le gros avantage est de désynchroniser les résolutions des différentes applications en leur affectant des critères d'adaptativité propres, comme illustré dans la figure 3.6. Ainsi, un endroit très déformé peut être animé finement, mais s'il est caché ou loin dans la scène, on peut l'afficher avec une résolution grossière sans tous les détails de déformations. Avoir un rendu plus détaillé que la mécanique peut aussi être utile dans le cas où l'on désire ajouter des détails de déformations procéduraux, tel que du bump mapping ou des jeux de textures dont la résolution dépend de la résolution du maillage.



**FIG. 3.6** – Arbrescences virtuelles

*Un maillage hexaédrique hiérarchique en 2D. De gauche à droite : tous les éléments les plus fins de la hiérarchie, un raffinement suivant un critère d'interaction, un autre raffinement suivant un critère de visualisation.*

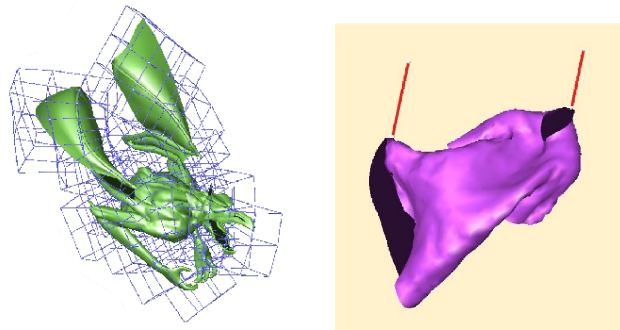
### Adaptativité

Suivant l'application et le type de maillage, on peut imaginer un grand nombre de critères de raffinement automatique. Nous ne nous y sommes pas attardé dans cette thèse ; nous proposons uniquement la structure et les outils permettant d'intégrer n'importe quel critère. Pour nos tests un raffinement automatique de la mécanique utilise le critère géométrique présenté dans [Debunne 01] relatif à la courbure à travers le laplacien des déplacements. Concernant la visualisation, aucun critère automatique n'a été intégré ; les modifications de résolutions ont été faites manuellement.

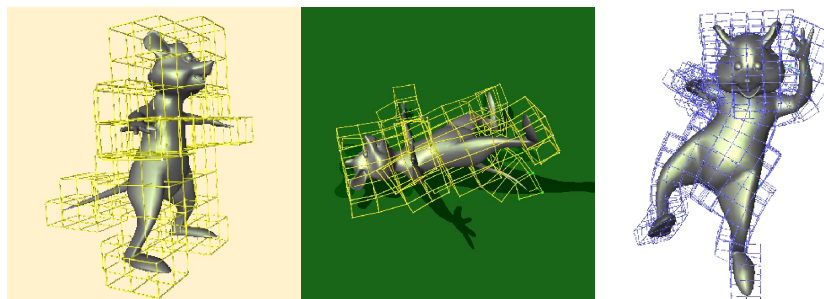
## 6 Résultats

Le maillage d'hexaèdres englobants est très facile à obtenir à une résolution désirée sur laquelle n'importe quelle méthode mécanique peut être appliquée pour animer l'objet. Comme la surface est interpolée dans les éléments, les résolutions des maillages utilisés pour la mécanique et l'affichage sont totalement indépendantes, et il est possible d'avoir un maillage mécanique grossier tout en conservant un objet très finement détaillé pour le rendu. La parfaite régularité des éléments obtenus est importante dans le contexte temps réel afin d'obtenir un système bien conditionné le plus rapidement solvable.

Une contribution intéressante de cette approche est de pouvoir gérer des surfaces non-fermées (comme les ailes du dragon ou le T-shirt de la figure 3.7), et même des éléments linéiques très fins (queue de la souris) de la même façon qu'un volume. Une forme au repos en 3D peut être assignée à ce type d'objets, comme celle qui est la plus naturelle. Par exemple, le T-shirt tend à retrouver la forme d'un vêtement porté. Avec les approches de maillages classiques, animer ces objets est bien plus compliqué voire impossible, le faire en temps réel est un vrai défi.



Les surfaces peuvent être animées comme des volumes



Les parties fines sont bien gérées (oreilles, queue)

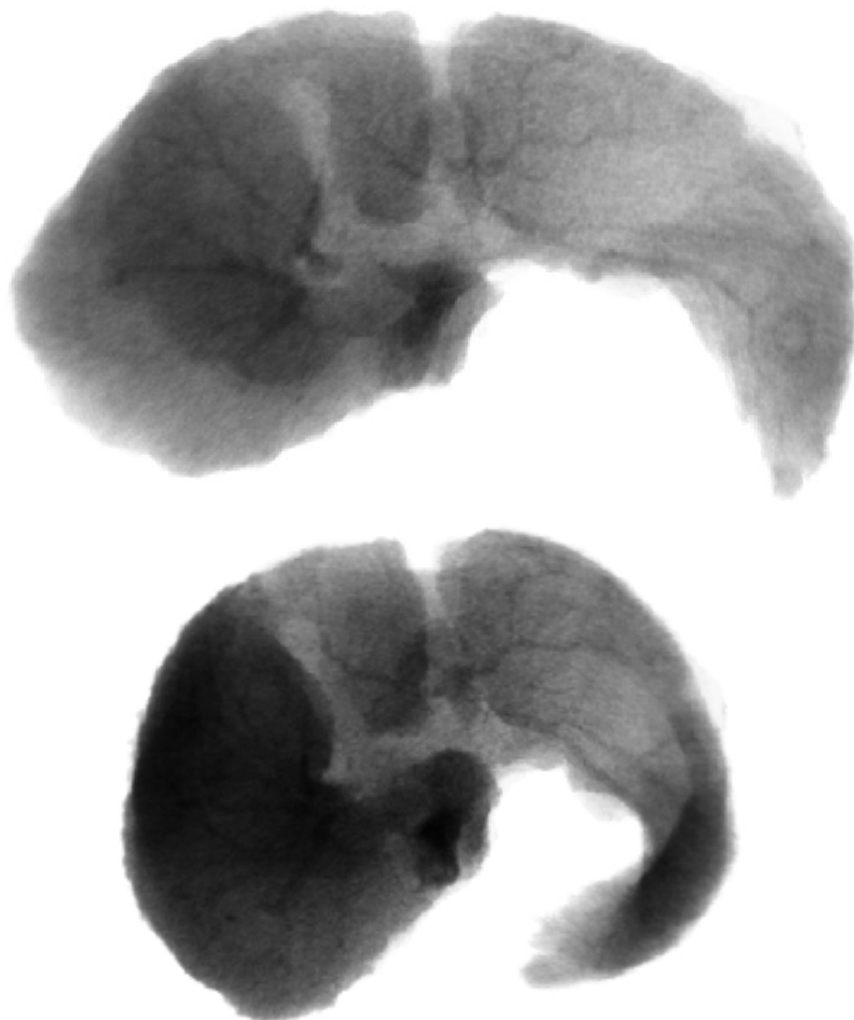
**FIG. 3.7** – Exemples de maillages classiquement difficiles à animer

Puisque des données volumiques de type scanner peuvent être directement animées grâce au principe d'un maillage défini par un arbre octal, il est intéressant de les utiliser également pour le rendu. Il se trouve que la structure hexaédrique utilisée pour la mécanique est très bien adaptée pour faire un rendu volumique efficace. L'implémentation que l'on propose pour ce rendu est décrite dans l'annexe C afin de ne pas surcharger ce chapitre. C'est un autre avantage intéressant de cette représentation, car





l'animation et le rendu peuvent utiliser la même structure de données, et ainsi une grande quantité de mémoire et de calculs peut être sauvée.



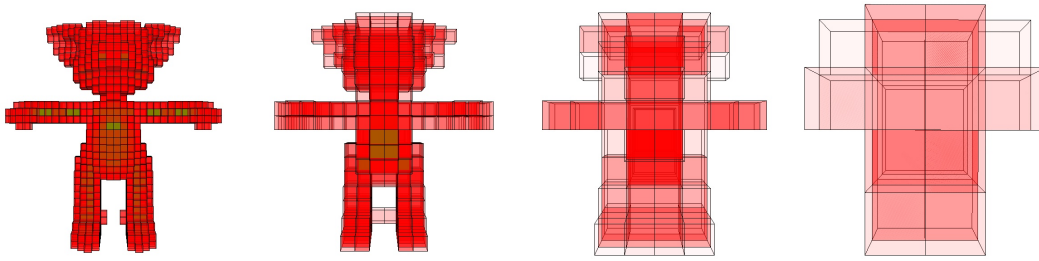
[vidéo 4](#)

**FIG. 3.8** – Rendu volumique d'un foie se déformant

## 7 Limitations

La principale limitation de notre approche réside dans le fait qu'à une résolution trop grossière, on va perdre certaines informations topologiques de l'objet, modifiant son comportement. Par exemple, dans la figure 3.9, une résolution trop grossière ne va plus permettre de dissocier les jambes du personnage.

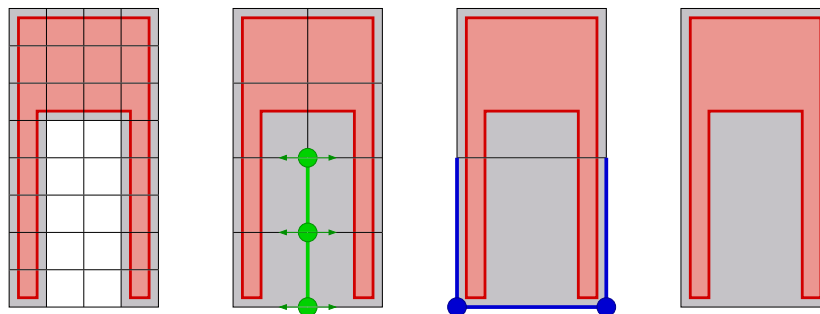
Un travail préliminaire utilisant l'idée des "virtual nodes" [Molino 04] proposés pour la découpe peut permettre de régler le problème des ramifications. L'idée est d'analyser la topologie de l'objet à travers un graphe de voisinage inter-éléments (un lien si deux éléments sont reliés par de la matière au niveau plus fin) et un graphe de voisinage des polygones à l'intérieur d'un élément. Si une telle



**FIG. 3.9** – Différentes résolutions de voxelisation

*A une trop grosse résolution les jambes se retrouvent liées*

solution est pratique pour une animation à résolution fixe, elle complique la multirésolution pour laquelle il faut choisir une politique de hiérarchisation adaptée. Plusieurs solutions de regroupement sont possibles et doivent être évaluées.



**FIG. 3.10** – Utilisation des virtual nodes pour un meilleur comportement des maillages grossiers

*Le maillage fin sépare bien les deux jambes, une simplification va coller les éléments non reliés, séparable en dédoublant les nœuds et les arêtes verts. Quand les deux bras sont dans le même élément, celui-ci doit entièrement être dédoublé et chaque bras appartiendra à son élément. Pour un seul élément, le comportement ne peut plus être traité.*

## 8 Conclusion

Il est habituellement compliqué d'obtenir un maillage mécanique régulier pour un objet quelconque, et particulièrement pour un organe issu de données volumiques médicales. Il est encore plus difficile de choisir sa résolution : les éléments devant coller au mieux à l'objet, leur taille et leur forme sont relativement imposées. Il est donc extrêmement complexe d'obtenir plusieurs maillages réguliers à différentes résolutions pour adopter une approche multirésolution, indispensable à notre sens en animation interactive. De plus le passage d'un niveau à l'autre peut être compliqué si les niveaux ne sont pas imbriqués.

Afin de ne pas se confronter à ces problèmes, nous reprenons l'idée de représenter un objet par sa voxelisation. On va ainsi pleinement profiter des propriétés de cette structure afin d'obtenir des algorithmes simples et efficaces. Elle facilite ainsi la construction de maillages volumiques multirésolutions hiérarchiques avec des arborescences virtuelles permettant de gérer parallèlement plusieurs



résolutions régies par des critères différents. L'animation physique pourra être calculée rapidement et de manière robuste sur ce maillage régulier, où des stratégies multirésolutions pourront y être employées. Dans cette approche, les résolutions mécanique et d'affichage sont découplées pour permettre aussi bien l'animation précise ou rapide d'un objet très détaillé visuellement. De plus ce maillage peut traiter des objets représentés par tout maillage surfacique, même non-manifold, comportant des trous, des lignes, des surfaces, des parties très fines. Ce genre de modèle est en général très difficile (voire impossible) à animer en temps réel en grande partie pour des problèmes de maillages. Et surtout, il est ainsi possible d'obtenir un maillage mécanique directement à partir de données scanner médicales, ce qui est des plus pratique pour fabriquer simplement et rapidement un organe virtuel spécifique à un patient. Pour finir, bien que les nœuds mécaniques d'un tel maillage ne soient pas sur la surface de l'objet, nous avons montré qu'il est possible d'appliquer de façon justifiée des conditions aux limites sur la surface afin de traiter les contacts avec réalisme.

Avec notre approche, l'utilisateur peut se contenter de fournir un objet géométrique à animer ; il obtient alors, sans se soucier des problèmes éventuels liés au maillage tétraédrique, un maillage multirésolution parfaitement régulier, avec des opérateurs simples de passages d'un niveau à l'autre, et des différenciations entre les éléments pleins, vides et sur la frontière de l'objet. La taille des éléments, et donc la résolution mécanique désirée est triviale à obtenir contrairement aux tétraèdres qui se basent très souvent sur la résolution de l'objet donné en entrée.

Cette représentation volumique semble selon nous très adaptée à l'animation temps-réel d'organes spécifiques à un patient. En effet, son aspect régulier lui confère un bon conditionnement et facilitera les résolutions numériques pour plus de rapidité et de stabilité, et sa possibilité d'adopter une approche multirésolution permettra d'économiser en temps de calcul. Sa construction directement depuis des données volumiques médicales facilite l'adaptativité au cas particulier.

Afin de déformer les éléments, on peut imaginer appliquer n'importe quelle méthode d'animation d'objets déformables présentée dans le chapitre 2. Dans le chapitre suivant, nous proposons un calcul des déformations qui profitera des caractéristiques de ce maillage pour fournir une méthode multirésolution rapide, stable et qui respectera au mieux le comportement de l'objet à toutes résolutions dans un souci de précision.

L'utilisation d'éléments non pleins (aux bords de l'objet) n'est pas habituelle en simulation mécanique. Nous allons montrer que leur usage est possible et avantageux, de telle manière que la formulation des éléments finis prenne en compte la répartition de la matière à l'intérieur de ceux-ci.



chapitre **4**

---

# Mécanique des hexaèdres englobants

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>76</b>
<b>2</b>	<b>Hexaèdres corotationnels</b> . . . . .	<b>76</b>
<b>3</b>	<b>Propriétés non-uniformes</b> . . . . .	<b>77</b>
<b>4</b>	<b>Multirésolution</b> . . . . .	<b>80</b>
<b>5</b>	<b>Résultats</b> . . . . .	<b>82</b>
	5.1 Vitesse . . . . .	82
	5.2 Robustesse . . . . .	85
	5.3 Comportement intuitif . . . . .	86
	5.4 Multirésolution . . . . .	88
<b>6</b>	<b>Conclusion</b> . . . . .	<b>89</b>

---

## 1 Introduction

A partir d'un maillage volumique multirésolution issu de voxelisations, nous allons présenter dans ce chapitre comment calculer des forces aux nœuds des hexaèdres en respectant au mieux les lois de la physique et en utilisant au maximum les simplifications offertes par la structure.

Dans notre problématique, on cherche à calculer des forces issues de la mécanique des milieux continus ; on le fera en utilisant la méthode des éléments finis. Pour que ceci soit réalisé rapidement, et avec une grande stabilité, on a adapté une méthode corotationnelle aux hexaèdres (section 2). Lors du chapitre précédent, nous avons identifié les limites de notre proposition de maillage concernant la représentation des bords de l'objet, puisque certains éléments ne sont pas complètement remplis de matière, ce qui est inhabituel pour la méthode des éléments finis. Pour rendre compte de cette spécificité, au niveau du calcul des forces, nous proposons de prendre en considération la répartition de la matière à l'intérieur des éléments de bords en proposant des matrices de raideur et de masse non-uniformes (section 3). Puis nous proposons une approche multirésolution pour exploiter la hiérarchie de maillages à différentes résolutions mise à disposition par les hexaèdres englobants (section 4). Enfin nous évaluons la rapidité, la robustesse et le comportement de ces éléments hexaédriques non-uniformes (section 5).

Pour rappel, nous nous limiterons ici au cas de matériaux purement élastiques linéaires suivant la loi de Hooke (chapitre 2 section 2.5) et aux éléments dont les fonctions d'interpolations sont linéaires. Concrètement, notre objectif consiste à proposer une brique de base robuste, simple et efficace, à partir de laquelle on espère pouvoir construire des modèles aux comportements plus complexes par la suite.

## 2 Hexaèdres corotationnels

Nous avons retenu les principes des méthodes corotationnelles pour leurs avantages notables. En particulier, ces méthodes permettent un calcul des déformations à partir de grands déplacements, tout en conservant un système linéaire. De plus elles ne demandent pas d'assemblage, ce qui est bien adapté aux méthodes dynamiques interactives et aux éventuels changements de topologie.

Jusqu'à présent les méthodes corotationnelles étaient proposées pour des triangles [Etzmuß 03b] et des tétraèdres [Müller 04a] car il est possible de trouver la rotation exacte *i.e.* celle donnant les plus petites déformations possible sur ces éléments. Dans le cas d'un hexaèdre, on ne peut pas trouver une unique rotation donnant les déformations les plus petites. Nous avons cependant proposé [Nesme 05a] de trouver une rotation fournissant de petites déformations, garantissant ainsi de rester dans le cadre applicatif du tenseur de déformations linéaire (eq. 2.2). L'approche géométrique, calant un repère orthogonal sur chaque élément a montré sa rapidité, sa grande stabilité, même face aux situations dégénérées [Nesme 05a]. Pour calculer ce repère local, nous avons décidé de nous baser sur toutes les arêtes de l'hexaèdre plutôt que d'appliquer directement la méthode des tétraèdres décrite en chapitre 2 section 3.2.2. On considère alors les moyennes des arêtes dans les trois directions comme illustré dans la figure 4.1, et sur ce tétraèdre, on peut facilement déduire un repère comme énoncé précédemment. A moindre coût, cela permet d'obtenir une meilleure rotation, garantissant de plus petites déformations en moyenne.

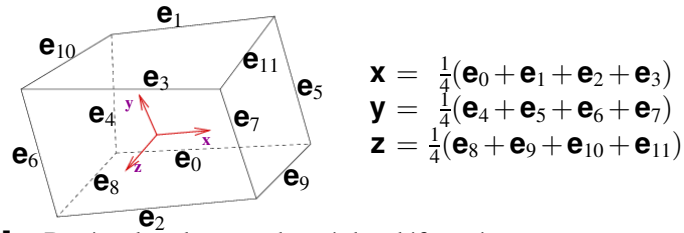


FIG. 4.1 – Repère local sur un hexaèdre déformé.

Pour former les arêtes d'un tétraèdre local, les moyennes des arêtes dans les trois directions sont considérées, avec lesquelles un repère orthogonal peut facilement être déduit comme dans [Nesme 05a].

### 3 Propriétés non-uniformes

Dans cette section, on traite le problème du comportement des éléments de bord, non entièrement pleins. On montre comment prendre en compte la répartition de matière en se basant sur leurs éléments fils plus fins. La différence obtenue par rapport à un moyennage uniforme des propriétés mécaniques est illustré dans la figure 4.2.

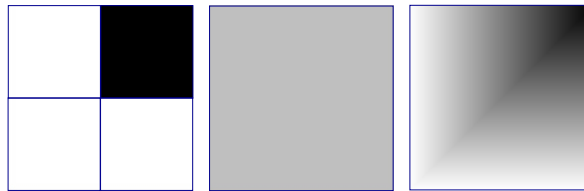


FIG. 4.2 – Schématisation en niveau de gris de la raideur en 2D

de gauche à droite : a) quatre petits éléments, trois vides, un plein. b) leur plus gros élément père regroupant ces éléments en utilisant un moyennage uniforme, l'élément est uniformément très mou. c) le même regroupement en utilisant la non-uniformité des propriétés va faire varier la raideur le long des arêtes.

Si l'on considère les éléments fins comme étant interpolés dans un élément grossier, il est possible de déduire l'influence d'un élément fin  $\mathbf{K}_{fils}$  sur son père  $\mathbf{K}_{père}$ . Il est alors possible de calculer les propriétés mécaniques (raideur et masse) de chaque parent en se basant sur ses enfants (au nombre de huit en trois dimensions).

Lorsque l'on ne considère plus que le gros élément à la place des petits, cela revient à supprimer les degrés de liberté des petits éléments. Une façon de voir est de se dire que les nœuds fils correspondants seront désormais contraints par rapport aux nœuds pères. Dans ce cas, on peut définir huit matrices  $\mathbf{L}_{fils}$  qui représentent l'interpolation des nœuds fils  $\mathbf{u}_{fils}$  en fonction des nœuds parents  $\mathbf{u}_{père}$  :  $\mathbf{u}_{fils} = \mathbf{L}_{fils}\mathbf{u}_{père}$ . Réciproquement, les forces appliquées aux nœuds fils doivent être remontées aux pères en utilisant la transposée de l'interpolation :  $\mathbf{f}_{père} = \mathbf{L}_{fils}^T \mathbf{f}_{fils}$ . En sommant l'influence de tous les enfants d'un élément père, on obtient

$$\mathbf{K}_{père} = \sum_{i=0}^7 \mathbf{L}_i^T \mathbf{K}_i \mathbf{L}_i$$

La même technique est utilisée pour les masses :  $\mathbf{M}_{père} = \sum_{i=0}^7 \mathbf{L}_i^T \mathbf{M}_i \mathbf{L}_i$ , où  $\mathbf{M}$  est la matrice de masse d'un élément. Notons qu'une fois remontée au niveau mécanique désiré, il est possible de "lumper" la matrice de masse pour obtenir une matrice diagonale, afin d'obtenir un système de particules classique avec la masse répartie sur les particules, permettant l'utilisation d'une intégration explicite. Bien sûr

cette étape n'est pas obligatoire afin de conserver un meilleur comportement pour les intégrations implicites qui ne nécessitent pas d'inversion de la matrice de masse.

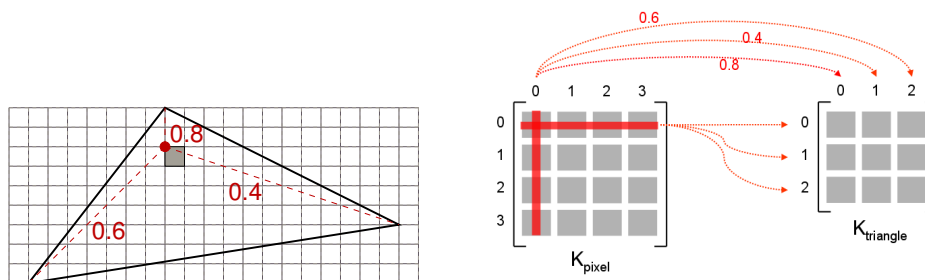
Les matrices  $L_{fils}$  dépendent seulement de la forme des éléments et du schéma de subdivision employé, elles peuvent donc être définies une fois pour toutes. Dans notre cas, les matrices d'interpolation dans une subdivision hexaédrique sont données en annexe D.

Le principe de condensation est imagé de façon plus concrète dans la figure 4.4. Des éléments fins sont représentés dans la colonne de gauche, et une schématisation de leurs matrices ( $M$  ou  $K$ ) dans la colonne de droite. Nous rappelons que ces matrices symétriques représentent l'interaction entre chaque couple de nœud, chaque ligne (et colonne) correspondant à un nœud. Ne considérer que les éléments grossiers revient à supprimer des nœuds donc des entrées dans les matrices. En considérant que les nœuds supprimés sont une interpolation des nœuds conservés, leurs contributions supprimées dans les matrices peuvent être réparties vers les contributions conservées.

Le point fort de cette non-uniformité, c'est qu'elle améliore le comportement mécanique des maillages grossiers sans aucun calcul supplémentaire durant l'animation, car les matrices  $K$  et  $M$  peuvent être précalculées. Par ailleurs, la condensation n'étant pas une étape très coûteuse, on pourrait envisager sa mise à jour dans le cas de matériaux non-linéaires.

**Prise en compte de différentes propriétés de matériaux :** Cette méthode de condensation peut s'avérer très pratique pour prendre en compte d'éventuelles propriétés mécaniques individuelles dont on pourrait disposer. Parmi les différentes méthodes de mesures rhéologiques in-vivo, l'élastométrie progresse rapidement, ce qui fait que l'on pourrait espérer disposer à moyen terme de paramètres mécaniques pour chaque voxels des données d'imagerie médicale. Le principe de ces méthodes d'élastométrie consiste à générer des ondes de cisaillement basse fréquence à l'intérieur de tissus biologiques ou de leur appliquer une contrainte statique, et d'analyser les mouvements de matières engendrés pour en extraire les propriétés élastiques de ces milieux [Ophir 91, Levinson 95, Dutt 96, Catheline 99, Sandrin 02]. Il serait donc très intéressant d'exploiter au mieux ces paramètres. Déjà, dans le cas de données scanner, on pourrait imaginer utiliser les intensités des niveaux de gris comme pondération de la raideur.

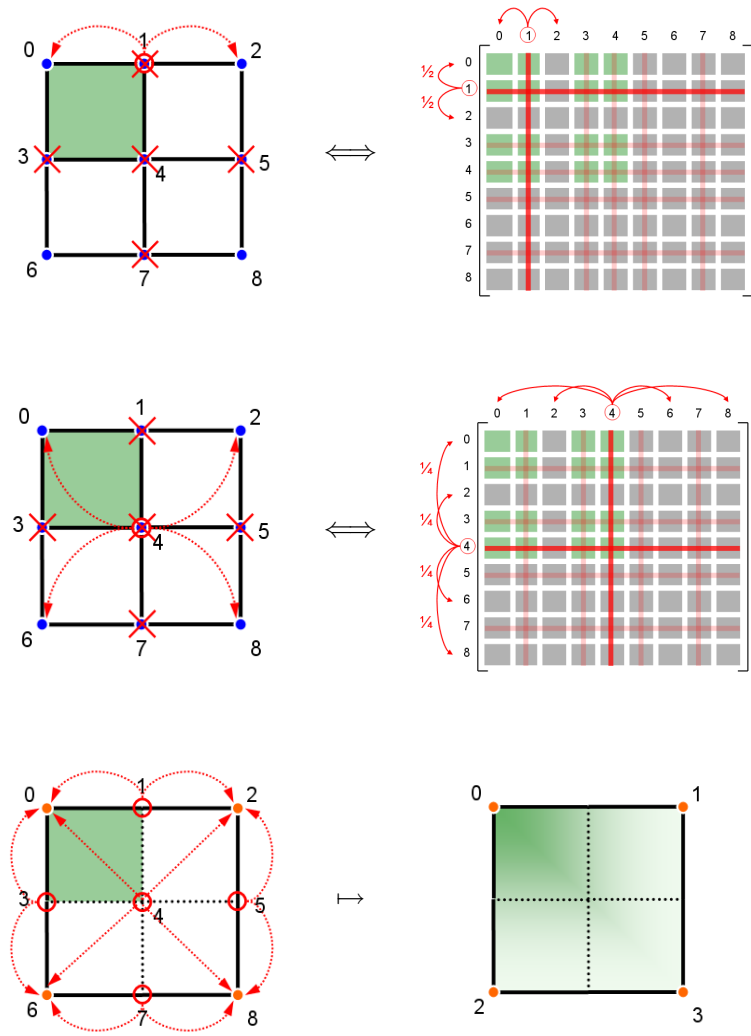
**Extension :** Il est intéressant de noter que la condensation proposée ne fonctionne pas seulement pour une subdivision hexaédrique, mais pour toute interpolation. Même si on dispose d'un maillage tétraédrique, les voxels initiaux peuvent être condensés vers les sommets du tétraèdre les englobant comme présenté dans la figure 4.3.



**FIG. 4.3** – Condensation d'un pixel donné vers un élément fini triangulaire

*Explication en 2D. Pour un certain pixel des données d'entrée, on peut calculer ses matrices de propriétés comme un élément fini rectangulaire en utilisant les paramètres mécaniques fournis. A partir des coordonnées barycentriques des sommets de ce pixel dans le triangle l'englobant, on peut répartir son influence. Attention, à bien prendre en compte les rapports d'aires (volumes en 3D) entre les deux.*





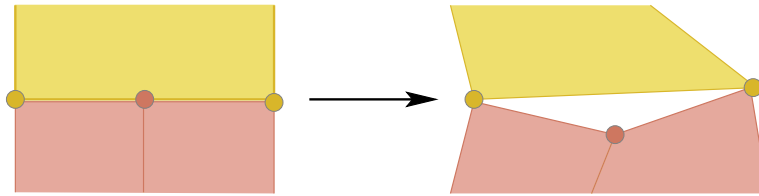
**FIG. 4.4** – Condensation non-uniforme des propriétés mécaniques

*Les influences des ddl supprimés sont remontées vers leurs parents en les pondérant par leurs coefficients d'interpolation. Dans les deux premières lignes, à droite on peut voir une schématisation des matrices, chaque carré étant un bloc  $3 \times 3$ .*

Les propriétés non-uniformes sont un bon moyen de prendre en compte des détails fins à une résolution d'animation grossière.

## 4 Multirésolution

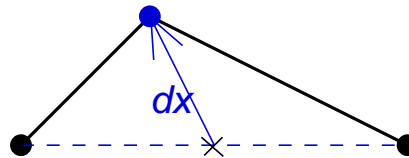
La principale difficulté dans la résolution numérique d'une approche multirésolution est le traitement des nœuds en T, qui se forment à l'interface de plusieurs niveaux de résolution (cf. figure 4.5). Ces nœuds doivent rester contraints sur l'arête (ou la face) sur laquelle ils se trouvent. Une telle contrainte n'est pas forcément évidente à traiter sans ajouts de calculs excessifs.



**FIG. 4.5** – Nœud en T à l'interface de divers niveaux de multirésolution (2 dimensions)

*Si aucune contrainte spécifique n'est gérée, après déformations le maillage devient non-conforme*

Principalement pour cette raison, [Grinspun 02] propose de traiter les éléments finis non pas par élément mais par nœud en considérant leurs fonctions d'interpolations. Ces fonctions sont définies hiérarchiquement, chaque niveau fin n'étant qu'un ajout de détail à une interpolation d'un niveau grossier (illustration figure 4.6). Les nœuds en T sont alors traités très facilement sans aucun surcoût, puisqu'il suffit que leur détail ajouté soit contraint à 0, ils ne peuvent ainsi se déplacer librement de leur interpolation par rapport au niveau grossier, et restent ainsi contraints sur l'arête (ou la face) définie par leurs parents.



**FIG. 4.6** – Approche hiérarchique

*Les nœuds fins (point bleu) sont définis comme étant un ajout de détail (flèche  $dx$ ) par rapport à une interpolation (croix noire) définie à partir des nœuds grossiers (points noirs). Le détail des nœuds en T peut être fixé à 0 avec la même simplicité que de fixer un nœud.*

Nous avons envisagé cette solution pour mieux propager les déformations (cf. chapitre 5 section 2), elle est très bien adaptée pour répondre au problème des nœuds en T, malheureusement elle ajoute une certaine lourdeur dans la représentation et dans la résolution mécanique, ce qui peut la rendre peu souple et difficile à utiliser.

Afin de traiter ce cas plus simplement, [Debunne 01, Wu 01, Dequidt 05] proposent de reprojeter le nœud en T déplacé de manière invalide d'après ses coordonnées barycentriques dans l'élément après un calcul des déformations libres de contrainte spécifique. Cette méthode a le mérite d'être simple et efficace pour toujours donner un maillage conforme. Malheureusement, elle ne respecte pas vraiment les principes de la mécanique et donne un comportement approché en omettant une partie des forces internes.

Pour gérer ce problème, notre approche profite de l'application simple des contraintes dans l'intégration implicite proposée par [Baraff 98] par filtrage des vecteurs dans le gradient conjugué employé. Nous étendons ce principe en proposant un filtre plus complexe permettant de traiter les nœuds en T

en se basant sur le principe des travaux virtuels déjà exploité dans le chapitre 3 section 4.2. L'idée est de ne pas considérer le nœud en T comme un degré de liberté à part entière. Au contraire, toutes les contraintes calculées sur ce nœud devront être remontées à ses parents (par la transposée de son interpolation). Ses variations d'états (déplacement, position, vitesse) seront alors déduites par interpolation de ses parents. Ces deux actions sont illustrées en figure 4.7. L'algorithme du gradient conjugué résultant est présenté ci-dessous (algorithme 1).

Cette idée semble équivalente aux *hard bindings* récemment publiés [Sifakis 07b].

**Adaptativité :** Lors d'un raffinement ou d'une simplification adaptative, les mêmes opérateurs de remontée et de propagation devront être utilisés. Lorsqu'une particule "inutile" disparaît, ses contributions doivent être réparties sur ses parents. Et inversement, quand une particule apparaît pour ajouter du détail, ses états en cours (position, vitesse) sont déduits par interpolation.

---

**Algorithme 1** GRADIENT CONJUGUÉ MULTIRÉSOLUTION  $\mathbf{Ax} = \mathbf{b}$

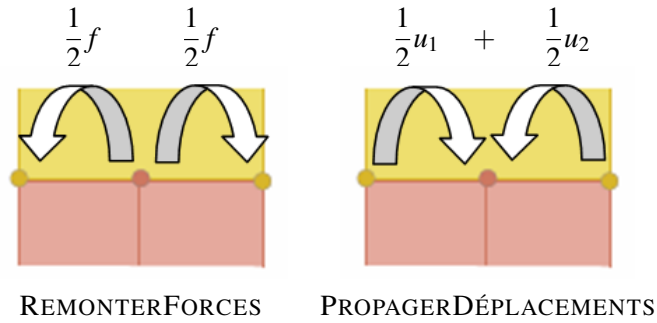
---

```

x = 0           // inconnues initialisées à 0
r = b
REMONTERFORCES( r ) // remonter les contributions des nœuds en T
FILTRER( r )       // filtrage classique [Baraff 98]
i = 0              // itération
loop
   $\rho = \mathbf{r} \cdot \mathbf{r}$ 
  if i == 0 then
    p = r
  else
     $\mathbf{p} *= \rho / \rho_{i-1}$ 
    p += r
  end if
  PROPAGERDÉPLACEMENTS( p ) // interpoler dx aux nœuds en T
  q = Ap
  REMONTERFORCES( q )
  FILTRER( q )
   $\alpha = \rho / (\mathbf{p} \cdot \mathbf{q})$ 
  x +=  $\alpha \mathbf{p}$ 
  r -=  $\alpha \mathbf{q}$ 
   $\rho_{i-1} = \rho$ 
  i += 1
end loop
PROPAGERDÉPLACEMENTS( x )

```

---



**FIG. 4.7** – Illustrations des étapes du filtre gérant les nœuds en T

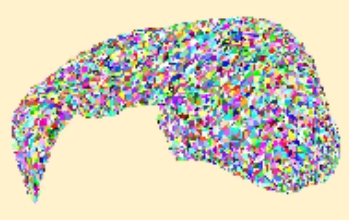
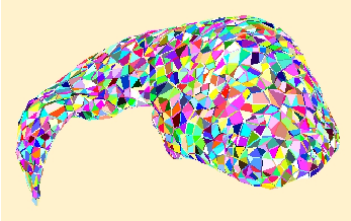
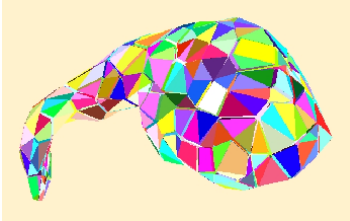
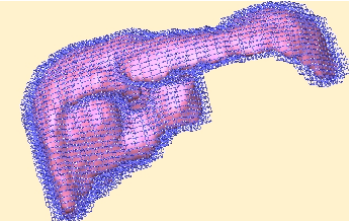
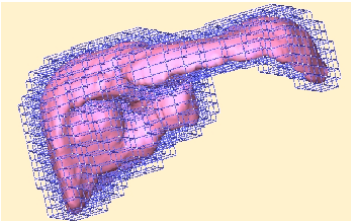
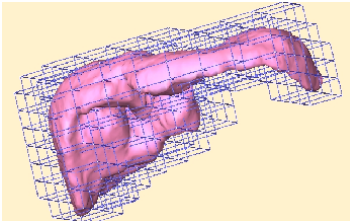
*Toute force calculée sur un nœud en T est remontée vers les degrés de libertés indépendants. La solution peut être déduite sur un nœud en T par interpolation depuis les degrés de libertés indépendants.*

## 5 Résultats

Dans cette section, nous étudions les résultats pour les principes mis en avant par notre calcul des forces, c'est-à-dire la rapidité et la stabilité des hexaèdres corotationnels et le comportement des propriétés non-uniformes pour des éléments non-pleins.

### 5.1 Vitesse

En considérant individuellement un seul élément bien conditionné, le temps de calcul pour évaluer les forces d'un hexaèdre est environ 15 fois plus long que pour un tétraèdre, essentiellement car un grand nombre de simplifications ne sont pas possibles [Nesme 05a]. A première vue, on pourrait donc penser qu'il va être difficile d'obtenir de bons résultats. Pourtant, comme nous allons le voir sur un exemple concret, ce n'est pas le cas. Grâce au maillage d'hexaèdres englobants, tous les éléments sont extrêmement régulier, permettant un système bien conditionné, donc facile à résoudre. Au contraire, un maillage tétraédrique va généralement produire des éléments de différentes tailles; le système moins bien conditionné est plus long à résoudre.

marching cube original 5541 sommets de la surface 11078 triangles	≈ 70% décimation 1455 sommets de surface 2904 triangles	≈ 95% décimation 229 sommets de surface, 452 triangles
7655 nœuds 25204 tétraèdres 1 Hz	2032 nœuds 6725 tétraèdres 5 Hz	309 nœuds 929 tétraèdres 40 Hz
		
11378 nœuds 8913 hexaèdres <1 Hz	2003 nœuds 1346 hexaèdres 13 Hz	436 nœuds 242 hexaèdres 70 Hz
		



vidéo 6

**FIG. 4.8** – Comparaison de temps de calculs entre des maillages tétraédriques classiques (haut) et nos hexaèdres englobants (bas) pour la même méthode corotationnelle (QR)

*La vitesse d'animation est donnée pour plusieurs résolutions mécaniques à peu près équivalentes. Le rendu non réaliste des tétraèdres est présenté ici afin de bien les distinguer.*

On peut voir dans la figure 4.8 des comparaisons entre maillages hexaédriques et maillages tétraédriques. Il ressort nettement que l'aspect régulier du maillage et le bon conditionnement sont des éléments très importants pour l'interactivité : même sur cet exemple réel d'un foie, qui n'a pourtant pas une topologie très complexe, et ne semble pas mettre spécialement les tétraèdres en défaut. Si individuellement, les hexaèdres sont plus lents, ils deviennent dans ce cas concret aussi rapides que les tétraèdres, voire plus. Un cas plus complexe, avec des toutes petites parties (tel que la souris en figure 3.7) mettrait encore plus en défaut les tétraèdres qui seraient encore moins bien conditionnés, malheureusement nous ne pouvons fournir de résultats devant la trop grande difficulté de fabriquer un maillage tétraédrique pour ce modèle... L'utilisation du maillage hexaédrique semble donc pleinement justifiée en terme de vitesse.

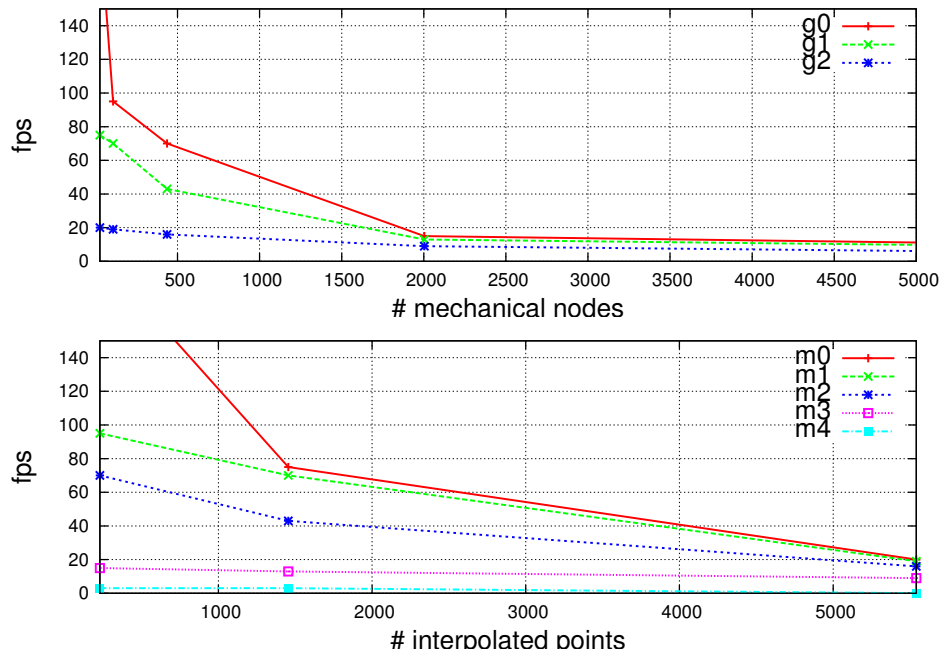
D'autres éléments de performance du maillage hexaédrique sur cet exemple du foie sont présentés dans la figure 4.9. Les résultats sont du même ordre pour d'autres objets d'une complexité équivalente.

Deux ensembles de tests ont été simulés. Dans le premier, seule la résolution mécanique varie. Dans le second c'est la résolution géométrique qui change. Les données de ces résolutions sont répertoriées dans le tableau 4.1. Dans tous les tests, le nombre d'itérations du gradient conjugué est fixé à 10. Bien sûr, les performances dépendent du nombre de degrés de liberté mécaniques ainsi que du nombre de points interpolés (points de la surface). C'est un très grand avantage, car il est possible de garder une animation rapide tout en ayant un affichage d'une grande qualité en réduisant le nombre d'éléments. Notons qu'utiliser un affichage raffiné aura un plus gros impact sur le nombre d'images par seconde pour un maillage mécanique grossier plutôt que fin, car pour ce dernier, ses calculs sont noyés dans une masse beaucoup plus grande.

résolutions mécaniques			résolutions géométriques		
nom	#nœuds	#éléments	nom	#points	#triangles
m0	27	8	g0	229	452
m1	107	46	g1	1455	2904
m2	436	242	g2	5541	11078
m3	2003	1346			
m4	11378	8913			

*Différentes résolutions mécaniques et géométriques*

**TAB. 4.1:** Ensembles de tests pour le foie de la figure 4.8



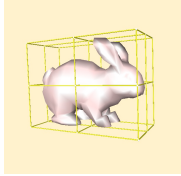
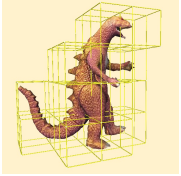
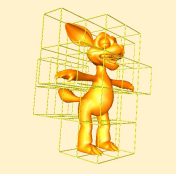
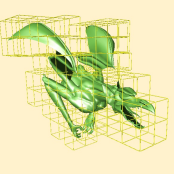
**FIG. 4.9** – Vitesse d'animation

*en nombre d'images par seconde. haut : fonction de la résolution mécanique. bas : fonction de la résolution géométrique.*

Dans le deuxième tableau de la figure 4.9, on peut voir que dans notre implémentation, une bonne partie du temps est employée à mettre à jour des points de la surface, afin d'évaluer les nouvelles

positions de la surface après une déformation des éléments. Il devrait être possible d'accélérer cette étape en effectuant l'interpolation sur le GPU. Des résultats préliminaires, utilisant un vertex shader très simple, calculant les nouvelles positions sans les rapatrier sur le CPU (gestion des collisions impossible) accélère les animations les plus détaillées par un facteur allant de deux à dix. Le rendu final utilise OpenGL avec un rendu de triangles. Pour comparaison, le foie présenté en figure 3.8 avec le rendu volumique est calculé à 10 images par seconde pour un maillage mécanique relativement fin de 291 éléments et 482 particules.

D'autres exemples de résultats en terme de vitesse pour notre méthode sont donnés dans le tableau 4.2. Des objets très détaillés peuvent être animés interactivement. Notons que la phase de précalcul des matrices non-uniformes est relativement longue, de l'ordre de quelques secondes, à cause d'une voxelisation inefficace.

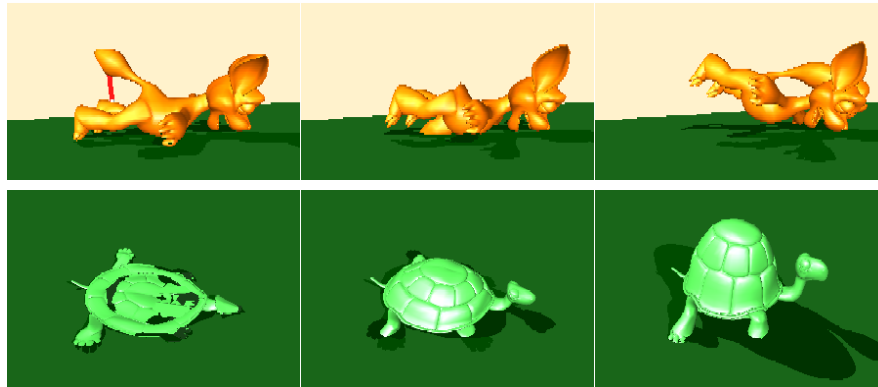
				
modèle	lapin	dino	renard	dragon
<b>nb points</b>	<b>453</b>	<b>2151</b>	<b>8564</b>	<b>21507</b>
nb triangles	901	4299	16777	42391
<b>nb éléments</b>	<b>8</b>	<b>33</b>	<b>30</b>	<b>125</b>
nb particules	27	87	79	292
résolution max	5	5	5	6
résolution animée	1	2	2	3
précalcul (s)	1.5	2	3	7
animation (ms)	1	2	3	7
interpolation (ms)	1	2	7	9
<b>Hz</b>	<b>270</b>	<b>60</b>	<b>18</b>	<b>7</b>

**TAB. 4.2:** Vitesses pour divers modèles détaillés à une résolution mécanique donnée



## 5.2 Robustesse

Grâce à la méthode corotationnelle employée basée sur le repère orthogonal, le modèle reste très stable et ce même face à des configurations extrêmes telles que l'inversion ou l'aplatissement d'éléments (figure 4.10). Ceci car le repère local, construit géométriquement n'est pas sensible aux problèmes numériques introduit dans des situations singulières [Nesme 05a].



**FIG. 4.10** – Robustesse face à des configurations dégénérées non-physiques

*haut : inversions d'éléments, bas : éléments aplatis retrouvant leur forme au repos en augmentant leur raideur initialement nulle*

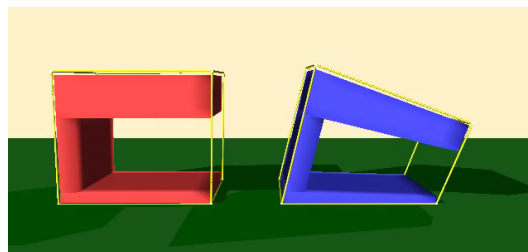
### 5.3 Comportement intuitif

Prendre en compte la répartition de la matière à l'intérieur d'un élément pas entièrement plein permet d'améliorer considérablement la qualité de l'animation sans ajouter aucune complexité. Ceci rend des animations plus plausibles, même en utilisant un nombre réduit d'éléments.

Un cas extrême est présenté en figure 4.11 où un objet très mou en forme de 'C' soumis à la gravité est animé entièrement avec un seul élément. Comme attendu, en utilisant des propriétés (raideur et masse) moyennées, toutes les parties de l'objet ont les mêmes propriétés, et avec une partie vide aussi raide que la partie pleine. Au contraire, en utilisant notre loi non-uniforme, le modèle prend en compte la répartition de la matière, résultant dans un comportement plus réaliste.

Dans l'exemple du lapin (figure 4.12), un peu moins extrême, mais tout de même avec un maillage très grossier, le changement de comportement se fait aussi ressentir. Avec les propriétés non-uniformes, le vide autour des oreilles est pris en compte.

Enfin, notons tout de même que dans un cas d'utilisation où la précision est désirée, le maillage mécanique est plus raffiné, le nombre total d'éléments est beaucoup plus important et les éléments non-pleins sont peu prépondérants.



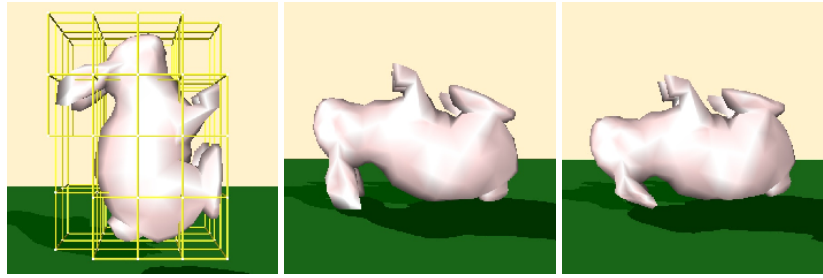
**FIG. 4.11** – Uniforme moyenné contre Non-uniforme

*Deux objets similaires (même maillage, mêmes propriétés mécaniques) soumis à la gravité sont animés par un seul élément. Celui de gauche utilise des propriétés uniformes moyennées, celui de droite utilise les propriétés non-uniformes.*



vidéo 8



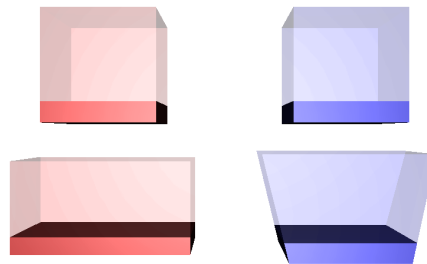


**FIG. 4.12** – Uniforme moyenné contre Non-uniforme

Deux lapins similaires avec le même maillage mécanique englobant (gauche) et les mêmes propriétés sont soumis à la gravité. Le lapin du milieu utilise une répartition uniforme, tandis que celui de droite utilise la répartition non-uniforme. On peut voir la différence de comportement au niveau des oreilles.

Il est aussi possible d'utiliser la condensation non-uniforme des matrices pour définir des propriétés de matériaux très locales, qui seront prise en compte à un niveau mécanique grossier.

Pour illustrer l'apport de cette méthode, nous considérons, dans la figure 4.13, l'exemple d'un cube dont une base est beaucoup plus raide que le reste. Ce cube est déformé par un seul élément hexaédrique. En considérant classiquement un matériau unique à tout l'élément, en négligeant la petite partie plus raide, celui-ci va se comporter de la même manière quelque soit la direction de déformation, tandis qu'avec la condensation non-uniforme, l'élément grossier conserve le fait que le matériau est plus raide d'un côté que de l'autre.



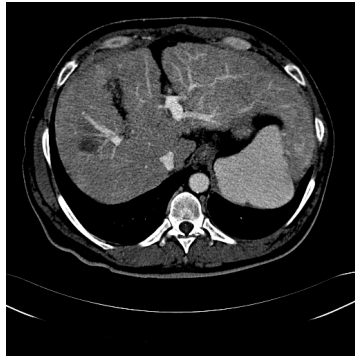
**FIG. 4.13** – Matériau uniforme contre matériau non-uniforme

haut : au repos, bas : en appliquant des forces d'élongation aux nœuds. gauche : élément uniforme, droite : élément non-uniforme. La tranche opaque est beaucoup plus raide que le pourtour transparent.

La figure 4.15 montre une condensation non-uniforme des matrices effectuée depuis des images scanner  $512 \times 512 \times 246$  (figure 4.14) jusqu'à un niveau grossier d'animation. Dans l'exemple du haut, pour chaque voxel de départ, le module d'Young et la densité sont choisis en fonction du niveau de gris (utilisation intuitive de l'unité de Hounsfield). Dans notre implémentation, cette fonction de phase est empirique, plus l'intensité est forte, plus l'élément est raide  $E_{with\_density} = (1 + 10 \times e^{1 - \frac{255}{intensity}}) \times E$ . Le niveau fin étant très fin, le nombre de matrices calculées est très important et le temps de précalcul assez élevé (de l'ordre de plusieurs minutes). En utilisant un matériau unique à tout l'objet (exemple du bas), seuls les éléments au bord de l'objet profitent de la condensation non-uniforme, et tous les éléments internes ont la même raideur.

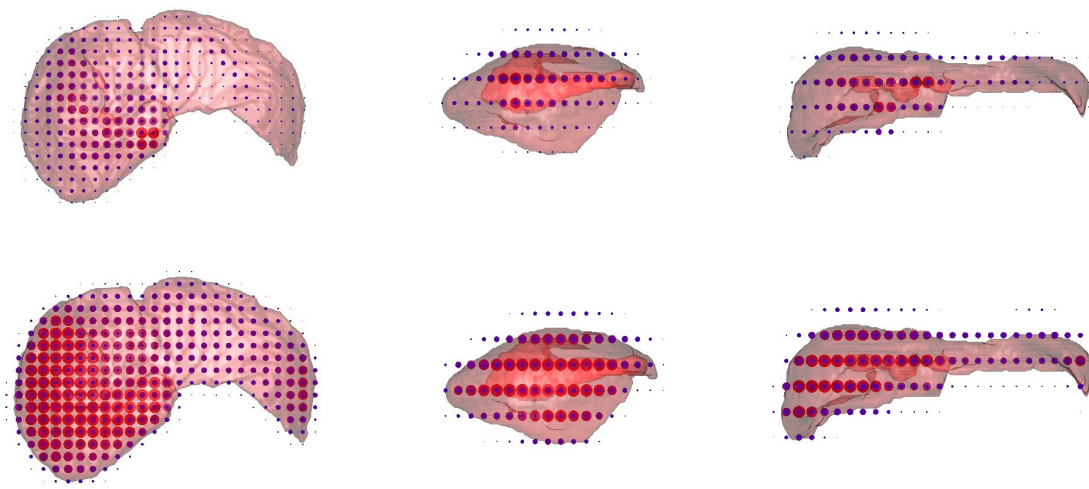


vidéo 9



**FIG. 4.14** – Une coupe axiale des données scanner d’entrée

*Certaines zones du foie (veineuses) apparaissent blanches et sont sûrement plus raides (données de l’IRCAD)*

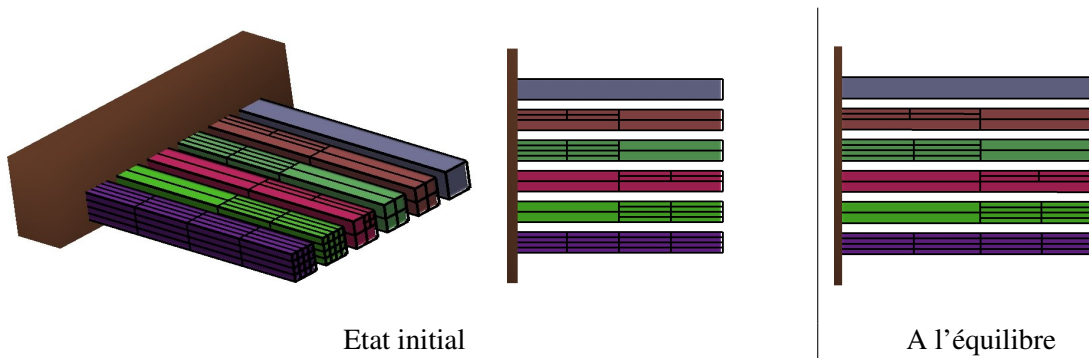


**FIG. 4.15** – haut : les raideurs prennent en compte les niveaux de gris des données scanner d’entrée.  
bas : un matériau unique est utilisé pour tout l’objet.

*Une sphère représente la raideur d’un nœud, mesurée en sommant toutes ses contributions (en valeurs absolues) dans la matrices de rigidité. Plus elle est grande et rouge, plus le nœud est raide. Les échelles ne sont pas les mêmes dans les deux exemples, les valeurs ne sont pas comparables, seulement les rapports des sphères dans une même scène ont du sens.*

## 5.4 Multirésolution

Pour tester en pratique notre gestion des nœuds en T, nous avons évalué le comportement d’une poutre complètement compressible (coefficient de Poisson à 0) soumise à des forces externes l’allongeant, en comparant une approche non multirésolution et une approche multirésolution, tel que les deux modèles doivent donner les mêmes résultats (figure 4.16).



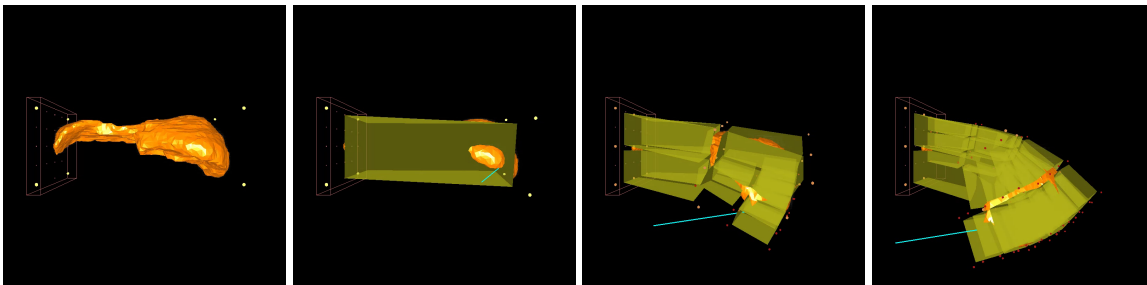
**FIG. 4.16** – Vérification sur la gestion des nœuds en T

*Plusieurs poutres identiques maillées de différentes manières, dont certaines avec des nœuds en T, sont soumises à une traction. Avec des matériaux identiques et un coefficient de Poisson nul, toutes donnent bien la même position finale.*

La gestion des nœuds en T que l'on propose permet bien de garder la cohérence des résultats, et tous les maillages, fin comme grossier et avec des nœuds en T donnent bien le même résultat.

Cet exemple est simple, et sûrement pas suffisant mais il n'est pas évident de trouver des expériences plus complexes permettant de comparer les résultats donnés par différentes résolutions.

La figure 4.17 montre des résultats obtenus avec une stratégie adaptative en adoptant le critère de raffinement automatique décrit dans [Debunne 01].



**FIG. 4.17** – Raffinements automatiques

*Un foie est animé au départ par un seul hexaèdre englobant, en appliquant une force externe il se déforme, les endroits de fortes courbures se raffinent automatiquement.*

## 6 Conclusion

Grâce à l'approche corotationnelle, nous proposons une méthode d'animation d'hexaèdres basée sur les éléments finis et reprenant les critères d'un simulateur interactif : rapidité et robustesse. La prise en compte de la répartition de la matière à travers la condensation des matrices de raideur et de masse des niveaux fins vers les niveaux grossiers permet d'obtenir des maillages offrant un comportement réaliste quelle que soit la résolution. Cette contribution permet de créer une hiérarchie de maillages qui ont le même comportement à la résolution près, les maillages grossiers peuvent plus proprement épauler les niveaux fins lors d'une approche multirésolution. De plus sa réalisation est simple à mettre

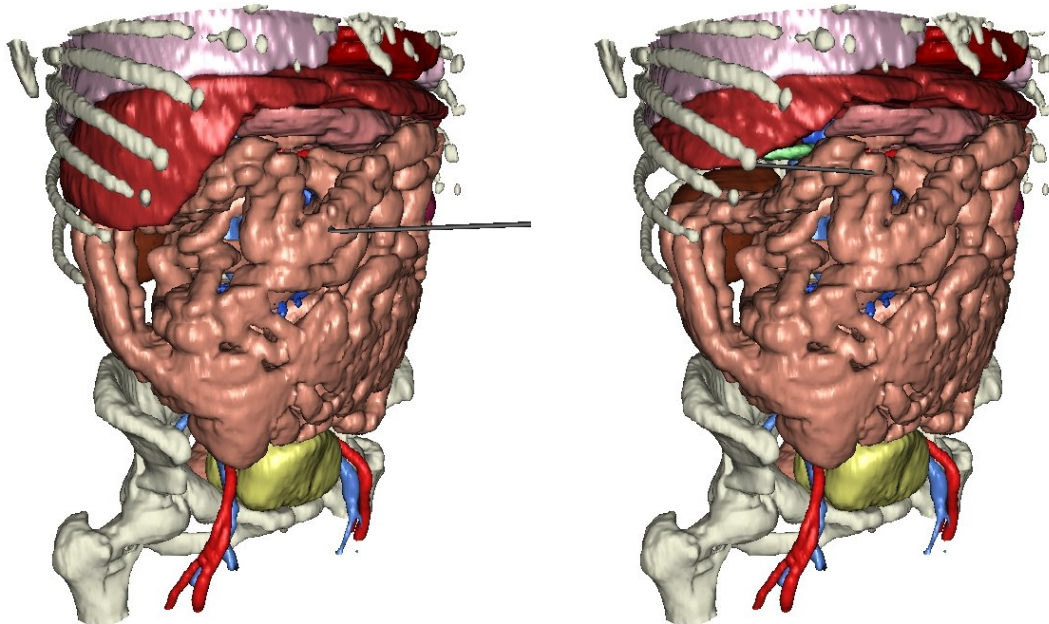


vidéo 10

en place. Tous les calculs étant effectués à l'initialisation, aucune complexité n'est ajoutée pour le calcul des forces durant l'animation.

Concernant la compatibilité avec les changements de topologie, tous les calculs étant locaux, les matrices n'ont pas à être assemblées. On stocke les bouts de matrices élément par élément. Si un élément est ajouté ou supprimé, tout ce qui concerne les autres éléments ne varie pas. Un avantage est que l'on peut si on le souhaite garder la vraie matrice de masse non lumpée, par élément.

A ce stade, on fournit au moins une solution appréciable en synthèse d'animations, car la tâche de l'utilisateur est simplifiée, il n'a plus à créer de maillage volumique, et il peut obtenir un comportement réaliste très facilement pour n'importe quel objet à la résolution voulue. Il a ainsi un réglage vitesse/précision (résolution d'animation) facile à choisir. Cette solution serait aussi très utile en simulation chirurgicale, car le modèle spécifique au patient peut être facilement exploité, même directement à partir des données volumiques (un exemple est donné en figure 4.18). Sa précision est évaluée dans le chapitre 6.



**FIG. 4.18** – Utilisation des maillages voxelisés sur une opération spécifique à un patient. Relèvement du foie (rouge foncé) pour atteindre la vésicule biliaire (verte).

*Construction des maillages mécaniques à partir des données scanner segmentées, affichage de surfaces construites par marching cubes (données scanner fournies par l'IRCAD présentées en figure 3.4). Seuls le foie, la vésicule biliaire, la veine cave (bleue) et l'aorte (rouge) sont déformables. La scène est animée en temps réel à 2 images/seconde, gestion des collisions incluse, sur un ordinateur personnel puissant.*



---

# Propagation dynamique

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>92</b>
<b>2</b>	<b>Éléments finis hiérarchiques</b> . . . . .	<b>94</b>
	2.1 Principe . . . . .	94
	2.2 Répercussions sur la dynamique . . . . .	95
	2.3 Mise en place . . . . .	97
	2.4 Résultats . . . . .	100
	2.5 Conclusion . . . . .	101
<b>3</b>	<b>Multigrid</b> . . . . .	<b>101</b>
	3.1 Principe . . . . .	102
	3.2 Mise en place . . . . .	103
	3.3 Adaptativité . . . . .	104
	3.4 Résultats . . . . .	105
<b>4</b>	<b>Conclusion</b> . . . . .	<b>109</b>

---

## 1 Introduction

Concernant les intégrateurs temporels, le problème soulevé précédemment est l'impossibilité d'avoir une bonne simulation des systèmes mal conditionnés (principalement des matériaux raides) en temps réel. Pour de tels matériaux, les schémas explicites requièrent de très petits pas de temps et donc énormément de petits calculs. Les schémas implicites sont moins sensibles à la taille des pas de temps, par contre le système mal conditionné demandera énormément d'itérations pour donner une solution approchée satisfaisante. Dans tous les cas, la simulation de matériaux mal conditionnés demande beaucoup de calculs, difficilement réalisable en temps réel. Il nous semble donc primordial d'être capable d'accélérer la "propagation" des déformations de long du milieu afin de mieux simuler ces matériaux en temps réel.

**Propagation des déformations :** Ce qu'on appelle "propagation" des déformations pour une méthode dynamique peut se voir de la même manière que la convergence d'une méthode de résolution itérative. Concrètement, quand on exerce une force sur un côté de l'objet, combien de temps faut-il pour que l'autre côté soit influencé ? Cette propagation est schématisée en une dimension dans la figure 5.1. A chaque pas de temps explicite, une particule influence sa voisine, de même qu'à chaque itération d'un schéma implicite. En une dimension, il faut  $O(n)$  pas de temps explicites / itérations implicites pour que la déformation se propage d'un bout à l'autre de la chaîne. Pour donner un exemple, si la chaîne comporte 100 éléments,  $O(100)$  itérations implicites devraient être calculées à chaque pas de temps pour bien retransmettre la déformation.

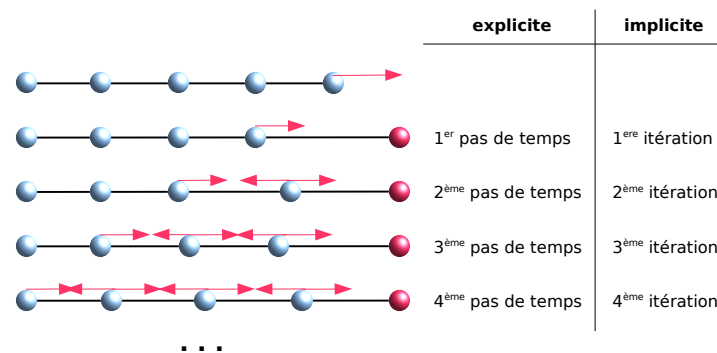
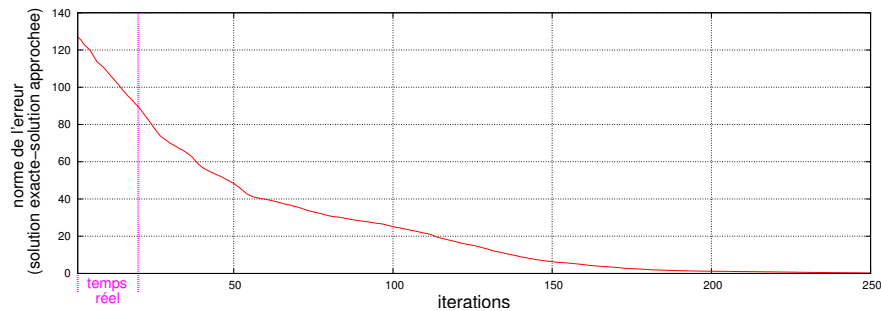


FIG. 5.1 – Propagation des déformations le long d'un objet discrétisé 1D.

Si les pas de temps explicites sont trop grands ou que le nombre d'itérations implicites est trop petit, comme c'est le cas dans un contexte d'interactivité, les déformations ne seront pas retransmises à tout l'objet à chaque instant. Une certaine "mollesse" artificielle est alors introduite, les matériaux réagissent moins vite à une contrainte.

Dans notre travail, on s'intéresse principalement à l'intégration implicite, moins sensible aux pas de temps et plus stable, donc bien adaptée au contexte de l'interactivité. Pour améliorer la propagation de cette intégration, on cherche une méthode qui relaxe énormément le système en début de résolution pour que les premières itérations soient très efficaces. Seules ces premières itérations peuvent être évaluées en temps réel et doivent donner la meilleure solution possible, où les déformations ont

été propagées le plus loin possible. La résolution par gradient conjugué est déjà une grande avancée, car sa convergence est beaucoup plus rapide que les méthodes de Gauss-Seidel ou de Jacobi initialement utilisées. Malgré tout, sa convergence n'avance que degré de liberté par degré de liberté ; pour un grand nombre de particules, le calcul est long. Les pistes classiques envisagées pour accélérer sa convergence consistent à utiliser un bon préconditionneur afin de démarrer la résolution itérative du système à partir d'une solution de départ assez bonne [Bridson 05]. Nous voulons proposer une méthode orthogonale, qui n'empêchera pas l'utilisation de tels préconditionneurs, bien au contraire, mais qui accélère réellement la convergence. Les préconditionneurs sont efficaces dans le cas de la recherche d'une solution exacte, hors du contexte temps réel. Dans le cas d'un court laps de temps imparti, la mise en place du préconditionnement coûte cher et laisse moins de temps à la résolution de s'appliquer, résultant en une solution par forcément meilleure.



**FIG. 5.2** – Convergence du gradient conjugué

*Un pas de temps sur l'exemple d'une poutre encastrée soumise à la gravité, 491 particules, 4096 éléments,  $E = 40000 \text{ Pa}$   $\nu = 0.49$ . Il faut 150 à 200 itérations pour obtenir une solution correcte.*

Pour améliorer la convergence en début de résolution, deux principes ont été envisagés. Les deux approches se basent massivement sur le fait qu'on dispose d'une description multirésolution. On part du principe que si on sait échantillonner les déformations à chaque niveau de résolution, la propagation sera globalement meilleure. En effet, à un niveau grossier, il semble relativement plus aisé et peu coûteux de propager les déformations d'un bout à l'autre de l'objet. La première idée consiste à faire un changement de base du système pour l'amener dans une base hiérarchique (basée sur une représentation en ondelettes chapeau) en utilisant le principe des éléments finis hiérarchiques [Grinspun 02]. La seconde piste est de résoudre le système à l'aide d'une méthode multigrid.

Ces différentes pistes correspondent toutes deux à des approches hiérarchiques, c'est-à-dire qui utilisent plusieurs niveaux de résolution d'un même modèle (cf chapitre 1 section 4.3.2). A notre sens, une des grosses barrières qui freine l'utilisation de telles approches concerne la difficulté de pouvoir disposer d'une représentation hiérarchique et d'avoir des maillages dont le comportement est globalement correct à tous les niveaux. Nous avons vu comment une représentation en arbre octal de voxelisations utilisant des éléments aux propriétés non-uniformes pouvait résoudre efficacement ce pallier. De plus, avec une telle représentation, le schéma de subdivision, *i.e.* de passage d'un niveau de résolution à l'autre, est extrêmement simple (un hexaèdre  $\mapsto$  8 hexaèdres), il est donc aisé de passer sans surcoût important d'un niveau à l'autre. Pour des maillages classiques non emboîtés, chaque niveau doit être interpolé dans son niveau plus grossier, demandant le précalcul et le stockage de coordonnées d'interpolation, comme par exemple dans [Debunne 01, Georgii 06] sur des maillages

tétraédriques.

A notre connaissance, [Wu 04] est le seul travail qui a souligné et essayé d'améliorer la propagation des déformations. Sa proposition reste toutes fois relativement empirique. Il se base sur un cycle en V de multigrad, qu'il réalise à l'envers, où pour chaque niveau il n'utilise pas une relaxation d'un système, mais effectue des intégrations explicites. Leur idée revenant plutôt à trouver une pré-solution au niveau grossier pour avoir un meilleur point de départ au niveau fin et ainsi se rapprocher plus facilement de la solution. L'amélioration de la convergence sans modifier le comportement n'est pas claire.

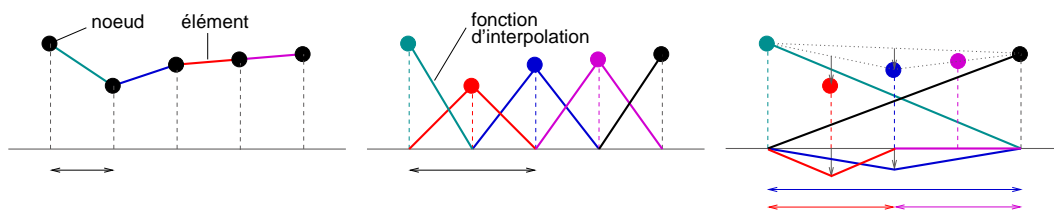
## 2 Éléments finis hiérarchiques

Les éléments finis hiérarchiques ont déjà été introduits en graphisme [Grinspun 02] principalement afin de gérer les approches multirésolutions introduisant des nœuds en T. Aucune considération dynamique n'a été envisagée, alors qu'à notre avis le fait que cette approche traite le problème à différentes résolutions est particulièrement bien adapté à la propagation des déformations.

### 2.1 Principe

L'idée consiste à considérer chaque nœud indépendamment plutôt que de considérer des éléments. Plus précisément, les fonctions d'interpolation en chaque nœud sont considérées. La figure 5.3 donne un exemple en une dimension où chaque couleur appartient à une entité indépendante ; à gauche le problème est représenté classiquement en considérant les éléments ; au milieu, l'approche est dite nodale, car les fonctions d'interpolation en chaque nœud sont considérées.

Avec cette vision il est alors aisé d'adopter une représentation hiérarchique (figure de droite), dans laquelle, le rayon d'influence des fonctions est plus ou moins grand, et où les petites fonctions viennent ajouter du détail aux grandes. Si le temps de calcul ne le permet pas, les détails peuvent être négligés. Au final, le système est inchangé, pour un raffinement maximum le résultat sera le même que l'approche nodale par éléments finis classiques, avec l'avantage de pouvoir facilement gérer des variations de résolutions locales.

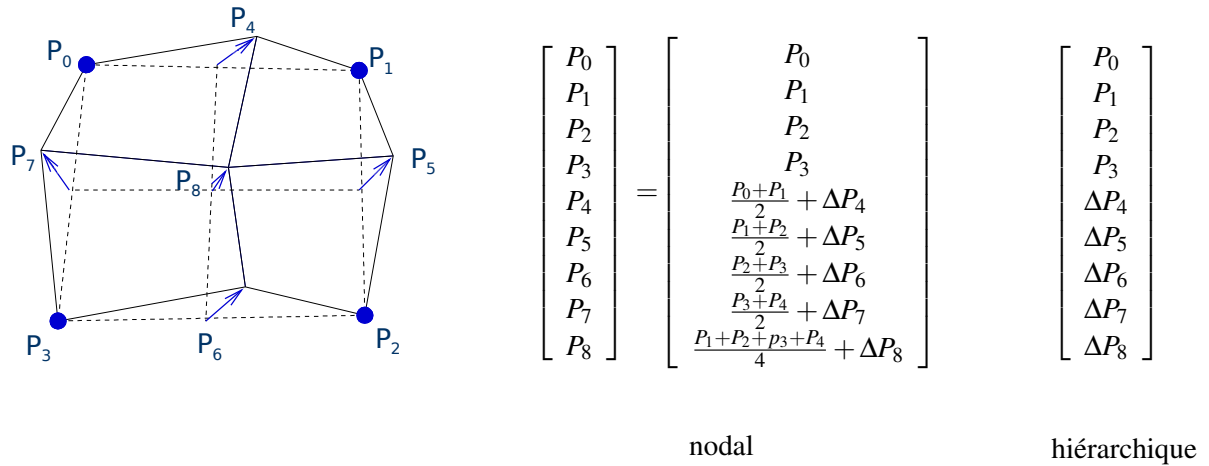


**FIG. 5.3** – Différentes représentations d'éléments finis 1D

*les traits verticaux correspondent aux particules. de gauche à droite : les éléments, leurs fonctions d'interpolation nodales classiques, ces mêmes fonctions sous une forme hiérarchique. Les flèches représentent leurs rayons d'influence.*

Sous cette forme hiérarchisée, les vecteurs d'états du système représentent les ajouts de détails à chaque niveau par rapport à une interpolation du niveau précédent. Concrètement, au niveau le plus grossier, les valeurs nodales sont stockées. Au niveau plus fin, seules les variations par rapport à une interpolation du niveau grossier sont stockées ; et ce pour tous les niveaux. Un vecteur d'état hiérarchique est schématisé en figure 5.4.





**FIG. 5.4** – Vecteurs d'états nodal et hiérarchique

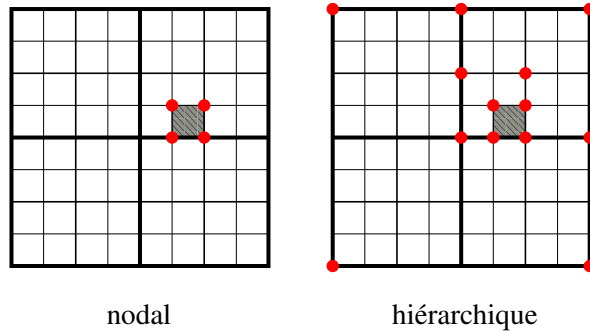
*En hiérarchie, à chaque niveau, n'est stocké que le détail par rapport à l'interpolation du niveau plus grossier*

**Nœud en T :** Comme nous l'avons vu précédemment, un des problèmes causé par la multirésolution concerne l'apparition de nœuds en T, qui ne sont pas des vrais degrés de liberté, mais ont un mouvement contraint par leur voisinage. L'approche hiérarchique traite sans aucun effort supplémentaire ces particules. Il suffit de fixer leur détail à 0, et ainsi elles seront toujours l'interpolation parfaite de leur parents. C'est le gros avantage d'une représentation hiérarchique, car aucune complexité ni calcul ne sont ajoutés pour gérer le niveau de détail, et une approche adaptative peut être trivialement être mise en place.

## 2.2 Répercussions sur la dynamique

Cette représentation ne donne aucune différence par rapport à la représentation nodale sur un plan statique (la solution finale est la même), mais elle donne deux différences majeures qui vont améliorer la dynamique.

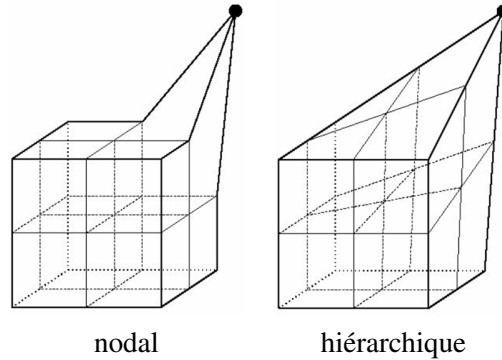
■ **Voisinage mécanique** Premièrement une particule n'est plus seulement connectée à son voisinage proche, mais aussi à toute sa hiérarchie (les matrices de raideur et de masse sont donc moins creuses). Les voisinages nodal et hiérarchique sont illustré dans la figure 5.5. La conséquence est que quand une particule subit une déformation, cette déformation est directement retransmise à ses voisins et à sa hiérarchie.



**FIG. 5.5** – Influence d'un élément

*En hiérarchique, un élément fin n'est pas simplement une connexion de nœuds voisins à un niveau donné, mais aussi une connexion de parenté à travers toute la hiérarchie.*

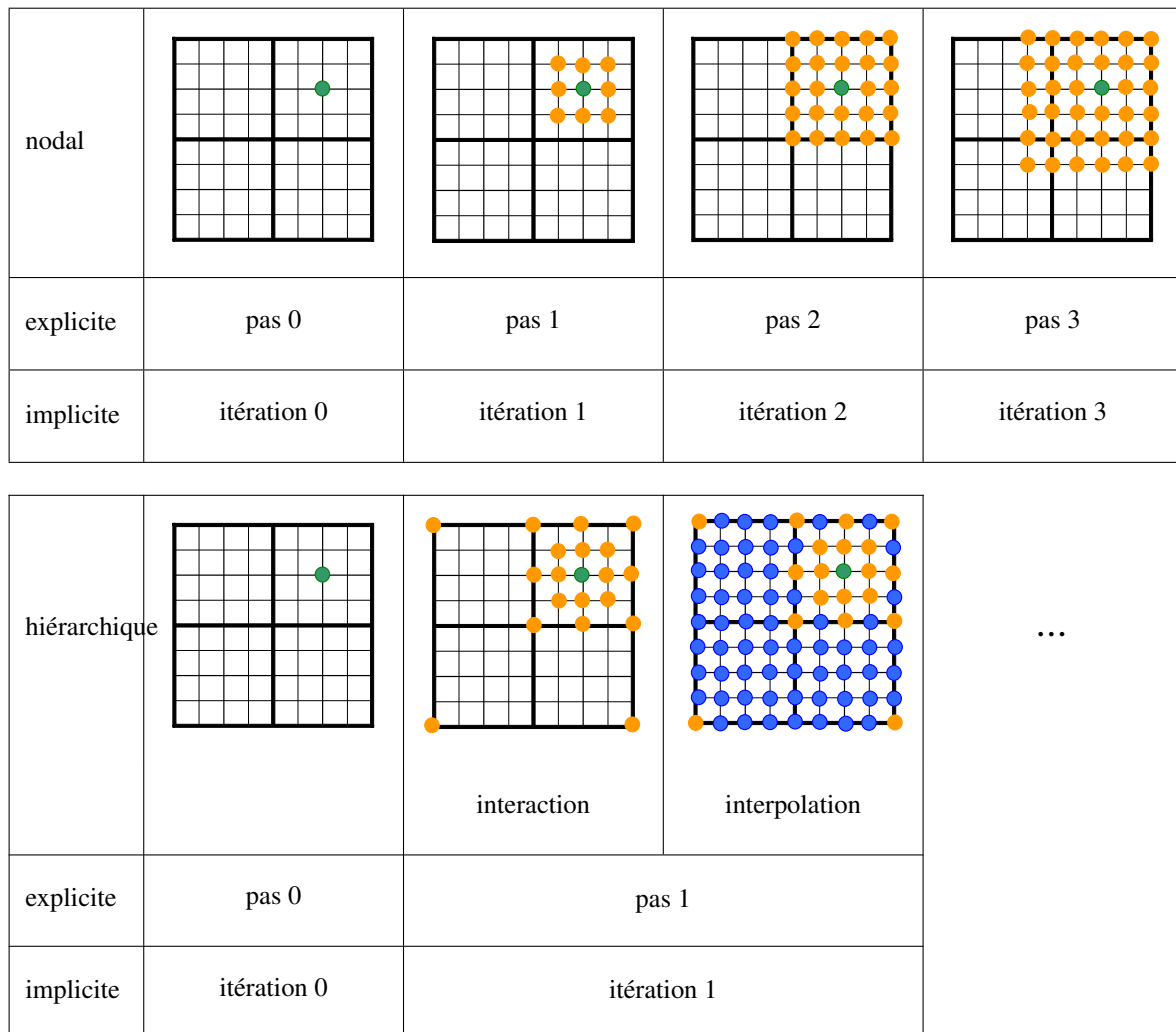
■ **Considérations géométriques** Le représentation hiérarchique des vecteurs d'états fait que même lorsque l'ajout de détails d'un niveau est nul, celui-ci va quand même suivre ses parents, et donc déjà donner géométriquement une meilleure approximation de la déformation. Même si la solution interpolée n'est pas la vraie solution, elle en est plus proche, donc est plus réaliste, et offre un meilleur point de départ pour la suite (pas de temps / itération suivante).



**FIG. 5.6** – Conséquences géométriques d'une représentation hiérarchique

*En nodal, quand un sommet est déplacé, il n'a aucune influence géométrique, tandis qu'en hiérarchique il va influencer toute sa "descendance" qu'il interpole. Instantanément, le hiérarchique va se rapprocher plus près de la solution finale.*

La conséquence de ces nouvelles interactions est représentée en figure 5.7 sur un exemple en deux dimensions.



**FIG. 5.7** – Comparaison de la propagation dynamique dans les cas nodal (haut) et hiérarchique (bas)

*Un stimulus est provoqué sur la particule verte. Dans le cas nodal, à chaque itération, la déformation est retransmise aux seuls voisins directs. Dans le cas hiérarchique, la plus grande connectivité mécanique, transmet la déformation jusqu'au niveau global, puis l'interpolation géométrique fournit une conséquence de cette déformation directement à toutes les particules.*

**Remarque :** Même si avec l'approche hiérarchique beaucoup plus de particules (dans le meilleur des cas, toutes) sont influencées en une itération, une seule itération ne suffit pas à fournir une solution suffisamment approchée. Par contre, son approximation est bien meilleure qu'en nodal, et donc l'itération suivante aura une solution de départ beaucoup plus proche de la solution exacte et pourra converger d'autant plus facilement.

### 2.3 Mise en place

La principale différence induite par cette formulation concerne les déplacements  $\mathbf{u}$  qui ne sont plus définis en coordonnées de l'espace global, mais définie hiérarchiquement. Seuls les déplacements des nœuds des éléments au niveau le plus grossier sont en coordonnées globales. Les déplacements des

autres nœuds sont relatifs à ceux de leurs parents. Pour construire le système correspondant, dans les matrices de masse  $\mathbf{M}$  et de raideur  $\mathbf{K}$ , non seulement les interactions entre les nœuds des éléments les plus fins sont pris en compte (comme dans la formulation classique), mais aussi les interactions de parenté, comme dans [Grinspun 02]. En effet, le long de la hiérarchie, toutes les fonctions d'interpolation agissant sur un élément sont considérées, comme exprimé dans les algorithmes 2 et 3.

---

**Algorithme 2** BUILD MATRICES  $\mathbf{K}$  AND  $\mathbf{M}$ 


---

```

for each cell do
  for each vertex  $i$  defined at level of cell do
    INTEGRATE( $\mathbf{B}_i, \mathbf{B}_i, \mathbf{H}_i, \mathbf{H}_i, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
    for each vertex  $j \neq i$  defined at level of cell do
      INTEGRATE( $\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
    end for
    for each ancestor of cell do
      for each vertex  $j$  defined at level of ancestor do
        take function  $h_j$  between range of cell in ancestor // detail in section 2.3.1
        INTEGRATE( $\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
        INTEGRATE( $\mathbf{B}_j, \mathbf{B}_i, \mathbf{H}_j, \mathbf{H}_i, \mathbf{C}_{cell}, \mathbf{J}_{cell}$ )
      end for
    end for
  end for
end for

```

---

**Algorithme 3** INTEGRATE( $\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}, \mathbf{J}$ )

---

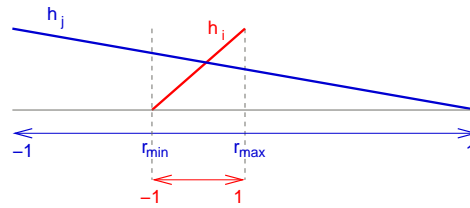
$$\mathbf{K}_{i,j} = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{B}_i^T \mathbf{C} \mathbf{B}_j \det \mathbf{J} \, dr \, ds \, dt$$

$$\mathbf{M}_{i,j} = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \mathbf{H}_i^T \mathbf{H}_j \det \mathbf{J} \, dr \, ds \, dt$$


---

### 2.3.1 Intervalle de définition des fonction d'interpolation

En MEF classique, grâce à l'utilisation des coordonnées naturelles, toutes les intégrations sont effectuées entre  $-1$  et  $1$ , qui correspondent par définition aux bornes d'un élément de référence (non déformé et de taille uniforme). Les matrices ainsi calculées sont ramenées sur l'élément au repos en considérant la jacobienne  $\mathbf{J}$  de la transformation géométrique qui ramène l'élément de référence vers l'élément considéré. Dans le cas hiérarchique, toutes les fonctions prises en compte n'ont pas le même champs d'action, et les fonctions correspondantes à un ancêtre ont un rayon d'influence plus grand que la taille de l'élément fils. Pour un élément fin, seule la partie correspondante à son champ d'action dans sa fonction ancêtre doit être prise en compte, comme illustré en une dimension dans la figure 5.8.



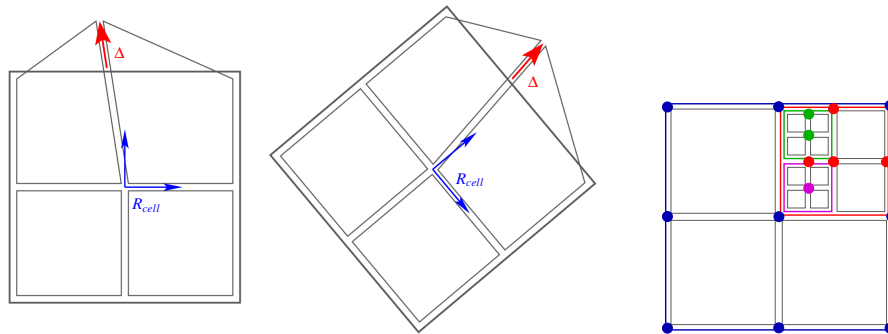
**FIG. 5.8** – Intégration sur des fonctions de rayons d'action différents

Seule une partie de la fonction ancêtre  $h_j$  doit être considérée pour le petit élément rouge situé au niveau hiérarchique  $i$

$B$  et  $H$  sont construites à partir de  $h$  et donc subissent ce changement de base.

### 2.3.2 Grands déplacements

Afin de gérer les grands déplacements, comme énoncé précédemment, nous employons une méthode corotationnelle. A cause de la représentation hiérarchique, chaque détail (c'est-à-dire chaque valeur hiérarchique) doit être indépendant aux rotations. Pour cela les détails sont représentés dans un repère local suivant les rotations comme illustré en figure 5.9. De cette façon, un repère local est attaché à chaque élément à chaque niveau de la hiérarchie et chaque nœud peut être défini dans un de ces repères locaux. Le repère correspondant à un nœud est défini comme celui du plus petit élément englobant tous les éléments incidents à ce nœud, comme illustré figure 5.9-droite. Pour chaque repère local, la méthode corotationnelle sur les hexaèdres est employée.



**FIG. 5.9** – Prise en compte des grands déplacements hiérarchiques

(gauche) Un repère local corotationnel pour chaque élément à tous les niveaux de hiérarchie. Les détails exprimés dans ces repères locaux deviennent invariants en rotation.

(droite) Repères dans lesquels sont exprimés les valeurs des nœuds sur un exemple multirésolution. La couleur du nœud correspond à la couleur de l'élément dans lequel il est exprimé.

## 2.4 Résultats

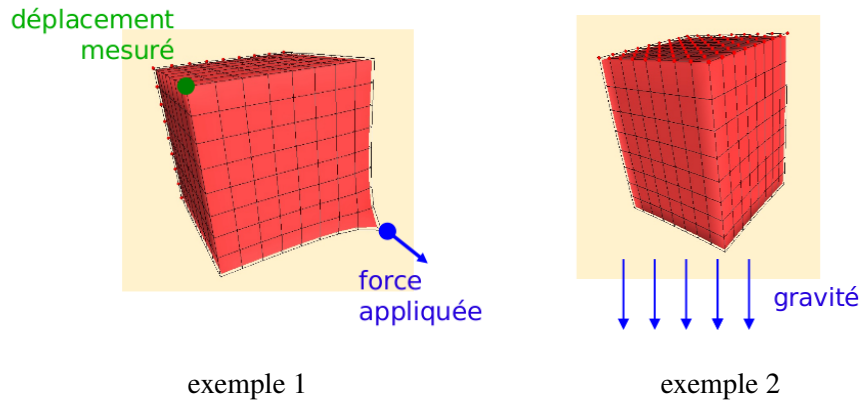


FIG. 5.10 – Les tests

Le tableau 5.1 compare le nombre d'itérations nécessaires pour converger entre la méthode nodale et la méthode hiérarchique. La convergence est atteinte lorsque le résidu considéré est inférieur au millième du résidu initial. Deux expériences sont mises en place, faisant intervenir des objets cubiques discrétisés avec plusieurs résolutions, simulés par un matériau raide ( $E = 20000 \text{ Pa}$ ,  $\nu = 0.4$ ). La première consiste en une poutre encastree pour laquelle une force est appliquée à un coin, la seconde est simplement soumise à la gravité. Comme attendu, la convergence est plus rapide en utilisant l'approche hiérarchique. Plus concrètement, quand un coin du cube est sollicité, le coin opposé bouge instantanément, alors que pour l'approche nodale, un temps de latence est nécessaire pour que la déformation se propage le long des éléments.

	Nombre d'éléments :	8	64	512
exemple 1 (force)	nodal	50	87	198
	hiérarchique	24	37	52
exemple 2 (gravité)	nodal	13	55	146
	hiérarchique	11	27	47

*Pour trois niveaux de raffinement du maillage mécanique*

TAB. 5.1: Nombres d'itérations du gradient conjugué pour converger

La convergence est meilleure en utilisant une base hiérarchique (*cf.* tableau 5.1), et ce d'autant plus que le nombre de d'éléments augmente. Par contre, la matrice étant plus dense et le calcul des interpolations étant nécessaire, chaque produit matrice/vecteur coûte plus cher et une itération hiérarchique est un peu plus longue qu'une itération nodale. Pour prendre en compte ce facteur, dans la figure 5.11, la convergence est donnée en fonction du temps.



vidéo 12

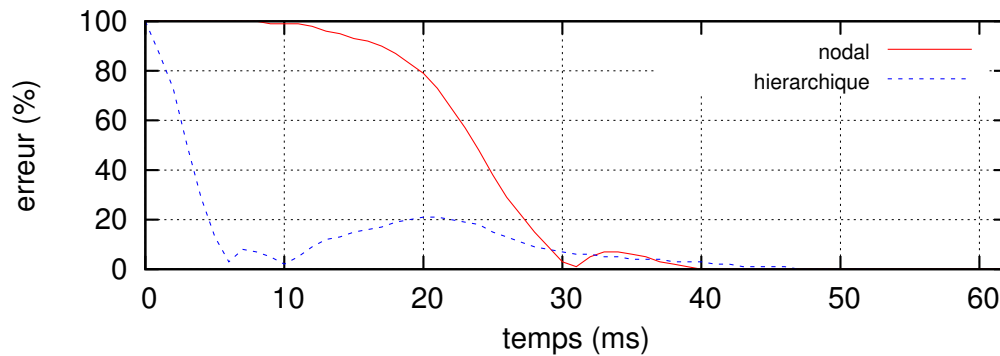


FIG. 5.11 – Vitesse de convergence sur l'exemple 1

*Analyse de l'erreur durant la convergence en fonction du temps de calcul et non pas en fonction des itérations*

Ces résultats sont importants, car dans les deux cas, les temps nécessaires pour obtenir la convergence finale sont identiques. Dans le cas d'une simulation statique précise sans contrainte de temps réel, la représentation hiérarchique n'est donc pas forcément intéressante. Par contre, elle l'est dans le cas de simulations interactives, car les premières itérations se rapprochent beaucoup plus de la solution finale. Dans un cas imparté très court, l'approche hiérarchique fournit une solution approchée bien meilleure.

## 2.5 Conclusion

L'approche hiérarchique semble intéressante car elle permet trivialement une gestion des nœuds en T pour la multirésolution, et il est très facile d'adopter une approche adaptative pour ajouter ou enlever du détail interactivement. Mais ces avantages sont à relativiser, car ceci implique une complexité algorithmique non négligeable. C'est pourquoi nous aimerions étudier une seconde approche, en l'occurrence le multigrid.

## 3 Multigrid

Le multigrid est une approche générique permettant la résolution de systèmes linéaires à une échelle de discrétisation variable [Brandt 77]. Elle a donc une multitude d'utilisations envisageables en informatique graphique où elle commence à faire son apparition [Bolz 03, Shi 06] ainsi que dans le domaine de l'animation physique [Georgii 06, Otaduy 07] où la bonne convergence est utilisée pour traiter interactivement des modèles mécaniques très détaillés. L'impact du multigrid en terme de propagation des déformations n'est pas étudié. De plus, la construction des maillages reste complexe. [Georgii 06] fabrique des maillages encore plus fins en subdivisant le maillage original. Cela permet des simulations plus précises mais limite l'utilisation du multigrid dans le but de mieux converger avec des maillages plus grossiers que le maillage initial. [Otaduy 07] utilise également des maillages grossiers, mais simplement construits individuellement sur quelques exemples. Les difficultés de mailler volumiquement à différentes échelles rendent l'approche multigrid très contraignante. Dans notre méthode nous n'avons pas cette contrainte et le multigrid semble s'y adapter naturellement.

### 3.1 Principe

La méthode multigrid est une méthode hiérarchique qui exploite des représentations à différentes échelles du problème. Cette approche se base sur le fait que les méthodes de relaxation classiques (telle que Gauss-Seidel) réduisent rapidement les hautes fréquences, alors que les basses fréquences vont demander beaucoup d'itérations. Or les basses fréquences d'un problème discrétisé finement correspondent aux hautes fréquences du même problème discrétisé grossièrement. L'idée qui en découle est de résoudre le système à différentes échelles pour lisser toutes les fréquences rapidement et ainsi obtenir une très bonne convergence. Son fonctionnement est très bien détaillé dans [Briggs 00]. Cette méthode peut être très utile en animation physique d'objets déformables afin d'améliorer la convergence de l'intégration implicite qui correspond à un système linéaire.

Le principe de base est de corriger la solution approximée au niveau fin en effectuant des relaxations aux niveaux plus grossiers. Une itération au niveau fin étant coûteuse, leur nombre est limité. Dans le multigrid, la solution approchée au niveau fin peut être améliorée à faible coût par les niveaux grossiers pour lesquels une itération est beaucoup moins coûteuse. Pour cela, l'idée est de transférer le résidu restant du niveau fin vers le niveau grossier (opérateur de restriction), de calculer l'erreur corrigeant ce résidu au niveau grossier, et d'interpoler cette erreur au niveau fin pour s'en servir de correction de la solution approchée initialement calculée. Les opérateurs de transferts sont ceux déjà évoqués pour la multirésolution. Leurs coefficients sont simples à évaluer dans nos grilles emboîtées, et les opérateurs peuvent être implémentés de façon très efficace en ajoutant un coût négligeable à la résolution par rapport aux relaxations.

Ce principe est schématisé étape par étape sur deux niveaux dans la figure 5.12.

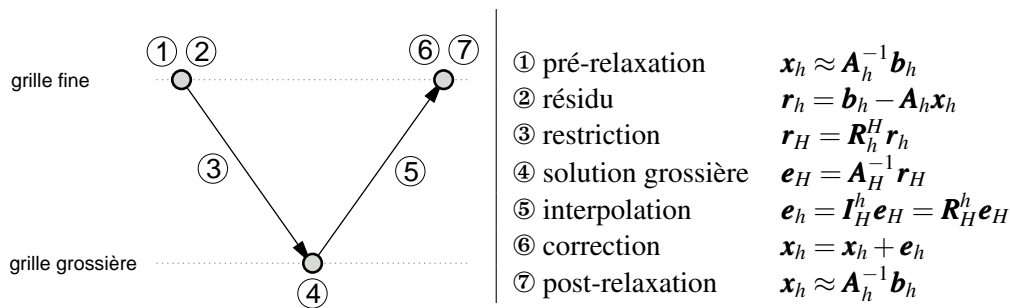


FIG. 5.12 – Cycle de multigrid à 2 niveaux

On essaie de trouver  $\mathbf{x}_h$  tel que  $\mathbf{A}_h \mathbf{x}_h = \mathbf{b}_h$ .  $h$  représente le niveau fin et  $H$  le niveau grossier.  $\mathbf{r}$  est le résidu à un certain niveau,  $\mathbf{e}$  représente l'erreur au niveau grossier et  $\mathbf{A}_H$  est la matrice équivalente à  $\mathbf{A}_h$  au niveau grossier (construite par la condensation des propriétés non-uniforme).

Ce principe peut être utilisé sur plus de deux niveaux comme illustré dans la figure 5.13. Le cycle en V correspond à un passage par chaque niveau, chacun des niveaux grossiers corrigeant son prédécesseur. Dans le cycle complet, une solution est d'abord approchée à chaque niveau (à l'aide d'un cycle multigrid en V), afin de débiter la résolution fine avec une solution déjà bien approchée.



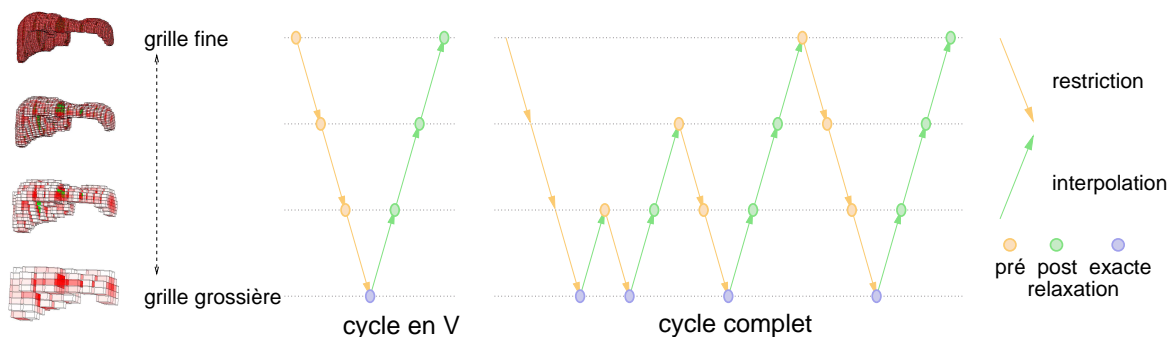


FIG. 5.13 – Les cycles de multigrid

### 3.2 Mise en place

Notre implémentation comporte quelques points originaux qui méritent d'être détaillés.

#### ■ Relaxation

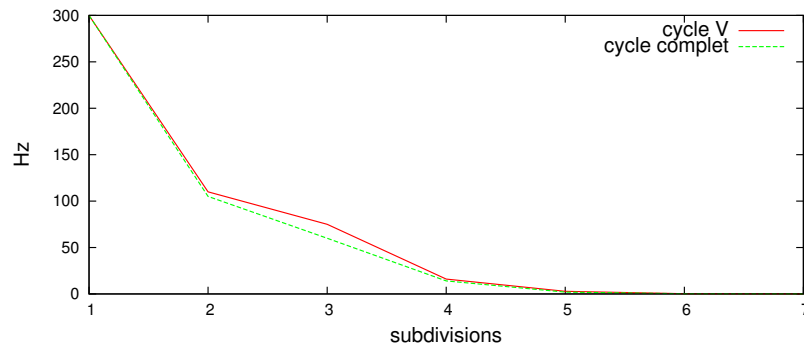
D'habitude, Gauss-Seidel (GS) est la méthode de relaxation préférée dans l'approche multigrid car elle lisse très bien les hautes fréquences. Sans avoir la justification théorique de cette même propriété concernant le Gradient Conjugué (GC), nous avons préféré GC à GS, car il offre des avantages non négligeables décrits dans [Baraff 98]. Tout d'abord, sa convergence est bien meilleure pour des systèmes symétriques définis positifs comme c'est le cas pour l'intégration implicite employée. De plus, sa gestion des contraintes par filtrage est simple et ne rajoute pas de surcoût. Enfin GC va être encore plus intéressant à nos yeux dans une approche multigrid grâce à ses critères d'arrêt facilement implémentables. Alors qu'avec une relaxation de GS, un certain nombre d'itérations fixe est effectué, GC peut, à moindre coût, s'arrêter si le résidu est assez petit ou si numériquement la solution ne peut être améliorée. Ainsi, à la descente les pré-relaxations peuvent être très rapides si le système est facile à résoudre. De même les post-relaxations (qui ne sont pas obligatoires) GC ne seront effectuées que si le résidu est encore trop grand, même après correction. Le GC offre donc un contrôle automatique entre la qualité de la solution et le nombre d'itérations à employer.

#### ■ Opérateurs de transferts

Grâce à la structure emboîtée des grilles utilisées, les opérateurs de transferts d'un étage à l'autre sont triviaux et très rapides. Les matrices correspondant aux discrétisations grossières sont calculées à l'aide des propriétés non-uniformes présentées dans le chapitre 4, section 3. Elles ont le gros avantage d'être une bonne approximation du système fin à tous les niveaux en respectant la propriété de Galerkin [Briggs 00] afin d'assurer une convergence optimale.

#### ■ Cycle

Nous avons comparé une implémentation utilisant un cycle en V face à un cycle complet (figure 5.14). Le cycle en V est celui qui semble offrir les meilleurs résultats pour le temps réel. Le cycle complet converge généralement un tout petit peu plus rapidement, mais quelque fois, sa descente est beaucoup plus lente. Ceci s'explique sûrement par le fait que nos systèmes convergent déjà assez vite dans les premières itérations au niveau fin, et que débiter par une bonne solution initiale prend du temps et n'améliore pas assez la convergence.



**FIG. 5.14** – Comparaison de vitesses entre des résolutions multigrid basées sur un cycle en V et un cycle complet

*Dans le cas d'une poutre encastree soumise à la gravité en flexion,  $E = 40000 \text{ Pa}$ ,  $\nu = 0.3$ , pour diverses tailles de maillages mécaniques données par le nombre de subdivisions.*

La figure 5.14 présente des vitesses de calcul dans un contexte interactif, avec peu d'itérations (7 à chaque niveau, et 150 au niveau grossier). Les cycles en V et complet offrent significativement les mêmes résultats, avec un léger avantage pour le cycle en V. Lorsque le cycle complet semble prendre l'avantage (>6 subdivisions), de très gros maillages mécaniques sont utilisés, très loin de ceux qui peuvent être animés en temps réel.

#### ■ Nombre de niveaux

Avec les grilles emboîtées, il est très facile de faire varier le nombre de résolutions. Au dessus d'un certain nombre de résolutions grossières, il semble que la convergence ne soit plus significativement améliorée. Dans nos études, le choix de trois/quatre niveaux de raffinement semblent un bon compromis simplicité/précision/vitesse (cf. les figures 5.15, 5.16, 5.17).

### 3.3 Adaptativité

On peut dire qu'une approche multigrid non-adaptative est relativement simple à implémenter pour une efficacité certaine. Intégrer une multirésolution adaptative complexifie fortement les structures de données et les calculs. En effet, dans le multigrid classique, une solution est calculée au niveau fin, et une correction est apportée au niveau grossier. Dans le cas d'une multirésolution, certaines parties de l'objet ne sont pas représentées au niveau fin, et le niveau grossier a donc des parties fines de solution et des parties grossières de correction. Dans [Otaduy 07], la méthode utilisée calcule deux solutions en parallèle, une précise issue du maillage grossier et une approchée issue du maillage fin pour en déduire une correction du maillage fin en certains endroits. Mais nous désirerions trouver mieux, en ne faisant qu'une seule relaxation par niveau. Pour cela, nous pensons qu'il est possible d'exploiter le fait que l'on peut intégrer une force à n'importe quel niveau, et ce d'autant plus facilement que les grilles sont imbriquées. Par exemple la gravité peut être traitée simplement au niveau le plus grossier, en quel cas, son ajout dans le résidu grossier correspond à une grosse erreur à corriger ; seule la correction la prend en compte et l'ajoute à la solution approchée au niveau fin. Cette approche est actuellement encore en cours de développement.

### 3.4 Résultats

Pour analyser les résultats de l'approche multigrid, nous comparons ses résultats en terme de vitesse de calcul et d'erreur. Nous prenons pour référence de qualité, la solution du système donnée par un gradient conjugué avec beaucoup d'itérations. Nous prenons pour référence de vitesse un gradient conjugué avec peu d'itérations. A chaque pas de temps, l'intégration implicite est résolue avec toutes les méthodes et leurs résultats sont comparés. Le schéma de comparaison se base sur la norme de la différence entre les solutions et la solution de référence de qualité :

$$erreur = \frac{norme(solution\_reference - solution\_approchee)}{norme(solution\_reference)} \times 100$$

Afin d'éviter un effet de dérive, la solution de référence de qualité est utilisée pour passer au pas de temps suivant. Chaque pas de temps est donc un problème indépendant et la cohérence temporelle n'a aucune incidence.

L'exemple de test, est une poutre encastree soumise à une très grande gravité (1000000G). Cette poutre est subdivisée récursivement 6 fois résultant dans un gros maillage mécanique de 32768 éléments et 35937 nœuds. Le nombre d'itérations pour chacune des méthodes est donné dans le tableau 5.2 (les résolutions des niveaux les plus grossiers devant être "exactes", nous fixons le nombre maximal d'itérations à 150, même si en pratique, pour les résolution très grossières, le GC aura convergé avant). L'utilisation de plus de 4 niveaux grossiers pour le multigrid n'est pas représentée pour ne pas surcharger, car à partir de 4 niveaux, les résultats ne s'améliorent plus. Les résultats nous permettent d'affecter une note à chaque méthode, en prenant le ratio entre la précision et le temps de calcul, suivant la formule :  $note = moyenne \left( \frac{100}{erreur \times temps\ de\ calcul} \right)$  pour les résultats des tests suivants.

niveau	CGmax	CGfast	MG1	MG2	MG3	MG4
<b>note</b>	-	0.5	1.5	5	10	176
<b>0</b> (fin)	150	10	7	7	7	7
<b>1</b>	-	-	150	7	7	7
<b>2</b>	-	-	-	150	7	7
<b>3</b>	-	-	-	-	150	7
<b>4</b> (grossier)	-	-	-	-	-	150

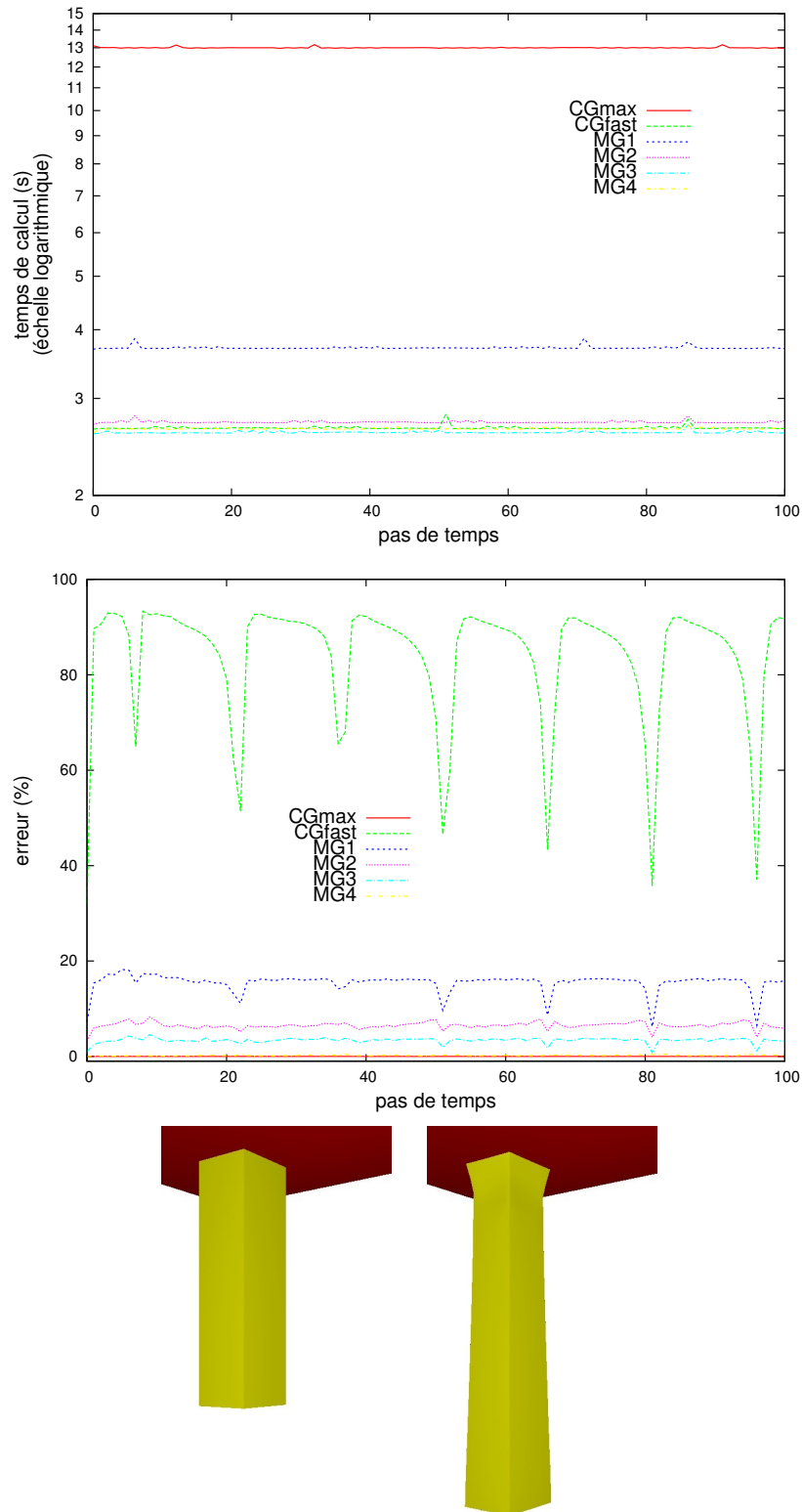
*CGmax=référence en précision, CGfast=référence en vitesse, MG=multigrid en utilisant différents nombres de niveaux grossiers*

**TAB. 5.2:** Nombres d'itérations employés à chaque niveaux pour chacune des méthodes

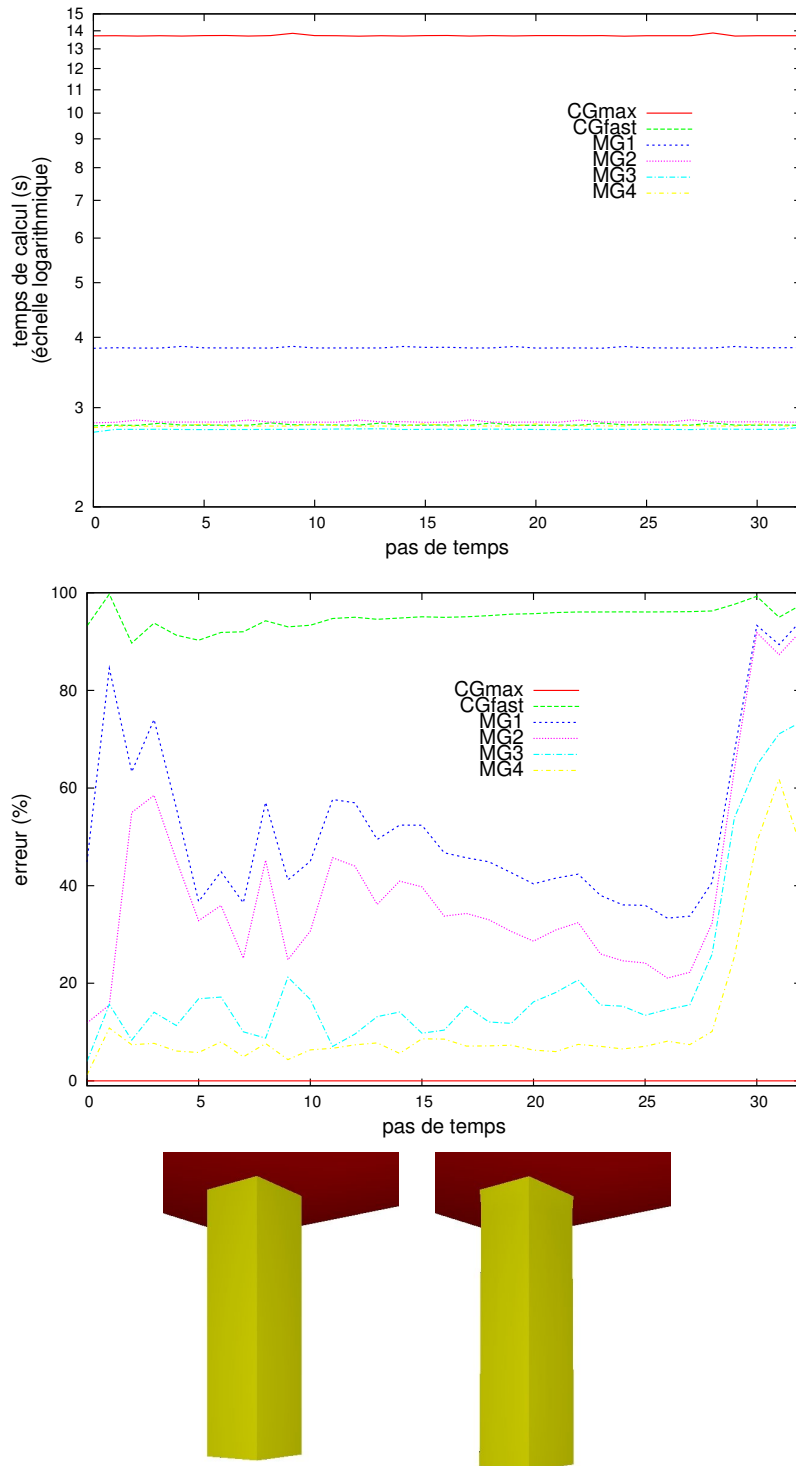
Dans un premier temps, la gravité est appliquée dans le sens de la longueur. Le matériau est fixé à un module d'Young à 40000 Pa et un coefficient de Poisson à 0.3, pour être ni trop raide, ni trop incompressible, afin de ne pas obtenir un système particulièrement mal conditionné. Les résultats sont donnés en figure 5.15.

Dans un second temps, nous avons raidi le matériau ( $E = 400000$  Pa,  $\nu = 0.45$ ) afin de détériorer le conditionnement pour la même expérience. Les résultats sont donnés en figure 5.16.

Dans un troisième temps, nous allons tester un cas où les niveaux grossiers semblent être intuitivement d'une moins grande aide, toujours dans le cas d'une poutre encastree soumise à la gravité, mais provoquant une flexion. Chaque niveau se comporte indépendamment différemment. Les résultats sont donnés en figure 5.17.

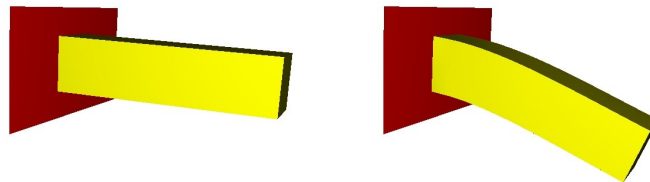
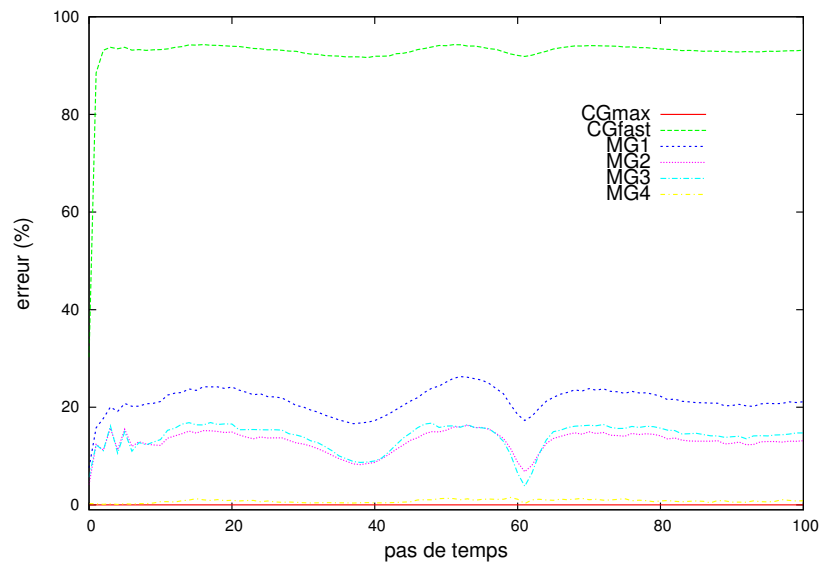
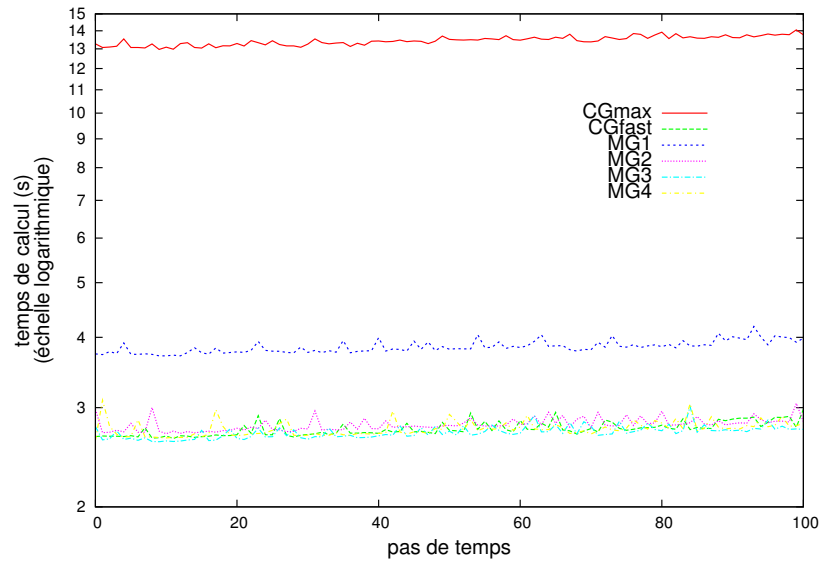


**FIG. 5.15** – Comparaison de la qualité (vitesse et erreur) des différentes approches pour le matériau "mou" dans le cas d'une poutre allongée sous la gravité



**FIG. 5.16** – Comparaison de la qualité (vitesse et erreur) des différentes approches pour le matériau "raide" dans le cas d'une poutre allongée sous la gravité

*La méthode de référence CGmax converge autour de la 27<sup>ième</sup> itération, fournissant alors une solution quasiment nulle, expliquant l'envolée des erreurs relatives.*



**FIG. 5.17** – Comparaison de la qualité (vitesse et erreur) des différentes approches pour la poutre en flexion



L'approche multigrid améliore clairement la qualité des résultats que ce soit en terme de précision de la solution ou en terme de vitesse de calcul. Utiliser simplement un seul niveau de subdivision divise par trois les temps de calculs pour une solution sans commune mesure par rapport à une solution approchée rapide. Trois/quatre niveaux de subdivision donnent des résultats impressionnants : la même vitesse que le gradient conjugué très approché pour une erreur très petite. D'après les notes du tableau 5.2, exprimant le rapport erreur/temps de calcul, le multigrid avec 4 niveaux raffinés est plus de 300 fois plus rapide pour la même précision (ou 300 fois plus précis pour la même vitesse) que le gradient conjugué rapide, utilisé classiquement pour les simulations interactives.

## 4 Conclusion

Les approches hiérarchiques semblent très prometteuses pour mieux retranscrire la raideur des matériaux interactivement, et ainsi pouvoir simuler plausiblement des objets en temps réel.

D'un côté les éléments finis hiérarchiques sont assez complexes à mettre en place, mais ils permettent d'améliorer sensiblement la convergence tout en assurant une multirésolution adaptative très facilement. De l'autre côté, l'approche multigrid donne un excellent compromis vitesse/précision, mais son extension à la multirésolution est plus complexe. Nous sommes persuadés que l'utilisation d'un multigrid adaptatif est une bonne direction à prendre pour animer les hiérarchies de voxelisations.





chapitre **6**

---

Validations du modèle

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>112</b>
<b>2</b>	<b>Cas des hexaèdres pleins</b> . . . . .	<b>113</b>
2.1	Poutre encastrée . . . . .	113
2.2	Truthcube . . . . .	115
<b>3</b>	<b>Cas des hexaèdres non-pleins</b> . . . . .	<b>118</b>
3.1	Poutre encastrée . . . . .	118
3.2	Cylindre . . . . .	119
3.3	Foie . . . . .	121
<b>4</b>	<b>Conclusion</b> . . . . .	<b>123</b>

---

## 1 Introduction

Avant d'utiliser une méthode numérique pour prédire un comportement, il nous semble nécessaire de vérifier la validité de cette méthode. Cette démarche est relativement nouvelle pour des objets très déformables et de formes complexes. En effet, les méthodes de simulations précises étaient jusqu'à présent principalement utilisées en mécanique pour des pièces de type métal ou béton, et les simulations interactives de l'informatique graphique n'avaient qu'un critère qualitatif sur la plausibilité du comportement et la beauté visuelle. Il nous apparaît donc important de mettre au point une phase conséquente de tests pour valider un modèle déformable.

Un modèle dynamique doit être évalué sur deux aspects : l'état d'équilibre sous contraintes (aspect statique) et le cheminement pour y arriver (aspect dynamique).

Dans un premier élan, afin de tester l'aspect statique, nous proposons de comparer nos résultats numériques à des résultats théoriques sur des problèmes simples que l'on peut résoudre analytiquement. Ensuite, le plus intéressant à notre sens est de comparer nos résultats numériques avec des résultats fournis par d'autres méthodes numériques certifiées, utilisées en ingénierie. Comme référence, nous allons utiliser le logiciel [Ansys], qui donne statiquement une solution précise d'équilibre respectant des contraintes. Au mieux, nous ne pourrions qu'approcher ces méthodes pour trouver une solution à l'équilibre, car elles suivent les mêmes lois mécaniques des milieux continus et emploient des méthodes à une précision numérique très poussée. Le test ultime est de se comparer avec des données réelles, issues d'expériences appropriées voire carrément d'opérations chirurgicales réelles, avec idéalement des mesures pré-, per- et post-opératoires. Dans un premier temps, il nous semble plus judicieux de se comparer avec des solutions numériques certifiées avec lesquelles il est possible de reproduire exactement la même simulation, avec les mêmes conditions aux limites... En effet, dans le cas de données réelles, énormément de facteurs entrent en jeu (imprécisions des surfaces, contacts approximatifs, matériaux mal connus...) pouvant fausser les simulations, sans permettre de valider ou d'infirmer un modèle. L'étape de confrontation avec des données réelles, semble plus adaptée à un simulateur final complet, mais moins à un modèle mécanique.

Concernant l'aspect dynamique, même si nous savons que la qualité de l'animation dans le temps dépend énormément du schéma d'intégration employé, nous n'avons pas cherché à évaluer la pertinence de ceux-ci. De plus, il n'existe pas de moyen simple à mettre en œuvre pour leur évaluation. Peut-être une idée pourrait être de prendre comme référence un intégrateur explicite réputé en simulation de crash test automobile avec de très petits pas de temps, car ils sont reconnus pour donner des résultats précis. Il est important de souligner que les méthodes permettant d'accélérer la propagation des déformations (présentées en chapitre 5) ne modifient pas le système, et n'entraînent donc aucune différence dans la solution à l'état d'équilibre statique. C'est pourquoi aucun test spécifique n'est effectué pour ces méthodes. Par contre elles modifient bien sûr l'aspect dynamique pour lequel nous savons qu'elles donnent des résultats plus plausibles en temps réel, même si nous ne sommes pas en mesure de quantifier cette plausibilité. Au mieux elles ne peuvent que mieux retranscrire l'intégrateur implicite qu'elles accélèrent pour fournir une meilleure solution approchée dans un court laps de temps, mais en aucun cas elle peuvent en donner une meilleure solution dans l'absolu. C'est donc le schéma d'intégration qu'il est important d'évaluer.

L'aspect statique de notre méthode va être mis à l'épreuve en deux temps. Dans un premier temps, nous allons tester uniquement la précision des éléments finis utilisés. Pour cela, nous allons les employer dans des maillages mécaniques classiques, qui représentent parfaitement l'objet et où tous les

éléments sont pleins de matière. Puis dans un deuxième temps, nous allons regarder l'effet des propriétés non uniformes sur des éléments non-pleins. L'idée va être de faire varier le positionnement du maillage mécanique autour de l'objet et de comparer ces résultats entre eux et avec des résultats issus de maillages classiques.

Pour les simulations dynamiques, les mesures sont faites après un certain temps de relaxation afin d'atteindre la convergence vers l'état d'équilibre.

Il est relativement difficile de comparer les temps de calcul, étant donné que Ansys utilise un solveur statique alors que les autres méthodes sont dynamiques, pour lesquelles la solution à l'équilibre est obtenue après avoir attendu une stabilisation.

Le critère de comparaison utilisé est basé sur la distance euclidienne entre les positions des nœuds :  $\|position\_de\_reference - position\_testee\|$  (mm). Ce critère en relatif par rapport à la taille de l'objet simulé donne plus de sens :

$$\left( \frac{\|position\_de\_reference - position\_testee\|}{taille\_objet} \times 100 \right) \%$$

Il est alors possible de déduire une erreur moyenne pour l'ensemble des nœuds, ainsi qu'une erreur minimale, maximale et l'écart type.

Les résultats en terme de performance proviennent d'un ordinateur portable muni d'un Pentium M 2 GHz avec 2 Go de RAM et une carte vidéo nVidia Quadro FX Go1400 (donc d'une puissance très raisonnable).

## 2 Cas des hexaèdres pleins

Dans cette partie on ne va tester que la précision pure du comportement des éléments hexaédriques corotationnels utilisés. Pour cela, on ne va considérer que des maillages mécaniques classiques qui correspondent exactement à l'objet simulé.

### 2.1 Poutre encastrée

La flèche (déplacement vertical) d'une poutre encastrée soumise à la gravité est un test intéressant, car il en existe une solution théorique approchée.

Dans cet exemple, nous allons évaluer la qualité des résultats du calcul corotationnel des déformations. On le teste à la fois des éléments tétraédriques et hexaédriques, pour deux méthodes corotationnelles, la polaire censée être très précise, et la QR plus rapide. Les résultats numériques sont comparés aux résultats théoriques.

La poutre étudiée est représenté sur le schéma 6.1 :  $l$  est sa longueur,  $H$  sa hauteur,  $T$  son épaisseur,  $\rho$  sa masse volumique et  $g$  la gravité.

Le problème théorique reste unidimensionnel, dans nos simulations seuls les points se trouvant donc sur la ligne moyenne seront utilisés pour la comparaison.

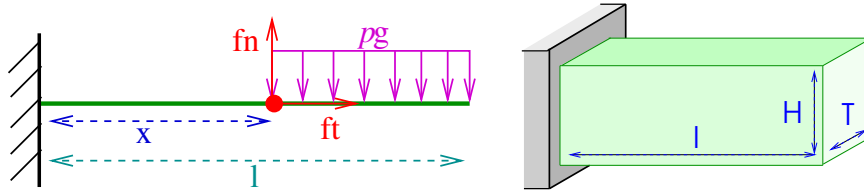


FIG. 6.1 – Notations utilisées pour le problème de la poutre encastree

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  $f_n$ .

D'après la théorie des poutres déformables, la flèche d'une poutre encastree soumise à la gravité est :

$$u = \frac{-1}{EI} \left( \rho g \frac{(l-x)^4}{24} \right) + \frac{1}{EI} \left( -\rho g \frac{l^3}{6} \right) x + \frac{1}{EI} \left( \rho g \frac{l^4}{24} \right)$$

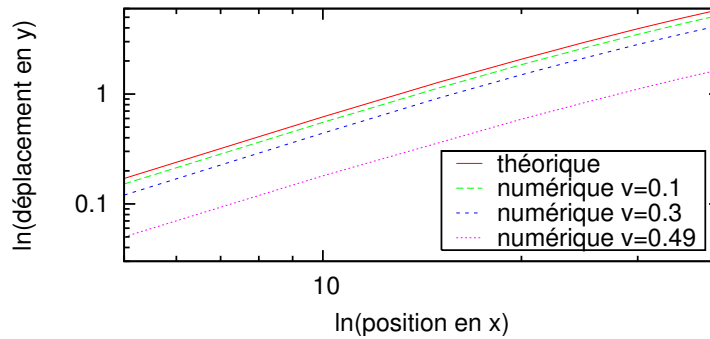
où  $E$  est le module d'Young, soit la rigidité du matériau, et  $I$  est le moment d'inertie de la poutre, fonction de la surface de coupe. Dans le cas d'une poutre rectangulaire,  $I = \frac{TH^3}{12}$ .

Cette solution théorique reste une approximation, car elle suppose que les déplacements vont être purement verticaux. Mais le résultat est très valide pour des petits déplacements. De plus, le coefficient de Poisson n'est pas pris en compte pour le matériau, la conservation du volume de la poutre n'est donc pas assurée. C'est pourquoi, plutôt que de comparer quantitativement les flèches des poutres, nous allons plutôt nous intéresser à comparer les courbures que prennent les poutres sous la gravité.

Différents résultats numériques sont comparés avec ce résultat analytique pour une poutre de 40 m de longueur sur 4 m de large et 2 m de haut, avec un module d'Young à 20 kPa et une densité de 1.

D'après la formulation analytique, la courbure de flexion suit un polynôme de degré 4, c'est pourquoi des échelles logarithmiques vont être utilisées, afin de comparer ce degré. La figure 6.2 fournit les résultats obtenus pour trois valeurs différentes du coefficient de Poisson. Pour un coefficient de Poisson donné, toutes les méthodes évaluées (tétraèdres et hexaèdres corotationnels QR et polaires) donnent des résultats quasi-similaires, c'est pourquoi nous n'en faisons apparaître qu'un seul.

Toutes ces courbes sont quasi-parallèles, on peut donc affirmer que les approches numériques suivent bien une loi de degré 4. De plus, les simulations convergent vers le résultat analytique quand le coefficient de Poisson tend vers zéro, ce qui reflète bien le fait que l'incompressibilité n'est pas prise en compte dans la théorie des poutres.



**FIG. 6.2** – Flèches de poutres sous la gravité obtenues par la théorie et par des calculs numériques pour différents coefficients de Poisson

Sur l'exemple de la poutre encastree, les approches corotationnelles (polaire et QR) semblent une bonne definition des deformations, y compris la version etendue aux hexaedres.

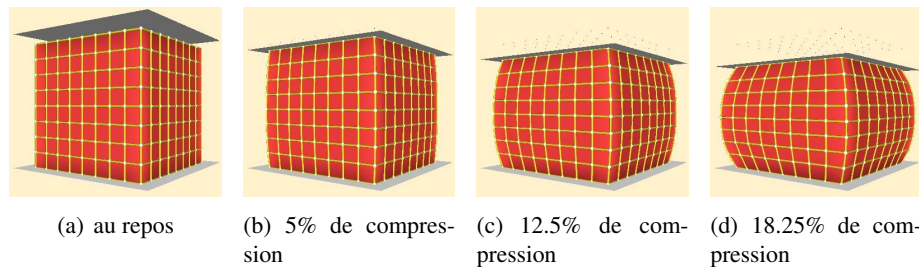
## 2.2 Truthcube

[Kerdok 03] fournit des donnees mesurees de deformations d'un cube de silicone. Dans l'exemple du truthcube, l'idee est de mesurer les deformations d'un cube de silicone compressé, en suivant les déplacements de petites billes de plastiques placees à l'interieur de celui-ci. Les billes sont localisees au cours de la compression puisque l'experience est realisee à l'interieur d'un scanner CT. Les propriétés materielles du silicone sont supposees connues (module d'Young à 14,9 KPa et coefficient de Poisson à 0,4999), ainsi que les conditions aux limites imposees. Il est ainsi possible de reproduire virtuellement l'experience avec diverses methodes numeriques et d'en evaluer leur precision.

Dans la simulation, le cube est modelise comme une grille de  $10 \times 10 \times 10$  hexaedres, pour lesquels les coins correspondent aux positionnements des billes. Ces billes sont assez petites pour être negligees durant la simulation ; le cube est donc considere comme totalement isotrope. Trois stades de compression de la face haute du cube sont realises (5%, 12.5% et 18.25%).

Pour tester les elements tetraedriques, nous choisissons de decouper chaque hexaedre en dix tetraedres en superposant les deux seules facons de decouper un hexaedre en cinq tetraedres (details en annexes D). La matiere étant dedoublée, le module d'Young est divise par deux.

Pour les simulations dynamiques, la compression est appliquee graduellement. En terme de parametrage, le pas de temps utilise est de 0,4 ms avec seulement 5 iterations du gradient conjuge de l'integration implicite, la gravite est de  $9,81 \text{ kg.m.s}^{-2}$ , le tout sans amortissement.



**FIG. 6.3** – Resultats des simulations de l'experience du Truthcube



video 14

Les résultats des simulations sont donnés dans le tableau 6.1. Plusieurs méthodes ont été testées. En dynamique, nous avons étudié le comportement d'une méthode purement linéaire et les méthodes corotationnelles à la fois basées sur les décomposition QR et polaire, aussi bien pour les tétraèdres que pour les hexaèdres. Parallèlement, nous avons effectué les mêmes tests en utilisant les méthodes statiques certifiées de Ansys, en utilisant un calcul linéaire et non-linéaire des déformations. Pour comparaison, les vitesses de calcul des méthodes dynamiques sont précisées.

**linéaire / non-linéaire** Les résultats entre méthodes purement linéaire et non-linéaire sont assez proche, signifiant que l'expérience de compression du Truthcube n'est pas assez discriminante car peu d'éléments subissent de grande rotation. Cela dit, au delà du seuil des petites déformations (10-15% de compression) les écarts commencent à grandir.

**corotationnelles QR / polaire** En offrant les plus petites déformations, la décomposition polaire offre des résultats un peu meilleurs (mais les écarts restent très faibles) que la décomposition QR, très proches du modèle Ansys non-linéaire. Par contre la décomposition QR est significativement plus rapide sur les tétraèdres, et surtout d'une stabilité à toute épreuve. L'approche corotationnelle semble être un élément indiscutable des simulations interactives, d'autant plus que cet exemple ne la met pas en valeur du fait des faibles rotations.

**tétraèdres / hexaèdres** Globalement, dans cette exemple l'utilisation d'hexaèdres ne donne pas significativement de meilleurs résultats pour un temps de calcul deux à trois fois plus important, ceci s'expliquant par le fait que tous les éléments sont bien réguliers dans cet exemple, en quel cas les tétraèdres sont plus rapides.

Attention, quelque soit la méthode employée, si l'erreur moyenne est acceptable (de l'ordre du millimètre), l'erreur maximale est plutôt grande, de l'ordre de plusieurs millimètres. Ce n'est pas un problème des méthodes dynamiques interactives, puisque Ansys donne les mêmes valeurs. D'une part, l'incompressibilité est mal simulée, mais ces erreurs soulignent aussi le fait que d'autres problèmes sont importants dans la modélisation, tels que les conditions aux limites qui ne semblent pas correspondre véritablement à la réalité de l'expérience.

		(mm)				(%)			
		Moy	Min	Max	E-T	Moy	Min	Max	E-T
<b>compression 5%</b>									
Tétraèdres	Linéaires	0.85	0.07	2.34	0.5	1.06	0.09	2.92	0.62
	QR	0.85	0.05	2.31	0.47	1.06	0.06	2.89	0.59
	Polaires	0.84	0.07	2.3	0.49	1.05	0.09	2.88	0.61
Hexaèdres	QR	0.84	0.07	2.3	0.49	1.05	0.09	2.88	0.61
	Polaires	0.85	0.07	2.3	0.48	1.06	0.09	2.88	0.6
	Ansys linéaire	0.86	0.02	2.35	0.5	1.07	0.03	2.94	0.63
	Ansys non-lin	0.85	0.03	2.34	0.5	1.06	0.04	2.93	0.63
<b>compression 12,5%</b>									
Tétraèdres	Linéaires	1.22	0.06	3.5	0.7	1.53	0.08	4.38	0.88
	QR	1.23	0.16	3.39	0.62	1.54	0.2	4.24	0.77
	Polaires	1.15	0.09	3.11	0.63	1.44	0.11	3.89	0.79
Hexaèdres	QR	1.15	0.07	3.12	0.64	1.44	0.09	3.9	0.8
	Polaires	1.18	0.07	3.1	0.62	1.47	0.09	3.87	0.78
	Ansys linéaire	1.19	0.08	3.43	0.7	1.49	0.1	4.29	0.87
	Ansys non-lin	1.16	0.04	3.3	0.66	1.45	0.05	4.12	0.83
<b>compression 18,25%</b>									
Tétraèdres	Linéaires	1.66	0.22	4.26	0.82	2.08	0.27	5.32	1.03
	QR	1.17	0.51	4.31	0.73	2.13	0.64	5.39	0.91
	Polaires	1.42	0.12	3.4	0.7	1.77	0.15	4.25	0.88
Hexaèdres	QR	1.41	0.07	3.49	0.74	1.76	0.09	4.36	0.92
	Polaires	1.46	0.11	3.33	0.7	1.82	0.14	4.16	0.87
	Ansys linéaire	1.58	0.35	4.14	0.8	1.97	0.44	5.18	1.00
	Ansys non-lin	1.42	0.16	3.86	0.73	1.78	0.2	4.83	0.91
<b>vitesse (ms par pas de temps)</b>									
Tétraèdres	Linéaires	40							
	QR	58							
	Polaires	80							
Hexaèdres	QR	150							
	Polaires	150							

Pour trois niveaux de compression, des éléments finis tétraédriques et hexaédriques linéaires, corotationnels (méthodes QR et polaire) + Ansys avec tenseurs de déformation linéaire et non-linéaire

**TAB. 6.1:** Résultats des simulations du Truthcube

### 3 Cas des hexaèdres non-pleins

On vient de voir que le calcul corotationnel offrait une certaine précision aux méthodes interactives ; nous allons voir ce qu'il en est quand les maillages hexaédriques comportent des éléments non-pleins. Pour cela, nous allons étudier l'influence du positionnement des hexaèdres par rapport à l'objet déformable, afin d'évaluer plus précisément l'effet des propriétés non-uniformes. L'objectif est de tester la cohérence de cette approche.

#### 3.1 Poutre encastrée

Nous reprenons ici l'exemple de la poutre encastrée soumise à la gravité. Contrairement à ce qui était fait au paragraphe 2.1, nous l'avons volontairement maillée de façon non-optimisée (éléments trop larges, tournés...) comme présenté dans la figure 6.4. Ces configurations ne sont bien sûr pas souhaitables, mais vont nous permettre de tester le comportement des éléments non-uniformes en les comparant au cas optimal, où la poutre est maillée de manière classique, sans élément non-plein. Même si la résolution des éléments est grossière en comparaison avec la poutre simulée, les résultats sont sensiblement similaires comme on peut le voir dans la figure 6.4. On a comparé les deux cas non-optimaux par rapport au cas optimal. Les résultats pour une poutre mesurant  $160 \times 40 \times 40 \text{ mm}^3$  sont donnés en figure 6.5.

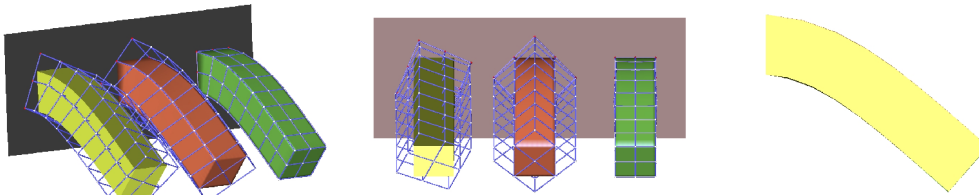


FIG. 6.4 – Différents maillages d'éléments englobant une poutre encastrée.

*vert : placement optimal (cas classique sans élément non-pleins), rouge et jaune : placements volontairement non-optimaux pour exagérer les éléments non-pleins. La figure de droite, vue de côté, superpose les résultats des simulations pour tous les cas.*

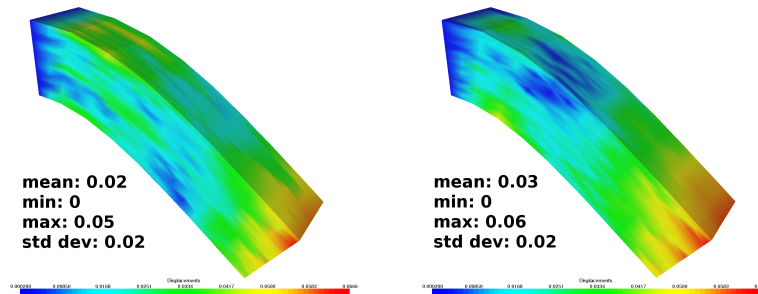


FIG. 6.5 – Erreur en distance euclidienne (mm) par rapport au maillage optimal

*gauche : poutre jaune, droite : poutre rouge de la figure 6.4*



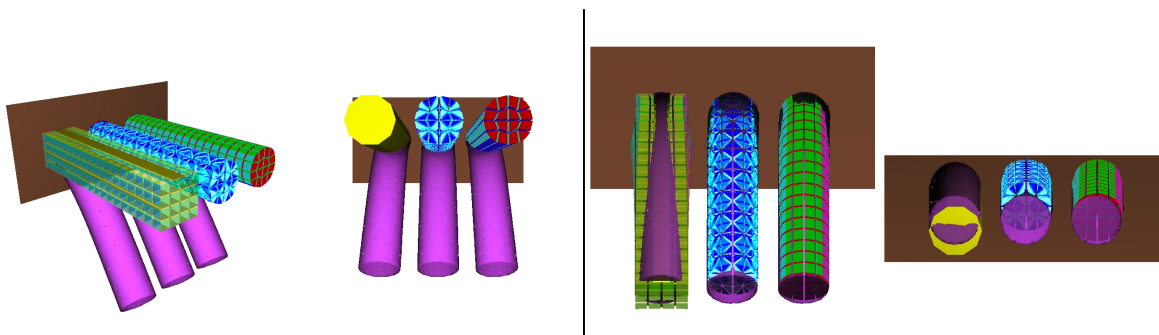
Notons tout de même un petit bémol, car les dynamiques (*i.e.* les chemins empruntés) pour arriver à cette position d'équilibre sont assez différentes suivant les maillages. Le maillage central (poutre rouge) est beaucoup plus amorti dans la direction de la courbure.

### 3.2 Cylindre

Dans le cas de la poutre encastree décrit ci-dessus, un maillage hexaédrique classique parait tout à fait adapté et donc l'utilisation d'éléments non-pleins peu pertinente. Par contre, pour des formes un peu plus compliquées à mailler, comme un cylindre, l'utilisation des éléments non-uniformes devient intéressante.

#### 3.2.1 Flexion

Dans la première expérience, nous allons comparer différents modèles interactifs face à des données réelles (issues de [SOFA]). Ces données consistent en un cylindre en silicone de longueur 35.56 et d'un rayon de 3.75 soumis à la gravité. Une comparaison quantitative de ces données avec les modèles numériques n'est pas attendue, car les paramètres du matériau du cylindre ne sont pas connus. Ils ont été déduits à partir de la théorie des poutres encastrees, qui n'est pas exacte pour des grandes déformations comme c'est le cas ici. Un module d'Young de 1116 Pa a été déduit en considérant le matériau comme incompressible (coefficient de Poisson de 0.49). Les modèles numériques interactifs mis en avant sont : nos éléments englobants non-uniformes (deux niveaux de subdivision), un maillage tétraédrique et un maillage hexaédrique, tous utilisant le même calcul des déformations corotationnel basé sur une décomposition polaire.



**FIG. 6.6** – Comparaison de simulations d'une poutre cylindrique encastree face à des données réelles

*Les données réelles sont en mauve, et de gauche à droite : les éléments englobants non-uniforme (objet embarqué en jaune), les tétraèdres, les hexaèdres. La partie gauche représentent les états initiaux, et la partie droite les résultats à l'équilibre.*

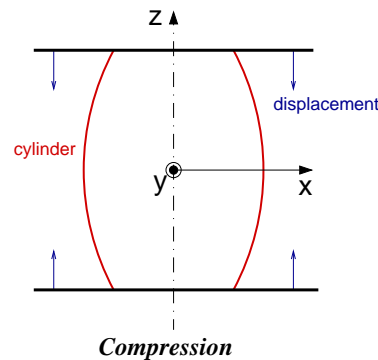
On peut remarquer dans les résultats illustrés en figure 6.6 qu'aucun modèle n'est parfait. Le maillage tétraédrique semble un peu trop raide (peut être dû au "locking effect"), alors que les éléments englobants non-uniformes semblent trop mous. Cette mollesse est certainement due à la simplification qui consiste à dire que les éléments frontières (du niveau le plus fin, deux niveaux en dessous non visualisé dans cet exemple) sont remplis de moitié, or ils prennent une place assez importante dans ce cas où ils sont trop nombreux par rapport au nombre total d'éléments.

### 3.2.2 Compression uni-axiale

La seconde expérience est illustrée dans la figure 6.7, on étudie le rayon central d'un cylindre écrasé dans le sens de son axe. Les solutions obtenues avec notre approche sont comparées à des solutions certifiées calculées à l'aide du logiciel Ansys [Ansys] en utilisant le tenseur de déformations non-linéaire de Green-Lagrange. Pour les simulations numériques, deux types de maillages sont utilisés, un premier, classique, est basé sur des prismes et des hexaèdres, le second utilise un maillage d'éléments englobants (présentés en figure 6.8). Le maillage mêlant prismes et hexaèdres a été construit à la main, il est nécessaire car le mailleur automatique en tétraèdres de Ansys donne un maillage trop complexe, avec lequel Ansys n'arrive pas à converger lorsque les matériaux sont quasi-incompressibles (coefficient de Poisson = 0.49). Nous pensons que ceci est dû aux non-symétries du maillage tétraédrique et au "locking effect". Ceci nous encourage encore plus à utiliser les éléments englobants pour ne plus être confronté à ce type de problèmes.

Pour tester l'influence du positionnement des hexaèdres, deux maillages ont été utilisés. Le premier est bien centré autour du cylindre, tandis que le second est décalé afin de ne pas être symétrique (cf. figure 6.8).

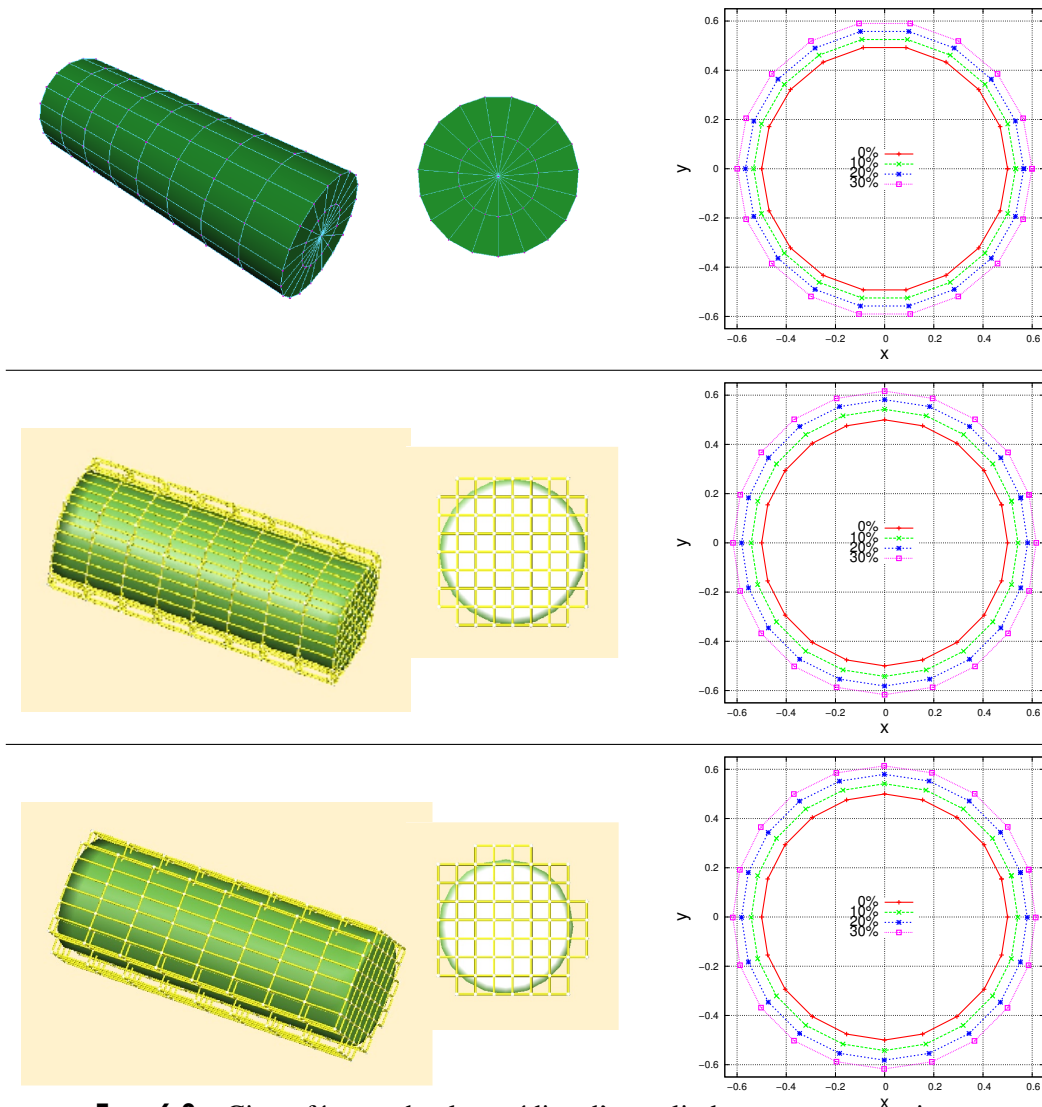
Dans les deux cas, bien que les éléments ne collent pas exactement à la surface, la circonférence du cylindre reste bien un cercle sous compression, comme présenté en figure 6.8, même si la boîte englobante considérée est décalée. Ces résultats encourageants sont vérifiés numériquement dans le tableau 6.2 où les rayons de chaque sommet du plan médian sont mesurés. Puisque le maillage utilisé sous Ansys est symétrique, il est cohérent que ses résultats soient parfaitement symétriques.



**FIG. 6.7** – Compression d'un cylindre composé d'un matériau quasi-incompressible (coefficient de Poisson = 0,49)

rayon	compression	moyenne	min	max	écart type
Ansys	10%	0,5331052	0,5331052	0,5331052	0
	20%	0,5662103	0,5662103	0,5662103	0
	30%	0,5993155	0,5993155	0,5993155	0
maillage bien positionné	10%	0,5435298	0,5424220	0,5446057	0,0008812
	20%	0,5835664	0,5815420	0,5855633	0,0016283
	30%	0,6194658	0,6167130	0,6222278	0,0022409
maillage mal positionné	10%	0,5429189	0,5412755	0,5445390	0,0009428
	20%	0,5828810	0,5796358	0,5859284	0,0018170
	30%	0,6189712	0,6141881	0,6232140	0,0026136

**TAB. 6.2:** Rayon central du cylindre pour trois niveaux de compression

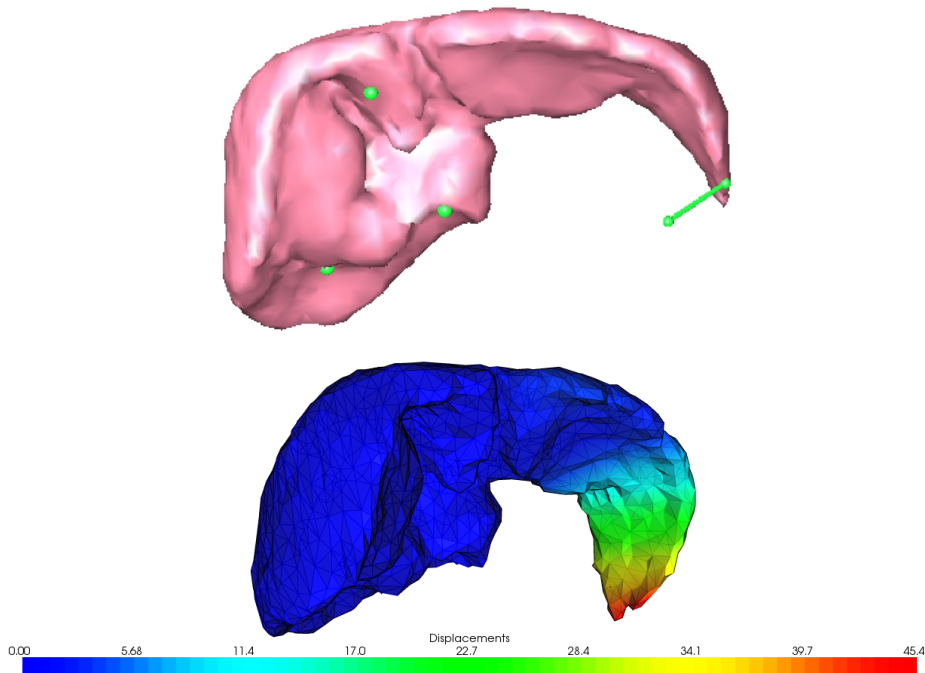


**FIG. 6.8** – Circonférence du plan médian d’un cylindre sous compression

*gauche : différents maillages du cylindre. haut : maillage classique en prismes et hexaèdres. milieu : éléments englobants idéalement placés. bas : éléments englobants mal placés. droite : circonférences du plan médian du cylindre au repos et après plusieurs compressions.*

### 3.3 Foie

Dans cette section, le cas d’un objet réel est considéré, en l’occurrence le foie introduit en figure 4.8. L’expérience est présentée en figure 6.9. Trois points de la surface sont fixés et un déplacement est imposé sur un quatrième. La taille du foie est d’environ  $100 \times 100 \times 200 \text{ mm}^3$ , et le déplacement est d’environ 40 mm. Le matériau hookéen employé a un module d’Young à 100 KPa et deux coefficients de Poisson sont considérés ( $\nu=0.3$  et  $0.4$ ). Ce déplacement imposé donne des déformations assez grandes, de l’ordre de 13-18%. Les résultats obtenus avec notre méthode sont comparés à des résultats numériques certifiés (Ansys).



**FIG. 6.9** – L’expérience de déplacements imposés sur le modèle du foie

*Trois points de la surface du foie sont fixés et un déplacement est imposé à un quatrième. Résultats en échelle thermique des déplacements résultants pour le maillage tétraédrique détaillé.*

On s’intéresse à comparer les positions des nœuds une fois l’état d’équilibre atteint. La solution de référence est calculée à l’aide du logiciel [Ansys] en utilisant le tenseur de déformation non-linéaire de Green-Lagrange afin de traiter les grands déplacements. Tous les résultats sont comparés à cette solution de référence.

Un traitement en petit déplacements est considéré afin d’évaluer l’importance de gérer les rotations, calculé aussi avec Ansys en utilisant le tenseur linéaire de déformations de Cauchy. Une comparaison avec des tétraèdres corotationnels est effectuée, afin de voir ce que donne le calcul corotationnel des déformations sur un maillage classique. De plus, cela permet une comparaison en terme de vitesse de calcul pour une autre approche dynamique interactive.

Les méthodes tétraédriques utilisent les maillages déjà présentés en figure 4.8.

Afin de permettre une bonne mesure de l’erreur, même les nœuds internes (*i.e.* pas seulement à la surface) sont utilisés dans la comparaison. Ceci impose d’interpoler plus de points que nécessaire avec les éléments englobants. Les résultats présentés dans le tableau 6.3 sont donnés en millimètres. L’expérience est fournie pour deux résolutions de maillages.

Résolution de la surface du foie	Méthode Mécanique	Erreur				Hz
		Moy	Min	Max	E.T.	
Grossière <sup>r1</sup> v = .3	Elem englobants grossiers <sup>m1</sup>	0.64	0.05	2.74	0.5	70
	Elem englobants détaillés <sup>m2</sup>	0.61	0.07	1.95	0.38	15
	Tétraèdres corotationnels <sup>t1</sup>	0.29	0.01	1.08	0.29	40
	Tétraèdres linéaires (Ansys Cauchy) <sup>t1</sup>	1.96	0.08	6.42	1.55	-
Détaillée <sup>r2</sup> v = .3	Elem englobants grossiers <sup>m1</sup>	0.65	0.04	1.84	0.35	43
	Elem englobants détaillés <sup>m2</sup>	0.98	0.04	1.7	0.36	13
	Tétraèdres corotationnels <sup>t2</sup>	0.07	0	0.27	0.07	5
	Tétraèdres linéaires (Ansys Cauchy) <sup>t2</sup>	2.09	0.04	7.4	1.77	-
Grossière <sup>r1</sup> v = .4	Elem englobants grossiers <sup>m1</sup>	0.8	0.05	3.24	0.73	70
	Elem englobants détaillés <sup>m2</sup>	0.66	0.06	2.4	0.49	15
	Tétraèdres corotationnels <sup>t1</sup>	0.24	0.03	0.77	0.19	40
	Tétraèdres linéaires (Ansys Cauchy) <sup>t1</sup>	1.94	0.1	6.36	1.54	-
Détaillée <sup>r2</sup> v = .4	Elem englobants grossiers <sup>m1</sup>	0.64	0.04	1.89	0.3	43
	Elem englobants détaillés <sup>m2</sup>	1.15	0.04	2.18	0.42	13
	Tétraèdres corotationnels <sup>t2</sup>	0.01	0	0.02	0	5
	Tétraèdres linéaires (Ansys Cauchy) <sup>t2</sup>	2.08	0.03	7.12	1.73	-

<sup>r1</sup> = 229 points, 452 triangles, <sup>r2</sup> = 1455 points, 2904 triangles

<sup>m1</sup> = 436 particules, 242 hexaèdres, <sup>m2</sup> = 2003 particules, 1346 hexaèdres

<sup>t1</sup> = 309 particules, 929 tétraèdres, <sup>t2</sup> = 2032 particules, 6725 tétraèdres

**TAB. 6.3:** Erreurs de diverses solutions numériques par rapport à une solution de référence (Ansys déformations non-linéaires de Green-Lagrange)

Comme attendu, le calcul linéaire des déformations donne des résultats très approximatifs, alors que le calcul corotationnel s'approche assez bien du tenseur non linéaire de Green-Lagrange. Notons que sur les tétraèdres, le calcul corotationnel est quasi-parfait, pour des temps de calcul relativement bons.

Plus intéressant : les résultats pour les éléments englobants sont bons avec une erreur moyenne en dessous de 1% (en prenant en compte la taille du foie).

Un comportement étrange peut être remarqué quand la résolution est trop importante. Un trop grand nombre de petits éléments fournit un système trop lourd à résoudre en quelques itérations, et la solution approchée donne des résultats moins bons que pour un maillage un peu plus grossier pour lequel on sait bien résoudre le système.

Néanmoins, les résultats des éléments non-uniformes sont proches de la solution de référence qui représente en quelque sorte la limite de précision que l'on saurait atteindre. Pour une précision juste un peu moindre que les tétraèdres corotationnels, l'accélération des temps de calculs n'est pas négligeable, surtout pour un affichage très détaillé ; sans compter la facilité de la construction du maillage ; même si pour cet exemple où le maillage tétraédrique est relativement bien conditionné, les éléments englobants ne sont pas énormément meilleurs que les tétraèdres.

## 4 Conclusion

Dans un premier temps, nous avons vérifié que les calculs corotationnels sont une bonne représentation des déformations, y compris pour la méthode QR, qui ne fournit pas les plus petites déformations pour être plus rapide et plus stable. Notre extension aux hexaèdres donne des résultats numériques corrects.

Dans un deuxième temps, nous avons analysé le comportement des éléments englobants, en prenant soin de les mettre à défaut dans des situations défavorables. La prise en compte de la répartition de la matière à l'aide des propriétés non-uniformes permet une relativement bonne cohérence du comportement, même sur des maillages très grossiers.

Grâce à ces tests, on peut conclure que les ordres de grandeur d'imprécision des calculs montrent que l'idée des maillages englobants avec la prise en compte de la répartition de la matière est une approche intéressante, qui semble très bien appropriée aux simulations interactives qui demandent une certaine précision, comme les simulateurs chirurgicaux.

Nous aurions maintenant grand besoin de disposer de données dynamiques réelles afin de tester les comportements dynamiques. Un exemple pertinent serait la compression du TruthCube sous IRM dynamique qui fournirait une vitesse pour le chargement et le dynamisme des déformations. On pourrait ainsi mieux évaluer les intégrations dynamiques, et pouvoir mesurer les apports des méthodes hiérarchiques employées pour améliorer les propagations.

Pour conclure, nous aimerions faire émerger un besoin de benchmark "universels", afin que chacun puisse comparer sa méthode avec exactement les mêmes expériences, les mêmes paramètres, et ne tester que sa nouveauté (par exemple prendre exactement le même schéma d'intégration temporelle pour évaluer des calculs de forces). Cette étape nous semble nécessaire si on désire évaluer et comparer quantitativement les différentes méthodes numériques proposées dans la littérature.







---

# Conclusion



## 1 Récapitulatif des contributions

Au cours de cette thèse, nous avons présenté une méthode simple et efficace pour animer des objets déformables en temps réel.

Un maillage mécanique basé sur une voxelisation englobant l'objet à animer a été proposé. Son principal avantage est sa simplicité de construction, qui le rend complètement automatique. La parfaite régularité des voxels retire les problèmes numériques liés aux éventuels mauvais conditionnements des tétraèdres et rend ce maillage idéalement adapté au temps réel. Nous avons montré comment coupler les grilles mécaniques avec des surfaces de contact quelconques et comment physiquement appliquer des forces et contraintes géométriques sur ces surfaces. Il permet d'animer facilement n'importe quel objet, quelle que soit sa complexité géométrique. Ce maillage permet d'ailleurs d'animer aussi bien les objets représentés par des surfaces que par des données volumiques. Nous l'avons utilisé pour modéliser mécaniquement des organes à partir d'images médicales avec des lois de comportement définies au voxel près, ainsi que des modèles polygonaux infographiques quelconques représentant des objets volumiques, surfaciques ou linéiques.

Pour animer efficacement ce maillage, nous avons proposé une méthode physique profitant pleinement de ses caractéristiques. La principale contribution est de prendre en considération la répartition non uniforme de la matière et des propriétés des matériaux à l'intérieur d'éléments hexaédriques, autorisant un comportement plausible même à une résolution grossière. Un calcul des déformations basé sur une méthode corotationnelle a été adapté aux hexaèdres afin de traiter les grands déplacements de façon rapide et robuste, même face à des déformations extrêmes introduisant des inversions.

Des représentations à diverses résolutions peuvent être facilement construites sous la forme d'un arbre octal fournissant des opérateurs de transitions hiérarchiques simples. Afin d'exploiter mécaniquement cette représentation multirésolution, nous avons proposé une gestion simple et physique des discontinuités entre niveaux de résolutions (*i.e.* nœuds en T) permettant un contrôle du niveau de détail par adaptativité. De plus, nous avons examiné deux méthodes permettant de propager plus rapidement les déformations, afin d'améliorer les comportements dynamiques en temps réel, par une résolution hiérarchique ou multigrid des équations.

Enfin, nous avons évalué qualitativement et quantitativement la précision de notre approche.

## 2 Analyse

### Qualités

- **souplesse** : diversité des objets représentables, de données volumiques issues d'images médicales à des modèles polygonaux comportant des détails très fins
- **précision** mécanique en prenant en compte la répartition de la matière et des propriétés des matériaux à l'intérieur des éléments
- **compromis** précision-temps de calcul par simple réglage de la résolution de la grille
- facilement extensible à des approches hiérarchiques et **multirésolutions**
- modélisation mécanique **automatique**

### Limitations

- nécessite des adaptations pour modéliser à basse résolution et hiérarchiquement des **objets ramifiés**
- l'application de **contraintes géométriques** dures nécessite de calculer des pseudo-inverses, coûteuses quand le nombre de contraintes par degré de liberté est élevé

## Discussion

Ces qualités suggèrent que la méthode est bien adaptée tant à l'animation d'objets infographiques qu'à la simulation médicale individualisée.

En effet, elle permet de "donner vie" automatiquement, et sans connaissance particulière, aussi bien à un modèle créé par un infographiste, qu'à des organes issus d'images médicales segmentées. Sa rapidité et sa robustesse autorisent des animations interactives où l'utilisateur peut imposer des sollicitations sévères. La possibilité de déformer rapidement des objets détaillés visuellement lui permet de rentrer dans le cadre d'applications où le rendu visuel est important. De plus, l'évaluation en pratique de son comportement montre une grande rigueur physique compatible avec une recherche de précision.

Pour plaire aux développeur, notre méthode repose sur des concepts relativement simples, qui lui confèrent une grande souplesse d'implémentation.

Même si l'animation interactive d'objets déformables voit le jour, modéliser des scènes réelles avec des matériaux mous reste un gros challenge. Si la modélisation physique arrive à progresser autant qu'en synthèse d'images, des applications très intéressantes pourront apparaître. La complexité géométrique et matérielle est telle que les applications traditionnelles issues de l'ingénierie mécanique sont insuffisantes. Nous sommes persuadés qu'il faut profiter de toutes les améliorations possibles, des simplifications calculatoires des modèles (par exemple la multirésolution adaptative ou les propriétés non-uniformes) aux optimisations algorithmiques (telle que la résolution multigrid) car elles offrent plus de gain que la simple attente de matériel plus puissant. Dans le même sens, nous pensons, qu'afin d'obtenir le meilleur compromis temps de calcul / précision, il est plus judicieux de se limiter dans le nombre de degrés de libertés et d'améliorer leur comportement, plutôt que d'essayer de traiter un maximum de degrés de libertés aux comportements simplistes.

## 3 Perspectives

### 3.1 A court terme

Afin de proposer des comportements pour tous types de tissus, il serait possible de prendre en compte d'autres lois de comportements de matériaux que l'élasticité linéaire. Les matériaux non-linéaires imposent des mises à jour du système, et donc la condensation des propriétés mécaniques permettant la prise en compte de la répartition de la matière à l'intérieur des éléments non pleins doit être refaite à chaque mise à jour. Cette étape étant relativement rapide, il ne semble pas qu'elle serait une entorse au temps réel.

Même si l'interpolation tri-linéaire de l'objet à l'intérieur de la grille mécanique procure des résultats visuellement agréables, il serait possible de lisser les transitions entre les éléments déformés, en utilisant une interpolation basée sur les polynômes de Bernstein par exemple.

### 3.2 A moyen terme

Il nous paraît indispensable de finaliser une approche multigrid adaptative, afin d'exploiter pleinement les avantages de la représentation hiérarchique, et ainsi fournir un très bon compromis précision/vitesse.

Dans ce document, il a été évoqué la possibilité d'utiliser le principe des "virtual nodes" dans le but de séparer les parties indépendantes d'un objet à une résolution mécanique grossière. Cette approche pose des problèmes pour créer une hiérarchie multirésolution. Vu son efficacité à une résolution donnée, nous pensons qu'il serait très utile de proposer des transitions hiérarchiques sur de tels maillages.

Dans certains cas où on dispose d'un maillage tétraédrique bien conditionné, l'apport actuel des éléments englobants n'est pas significatif. Il pourrait alors être intéressant d'étendre la condensation des propriétés non-uniformes aux tétraèdres, afin de prendre en compte différentes propriétés mécaniques dans un même tétraèdre, au voxel près pour des données médicales. On peut même imaginer fabriquer des maillages de tétraèdres grossiers englobants l'objet.

### 3.3 A long terme

La découpe est indispensable à certains types d'applications, comme la plupart des simulations chirurgicales ; il faudrait donc l'intégrer dans notre modèle. Nous sommes convaincus que des méthodes déjà existantes pourraient s'y intégrer efficacement, telles que les "virtual nodes" ou les X-FEM, mais pour lesquelles le problème d'une représentation multirésolution n'a pas été envisagé.

Afin de mettre au point des simulations très précises, il est indispensable d'évaluer le comportement dynamique des modèles numériques avec entre autre des confrontations à des données dynamiques réelles. Pour cela il faudrait être capable d'accéder à des images volumiques dynamiques, type IRM dynamique, à une bonne résolution temporelle (actuellement d'environ une image par seconde).

Dans ce but, mais aussi afin de tester le comportement statique des modèles numériques, il semblerait très utile de mettre au point un jeu d'expériences, avec toutes les données nécessaires, permettant de quantitativement comparer les méthodes numériques entre elles et face à la réalité.

Pour finir, on peut vraisemblablement imaginer que les grilles de voxels vont de plus en plus être utilisées comme primitives volumiques, de la même manière que les pixels ont remplacé les écrans vectoriels. Cette évolution repose sur du matériel spécialisé, utilisant un type d'algorithmique massivement parallèle qui commence à être adapté au traitement des textures 3D, très proches des grilles volumiques. Ces accélérations devraient pouvoir être employées sur des grilles d'éléments englobants déformables.



# Bibliographie

---

- [Abolhassani 07] N Abolhassani, R Patel & M Moallem. *Needle insertion into soft tissue : a survey*. In *Medical Engineering and Physics*, volume 29, pages 413–431, 2007.
- [Alexa 05] Marc Alexa, Alexis Angelidis, Marie-Paule Cani, Karan Singh & Denis Zorin. *Interactive shape modelling*, volume 5 of *Eurographics Tutorial series*. 2005.
- [Alexa 06] Marc Alexa. *Mesh editing based on discrete Laplace and Poisson models*. In *ACM SIGGRAPH Courses*, pages 51–59, 2006.
- [Alexopoulos 05] L.G. Alexopoulos, G.M. Williams, M.L. Upton, L.A. Setton & G. Guilak. *Osteoarthritic changes in the biphasic mechanical properties of the chondrocyte pericellular matrix in articular cartilage*. *Journal of Biomechanics*, vol. 38, pages 509–517, 2005.
- [Allard 07] Jérémie Allard, Stéphane Cotin, François Faure, Pierre-Jean Bensoussan, François Poyer, Christian Duriez, Hervé Delingette & Laurent Grisoni. *SOFA an Open Source Framework for Medical Simulation*. In *Medicine Meets Virtual Reality (MMVR)*, 2007.
- [AnimAL] *AnimAL Animation Algorithm Library*.
- [Ansys] *Ansys engineering simulation software*. <http://www.ansys.com>.
- [Baraff 98] David Baraff & Andrew Witkin. *Large steps in cloth simulation*. In *SIGGRAPH '98*, 1998.
- [Barbič 05] Jernej Barbič & Doug L. James. *Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models*. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 24 (3), pages 982–990, 2005.
- [Barr 84] A. H. Barr. *Global and Local Deformations of Solid Primitives*. In *Proc SIGGRAPH '84*, pages 21–30, 1984.
- [Basar 00] Yavuz Basar & Dieter Weichert. *Nonlinear continuum mechanics of solids*. Springer, 2000.
- [Bathe 82] K.J. Bathe. *Finite element procedures in engineering analysis*. Prentice-Hall, 1982.
- [Batter 71] J. J. Batter & F. P. Brooks Jr. *Grope-1 : A computer display to the sense of feel*. In *Information Processing (Proc of IFIP Congress)*, pages 759–763, 1971.
- [Baur 98] C. Baur, D. Guzzoni & O. Georg. *Virgy : A virtual reality and force feedback based endoscopic surgery simulator*, 1998.
- [Belytschko 76] T. Belytschko. *A Survey of Numerical Methods and Computer Programs for Dynamic Structural Analysis*. *Nuclear Engineering and Design*, vol. 37, pages 23–24, 1976.
- [Benabid 87] A. L. Benabid, P. Cinquin, S. Laval, J. F. Le Bas, J. Demongeot, & J. de Rougemont. *Computer-driven robot for stereotactic surgery connected to CT scan and magnetic resonance imaging. Technological design and preliminary results*. *Applied Neurophysiology*, vol. 50, pages 153–154, 1987.

- [Benzley 95] S.E. Benzley, E. Perry, K. Merkley & B. Clark. *A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis*. In Proc. 4th International Meshing Roundtable, pages 179–191, 1995.
- [Bern 00] Marshall Bern & Paul Plassmann. *Mesh Generation*. In Handbook of Computational Geometry. Elsevier Science, 2000.
- [Bielser 04] D. Bielser, P. Glardon, M. Teschner & M. Gross. *A state machine for real-time cutting of tetrahedral meshes*. Graph. Models, vol. 66, no. 6, pages 398–417, 2004.
- [Blender] Blender. <http://www.blender.org/>.
- [Bolz 03] Jeff Bolz, Ian Farmer, Eitan Grinspun & Peter Schröder. *Sparse matrix solvers on the GPU : Conjugate gradients and multigrid*. In Proceedings of SIGGRAPH, pages 917–924, 2003.
- [Botsch 06] M. Botsch, R. W. Sumner, M. Pauly & M. Gross. *Deformation transfer for detail-preserving surface editing*. In Vision, Modeling, Visualization, 2006.
- [Boudou 06] T. Boudou, J. Ohayon, Y. Arntz, G. Finet ans C. Picart & P. Tracqui. *An extended modeling of the micropipette aspiration experiment for the characterization of the Young's modulus and Poisson's ratio of adherent thin biological samples : Numerical and experimental studies*. Journal of Biomechanics, vol. 39, pages 1677–1685, 2006.
- [Bourguignon 00] David Bourguignon & Marie-Paule Cani. *Controlling Anisotropy in Mass-Spring Systems*. In Eurographics Workshop on Computer Animation and Simulation (EGCAS), Springer Computer Science, pages 113–123, 2000.
- [Bouthors 07] Antoine Bouthors & Matthieu Nesme. *Twinned Meshes for Dynamic Triangulation of Implicit Surfaces*. In Graphics Interface, 2007.
- [Brandt 77] A. Brandt. Multi-level adaptive solutions to boundary value problems, volume 31 of *Mathematics of Computation*. 1977.
- [Breen 92] D. Breen, D. House & P. Getto. *A physically-based particle model of woven cloth*. The Visual Computer, vol. 8, no. 5–6, pages 264–277, 1992.
- [Breen 94] David E. Breen, Donald H. House & Michael J. Wozny. *Predicting the drape of woven cloth using interacting particles*. In Proc SIGGRAPH'94, pages 365–372, New York, NY, USA, 1994. ACM Press.
- [Bridson 03a] R. Bridson, R. Fedkiw & J. Anderson. *Robust treatment of collisions, contact and friction for cloth animation*. In ACM Transactions on Graphics (Proc of SIGGRAPH), volume 21(3), pages 594–603, 2003.
- [Bridson 03b] R. Bridson, S. Marino & R. Fedkiw. *Simulation of clothing with folds and wrinkles*. In SCA, 2003.
- [Bridson 05] R. Bridson & Chen Greif. *A multi-preconditioned conjugate gradient algorithm*. SIAM Journal on Matrix Analysis and Applications, vol. 27, no. 4, pages 1056–1068, 2005.
- [Briggs 00] William L. Briggs, Van Emden Henson & Steve F. McCormick. *A multigrid tutorial (second edition)*. SIAM. 2000.
- [Bro-Nielsen 96] M. Bro-Nielsen & S. Cotin. *Real-time volumetric deformable models for surgery simulation using finite elements and condensation*. In Proc Eurographics, 1996.



- [Brown 94] J. M. Brown & J. E. Colgate. *Physics-Based Approach to Haptic Display*. In Proc of ISMCR 94, Topical Workshop on Virtual Reality, 1994.
- [Bucki 07] M. Bucki, C. Lobos & Y. Payan. *Framework for a Low-cost Intra-operative Image-guided Neuronavigator Including Brain Shift Compensation*. In IEEE Engineering in Medicine and Biology Society, volume 29, 2007.
- [Bullet] Bullet. *Bullet Physics Library*. <http://www.bulletphysics.com/Bullet/>.
- [Cadoz 99] C. Cadoz. *sound, music, and image creation with physical model- the CORDIS and GENESIS tools from ACROE*. 1999.
- [Cani 93] Marie-Paule Cani. *An implicit formulation for precise contact modeling between flexible solids*. In Computer Graphics Proceedings, Proc of SIGGRAPH, pages 313–320, 1993.
- [Capell 02a] Steve Capell, Seth Green, Brian Curless, Tom Duchamp & Zoran Popović. *Interactive Skeleton-Driven Dynamic Deformations*. In Proc SIGGRAPH'02, 2002.
- [Capell 02b] Steve Capell, Seth Green, Brian Curless, Tom Duchamp & Zoran Popović. *A multiresolution framework for dynamic deformations*. In SCA '02, 2002.
- [Castellano-Smith 01] A. Castellano-Smith, T. Hartkens, J. Schnabel, D. Hose, H. Liu, W. Hall, C. Truwit, D. Hawkes & D. Hill. *Constructing patient specific models for correcting intraoperative brain deformation*. In Lecture Notes in Computer Science, volume 2208, pages 1091–1098, 2001.
- [Catheline 99] S. Catheline, J.L. Thomas, F. Wu & M. Fink. *Diffraction Field of a Low Frequency Vibrator in Soft Tissues Using Transient Elastography*. IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, 1999.
- [Chabanas 02] M Chabanas. *Modélisation des tissus mous de la face pour la chirurgie orthognatique assistée par ordinateur*. In Doctorat de l'UJF (ED ISCE), 2002.
- [Chabanas 03] M. Chabanas, V. Luboz & Y. Payan. *Patient Specific Finite Element Model of the Face Soft Tissue for Computer-assisted Maxillofacial Surgery*. In Medical Image Analysis, volume 7/2, pages 131–151, 2003.
- [CHAI 3D] CHAI 3D *An Open-Source framework for Haptics and Dynamics Simulations*. <http://www.chai3d.org>.
- [Chenney 97] Stephen Chenney & David A. Forsyth. *View-Dependent Culling of Dynamic Systems in Virtual Environments*. In Symposium on Interactive 3D Graphics, pages 55–58, 1997.
- [Chouly 03] F. Chouly, A. Van Hirtum, X. Pelorson & Y. Payan. *Obstructive sleep apnea syndrome. Part 2 : computer simulation of the fluid-structure interaction*. In Arch. of Physiology and Biochemistry, volume 111, page 54, 2003.
- [Chua 00] Clint Chua & Ulrich Neumann. *Hardware-accelerated free-form deformation*. In HWWS '00, pages 33–39, 2000.
- [Clatz 05] O. Clatz, M. Sermesant, P.-Y. Bondiau, H. Delingette, S. K. Warfield, G. Malandain & N. Ayache. *Realistic simulation of the 3D growth of brain tumors in mr images including diffusion and mass effect*. In IEEE Transactions on Medical Imaging, volume 24(10), pages 1334–1346, 2005.

- [Clavet 05] Simon Clavet, Philippe Beaudoin & Pierre Poulin. *Particle-based Viscoelastic Fluid Simulation*. In Symposium on Computer Animation, pages 219–228, 2005.
- [Cohen 95] J. D. Cohen, M. C. Lin, D. Manocha & M. K. Ponamgi. *I-COLLIDE : An interactive and exact collision detection system for large-scale environments*. In Symposium on Interactive 3D Graphics, pages 189–196, 1995.
- [Cotin 96] S. Cotin, H. Delingette, J.-M. Clement, V. Tasseti, J. Marescaux & N. Ayache. *Volumetric Deformable Models for Simulation of Laparoscopic Surgery*. In Computer Assisted Radiology, 1996.
- [Cotin 97] S. Cotin. *Modèles anatomiques déformables en temps réel : Application à la simulation de chirurgie avec retour d'effort*. PhD thesis, Université de Nice Sophia-Antipolis, 1997.
- [Cotin 99] Stéphane Cotin, Hervé Delingette & Nicholas Ayache. *Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation*. IEEE Transactions on Visualization and Computer Graphics, vol. 5 (1), pages 62–73, 1999.
- [Cotin 00] S. Cotin, H. Delingette & N. Ayache. *A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation*. The Visual Computer, vol. 16, no. 8, 2000.
- [Couteau 00] B. Couteau, Y. Payan & S. Lavallée. *The Mesh-Matching algorithm : an automatic 3D mesh generator for finite element structures*. In Journal of Biomechanics, volume 33/8, pages 1005–1009, 2000.
- [Cover 93] Steven Cover, Norberto Ezquerro, James O'Brien, Richard Rowe, Thomas Gaudacz & Ellen Palm. *Interactively Deformable Models for Surgery Simulation*. IEEE Comput. Graph. Appl., vol. 13, no. 6, pages 68–75, 1993.
- [Debunne 99] G. Debunne, M. Desbrun, A. H. Barr & M-P. Cani. *Interactive multiresolution animation of deformable models*. In Eurographics Workshop on Computer Animation and Simulation, 1999.
- [Debunne 00a] G. Debunne, M. Desbrun, M-P. Cani & A. H. Barr. *Adaptive Simulation of Soft Bodies in Real-Time*. In Computer Animation 2000, 2000.
- [Debunne 00b] Gilles Debunne. *Animation multirésolution d'objets déformables en temps-réel, Application à la simulation chirurgicale*. PhD thesis, Institut National Polytechnique de Grenoble, 2000.
- [Debunne 01] G. Debunne, M. Desbrun, M-P. Cani & A. H. Barr. *Dynamic Real-Time Deformations using Space and Time Adaptive Sampling*. In SIGGRAPH '01, 2001.
- [Delingette 98] Herve Delingette. *Towards Realistic Soft Tissue Modeling in Medical Simulation*. Rapport technique RR-3506, INRIA, 1998.
- [Delingette 99] H. Delingette, S. Cotin & N. Ayache. *Efficient linear elastic models of soft tissues for real-time surgery simulation*. Proc. MMVR (Medicine Meets Virtual Reality), no. 7, pages 139–151, 1999.
- [Dequidt 05] J. Dequidt, D. Marchal & L. Grisoni. *Time Critical animation of deformable solids*. Journal of Computer Animation and Virtual Worlds, vol. 16, pages 177–187, 2005.

- [Desbrun 95] Mathieu Desbrun & Marie-Paule Cani. *Animating soft substances with implicit surfaces*. In Computer Graphics Proceedings, Proc of SIGGRAPH, pages 287–290, 1995.
- [Desbrun 99] Mathieu Desbrun, Peter Schröder & Alan Barr. *Interactive Animation of Structured Deformable Objects*. In Graphics Interface, pages 1–8, 1999.
- [Dingliana 00] J. Dingliana & C. O’Sullivan. *Graceful degradation of collision handling in physically based animation*. In Computer Graphics Forum, volume 19(3), pages 239–248, 2000.
- [Dong 04] Zhao Dong, Wei Chen, Hujun Bao, Hongxin Zhang & Qunsheng Peng. *Real-time Voxelization for Complex Polygonal Models*. In Proceedings of the Computer Graphics and Applications, 12th Pacific Conference, pages 43–50, 2004.
- [Dutt 96] V. Dutt, R. R. Kinnick & J.F. Greenleaf. *Acoustic Shear Wave Displacement Measurement Using Ultrasound*. IEEE Ultrasonics Symposium, vol. 2, pages 1185–1188, 1996.
- [Eberhardt 00] B. Eberhardt, O. Etmuss & M. Hauth. *Implicit-Explicit Schemes for Fast Animation with Particles Systems*. In Proceedings of the Eurographics workshop on Computer Animation and Simulation, pages 137–151, 2000.
- [Eisemann 06] Elmar Eisemann & Xavier Décoret. *Fast Scene Voxelization and Applications*. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pages 71–78, 2006.
- [Etzmuß 03a] O. Etmuß & M. Keckeisen. *A Linearised Finite Element Model for Cloth Animation*. Technical Report WSI-2003-2, Universität Tübingen, 2003.
- [Etzmuß 03b] O. Etmuß, M. Keckeisen & W. Straßer. *A Fast Finite Element Solution for Cloth Modelling*. Proc Pacific Graphics, 2003.
- [Faloutsos 97] P. Faloutsos, M. van de Panne & D. Terzopoulos. *Dynamic Free-Form Deformations for Animation Synthesis*. IEEE Transactions on Visualization and Computer Graphics, vol. 3, no. 3, pages 201–214, 1997.
- [Faure 07] François Faure, Jérémie Allard & Matthieu Nesme. *Eulerian Contact for Versatile Collision Processing*. Rapport technique RR-6203, INRIA, 2007.
- [Felippa 00] C.A. Felippa. *Nonlinear finite element methods*. Lecture Notes of The University of Colorado, 2000.
- [Fung 93] Y.C. Fung. *Biomechanics, mechanical properties of living tissues*. Springer Verlag, second edition, 1993.
- [Gain 01] James E. Gain & Neil A. Dodgson. *Preventing Self-Intersection under Free-Form Deformation*. IEEE Transactions on Visualization and Computer Graphics, vol. 7, no. 4, pages 289–298, 2001.
- [Ganovelli 99] F. Ganovelli, P. Cignoni & R. Scopigno. *Introducing multiresolution representation in deformable object modeling*. In ACM Spring Conference on Computer Graphics, pages 149–158, 1999.
- [Ganovelli 00a] F. Ganovelli, P. Cignoni, C Montani & R. Scopigno. *Enabling Cuts on Multiresolution Representation*. In CGI ’00 : Proceedings of the International Conference on Computer Graphics, page 183, 2000.

- [Ganovelli 00b] Fabio Ganovelli, Paolo Cignoni, Claudio Montani & Roberto Scopigno. *A Multiresolution Model for Soft Objects Supporting Interactive Cuts and Lacerations*. Computer Graphics Forum (Proc. of EG), vol. 19, no. 3, 2000.
- [Ganovelli 01] F. Ganovelli & C. O’Sullivan. *Animating cuts with on-the-fly re-meshing*, 2001.
- [Garrigues 00] Jean Garrigues. *Initiation à la méthode des éléments finis*, 2000.
- [Georgii 05] Joachim Georgii, Florian Echtler & Rüdiger Westermann. *Interactive Simulation of Deformable Bodies on GPUs*. In Proceedings of Simulation and Visualisation 2005, pages 247–258, 2005.
- [Georgii 06] Joachim Georgii & Rüdiger Westermann. *A Multigrid Framework for Real-Time Simulation of Deformable Bodies*. Computer & Graphics, vol. 30, pages 408–415, 2006.
- [Gerard 03] J.M. Gerard, R. Wilhelms-Tricarico, P. Perrier & Y. Payan. *A 3D dynamical biomechanical tongue model to study speech motor control*. In Recent Research Developments in Biomechanics, pages 49–64, 2003.
- [Gerard 04] J.M. Gerard. *Un modèle biomécanique tridimensionnel de la langue*. In Doctorat de l’INPG (Spécialité : Signal, Image, Parole et Télécom), 2004.
- [Gerard 05] J.M. Gerard, J. Ohayon, V. Luboz, P. Perrier & Y. Payan. *Nonlinear elastic properties of the lingual and facial tissues assessed by indentation technique. Application to the biomechanics of speech production*. Medical Engineering and Physics, vol. 27, pages 884–892, 2005.
- [Gibson 97] Sarah F. Frisken Gibson. *3D ChainMail : A Fast Algorithm for Deforming Volumetric Objects*. In Symposium on Interactive 3D Graphics, pages 149–154, 1997.
- [GiPSi] GiPSi *An Open Source/Open Architecture Software Development Framework for Surgical Simulation*. <http://gipsi.case.edu>.
- [Gortler 95] Steven J. Gortler & Michael F. Cohen. *Hierarchical and variational geometric modeling with wavelets*. In SI3D ’95, 1995.
- [Gottschalk 96] S. Gottschalk, M. C. Lin & D. Manocha. *OBTree : a hierarchical structure for rapid interference detection*. In SIGGRAPH : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 171–180, 1996.
- [Goulette 07] François Goulette. *Hyper Elastic Mass Links*. In Groupe de Travail Animation et Simulation. Ecole des Mines de Paris, 2007.
- [Govindaraju 07] Naga K. Govindaraju, Ilknur Kabul, Ming C. Lin & Dinesh Manocha. *Fast continuous collision detection among deformable models using graphics processors*. In Computer Graphics, volume 31(1), 2007.
- [Grinspun 02] Eitan Grinspun, Petr Krysl & Peter Schröder. *CHARMS : a simple framework for adaptive simulation*. In SIGGRAPH ’02, 2002.
- [Guendelman 03] E. Guendelman, R. Bridson & R. Fedkiw. *Nonconvex Rigid Bodies with Stacking*. In ACM TOG, Proc. of SIGGRAPH, volume 22, pages 871–878, 2003.
- [Guendelman 05] E. Guendelman, A. Selle, F. Losasso & R. Fedkiw. *Coupling Water and Smoke to Thin Deformable and Rigid Shells*. In SIGGRAPH, pages 973–981, 2005.

- [Hauser 03] Kris K. Hauser, Chen Shen & James F. O'Brien. *Interactive Deformation Using Modal Analysis with Constraints*. In Graphics Interface, pages 247–256, 2003.
- [Hauth 01] M. Hauth & O. Eitzmuss. *A High Performance Solver for the Animation of Deformable Objects using Advanced Numerical Methods*. In Proceedings of Eurographics, 2001.
- [Hauth 04a] M. Hauth & W. Straßer. *Corotational Simulation of Deformable Solids*. In Proc WSCG, 2004.
- [Hauth 04b] Michael Hauth. *Visual Simulation of Deformable Models*. Phd thesis, Wilhelm-Schickard-Institut für Informatik, University of Tübingen, Germany, July 2004.
- [Havok] Havok. *Havok Physics*. <http://www.havok.com/>.
- [Heidelberger 04] Bruno Heidelberger, Matthias Teschner & Markus Gross. *Detection of collisions and self-collisions using image-space techniques*. In WSCG, 2004.
- [Hendriks 06] F.M. Hendriks, D. Brokken, C.W.J. Oomens, D.L. Bader & F.P.T. Baaijens. *The relative contributions of different skin layers to the mechanical behavior of human skin in vivo using suction experiments*. Medical Engineering and Physics, vol. 28, pages 259–266, 2006.
- [Hirota 99] Gentaro Hirota, Renee Maheshwari & Ming C. Lin. *Fast Volume-Preserving Free Form Deformation Using Multi-Level Optimization*. In ACM SOLID MODELING, 1999.
- [Horton 06] Ashley Horton, Adam Wittek & Karol Miller. *Towards Meshless Methods for Surgical Simulation*. In MICCAI Workshop, 2006.
- [Howlett 98] P. Howlett & W. T. Hewitt. *Mass-Spring Simulation using Adaptive Non-Active Points*. Computer Graphics Forum (Proc. of EG), vol. 17, no. 3, 1998.
- [Hsu 92] William M. Hsu, John F. Hughes & Henry Kaufman. *Direct manipulation of free-form deformations*. In Proc of SIGGRAPH '92, pages 177–184, New York, NY, USA, 1992. ACM Press.
- [Hughes 87] T. J. R. Hughes. *The finite element method, linear static and dynamic finite element analysis*. Englewood Cliffs. Prentice-Hall, 1987.
- [Hugues 87] T.J.R. Hugues. *The finite element method : Linear static and dynamic finite element analysis*. Prentice-Hall, 1987.
- [Hunter 05] P. Hunter. *FEM/BEM Notes*. In University of Oakland, New Zealand, 2005.
- [Hutchinson 96] D. Hutchinson, M. Preston & T. Hewitt. *Adaptive refinement for mass/spring simulations*. In Eurographics Workshop on Computer Animation and Simulation, pages 31–45, 1996.
- [Irving 04] G. Irving, J. Teran & R. Fedkiw. *Invertible Finite Elements For Robust Simulation of Large Deformation*. In Proc SCA, 2004.
- [Irving 07] G. Irving, C. Schroeder & R. Fedkiw. *Volume Conserving Finite Element Simulations of Deformable Models*. In Proc SIGGRAPH, 2007.
- [James 99] D. James & D. Pai. *Accurate Real Time Deformable Objects*. In Proc SIGGRAPH, 1999.

- [James 04a] D. L. James & D. K. Pai. *BD-Tree : Output-Sensitive Collision Detection for Reduced Deformable Models*. Proc of ACM SIGGRAPH, 2004.
- [James 04b] Doug L. James, Jernej Barbič & Christopher D. Twigg. *Squashing Cubes : Automating Deformable Model Construction for Graphics*. In Proc SIGGRAPH'04 Conference on Sketches & Applications, 2004.
- [Jerabkova 07] Lenka Jerabkova, Jakub Jerabek, Rostislav Chudoba & Torsten Kuhlen. *A Stable Cutting Method for Finite Elements based Virtual Surgery Simulation*. In Proc of Medicine Meets Virtual Reality (MMVR), 2007.
- [Joldes 06] G.R. Joldes, A. Wittek & K.S Miller. *Improved Linear Tetrahedral Element for Surgical Simulation*. In Medical Image Computing and Computer Assisted Intervention Conference (MICCAI), pages 54–65, 2006.
- [Kerdok 03] A. E. Kerdok, S. M. Cotin, M. P. Ottensmeyer, A. M. Galea, R. D. Howe & S. L. Dawson. *Truth Cube : Establishing Physical Standards for Real Time Soft Tissue Simulation*. Medical Image Analysis, vol. 7, 2003.
- [Kirkpatrick 01] S. W. Kirkpatrick, M. Schroeder & J. W. Simons. *Evaluation of Passenger Rail Vehicle Crashworthiness*. International Journal of Crashworthiness, vol. 6, pages 95–106, 2001.
- [Klein 03] J. Klein & G. Zachmann. *Adb-trees : Controlling the error of time-critical collision detection*. In 8th International Fall Workshop Vision, Modeling, and Visualization (VMV), pages 19–21, 2003.
- [Kuhnappel 00] U. Kuhnappel, H. Cakmak & H. Maass. *Endoscopic surgery training using virtual reality and deformable tissue simulation*. Computers & Graphics, vol. 24, pages 671–682, 2000.
- [Kwoh 88] Y. S. Kwoh, J. Hou, E. A. Jonckheere & S. Hayaty. *A robot with improved absolute positioning accuracy for CT guided stereotactic brain surgery*. IEEE Trans. on Biomedical Engineering, vol. 35, pages 153–160, 1988.
- [Lavallée 89] Stéphane Lavallée. *Gestes Medico-Chirurgicaux Assistés par Ordinateur : Application à la Neurochirurgie Stéréotaxique*. PhD thesis, Université de Grenoble, 1989.
- [Lennard-Jones 31] J. E Lennard-Jones. *Cohesion*. Proceedings of the Physical Society, vol. 43, pages 461–482, 1931.
- [Lensch 02] Hendrik P. A. Lensch, Katja Daubert & Hans-Peter Seidel. *Interactive Semi-Transparent Volumetric Textures*. In Proc Vision, Modeling and Visualization, pages 505–512, 2002.
- [Levinson 95] S.F. Levinson, M. Shinagawa & T. Sato. *Sonoelastic determination of human skeletal muscle elasticity*. Journal of Biomechanics, vol. 28, pages 1145–1154, 1995.
- [Levoy 90] Marc Levoy. *Efficient ray tracing of volume data*. ACM Trans. Graph., vol. 9, no. 3, pages 245–261, 1990.
- [Liu 94] A. Liu & B. Joe. *Relationship between tetrahedron shape measures*. In BIT, volume 34, pages 268–287, 1994.
- [Lloyd 07] B. Lloyd, G. Szekely & M. Harders. *Identification of Spring Parameters for Deformable Object Simulation*. IEEE Transactions on Visualization and Computer Graphics, vol. 13, 2007.

- [Loehner 88] R. Loehner & P. Parikh. *Three-dimensional grid generation via the advancing-front method*. In *International Journal for Numerical Methods in Fluids*, pages 1135–1149, 1988.
- [Lorensen 87] William E. Lorensen & Harvey E. Cline. *Marching Cubes : A High Resolution 3D Surface Construction Algorithm*. In *Computer Graphics (Proceedings of SIGGRAPH 87)*, 1987.
- [Luboz 04] V. Luboz, A. Pedrono, D. Ambard, F. Boutault, Y. Payan & P. Swider. *Prediction of tissue decompression in orbital surgery*. In *Clinical Biomechanics*, volume 19, pages 202–208, 2004.
- [Luciani 84] A. Luciani & C. Cadoz. *Modélisation et animation gestuelle d'objets - Le système ANIMA*. Rapport technique, ACROE, 1984.
- [Luciani 91] A. Luciani, S. Jimenez, C. Cadoza, J.-L. Florens & O. Raoult. *An Unified View of Multitude Behaviour, Flexibility, Plasticity and Fractures Balls, Bubbles and Agglomerates*. *Modeling in Computer Graphics (Proc of IFIP WG 5.10 Working Conference)*, 1991.
- [Marchal 06a] M. Marchal, M. Chabanas, Y. Payan, C. Marçaux, P. Swider, F. Boutault, E. Promayon & J. Troccaz. *Comparison Of Different Soft Tissue Modelling Methods With Post-Operative CT Scan In Maxillofacial Surgery*. In *7th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering, CMBBE06*, March 2006.
- [Marchal 06b] Maud Marchal. *Modélisation des tissus mous dans leur environnement pour l'aide aux gestes médico-chirurgicaux*. PhD thesis, University Joseph Fourier, Grenoble, 2006.
- [Marescaux 01] Jacques Marescaux, Joel Leroy, Michel Gagner, Francesco Rubino, Didier Mutter, Michel Vix, Steven E. Butner & Michelle K. Smith. *Transatlantic Robot-Assisted Telesurgery*. In *Nature*, numéro 413 :379-380, 2001.
- [Matyka 03] M. Matyka & M Ollila. *A pressure model for soft body simulation*. In *Sigrad*, 2003.
- [Maya] Autodesk Maya. <http://www.autodesk.fr/maya>.
- [McCormick 89] Stephen F. McCormick. *Multilevel adaptive methods for partial differential equations*. SIAM. 1989.
- [Mendoza 05] C. Mendoza & C. O'Sullivan. *Towards Time-Critical Collision Detection for Deformable Objects Based on Reduced Models*. In *CASA*, 2005.
- [Meseure 03] P. Meseure, J. Davanne, L. Hilde, J. Lenoir, L. France, F. Triquet & C. Chaillou. *A Physically-Based Virtual Environment dedicated to Surgical Simulation*. In *Surgery Simulation and Soft Tissue Modeling (IS4TM)*, pages 38–47, 2003.
- [Meyer 98] Alexandre Meyer & Fabrice Neyret. *Interactive Volumetric Textures*. In *Rendering Techniques (Eurographics Workshop on Rendering - EGSR)*, pages 157–168, Jul 1998.
- [Meyer 02] M. Meyer, M. Desbrun, P. Schröder & A. Barr. *Discrete Differential Geometry Operators for Triangulated 2-Manifolds*. In *VisMath*, 2002.

- [Mezger 06] Johannes Mezger & Wolfgang Straßer. *Interactive Soft Object Simulation with Quadratic Finite Elements*. In Lecture Notes on Computer Science (Proc of IV Conference on Articulated Motion and Deformable Objects (AMDO)), volume 4069, pages 434–443, 2006.
- [Miller 89] Gavin Miller & Andrew Pearce. *Globular Dynamics : A Connected Particle System for Animating Viscous Fluids*. Computers and Graphics, vol. 13, no. 3, pages 305–309, 1989.
- [Miller 00] K. Miller, K. Chinzei, G. Orssengo & P. Bednarz. *Mechanical properties of brain tissue in-vivo : experiment and computer simulation*. Journal of Biomechanics, vol. 33, pages 1369–1376, 2000.
- [Miller 01] Karol Miller. *How to test very soft biological tissues in extension ?* J. Biomechanics, vol. 34/5, 2001.
- [Miller 05] Karol Miller. *Method of testing soft biological tissues in compression*. J. Biomechanics, vol. 38, 2005.
- [Miller 06] K.S. Miller, G.R. Joldes & Wittek. *A New Finite Element Algorithm for Surgical Simulation*. In Workshop on Computer Assisted Diagnosis and Surgery, pages 1–4, 2006.
- [Miller 07] Karol Miller, Grand Joldes, Dane Lance & Adam Wittek. *Total Lagrangian Explicit Dynamics Finite Element Algorithm for Computing Soft Tissue Deformation*. Communications in Numerical Methods in Engineering, vol. 23, pages 121–134, 2007.
- [Molino 04] Neil Molino, Zhaosheng Bao & Ron Fedkiw. *A Virtual Node Algorithm for Changing Mesh Topology During Simulation*. In SIGGRAPH, volume 23, pages 385–392, 2004.
- [Monserrat 01] 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.
- [Mor 00] Andrew B. Mor & Takeo Kanade. *Modifying Soft Tissue Models : Progressive Cutting with Minimal New Element Creation*. In MICCAI '00 : Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 598–607, 2000.
- [Myszkowski 95] K. Myszkowski, O. Okunev & T. Kunii. *Fast collision detection between complex solids using rasterizing graphics hardware*. In The Visual Computer, volume 11(9), pages 497–512, 1995.
- [Müller 02] M. Müller, J. Dorsey, L. McMillan, R. Jagnow & B. Cutler. *Stable real-time deformations*. In Proc SCA, 2002.
- [Müller 03] M. Müller, D. Charypar & M. Gross. *Particle-Based Fluid Simulation for Interactive Applications*. In Symposium on Computer Animation, pages 154–159, 2003.
- [Müller 04a] M. Müller & M. Gross. *Interactive Virtual Materials*. In Proc Graphics Interface, 2004.
- [Müller 04b] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross & M. Alexa. *Point based animation of elastic, plastic and melting objects*. In Proc SCA, pages 141–151, 2004.



- [Müller 04c] M. Müller, M. Teschner & M. Gross. *Physically based simulation of objects represented by surface meshes*. In Proc SCA, 2004.
- [Müller 05] M. Müller, B. Heidelberger, M. Teschner & M. Gross. *Meshless Deformations Based on Shape Matching*. In Proc SIGGRAPH'05, pages 471–478, July 2005.
- [Müller 06] M. Müller, B. Heidelberger, M. Hennix & J. Ratcliff. *Position Based Dynamics*. In Proceedings of Virtual Reality Interactions and Physical Simulations (VRIPhys), pages 71–80, 2006.
- [Nealen 05] A. Nealen, M. Müller, R. Keiser, E. Boxerman & M. Carlson. *Physically Based Deformable Models in Computer Graphics*. In Computer Graphics Forum, volume 25 (4), pages 809–836, 2005.
- [Nesme 05a] M. Nesme, Y. Payan & F. Faure. *Efficient, Physically Plausible Finite Elements*. In Eurographics (short papers), pages 77–80, August 2005.
- [Nesme 05b] Matthieu Nesme, Maud Marchal, Emmanuel Promayon, Matthieu Chabanas, Yohan Payan & François Faure. *Physically Realistic Interactive Simulation for Biological Soft Tissues*. Recent Research Developments in Biomechanics, vol. 2, 2005.
- [Nesme 06a] Matthieu Nesme & Antoine Bouthors. *Dynamic Triangulation of Implicit Surfaces : towards the handling of topology changes*. Rapport technique RR-6128, INRIA, 2006.
- [Nesme 06b] Matthieu Nesme, François Faure & Yohan Payan. *Hierarchical Multi-Resolution Finite Element Model for Soft Body Simulation*. Lecture Notes in Computer Science (Proc. of Symposium on Biomedical Simulation), 2006.
- [Nesme 06c] Matthieu Nesme, Yohan Payan & François Faure. *Animating Shapes at Arbitrary Resolution with Non-Uniform Stiffness*. In Eurographics Workshop in Virtual Reality Interaction and Physical Simulation (VRIPHYS), 2006.
- [O'Brien 99] James F. O'Brien & Jessica K. Hodgins. *Graphical modeling and animation of brittle fracture*. In SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 137–146, 1999.
- [O'Brien 02] James F. O'Brien, Adam W. Bargteil & Jessica K. Hodgins. *Graphical modeling and animation of ductile fracture*. ACM Trans. Graph., vol. 21, no. 3, pages 291–294, 2002.
- [Ohayon 00] Jacques Ohayon. *Mécanique des milieux continus et méthodes des éléments finis*, 2000.
- [Ophir 91] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yasdi & X. Li. *Elastography : A Quantitative Method for Imaging the Elasticity of Biological Tissues*. Ultrasonic imaging, vol. 13, pages 186–210, 1991.
- [Otaduy 07] Miguel A. Otaduy, Daniel Germann, Stéphane Redon & Markus Gross. *Adaptive deformations with fast tight bounds*. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer animation, pages 181–190, 2007.
- [Ottensmeyer 04] M.P. Ottensmeyer, A.E. Kerdok, R.D. Howe & S.L. Dawson. *The effects of testing environment on the viscoelastic properties of soft tissues*. Second International Symposium on Medical Simulation, pages 9–18, 2004.

- [Paul 92] H. A. Paul, W. L. Bargar & B. Mittelstadt. *Development of a surgical robot for cementless total hip replacement*. Clin Orthop, vol. 57, page 285, 1992.
- [Pauly 05] Mark Pauly, Richard Keiser, Bart Adams, Philip Dutré, Markus Gross & Leonidas J. Guibas. *Meshless animation of fracturing solids*. ACM Trans. Graph., vol. 24, no. 3, pages 957–964, 2005.
- [Pauwels 89] F. Pauwels. *Biomécanique de l'appareil locomoteur*. Berlin. Springer, 1989.
- [Pentland 89] A. Pentland & J. Williams. *Good vibrations : modal dynamics for graphics and animation*. In ACM SIGGRAPH, volume 23 (3), pages 215–222, 1989.
- [Phai 82] Nguyen Van Phai. *Automatic mesh generation with tetrahedral elements*. In International Journal for Numerical Methods in Engineering, volume 18, pages 273–289, 1982.
- [PhysX] NVIDIA PhysX. [http://www.nvidia.com/object/nvidia\\_physx.html](http://www.nvidia.com/object/nvidia_physx.html).
- [Picinbono 00a] G. Picinbono, H. Delingette & N. Ayache. *Improving Realism of a Surgery Simulator*. Rapport technique, INRIA, 2000.
- [Picinbono 00b] Guillaume Picinbono, Hervé Delingette & Nicholas Ayache. *Real-Time Large Displacement Elasticity for Surgery Simulation : Non-linear Tensor-Mass Model*. MICCAI, pages 643–652, 2000.
- [Picinbono 03] G. Picinbono, H. Delingette & N. Ayache. *Non-linear anisotropic elasticity for real-time surgery simulation*. Graph. Models, 2003.
- [Platt 88] J. C. Platt & A. H. Barr. *Constraint Methods for Flexible Models*. In In Computer Graphics (SIGGRAPH '88), volume 22 (4), pages 279–288, 1988.
- [Press 92] Press, Teukolski, Vetterling & Flannery. *Numerical recipes in c*. Cambridge University Press, 1992.
- [Promayon 96] E. Promayon, P. Baconnier & C. Puech. *Physically-Based Deformations Constrained in Displacements and Volume*. Computer Graphics Forum, Eurographics96, vol. 15, no. 3, pages 155–164, August 1996.
- [Provot 95] Xavier Provot. *Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior*. In Graphics Interface, pages 147–154. Canadian Human-Computer Communications Society, 1995.
- [Raghupathi 03] L. Raghupathi, V. Cantin, F. Faure & M.P. Cani. *Real-time Simulation of Selfcollisions for Virtual Intestinal Surgery*. In Int. Symposium on Surgery Simulation and Soft Tissue Modeling, volume 2673, pages 15–26, 2003.
- [Rivers 07] Alec R. Rivers & Doug L. James. *FastLSM : Fast Lattice Shape Matching for Robust Real-Time Deformation*. In ACM Transactions on Graphics (Proc of SIGGRAPH), 2007.
- [Rohlman 81] A. Rohlman, G. Bergman & R. Kolbel. *The relevance of stress computation in the femur with and without endo-prostheses*. Gallacher RH, vol. 2, pages 361–372, 1981.
- [Rust 72] Bert W. Rust & Walter R Burrus. *Mathematical programming and the numerical solution of linear equations*, volume 38 of *Modern Analytic and Computational Methods in Science and Mathematics*. American Elsevier Publishing Company, 1972.

- [Salençon 05] Jean Salençon. *Mécanique des milieux continus - Tome 1 - Concepts généraux*, 2005.
- [Sandrin 02] L. Sandrin, M. Tanter, J.L. Gennisson, S.Catheline & M. Fink. *Shear Elasticity Probe for Soft Tissue with 1D Transient Elastography*. IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, vol. 49, pages 436–446, 2002.
- [Schneider 03] Philip Schneider & David H. Eberly. *Geometric tools for computer graphic*. Morgan Kaufmann Publishers, Inc., 2003.
- [Schwartz 01] Jean-Marc Schwartz, Ève Langelier, Christian Moisan & Denis Laurendeau. *Non-linear Soft Tissue Deformations for the Simulation of Percutaneous Surgeries*. In Lecture Notes on Computer Science, Proc. MICCAI, volume 2208, pages 1271–1272, 2001.
- [Sederberg 86] T. W. Sederberg & S. R. Parry. *Free-Form Deformation of Solid Geometric Models*. In Proc SIGGRAPH'86, pages 151–159, 1986.
- [Shade 98] Jonathan W. Shade, Steven J. Gortler, Li-Wei He & Richard Szeliski. *Layered Depth Images*. Computer Graphics, vol. 32, pages 231–242, 1998.
- [Shi 06] Lin Shi, Yizhou Yu, Nathan Bell & Wei-Wen Feng. *A fast multigrid algorithm for mesh deformation*. In ACM Transaction on Graphics, volume 25(3), pages 1108–1117, 2006.
- [Sifakis 07a] E. Sifakis, K. Der & R. Fedkiw. *Arbitrary Cutting of Deformable Tetrahedralized Objects*. In Symposium on Computer Animation, pages 73–80, 2007.
- [Sifakis 07b] Eftychios Sifakis, Tamar Shinar, Geoffrey Irving & Ronald Fedkiw. *Hybrid simulation of deformable solids*. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 81–90, 2007.
- [SOFA] SOFA *Simulation Open Framework Architecture*. <http://www.sofa-framework.org>.
- [SPRING] SPRING *Framework for Building Networked Surgical Simulators*. <http://spring.stanford.edu>.
- [Stollnitz 96] Eric J. Stollnitz, Tony D. DeRose & David H. Salesin. *Wavelets for Computer Graphics : Theory and Applications*. Morgan Kaufmann Publishers, Inc., 1996.
- [Sukumar 00] N. Sukumar, N. Moës, B. Moran & T. Belytschko. *Extended Finite Element Method for Three-Dimensional Crack Modeling*. In International Journal for Numerical Methods in Engineering, volume 48 (11), pages 1549–1570, 2000.
- [Sundaraj 02] K. Sundaraj, C. Mendoza & C. Laugier. *A fast method to simulate virtual deformable objects with force feedback*. In Control, Automation, Robotics and Vision (ICARCV), volume 1, pages 413–418, 2002.
- [Szekely 00] G. Szekely, Ch. Brechbuhler, R. Hutter, A. Rhomberg & P. Schmid. *Modelling of soft tissue deformation for laparoscopic surgery simulation*. In Medical Image Analysis, volume 4, pages 57–66, 2000.
- [Taylor 96] R. Taylor, S. Lavallée, G. Burdea & R. Mosges. *Computer Integrated Surgery : Technology and clinical applications*. Rapport technique, Cambridge, MA : MIT Press, 1996.

- [Terzopoulos 87] Demetri Terzopoulos, John Platt, Alan Barr & Kurt Fleischer. *Elastically deformable models*. In SIGGRAPH '87, 1987.
- [Terzopoulos 88a] Demetri Terzopoulos & Kurt Fleischer. *Modeling inelastic deformation : viscoelasticity, plasticity, fracture*. In SIGGRAPH '88, 1988.
- [Terzopoulos 88b] Demetri Terzopoulos & Andrew Witkin. *Physically Based Models with Rigid and Deformable Components*. IEEE Comput. Graph. Appl., vol. 8, no. 6, pages 41–51, 1988.
- [Terzopoulos 91] Demetri Terzopoulos, John Platt & Kurt Fleischer. *Heating and melting deformable models*. The Journal of Visualization and Computer Animation, vol. 2, no. 2, pages 68–73, 1991.
- [Teschner 05] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, Laks Raghupathi, A. Fuhrmann, Marie-Paule Cani, François Faure, N. Magnenat-Thalmann, W. Strasser & P. Volino. *Collision Detection for Deformable Objects*. Computer Graphics Forum, vol. 24, no. 1, pages 61–81, 2005.
- [TetGen] TetGen. *A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*. <http://tetgen.berlios.de>.
- [Theret 88] D.P. Theret, M.J. Levesque, M. Sato, R.M. Nerem & L.T. Wheeler. *The application of a homogeneous half-space model in the analysis of endothelial cell micropipette measurements*. Journal of Biomechanics Engineering, vol. 110, pages 190–199, 1988.
- [Tsingos 07] Nicolas Tsingos, Carsten Dachsbacher, Sylvain Lefebvre & Matteo Dellepiane. *Instant Sound Scattering*. In Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering), 2007.
- [van den Doel 01] Kees van den Doel, Paul G. Kry & Dinesh K. Pai. *FoleyAutomatic : physically-based sound effects for interactive simulation and animation*. In SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 537–544, 2001.
- [Viceconti 04] Marco Viceconti, Mario Davinelli, Fulvia Taddei & Angelo Cappello. *Automatic generation of accurate subject-specific bone finite element models to be used in clinical studies*. In Journal of Biomechanics, volume 37, pages 1597–1605, 2004.
- [Vigneron 04] L.M. Vigneron, J. G. Verly & S. K. Warfield. *Modelling Surgical Cuts, Retractions, and Resections via Extended Finite Element Method*. In Lecture Notes in Computer Science, volume 3217, pages 311–318, 2004.
- [Volino 95] Pascal Volino, Martin Courchesne & Nadia Magnenat Thalmann. *Versatile and efficient techniques for simulating cloth and other deformable objects*. In SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages 137–144, 1995.
- [Volino 01] P. Volino & N. Magnenat-Thalmann. *Comparing Efficiency of Integration Methods for Cloth Animation*. In Proceedings of IEEE Computer Graphics International (CGI), pages 265–274, 2001.
- [Volino 05] P. Volino & N. Magnenat-Thalmann. *Implicit Midpoint Integration and Adaptive Damping for Efficient Cloth Simulation*. In Computer Animation and Virtual Worlds, volume 16 (3-4), pages 163–175, 2005.

- [VRASS] VRASS *Simulation Platform*. [http://www.kuhp.kyoto-u.ac.jp/~mi/en/index.php?\\_research\\_\\_med\\_vr](http://www.kuhp.kyoto-u.ac.jp/~mi/en/index.php?_research__med_vr).
- [Waters 90] K. Waters & D. Terzopoulos. *A Physical Model of Facial Tissue and Muscle Articulation*. IEEE, 1990.
- [Watson 81] D. F. Watson. *Computing the n-dimensional Delaunay-tessellation with application to Voronoi polytopes*. In *Computer Journal*, volume 24, pages 167–171, 1981.
- [Witkin 90] A. Witkin & W. Welch. *Fast animation and control for non-rigid structures*. In *ACM SIGGRAPH*, volume 24 (4), pages 243–252, 1990.
- [Wu 01] Xunlei Wu, Michael S. Downes, Tolga Goktekin & Frank Tendick. *Adaptive Nonlinear Finite Elements for Deformable Body Simulation Using Dynamic Progressive Meshe*. In *Computer Graphics Forum (Proc. of EG)*, volume 20(3), pages 349–358, 2001.
- [Wu 04] Xunlei Wu & Frank Tendick. *Multigrid Integration for Interactive Deformable Body Simulation*. In *ISMS*, pages 92–104, 2004.
- [Yerry 84] M. A. Yerry & M. S. Shephard. *Automatic three-dimensional mesh generation by the modified-octree technique*. In *International Journal for Numerical Methods in Engineering*, volume 20, pages 1965–1990, 1984.
- [Young 87] Ronald F. Young. *Application of robotics to stereotactic neurosurgery*. *Neurological Research*, vol. 9, pages 123–128, 1987.
- [Zara 03] Florence Zara, Jean-Marc Vincent & François Faure. *Coupling Parallel Simulation and Parallel Visualization on PC Clusters*. In *Commodity Clusters For Virtual Reality (VR)*, 2003.
- [Zara 04] Florence Zara, François Faure & Jean-Marc Vincent. *Parallel Simulation of Large Dynamic System on a PCs Cluster : Application to Cloth Simulation*. *International Journal of Computers and Applications*, 2004.
- [Zhou 03] Jianlong Zhou & Klaus D. Tonnies. *State of The Art for Volume Rendering*. Rapport technique, Institute for Simulation and Graphics, University of Magdeburg, Germany, 2003.
- [Zienkiewicz 67] O. C. Zienkiewicz & Y. K. Cheung. *The finite element method in structural and continuum mechanics*. McGraw-Hill Publ, 1967.
- [Zienkiewicz 82] O.C. Zienkiewicz, D.W. Kelly, J. Gago & I. Babuska. *Hierarchical Finite Element Approaches, Error Estimates and Adaptive Refinement*. *Mathematics of Finite Elements and Applications*, vol. 4, pages 313–346, 1982.



---

# Annexes





# Conditionnement d'un système linéaire

## 1 Définition

Un système linéaire  $\mathbf{Ax} = \mathbf{b}$  est dit "bien conditionné" lorsque de faibles variations sur les entrées du système (composantes de la matrice  $\mathbf{A}$  et/ou du vecteur  $\mathbf{b}$ ) entraînent de faibles variations sur la solution du système. Inversement, un système mal conditionné est susceptible de fournir une solution dont l'erreur est largement supérieure à celle apportée par les variations de  $\mathbf{A}$  ou  $\mathbf{b}$ . Le conditionnement caractérise la stabilité de la solution d'un système linéaire. Sa mesure, le nombre de conditionnement, est définie par

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$$

où  $\|\cdot\|$  est une norme matricielle.

Par exemple avec une norme matricielle euclidienne, on obtient

$$\kappa_2(\mathbf{A}) = \frac{w_{max}}{w_{min}}$$

où  $w_{max}$  et  $w_{min}$  sont respectivement les valeurs singulières maximale et minimale de  $\mathbf{A}$ .  $\kappa_2(\mathbf{A}) \geq 1.0$  et  $\mathbf{A}$  est d'autant mieux conditionnée que  $\kappa_2(\mathbf{A})$  est proche de 1, et d'autant moins bien conditionné que  $\kappa_2(\mathbf{A})$  est grand.

## 2 Conséquence

Pour une solution approchée  $\tilde{\mathbf{x}}$ , l'erreur est  $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$  et le résidu est  $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$ . L'inégalité suivante est vérifiée [Rust 72] :

$$\frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq \kappa(\mathbf{A}) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}$$

Si le conditionnement est bon, le résidu est une bonne estimation de l'erreur, en revanche, un résidu faible ne garantit pas une petite erreur pour un grand nombre de conditionnement.

## 3 Conditionnement du système implicite en fonction du matériau

On peut analyser le conditionnement de la matrice du système implicite en fonction des propriétés du matériau simulé (avec un objet composé d'un matériau uniforme). Dans notre contexte, la matrice du système de l'intégration d'Euler implicite [Baraff 98] considérée est  $\mathbf{M} - h^2\mathbf{K}$  avec  $h$  le pas de temps et sans amortissement de Rayleigh. La matrice de raideur  $\mathbf{K}$  est construite pour un matériau hookéen, défini par le coefficient de Poisson  $\nu$  et le module d'Young  $E$ . La matrice de masse  $\mathbf{M}$  dépend de la masse volumique de l'objet  $\rho$ . Nous avons testé la variation du conditionnement toujours

$h = 0.04$		$E$					
$\rho = 200$		100	1000	5000	10000	25000	50000
$\nu$	0	113370	30666	19037	70038	115600	218370
	0.1	36361	75086	13129	76790	226770	296990
	0.3	1037300	15494	14564	136340	65395	784520
	0.4	44590	702110	23712	239760	87721	1457300
	0.45	88061000	2513100	43103	343750	148690	1997900
	0.49	291150	517230	205210	668370	690890	2899000

$h = 0.04$		$E$					
$\rho = 1000$		100	1000	5000	10000	25000	50000
$\nu$	0	105950	36205	29094	83833	25991	52540
	0.1	83481	83211	58074	73995	98642	21930
	0.3	240530	24308	59574	1108700	1253600	47868
	0.4	400750	251960	70488	246950	107220	69776
	0.45	132370	453860	50234	43950000	82478	205920
	0.49	225350	310230	384700	422560	4276900	314880

$h = 0.01$		$E$					
$\rho = 1000$		100	1000	5000	10000	25000	50000
$\nu$	0	79603	36205	23220	83833	25991	52540
	0.1	83481	83211	58074	73995	99603	22252
	0.3	240530	29639	59574	1108700	1253600	47868
	0.4	400750	251960	94926	246950	107220	69776
	0.45	129670	453860	50234	43950000	82478	205920
	0.49	225350	310230	502470	422560	4276900	314880

**FIG. A.1** – Analyse du conditionnement du système implicite en fonction des paramètres de matériau et du pas de temps

Basée sur la norme matricielle euclidienne. Prise en compte d'un jeu de coefficients de Poisson  $\nu$  et de modules d'Young  $E$  pour deux masse volumique  $\rho$  et deux pas de temps  $h$ .

pour le même maillage hexaédrique régulier subdivisant une poutre en  $10 \times 10 \times 10$ , revenant au même type de maillage que les hexaèdres englobants utilisés dans notre méthode. Les résultats sont donnés en figure A.1.

On peut conclure que le conditionnement du système est très variable suivant les paramètres utilisés. Les matériaux ont une grande influence. Grossièrement certains couples raideur/incompressibilité vont mieux que d'autres. Il est assez difficile de tirer des conclusion sur un seul paramètre indépendamment. Le but de cette analyse est d'affirmer que pour un même objet, suivant le matériau simulé, les résultats vitesse/précision peuvent être très différents. Par abus de langage, nous qualifions les matériaux résultants en un système mal conditionné de *matériaux mal conditionnés*.

## Informatique et médical

Le monde médical est une source de motivation importante pour un grand nombre de domaines scientifiques, y compris l'informatique. Les nouvelles technologies y ont apporté une contribution très importante, si bien qu'aujourd'hui les outils informatiques sont devenus indispensables à la médecine moderne. Les appareils d'imagerie permettent d'observer l'intérieur d'un corps sans le traumatiser. Les cliniciens peuvent disposer de structures anatomiques, voire fonctionnelles de leurs patients. Ces grands progrès en imagerie médicale permettent aujourd'hui de disposer "relativement facilement" d'une grande quantité de données.

Mais l'informatique médicale n'est pas uniquement cantonnée à un rôle de capture de données. Elle peut par exemple apporter une aide à la formation, car elle permet de stocker et de gérer de grosses quantités d'informations virtuelles, très pratiques dans l'apprentissage de l'anatomie. On essaie maintenant de faire entrer l'ordinateur au plus proche de l'intervention, par exemple dans des systèmes de guidage aidant le chirurgien à atteindre une cible en neurochirurgie [Young 87, Benabid 87, Kwoh 88] ou à directement effectuer l'opération en chirurgie de la hanche [Paul 92]. On a même vu la possibilité de faire des opérations à distance. Les gestes du chirurgien sont analysés puis transmis à travers un réseau informatique à un télémanipulateur en contact avec le patient qui réalise l'intervention chirurgicale [Marescaux 01].

Ainsi sont nés les gestes chirurgicaux assistés par ordinateur. Ils ont pour objectifs d'aider les médecins et les chirurgiens à planifier et réaliser des procédures médicales ou chirurgicales. La méthodologie introduite dans [Lavallée 89] repose sur une boucle de trois étapes : perception, décision et action. La réalisation d'un système GMCAO repose donc sur la capacité à acquérir, traiter, fusionner des données multimodales provenant d'examens sur le patient, d'atlas préétablis ou de modèles numériques. Dans cette thèse, nous ciblons la partie décisionnelle, l'objectif étant de permettre la création d'applications visant à aider le chirurgien à profiter de son expérience pour définir la stratégie optimale à partir de données relatives au patient. Pour cela, il faut aider le chirurgien à spécifier les étapes et les gestes à réaliser pour son intervention, puis à anticiper les conséquences de ces gestes. La mise au point de telles planifications est depuis quelques années un domaine de recherche en pleine effervescence, dans le but de prévoir les conséquences d'une intervention, afin d'optimiser et de planifier la meilleure stratégie opératoire.

Toute une famille de ces aides informatiques s'essaie à modéliser physiquement les différentes structures anatomiques d'un patient. L'intégration de la physique pourrait permettre d'élargir considérablement le champs d'application de l'informatique dans le monde médical. On peut se prendre à rêver à de multiples applications. Par exemple, des simulateurs calculant les points de coupures les plus adaptés à un patient en particulier, ou encore des chirurgiens s'entraînant à l'avance sur une opération à effectuer plus tard, totalement immergés dans un environnement virtuel avec les modèles individualisés du patient à traiter.

Ce genre d'applications ne demande plus seulement une représentation visuelle du patient, mais aussi sa modélisation physique. Cette modélisation physique du corps a débuté par la simulation des parties solides, tels que les os [Rohlman 81, Pauwels 89]. On essaie d'étendre ces techniques aux

parties déformables telles que la peau ou les organes. Mais ces corps mous sont très complexes à synthétiser, de part leurs propriétés mécaniques, la complexité de leurs déformations et de leurs interactions. Ces modélisations sont aussi calculatoirement très coûteuses et se pose un réel problème lié au temps nécessaire pour obtenir le résultat d'une simulation.

Il faudra aussi être capable de modéliser d'autres lois de la physique, telles que les couplages avec des liquides (comme le sang) et pourquoi pas la thermodynamique...



**FIG. B.1** – Laparoscopie

*gauche : champ opératoire d'une laparoscopie. Des outils et une caméra sont insérés dans la patient. Le chirurgien n'interagit qu'indirectement, avec des gestes complexes et une vision limitée à un écran. (Leids Universitair medisch centrum) - droite : une idée d'un simulateur de laparoscopie. Des outils à retour d'effort commandent des organes virtuels affichés sur un écran (Laparoscopy Hospital)*

Les évolutions technologiques ont modifié les pratiques médicales, et la chirurgie minimalement invasive est de plus en plus employée. Elle permet de réduire la taille des voies d'abord et ainsi de diminuer le traumatisme subi par le patient, ainsi que son temps de rétablissement. L'intérêt de ce genre d'opérations est évident pour le patient, le contre-coup touche le praticien, car elles lui compliquent la tâche en diminuant ses capacités de perception et d'exécution. En effet, l'accès aux éléments internes par des voies naturelles ou des trocarts (petits trous percés dans l'abdomen) réduit la perception visuelle (qui ne se fait plus qu'en deux dimensions, avec une caméra et un écran à une qualité relative) et tactile (les outils sont longs, et le touché moins subtil) et limite les mouvements du chirurgien qui ne voit plus et ne touche plus directement les parties opérées. Ce genre d'intervention est appelé laparoscopie ou cœlioscopie. Un liché de laparoscopie est présenté en figure B.1-gauche. Il semble donc utile d'assister le chirurgien, soit en l'aidant à acquérir un plus grand savoir faire, soit en lui proposant les solutions qui semblent les plus valables, voire carrément en guidant ses gestes en temps réel.

Ces techniques demandent un grand savoir faire, les gestes sont très complexes et l'expérience d'une importance capitale, mais l'entraînement n'est pas des plus aisés. Ce dernier se fait actuelle-

ment sur des cadavres, mais si l'anatomie est identique, les comportements ne sont bien sûr pas les mêmes que sur un patient vivant. Pour une meilleure réactivité, il est possible d'opérer des cochons, mais dans ce cas, si les mouvements peuvent mieux tester la dimension dynamique, l'anatomie n'est pas strictement la même qu'un être humain. De plus, peu de centres élèvent de tels animaux dédiés à ces procédures d'entraînement. Le coût est très élevé, et d'évidents problèmes d'éthique sont soulignés. Il existe aussi des phantoms artificiels, qui reproduisent synthétiquement un abdomen, mais là encore, même si une première appréhension du matériel peut être révélée, les comportements sont très différents des tissus vivants. De plus ces phantoms en matière plastique reviennent chers, car il faut les remplacer dès qu'ils sont découpés, sans parler des problèmes de pollution engendrés. Il semble donc intéressant de trouver un nouveau moyen de formation utilisant les moyens informatiques.

Comme le praticien ne touche plus directement le patient, mais manipule des outils et regarde un écran de télévision, on peut simuler l'opération avec un ordinateur remplaçant le patient, un rendu d'une scène virtuelle à la place de la caméra et des outils à retour de force pour interagir. Un exemple est présenté en figure B.1-droite. Une telle solution peut clairement offrir une alternative très intéressante au problème de l'entraînement. Cela demande peu de matériel non destructible avec lequel on peut réaliser autant d'entraînements que l'on veut, pour un coût très réduit. Les erreurs n'ont aucune conséquence, et un aspect didactique virtuel peut même être directement relié aux erreurs. Mieux, différentes techniques peuvent être employées sur un même problème, afin de déterminer la technique optimale. Les compétences peuvent aussi être évaluées, ainsi que la capacité à réaliser un certain type d'opération. Pour aller plus loin, l'entraînement peut être très spécifique, sur un problème particulier très rare sur lequel il est impossible de s'entraîner avec les moyens actuels et que l'on rencontre en cas réel dès la première fois. L'idéal serait de s'entraîner sur des modèles individualisés d'un patient, pour faire son opération à l'avance.

Il est important de savoir précisément quel type de simulateurs on désire réaliser afin d'en déduire un modèle physique adapté. En effet, un simulateur mécanique du corps pourrait permettre une grande quantité d'applications médicales qu'on peut classer dans les familles suivantes aux besoins techniques variés :

- les planifications (pré-opératoires),
- les re-planifications (per-opératoires),
- les planifications en urgence,
- les simulateurs d'entraînement,
- les simulateurs d'entraînement dépendants patient,
- les simulations per-opératoires.

Les **planificateurs pré-opératoires** permettent de tester et d'optimiser des stratégies opératoires avant l'opération. L'idée est de déterminer une configuration d'équilibre en réponse aux contraintes appliquées au système. Les planificateurs actuels disposent de modèles mécaniques "précis" mais complexes et lents, requérant de nombreux calculs. Leur précision est très importante puisqu'on va énormément calquer l'intervention réelle sur ces résultats. Les planifications étant effectuées sur des données d'un patient avant opération, le jour J, une multitude de paramètres ont pu varier, tels que des organes se sont déplacés, la taille de la vessie qui a changé... Il est dans ce cas nécessaire d'effectuer une **re-planification** en prenant en compte ces variations. Pour ce faire, il faut être capable d'identifier les changements (étape de perception [Lavallée 89]), étape difficile car souvent multimodale, puisque généralement l'échographie peu onéreuse est utilisée durant l'opération, alors que le scanner ou l'IRM sont plutôt utilisés pour les planifications. Puis, à partir de ces changements il faut être en

mesure de réévaluer une nouvelle solution dans un temps assez court, le patient étant déjà sur le lit d'opération. Les planificateurs actuels sont trop lents pour permettre ce genre de réévaluation, ce qui compromet fortement leur utilisation en cas réel. Dans le **cas d'une urgence**, il faut en plus être capable de synthétiser le corps spécifique du patient dans un laps de temps très court et d'être capable de fournir plusieurs solutions très rapidement, pour évaluer la meilleure ; donc de disposer d'un modèle relativement rapide.

Pour les **simulateurs d'entraînement**, la principale difficulté se trouve au niveau du modèle mécanique, qui doit être très rapide pour permettre l'interactivité. Il doit aussi être très stable, c'est-à-dire, ne jamais exploser, même sur des sollicitations extrêmes ou non-physiques comme peut en imposer un utilisateur. Dans le cas de simulateurs d'entraînement aux gestes chirurgicaux en général, la précision n'est pas obligatoirement de rigueur, un comportement physiquement réaliste peut-être suffisant. On entend par là, "cela pourrait se passer comme ça", et non pas, "cela se passerait exactement comme ça". De plus, il n'y a pas besoin d'avoir des modèles individualisés, alors que l'entraînement à une opération précise demandera bien sûr un modèle individualisé du patient avec une grande précision du modèle mécanique. Un simulateur précis, rapide et stable pourrait même servir à guider le chirurgien au cours de l'intervention, dans le cas où la visibilité est très limitée et où tout peut se modifier, comme par exemple dans le cas des biopsies ou curiethérapies de la prostate, pour lesquelles le praticien ne dispose généralement que d'un échographe qui limite énormément la compréhension et le positionnement alors que la prostate se déforme et bouge beaucoup [Marchal 06b]. Si on est capable de mettre en entrée ses mouvements réels, de se recalculer avec une technique d'imagerie pré-opératoire, on pourrait en déduire leurs influences sur le modèle virtuel et ainsi fournir au chirurgien un support visuel pour l'aider dans sa gestuelle, l'aider à se localiser précisément...

Les simulations chirurgicales semblent donc être très utiles pour la formation et l'aide aux gestes chirurgicaux. Malheureusement, les techniques à mettre en œuvre pour les réaliser sont très complexes et demandent beaucoup d'avancées dans un grand nombre de domaines (simulation physique, rendu réaliste, retour haptique, matériel spécifique...). Ne serait-ce au niveau de la simulation des parties molles, qui nous concerne particulièrement, il est important de savoir ce qu'on veut faire exactement, car même si tous les simulateurs demandent de simuler des corps déformables, les propriétés à mettre en œuvre sont complètement différentes et requièrent des méthodes variées suivant les besoins. Les principaux besoins sont résumés dans le tableau B.1.

type application	construction rapide d'un maillage spécifique au patient	simulation		
		vitesse	précision	robustesse
planification	+	+	+++	+
re-planification	+	++	+++	+
planification en urgence	+++	++	+++	+
simulateur d'entraînement	-	+++	+	+++
simulateur d'intervention	+	+++	+++	+++
simulateur per-opératoire	+	+++	+++	+++

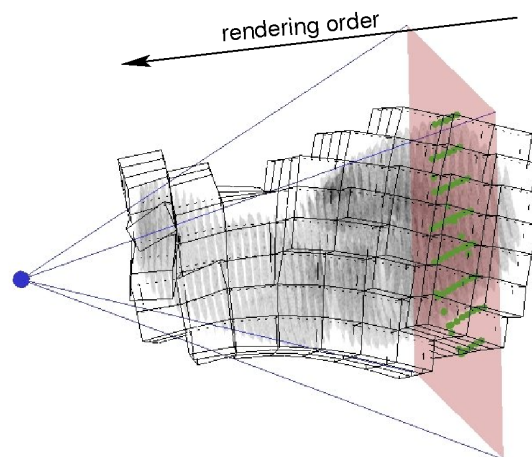
- : pas nécessaire, + : pas crucial, ++ : important, +++ : crucial

**TAB. B.1:** Besoins des différents types d'application

## Rendu volumique adapté à la structure hexaédrique

Plusieurs techniques pour afficher des données volumiques ont été élaborées, un état de l'art complet peut être trouvé dans [Zhou 03]. Nous proposons de suivre l'approche classique permettant un rendu interactif en utilisant une texture 3D avec l'accélération matérielle de mapping des coordonnées de texture [Meyer 98, Lensch 02]. L'idée consiste à couper le volume en plusieurs tranches pour déduire l'intensité du pixel affiché à l'écran. Ces tranches sont des polygones semi-transparents sur lesquels la texture 3D est appliquée. Différentes solutions existent pour choisir l'orientation des tranches, car des artefacts peuvent être introduits pour certains angles de vue. Une bonne qualité est obtenue lorsque les tranches sont choisies comme étant toujours orthogonales à la direction du regard.

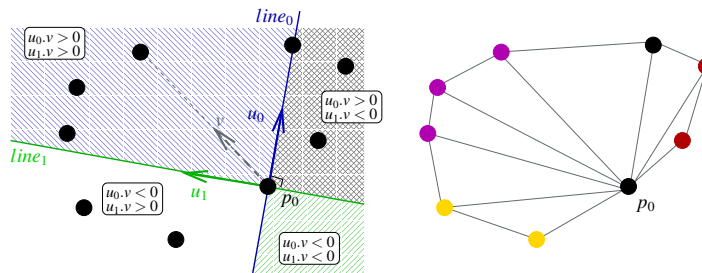
Nous avons adopté une approche assez similaire à [Lensch 02] qui détecte les points d'intersection entre des plans de coupe et des prismes englobant l'objet et propose une implémentation entièrement effectuée sur le GPU. Dans notre approche encore logicielle, une nouvelle façon de générer les plans de coupe de l'arrière vers l'avant est proposée en calculant les intersections entre les plans de coupes et les hexaèdres déformés (figure C.1). Ceci nous permet d'afficher interactivement des objets déformés. Pour accélérer les calculs, une implémentation matérielle pourrait être réalisée, encore plus efficacement que [Lensch 02] grâce à la nouvelle génération de cartes vidéo dont les "geometric shader" permettent de créer de nouveaux sommets et triangles à la volée.



**FIG. C.1** – Descriptif d'un rendu volumique d'un foie déformé

*Un plan de coupe (en rouge), orthogonal au point de vue (en bleu), coupe un maillage d'hexaèdres déformés (en noir). Toutes les intersections sont représentées par les points verts. Les triangles entre ces intersections sont affichés en semi-transparence de l'arrière vers l'avant. Les coordonnées de textures aux sommets des triangles (i.e. aux points d'intersection) sont interpolées entre les coins des hexaèdres.*

L'idée de base consiste à afficher des plans semi-transparents tout au long de l'objet. Ces plans sont orientés en face du point de vue afin de ne pas avoir d'artefacts dus à l'angle de vue. Pour gérer la semi-transparence en utilisant OpenGL, les plans doivent s'afficher les uns par dessus les autres, c'est-à-dire de l'arrière vers l'avant. Pour cela, le premier plan est celui qui passe par le coin le plus loin de la boîte englobante de l'objet par rapport au point de vue. Les plans suivants sont choisis en se rapprochant petit à petit du point de vue jusqu'à ne plus intersecter l'objet. Les plans peuvent ainsi être rendus au fur et à mesure. Ces plans sont texturés en fonction de la partie qu'ils représentent dans la texture 3D. Pour être rendus efficacement avec OpenGL, ces plans doivent être triangulés. Pour calculer cette triangulation, toutes les intersections entre un plan et les arêtes de tous les hexaèdres déformés sont considérées comme illustré dans la figure C.1 (l'algorithme rapide de calcul d'intersections utilisé est [Schneider 03]). On peut remarquer que le nombre de tests d'intersection peut être optimisé, car chaque arête qui a déjà été intersectée mais qui ne l'est plus, ne le sera pas pour les plans suivants. La triangulation est déduite en travaillant hexaèdre par hexaèdre pour ramener le problème à un cas plus simple de triangulation d'un contour dans le plan. Si les déformations des hexaèdres ne sont pas exagérés, on peut considérer que le polygone résultant est convexe. Nous proposons alors un moyen très rapide pour le trianguler. L'idée est de trier les sommets en les séparant en quatre groupes en fonction des valeurs du cosinus de l'angle qu'ils forment par rapport à deux axes d'un repère orthogonal quelconque. Déduire une bande de triangles de cette liste ordonnée est alors très facile, comme présenté en figure C.2. Pour plaquer la texture sur ces triangles, il faut trouver les bonnes coordonnées de textures. Pour cela, à l'initialisation, ces coordonnées sont stockées pour chaque sommet des hexaèdres (qu'on a directement puisque les hexaèdres sont construits à partir des données volumiques). Ces valeurs ne varient pas pendant l'animation. Ainsi, pour trouver les coordonnées de texture au sommet d'un triangle, il suffit d'interpoler linéairement les coordonnées de texture aux sommets de l'arête intersectée. Trouver la texture 2D correspondante à partir de coordonnées de texture 3D est directement traité par OpenGL.



**FIG. C.2** – Triangulation d'un contour dans le plan

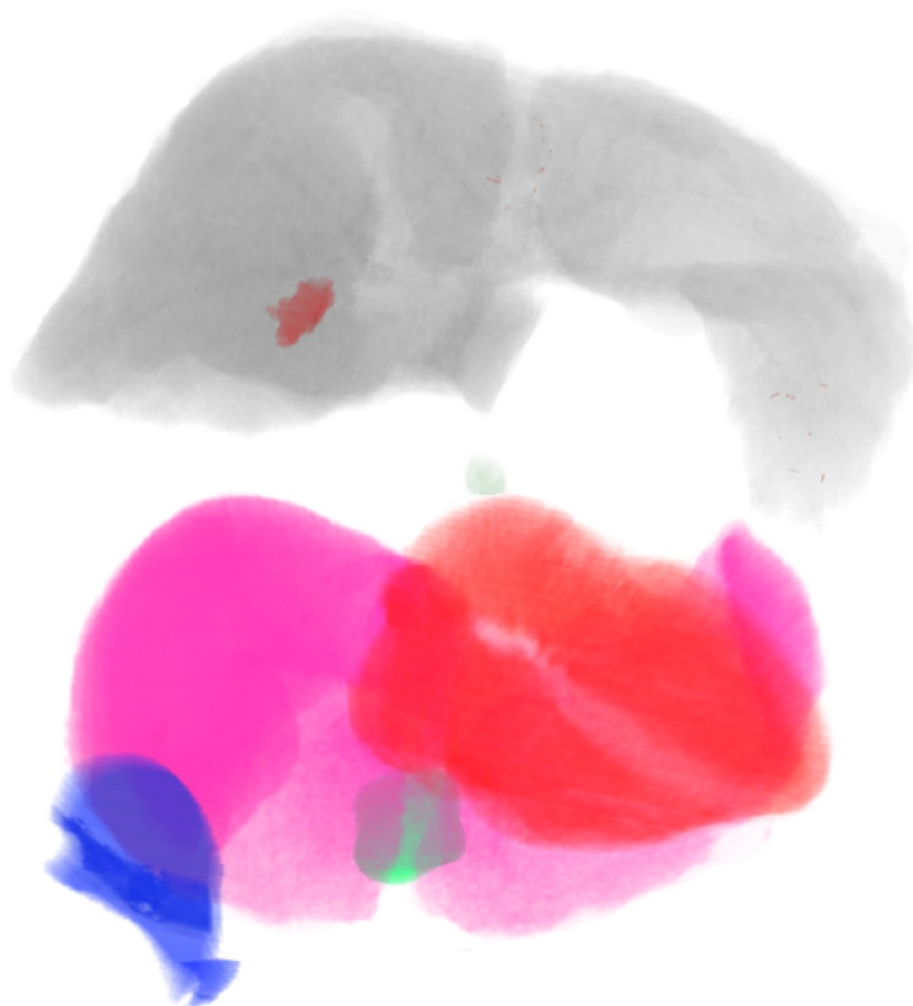
*Les sommets sont séparés en quatre groupes dépendamment du signe du cosinus de leur angle avec les deux axes d'un repère orthogonal de centre un sommet  $P_0$ . La bande de triangle résultante démarre avec  $P_0$  et parcourt les quatre groupes dans l'ordre en prenant en compte les valeurs des cosinus dans chaque groupe.*

Dans ce type de rendu, la qualité dépend fortement du nombre de plan affichés. Nous avons remarqué qu'une centaine de plans est suffisante pour obtenir des résultats satisfaisants, comme dans la figure 3.8 où 101 plans sont utilisés. Pour obtenir la même opacité de l'objet, quel que soit le nombre de plans affichés, le coefficient de transparence doit être adapté en suivant :

$$\alpha_{new} = \frac{\alpha_{old} \times nbplans_{old}}{nbplans_{new}}$$

On peut, grâce au rendu volumique, imaginer pouvoir ajouter une grande variété d'outils de visualisation, par exemple en figure C.3 sont ciblés une tumeur à l'intérieur d'un organe et différents organes.





**FIG. C.3** – Des outils de visualisation

*haut : une tumeur est ciblée en rouge. bas : plusieurs organes sont localisés par des couleurs différentes (cœur en rouge, foie en rose, partie du poumon gauche en bleu, vessie en vert)*



## 1 Tetraèdre

### 1.1 Matrice déplacements-déformations linéaire

Pour un tétraèdre  $(a, b, c, d)$ , la matrice de déplacements-déformations linéaire est de la forme :

$$B = \begin{bmatrix} \frac{\partial N_a}{\partial x} & 0 & 0 & \frac{\partial N_d}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_a}{\partial y} & 0 & 0 & \frac{\partial N_d}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_a}{\partial z} & 0 & 0 & \frac{\partial N_d}{\partial z} \\ \frac{\partial N_a}{\partial y} & \frac{\partial N_a}{\partial x} & 0 & \dots & \frac{\partial N_d}{\partial y} & \frac{\partial N_d}{\partial x} & 0 \\ 0 & \frac{\partial N_a}{\partial z} & \frac{\partial N_a}{\partial y} & 0 & \frac{\partial N_d}{\partial z} & \frac{\partial N_d}{\partial y} & \frac{\partial N_d}{\partial x} \\ \frac{\partial N_a}{\partial z} & 0 & \frac{\partial N_a}{\partial x} & \frac{\partial N_d}{\partial z} & 0 & \frac{\partial N_d}{\partial x} \end{bmatrix} = \frac{1}{6V} \begin{bmatrix} \beta_a & 0 & 0 & \beta_d & 0 & 0 \\ 0 & \delta_a & 0 & 0 & \delta_d & 0 \\ 0 & 0 & \gamma_a & \dots & 0 & 0 & \gamma_d \\ \gamma_a & \beta_a & 0 & \dots & \gamma_d & \beta_d & 0 \\ 0 & \delta_a & \gamma_a & 0 & \delta_d & \gamma_d \\ \delta_a & 0 & \beta_a & \delta_d & 0 & \beta_d \end{bmatrix}$$

$$6V = \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_a & x_b & x_c & x_d \\ y_a & y_b & y_c & y_d \\ z_a & z_b & z_c & z_d \end{vmatrix} \quad \beta_a = - \begin{vmatrix} 1 & 1 & 1 \\ y_b & y_c & y_d \\ z_b & z_c & z_d \end{vmatrix} \quad \gamma_a = \begin{vmatrix} 1 & 1 & 1 \\ x_b & x_c & x_d \\ z_b & z_c & z_d \end{vmatrix} \quad \delta_a = - \begin{vmatrix} 1 & 1 & 1 \\ x_b & x_c & x_d \\ y_b & y_c & y_d \end{vmatrix}$$

Les coefficients des autres fonctions d'interpolation sont les mêmes par un changement cyclique des indices ( $a$  devient  $b$ ,  $b$  devient  $c$ , etc, et les signes changent aussi).

### 1.2 Matrice déplacements-déformations linéaire après simplifications du repère local donnée par la méthode corotationnelle QR

Le repère QR permet de simplifier les calculs précédents, car le sommet  $a$  est à l'origine,  $b$  est sur le premier axe et  $c$  est dans le plan formé par les deux premières arêtes. Ces termes nuls résultent dans les calculs de coefficients suivants :

$$\begin{aligned} \beta_a &= -y_c z_d & \beta_b &= y_c z_d & \beta_c &= 0 & \beta_d &= 0 \\ \gamma_a &= (x_c z_d) - (x_b z_d) & \gamma_b &= x_c z_d & \gamma_c &= z_d x_b & \gamma_d &= 0 \\ \delta_a &= y_c x_d - x_c y_d + x_b y_d - x_b y_c & \delta_b &= y_c x_d - x_c y_d & \delta_c &= -y_d x_b & \delta_d &= -x_b y_c \end{aligned}$$

## 2 Hexaèdre

### 2.1 Matrices d'interpolation d'une subdivision hexaédrique

Les huit matrices suivantes sont les matrices d'interpolation  $L_c$  qui donnent les valeurs aux noeuds des éléments fils  $c$  en fonction des noeuds de leur élément père. Les indices sont présentés en Figure D.1.

Si  $a_{i \rightarrow c_j}$  est l'influence du  $i^{\text{ème}}$  noeud de l'élément père sur le  $j^{\text{ème}}$  noeud de son  $c^{\text{ème}}$  fils, alors :

$$L_c = \begin{bmatrix} a_{c_0 0} & \dots & a_{c_0 7} \\ \dots & \dots & \dots \\ a_{c_7 0} & \dots & a_{c_7 7} \end{bmatrix}$$

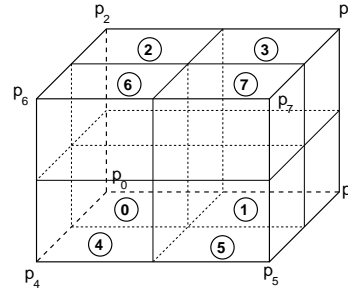


FIG. D.1 – Indices.

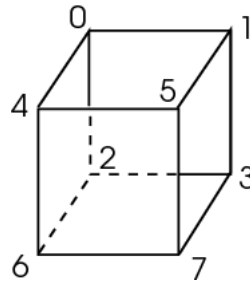
$$L_0 = \frac{1}{8} \begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \\ 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$L_4 = \frac{1}{8} \begin{bmatrix} 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 2 & 2 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 \\ 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

## 2.2 Découpe d'un hexaèdre en cinq tétraèdres

Un hexaèdre a deux, et seulement deux, découpes en cinq tétraèdres. Pour un hexaèdre d'indices présentés en Figure D.2, les découpes possibles sont :

découpe 1	découpe 2
(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. D.2** – Un hexaèdre avec ses sommets indexés.



---

# Tables détaillées





# Table des figures

---

1.1	Diverses déclinaisons de la multirésolution . . . . .	21
2.1	Un milieu continu, sa surface, des particules sur lesquelles s'appliquent des forces externes et des déplacements. . . . .	31
2.2	Déplacement $u$ d'un point de position $x$ . . . . .	31
2.3	Différentes déformations d'un hexaèdre . . . . .	32
2.4	Déformation : variation de longueurs dans un corps déformé. . . . .	32
2.5	Comparaison des tenseurs de déformations linéaire et non-linéaire. . . . .	34
2.6	Approche corotationnelle . . . . .	34
2.7	Force élémentaire . . . . .	35
2.8	Contrainte . . . . .	35
2.9	Divers tenseurs de contraintes en fonction des configurations considérées . . . . .	36
2.10	Des lois de comportement élastique . . . . .	37
2.11	Approximation linéaire (rouge) d'une loi de comportement élastique sublinéaire (bleue) . . . . .	37
2.12	Influences du module d'Young et du coefficient de Poisson . . . . .	38
2.13	Matériau viscoélastique . . . . .	39
2.14	Matériau plastique . . . . .	39
2.15	Relation déplacements/déformations/contraintes/forces en fonction de la géométrie et du matériau . . . . .	40
2.16	Une solution exacte (gauche) approximée par une simulation numérique sur une représentation discrète (droite). . . . .	41
2.17	Principe de la Méthode des Elements Finis . . . . .	42
2.18	Ajouter des nœuds aux éléments permettent d'avoir des degrés d'interpolation plus élevés . . . . .	43
2.19	Une seule rotation pour tout l'objet . . . . .	46
2.20	Repère au sommet . . . . .	46
2.21	Elimination de la rotation par sommet . . . . .	46
2.22	Elimination de la rotation par élément . . . . .	47
2.23	Différents repères locaux . . . . .	48
2.24	Schématisation des convergences avec et sans préconditionnement . . . . .	51
2.25	Couple fantôme. . . . .	51
2.26	Conséquence d'un moment fantôme . . . . .	52
3.1	Des maillages mécaniques en deux dimensions . . . . .	65
3.2	Maillage d'un dragon. . . . .	66
3.3	Détection des éléments vides/pleins/frontières . . . . .	66
3.4	Plusieurs résolutions de voxelisations à partir de données volumiques segmentées issues d'un scanner médical (ici pour un foie) . . . . .	67
3.5	Interpolation . . . . .	67
3.6	Arborescences virtuelles . . . . .	69
3.7	Exemples de maillages classiquement difficiles à animer . . . . .	70
3.8	Rendu volumique d'un foie se déformant . . . . .	71

3.9	Différentes résolutions de voxelisation . . . . .	72
3.10	Utilisation des virtual nodes pour un meilleur comportement des maillages grossiers . . . . .	72
4.1	Repère local sur un hexaèdre déformé. . . . .	77
4.2	Schématisation en niveau de gris de la raideur en 2D . . . . .	77
4.3	Condensation d'un pixel donné vers un élément fini triangulaire . . . . .	78
4.4	Condensation non-uniforme des propriétés mécaniques . . . . .	79
4.5	Nœud en T à l'interface de divers niveaux de multirésolution (2 dimensions) . . . . .	80
4.6	Approche hiérarchique . . . . .	80
4.7	Illustrations des étapes du filtre gérants les nœuds en T . . . . .	82
4.8	Comparaison de temps de calculs entre des maillages tétraédriques classiques (haut) et nos hexaèdres englobants (bas) pour la même méthode corotationnelle (QR) . . . . .	83
4.9	Vitesse d'animation . . . . .	84
4.10	Robustesse face à des configurations dégénérées non-physiques . . . . .	86
4.11	Uniforme moyenné contre Non-uniforme . . . . .	86
4.12	Uniforme moyenné contre Non-uniforme . . . . .	87
4.13	Matériau uniforme contre matériau non-uniforme . . . . .	87
4.14	Une coupe axiale des données scanner d'entrée . . . . .	88
4.15	haut : les raideurs prennent en compte les niveaux de gris des données scanner d'entrée. bas : un matériau unique est utilisé pour tout l'objet. . . . .	88
4.16	Vérification sur la gestion des nœuds en T . . . . .	89
4.17	Raffinements automatiques . . . . .	89
4.18	Utilisation des maillages voxelisés sur une opération spécifique à un patient. Relèvement du foie (rouge foncé) pour atteindre la vésicule biliaire (verte). . . . .	90
5.1	Propagation des déformations le long d'un objet discrétisé 1D. . . . .	92
5.2	Convergence du gradient conjugué . . . . .	93
5.3	Différentes représentations d'éléments finis 1D . . . . .	94
5.4	Vecteurs d'états nodal et hiérarchique . . . . .	95
5.5	Influence d'un élément . . . . .	96
5.6	Conséquences géométriques d'une représentation hiérarchique . . . . .	96
5.7	Comparaison de la propagation dynamique dans les cas nodal (haut) et hiérarchique (bas) . . . . .	97
5.8	Intégration sur des fonctions de rayons d'action différents . . . . .	99
5.9	Prise en compte des grands déplacements hiérarchiques . . . . .	99
5.10	Les tests . . . . .	100
5.11	Vitesse de convergence sur l'exemple 1 . . . . .	101
5.12	Cycle de multigrid à 2 niveaux . . . . .	102
5.13	Les cycles de multigrid . . . . .	103
5.14	Comparaison de vitesses entre des résolutions multigrid basées sur un cycle en V et un cycle complet . . . . .	104
5.15	Comparaison de la qualité (vitesse et erreur) des différentes approches pour le matériau "mou" dans le cas d'une poutre allongée sous la gravité . . . . .	106
5.16	Comparaison de la qualité (vitesse et erreur) des différentes approches pour le matériau "raide" dans le cas d'une poutre allongée sous la gravité . . . . .	107
5.17	Comparaison de la qualité (vitesse et erreur) des différentes approches pour la poutre en flexion . . . . .	108

6.1	Notations utilisée pour le problème de la poutre encastrée . . . . .	114
6.2	Flèches de poutres sous la gravité obtenues par la théorie et par des calculs numériques pour différents coefficients de Poisson . . . . .	115
6.3	Résultats des simulations de l'expérience du Truthcube . . . . .	115
6.4	Différents maillages d'éléments englobant une poutre encastrée. . . . .	118
6.5	Erreur en distance euclidienne (mm) par rapport au maillage optimal . . . . .	118
6.6	Comparaison de simulations d'une poutre cylindrique encastrée face à des données réelles . . . . .	119
6.7	Compression d'un cylindre composé d'un matériau quasi-imcompressible (coefficient de Poisson = 0,49) . . . . .	120
6.8	Circonférence du plan médian d'un cylindre sous compression . . . . .	121
6.9	L'expérience de déplacements imposés sur le modèle du foie . . . . .	122
A.1	Analyse du conditionnement du système implicite en fonction des paramètres de matériau et du pas de temps . . . . .	152
B.1	Laparoscopie . . . . .	154
C.1	Descriptif d'un rendu volumique d'un foie déformé . . . . .	157
C.2	Triangulation d'un contour dans le plan . . . . .	158
C.3	Des outils de visualisation . . . . .	159
D.1	Indices. . . . .	162
D.2	Un hexaèdre avec ses sommets indexés. . . . .	163



# Table des matières

---

<b>Introduction</b>	<b>1</b>
1 Motivation . . . . .	3
2 Contexte . . . . .	4
3 Problématique . . . . .	5
4 Contributions . . . . .	6
5 Organisation du document . . . . .	7
<b>Contexte scientifique</b>	<b>9</b>
<b>1 Etat de l'art</b> . . . . .	<b>11</b>
1 Introduction . . . . .	12
2 Critères de qualité . . . . .	12
3 Modèles non-physiques . . . . .	14
4 Modèles physiques . . . . .	14
4.1 Champs de forces . . . . .	15
4.2 Intégration du temps . . . . .	18
4.3 Améliorations . . . . .	20
4.3.1 Modèles hybrides . . . . .	20
4.3.2 Multirésolution . . . . .	21
4.3.3 Parallélisme . . . . .	23
4.4 Déchirure / découpe . . . . .	23
4.5 Gestion des collisions . . . . .	24
5 Applications . . . . .	25
5.1 Informatique graphique . . . . .	26
5.2 Simulateurs chirurgicaux . . . . .	26
5.3 Plates-formes d'animation physique . . . . .	27
6 Discussion . . . . .	27
<b>2 Notions de mécanique</b> . . . . .	<b>29</b>
1 Introduction . . . . .	30
2 Notions théoriques . . . . .	30
2.1 Mécanique des milieux continus . . . . .	30
2.2 Déplacements . . . . .	31
2.3 Déformations . . . . .	32
2.4 Contraintes . . . . .	35
2.5 Lois de comportement des matériaux . . . . .	36
2.6 Modèles répandus . . . . .	40
3 Notions numériques . . . . .	41
3.1 Méthode des éléments finis . . . . .	42
3.1.1 Principe . . . . .	42

3.1.2	Mailles . . . . .	42
3.1.3	Calcul des forces . . . . .	43
3.2	Implémentations . . . . .	44
3.2.1	Vitesse . . . . .	44
3.2.2	Grands déplacements . . . . .	45
3.2.3	Changements de topologie . . . . .	48
3.2.4	Stabilité . . . . .	49
3.2.5	Matériaux . . . . .	52
3.2.6	Eléments quadratiques . . . . .	53
3.3	Validation . . . . .	53
4	Conclusion . . . . .	54

## **Milieu mécanique multirésolution . . . . . 57**

<b>3</b>	<b>Animation d'objets par grilles de déformations . . . . .</b>	<b>63</b>
1	Introduction . . . . .	64
2	Principe . . . . .	65
3	Interpolation de la surface . . . . .	67
4	Conditions aux limites sur la surface . . . . .	68
4.1	Contraintes "dures" . . . . .	68
4.2	Forces externes . . . . .	68
5	Multirésolution . . . . .	69
6	Résultats . . . . .	70
7	Limitations . . . . .	71
8	Conclusion . . . . .	72
<b>4</b>	<b>Mécanique des hexaèdres englobants . . . . .</b>	<b>75</b>
1	Introduction . . . . .	76
2	Hexaèdres corotationnels . . . . .	76
3	Propriétés non-uniformes . . . . .	77
4	Multirésolution . . . . .	80
5	Résultats . . . . .	82
5.1	Vitesse . . . . .	82
5.2	Robustesse . . . . .	85
5.3	Comportement intuitif . . . . .	86
5.4	Multirésolution . . . . .	88
6	Conclusion . . . . .	89
<b>5</b>	<b>Propagation dynamique . . . . .</b>	<b>91</b>
1	Introduction . . . . .	92
2	Eléments finis hiérarchiques . . . . .	94
2.1	Principe . . . . .	94
2.2	Répercussions sur la dynamique . . . . .	95
2.3	Mise en place . . . . .	97
2.3.1	Intervalle de définition des fonction d'interpolation . . . . .	98
2.3.2	Grands déplacements . . . . .	99

2.4	Résultats . . . . .	100
2.5	Conclusion . . . . .	101
3	Multigrid . . . . .	101
3.1	Principe . . . . .	102
3.2	Mise en place . . . . .	103
3.3	Adaptativité . . . . .	104
3.4	Résultats . . . . .	105
4	Conclusion . . . . .	110
<b>6</b>	<b>Validations du modèle . . . . .</b>	<b>111</b>
1	Introduction . . . . .	112
2	Cas des hexaèdres pleins . . . . .	113
2.1	Poutre encastrée . . . . .	113
2.2	Truthcube . . . . .	115
3	Cas des hexaèdres non-pleins . . . . .	118
3.1	Poutre encastrée . . . . .	118
3.2	Cylindre . . . . .	119
3.2.1	Flexion . . . . .	119
3.2.2	Compression uni-axiale . . . . .	120
3.3	Foie . . . . .	121
4	Conclusion . . . . .	123
	<b>Conclusion . . . . .</b>	<b>127</b>
1	Récapitulatif des contributions . . . . .	129
2	Analyse . . . . .	129
3	Perspectives . . . . .	130
3.1	A court terme . . . . .	130
3.2	A moyen terme . . . . .	130
3.3	A long terme . . . . .	131
	<b>Bibliographie . . . . .</b>	<b>133</b>
	<b>Annexes . . . . .</b>	<b>149</b>
<b>A</b>	<b>Conditionnement d'un système linéaire . . . . .</b>	<b>151</b>
1	Définition . . . . .	151
2	Conséquence . . . . .	151
3	Conditionnement du système implicite en fonction du matériau . . . . .	151
<b>B</b>	<b>Informatique et médical . . . . .</b>	<b>153</b>
<b>C</b>	<b>Rendu volumique adapté à la structure hexaédrique . . . . .</b>	<b>157</b>

---

<b>D Matrices utiles</b>	<b>161</b>
1 Tetraèdre	161
1.1 Matrice déplacements-déformations linéaire	161
1.2 Matrice déplacements-déformations linéaire après simplifications du repère local donnée par la méthode corotationnelle QR	161
2 Hexaèdre	162
2.1 Matrices d'interpolation d'une subdivision hexaédrique	162
2.2 Découpe d'un hexaèdre en cinq tétraèdres	163
 <b>Tables détaillées</b>	 <b>165</b>
<b>Table des figures</b>	<b>167</b>
<b>Table des matières</b>	<b>171</b>



Les modèles dynamiques sont incontournables en synthèse d'animations car ils permettent la simulation réaliste de phénomènes physiques et accordent une meilleure immersion dans un monde virtuel.

Plusieurs approches performantes permettent l'animation d'objets déformables, mais les scènes sont souvent complexes à modéliser rendant leur utilisation difficile en pratique.

Dans cette thèse nous proposons une solution simplifiant l'animation physique interactive d'objets déformables. Nous suggérons de plonger et d'interpoler l'objet dans une grille déformable sur laquelle s'appliquent des lois mécaniques. Une méthode d'éléments finis rapides et robustes a été étendue afin de prendre en compte la répartition de la matière et plusieurs propriétés de matériaux à l'intérieur d'un élément, et ainsi offrir un comportement amélioré à des résolutions grossières. Afin de concentrer les calculs là où ils sont le plus nécessaires, une formulation multirésolution simple est proposée.

Puis nous analysons deux méthodes permettant d'améliorer la propagation des déformations pour des matériaux "mal conditionnés" : une formulation hiérarchique des éléments finis, lourde à mettre en place mais permettant facilement la multirésolution, et une formulation multigrid, élégante et performante, mais plus difficile à décliner en multirésolution.

Enfin nous validons la précision de notre méthode en la soumettant à diverses expériences.

Il en résulte une méthode rapide, robuste, précise et facile d'utilisation aussi bien pour un infographiste, qui peut animer n'importe quel modèle sans connaissances préalables, que pour la modélisation individualisée d'un patient à partir d'images médicales segmentées.

---

Physically based models are invaluable in the field of animation synthesis because they permit realistic simulation of physical phenomena and allow for better immersion in a virtual world.

Several approaches exist for animating soft bodies, but the modelling is often complex and difficult to use in practice.

In this thesis, we introduce simplified process for the interactive, realistic physical animation of deformable objects. In this approach, the object is imbedded within a deformable grid to which mechanical laws are applied. A fast and robust finite element method is extended to take into account the distribution of material and material properties inside an element, thereby offering improved behavior for coarse resolutions. To allow for interactive running times, a simple multiresolution formulation is proposed to focus computations where they are most necessary. To improve the propagation of deformations for "poorly conditioned" materials, two methods are evaluated : a hierarchical finite elements formulation, which is difficult to set-up but facilitates the multiresolution approach, and a multigrid formulation, which is elegant and efficient but more difficult to use in the multiresolution framework. Finally the precision of our method is experimentally validated and examples are shown for a wide-range of models.

In summary, this thesis presents a fast, robust, accurate, and easy-to-use approach for the physically realistic animation of deformable objects. This method simplifies both the task of the graphic artist, since prior knowledge in mechanics is not required, and the task of patient-specific modelling since segmented medical images are directly exploitable.