

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Guillaume Bousquet

Thèse dirigée par **François FAURE**

Préparée au sein du **Laboratoire Jean Kuntzmann**
et de **l'école doctorale MSTII**

Déformation et découpe interactive de solides à géométrie complexe

Thèse soutenue publiquement le **25 octobre 2012**,
devant le jury composé de :

Denis Favier

Professeur des Universités à l'UJF, Grenoble, Président

Hervé Delingette

Directeur de Recherches à l'INRIA - Sophia-Antipolis, Rapporteur

Yannik Rémion

Professeur des universités à l'IUT de Reims, Rapporteur

Jérémie Allard

Chargé de Recherches à l'INRIA - Lille, Examineur

Yohan Payan

Directeur de Recherche CNRS, TIMC, La Tronche, Examineur

François FAURE

Professeur des Universités à l'UJF, Grenoble, Directeur de thèse



Résumé La simulation physique des modèles déformables est devenue omniprésente dans l'infographie. Elle permet de synthétiser des comportements réels, en se basant sur des lois physiques issues de la mécanique des milieux continus. Dans cette thèse, nous nous intéressons aux simulations interactives telles qu'appliquées aux jeux vidéo ou aux simulateurs chirurgicaux.

La majorité des travaux se sont focalisés jusqu'ici sur l'animation d'objets composés de matériaux homogènes. Néanmoins, bon nombre d'objets réels, comme les structures biologiques, sont composés de multiples matériaux imbriqués. Leur décomposition en zones homogènes requiert une discrétisation spatiale à haute résolution pour résoudre les variations des propriétés du matériau, ce qui aboutit à des temps de calcul prohibitifs.

Dans ce contexte, nous présentons de nouvelles techniques de simulation en temps réel d'objets déformables qui peuvent être découpés.

Tout d'abord, une méthode de découpe d'objets déformables en temps réel dans laquelle, contrairement aux méthodes précédentes, l'objet se déforme au contact d'un outil coupant et ne se coupe que lorsque la pression atteint un certain niveau. L'indépendance des modèles physique, de collision et de rendu facilitent les changements topologiques. L'exploitation du GPU et les modifications locales permettent une exécution rapide.

Ensuite, une méthode dynamique sans maillage, utilisant des repères comme nœuds au lieu de points, dont la formulation du champ de déplacement s'apparente au *skinning* ; héritant ainsi de sa souplesse de mise en œuvre et d'une rigueur équivalente aux méthodes physiques telles que les éléments finis. L'introduction des points d'intégration, réduisant le nombre d'échantillons par une approximation aux moindres carrés, accélère les intégrations spatiales. D'autres pré-calculs sont présentés afin d'accélérer le temps de simulation.

Enfin, la définition de fonctions de forme anisotropes encodant les variations des propriétés du matériau grâce à l'introduction d'une distance de compliance. Ces fonctions de forme complexes découplent la résolution matérielle de celles des fonctions de déplacements et permettent un échantillonnage des nœuds extrêmement faible. L'utilisation de la distance de compliance permet une distribution automatique des nœuds de simulation relativement aux propriétés matérielles.

Abstract Physically based deformable models have become ubiquitous in computer graphics. It allow to synthetize real behaviors, based on the physical laws from continuum mechanics. In this thesis, we focus on interactive simulations such as to video games or surgical simulators.

The majority of the existing works focused up to here on the animation of objects made of homogeneous materials. Nevertheless, plenty of real objects, for instance like the biological structures, consist of multiple imbricated materials. Their decomposition in homogeneous zones requires a high-resolution spatial discretization to solve the variations of the material properties, which requires prohibitive computation time.

In this context, we present new real time simulation techniques for deformable objects which can be cut.

First of all, we present a real time method for cutting deformable objects in which, contrary to the previous methods, the object deforms on the cutting tool contact and cuts occur only when the pressure reaches a certain level. The independence of the physical,

collision and visual models makes the topological changes easier. The GPU computing and local modifications enable fast execution.

Then, a dynamic meshless method is described, which uses reference frames as control nodes instead of using points, with a displacement field formulation similar to skinning. It allows to easily tune the weights and benefits from the rigor of physical methods as the finite elements. The introduction of integration points, reducing the samples number by a least squares approximation, speeds up the spatial integrations. Other pre-computations are proposed in order to speed up the simulation time.

Finally, new anisotropic shape functions are defined to encode the variations of material properties thanks to the introduction of the compliance distance. These complex shape functions uncouple the material resolution of the displacement functions ones. It allow an extremely sparse nodes sampling. The use of the compliance distance allows an automatic nodes distribution with regard to the material properties.

Notations

Écriture :

a	Scalaire
\mathbf{x}	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 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

$$\nabla u = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} & \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} & \frac{\partial u_z}{\partial y} & \frac{\partial u_z}{\partial z} \end{bmatrix}$$

Gradient de a

Symboles :

\mathbf{M}	Matrice de masse
--------------	------------------

Sommaire

Introduction	1
<hr/>	
Contexte scientifique	7
<hr/>	
1 Introduction à la théorie des milieux continus	7
2 Modèles déformables discrets	19
<hr/>	
Déformation par repères déformables	45
<hr/>	
3 Simulation de découpe	47
4 Fonctions de forme relatives aux matériaux et Echantillonnage	57
5 Dynamique par repères déformables	71
6 Résultats et autres travaux	85
<hr/>	
Conclusion	99
<hr/>	
Bibliographie	103
<hr/>	
Annexes	111
<hr/>	
Formules utiles au mélange par quaternions duaux	111
<hr/>	
Tables détaillées	116
<hr/>	
Table des figures	116

Table des matières 117

Introduction

Motivation

Les simulations d'objets déformables sont présentes dans un nombre sans cesse plus important de domaines. Elles permettent de synthétiser des comportements réels, en se basant sur des lois physiques issues de la mécanique des solides déformables. La résolution analytique, décrite par ces modèles mathématiques, est complexe à réaliser en informatique où une discrétisation intervient et extrêmement coûteuse en temps de calcul. Il existe ainsi de nombreux modèles de simulation, proposant différents compromis entre rapidité et précision, par le biais d'approximations.

Certains domaines, comme l'étude industrielle (automobile, nautique, aéronautique...) ou scientifique (physique nucléaire, biologie, chimie...) ciblent la précision du comportement, quel que soit le temps de calcul. Certains, tels que les jeux vidéo ou la simulation chirurgicale, où un utilisateur interagit avec la simulation, privilégient l'exécution en temps réel, au détriment de la précision. D'autres, tel que les effets spéciaux, où le temps réel comme la précision ne sont pas critiques, choisissent un compromis entre les deux, de manière à obtenir un comportement visuellement plausible pour un temps de calcul ne nuisant pas à la productivité. Cette thèse laisse de côté les applications axées sur la précision et se concentre sur celles s'exécutant en temps réel ou interactifs.

Concernant le domaine des jeux vidéo, la simulation dynamique est de plus en plus présente depuis une décennie. Avant cela, la majorité des jeux proposaient des animations 2D, puis ont utilisé la cinématique sur des modèles articulés simples. Aujourd'hui, la plupart des jeux intègre une physique simplifiée basée sur la mécanique des solides rigides et peu d'entre eux commencent à proposer une interaction avec l'environnement à l'aide d'objets déformables. Le passage de la cinématique à la dynamique demande un travail trop important. Les effets spéciaux, quant à eux, sont partagés. Une minorité exploite déjà les simulations d'objets déformables et contribuent activement à la recherche dans ce domaine. Mais la plupart ont recours à la mécanique des solides rigides.

Ceci est dû à plusieurs facteurs. Concernant les jeux, le temps de calcul allouable aux simulations est faible et leur coût est important. Le temps de calcul est partagé pour l'exécution en temps réel du graphisme, de l'audio, de la mise en réseau, de l'intelligence artificielle, et autres. Ensuite, la simulation d'objets déformables nécessite l'intervention d'un expert. L'échantillonnage détermine la qualité de la simulation. La majorité des méthodes actuelles requièrent la création d'un modèle physique et sa configuration afin d'éviter les instabilités numériques et les artefacts visuels.

Dans le domaine médical, la demande de simulateurs chirurgicaux est croissante. Les cliniciens souhaiteraient disposer de simulateurs rapides pour effectuer des planifications pré-opératoires, dans le cas de prise en charge de patients en urgence, ou des replanifications en cours d'opération, lors d'un imprévu empêchant le déroulement classique de l'intervention. Ils permettraient aussi aux chirurgiens en formation, de disposer d'un outil d'entraînement, évitant ainsi d'avoir systématiquement recours à des cadavres tout en donnant la possibilité de mettre en avant les structures anatomiques ou les points importants d'une intervention à l'aide d'une visualisation plus schématique et pédagogique.

Ainsi, malgré les progrès continus du matériel informatique, la recherche de nouvelles techniques d'animation est nécessaire dans le but d'améliorer la rapidité, la précision, la robustesse et la simplicité d'utilisation des techniques sur des scènes toujours plus complexes et volumineuses.

Contexte

Dans le cadre d'un programme de prévention et de traitement des maladies du foie, la commission européenne a lancé le projet "PASSPORT - PATient Specific Simulation and PreOperative Realistic Training for liver surgery". Ce vaste projet implique la participation de nombreux partenaires experts incluant l'Ircad, Karl Storz, l'Inria, l'Iserm et les universités de Munich (MRI TUM), Londres (Imperial College et UCL), Leipzig (IZBI) et Zurich (ETH) dans le but de développer des outils et des techniques en médecine, informatique, biologie et robotique. Un des objectifs ambitieux visé par le projet PASSPORT est la réalisation d'un simulateur chirurgical en temps réel, basé sur les données d'une IRM du patient, dont toutes les étapes seraient automatisées.

Les travaux de cette thèse portent sur la simulation de corps déformables en temps réel, qui relève de différents domaines scientifiques, principalement de la mécanique, de l'informatique et des méthodes numériques. Elle a été réalisée au sein de l'équipe *Imagine* (anciennement *Evasion*) du laboratoire *LJK*, qui possède une grande expérience dans ces domaines. La thématique principale de l'équipe porte sur la modélisation des phénomènes naturels. L'éventail des connaissances est large, incluant des recherches sur la simulation physique en temps réel, l'analyse de croquis, le traitement géométrique ou encore la modélisation d'ontologie anatomique.

Sofa (Simulation Open Framework Architecture) est une plateforme, au code source libre, initialement orientée vers la simulation en temps réel. Développée par trois équipes Inria (Shacra, Evasion et Asclepios), SOFA a acquis au fil des années de nombreuses méthodes de simulations en temps réel. Elle propose ainsi de nombreux solveurs d'équations différentielles (explicites, implicites et statiques), modèles physiques (masses ressorts, éléments finis, meshless), systèmes de détection de collisions, de rendus, l'exploitation du calcul sur GPU, la parallélisation d'algorithmes et autres. Elle repose sur la modularité des composants et le découplage du modèle physique, de collision et de rendus ; et propose ainsi à la communauté scientifique un outil de recherche avancé. Les algorithmes développés au cours de cette thèse sont intégrés dans cette plateforme et disponible à tous à l'adresse <http://www.sofa-framework.org>. Bien qu'ils aient été conçus dans l'optique d'une utilisation par les applications chirurgicales du projet PASSPORT, leur genericité leur permet d'être employés dans d'autres domaines tel que le développement de jeux vidéos ou de films d'animation.

Problématique

Les simulateurs chirurgicaux en temps réel à l'heure actuelle sont très basiques et aucun ne propose une simulation spécifique au patient de manière entièrement automatisée. Les raisons sont nombreuses. Outre le problème de traitement des données initiales (segmentation automatique requérant des données a priori), développer une telle application nécessite la mise en place d'un modèle physique. L'échantillonnage de la matière est une étape cruciale pour la simulation. La plupart des méthodes actuelles, comme les éléments finis, reposent sur un maillage. Sa qualité détermine la précision du comportement et la stabilité de la méthode. Or, la génération automatique de celui-ci est complexe et reste un problème ouvert.

Ensuite, la modification des modèles survenant suite à une découpe au scalpel est une fonctionnalité indispensable pour la planification chirurgicale. Les travaux existants dans ce domaine présentent deux approches. La première consiste à procéder à une découpe nette et précise du modèle. Ce qui implique une opération de remaillage, qui est complexe et coûteuse. La seconde repose sur la suppression d'éléments, évitant ainsi l'étape de remaillage, mais il en résulte une découpe grossière dont la qualité dépend du maillage initial. Un autre aspect important de la découpe est la gestion des contacts. Il est en effet souhaitable que le modèle se déforme sous la pression du scalpel avant que la découpe ne survienne, une fois que la pression exercée par le chirurgien devient suffisante.

Enfin, les méthodes de simulation développées pour le temps réel sont souvent destinées à des applications de type jeux vidéos ou films d'animation. Les modèles sont simples et présentent des matériaux à propriétés uniformes (raideur, densité, etc). Or dans le cadre médical, les matériaux sont pratiquement tous hétérogènes et certains anisotropes. Il serait donc intéressant de disposer d'une méthode de simulation temps réel prenant en compte les variations des propriétés de la matière.

Contributions

Cette thèse présente ainsi une nouvelle technique de simulation en temps réel, qui permet la découpe avec réponse aux contacts. Le modèle physique présenté dans cette méthode est basée sur des éléments finis par grille régulière. L'utilisation de nœuds virtuels, étendus aux changements topologiques, garantit le respect de la topologie initiale du modèle. Un second modèle de simulation est ensuite présenté pour pallier à certains inconvénients du premier. Il repose sur un pipeline meshless basé sur l'énergie de déformation, utilisant des repères comme nœuds, et permettant la simulation de modèles à propriétés hétérogènes de manière simple. Voici les principales contributions décrites dans la suite de ce mémoire :

◆ **Découpe** Nous proposons une technique de découpe en temps réel sur des scènes de taille importante. Les modèles physiques et visuels sont automatiquement générés à partir de données issues d'IRM. La découpe est réalisée par suppression des voxels composant ces données, permettant une découpe plus fine que celle réalisée par suppression d'éléments du modèle physique. Pour améliorer l'immersion de l'utilisateur, la réponse aux contacts est prise en compte lors de la découpe.

◆ **Echantillonnage** Notre échantillonnage de nœuds est réalisé à l'aide de repères. L'absence de maillage et l'utilisation de fonctions de forme plus flexibles, permet d'obtenir une simulation à l'aide d'un nombre très réduit de nœuds, répartis en fonction des propriétés matérielles. Sa génération est automatique. L'utilisateur peut aussi choisir l'emplacement des repères, une complétion automatique étant réalisée ensuite pour garantir de bons résultats.

◆ **Matériaux à propriétés hétérogènes** Nos fonctions de forme encodent les propriétés de la matière, afin de correctement simuler des matériaux à propriétés hétérogènes, malgré un échantillonnage faible des nœuds. Cette étape est réalisée automatiquement à partir de cartes de densité (incluant les IRM). Elle permet par exemple la simulation automatique correcte d'une articulation du genou, sans connaissance a priori des structures

fonctionnelles qui la composent.

Organisation du document

Ce document se divise en deux parties décrivant le contexte scientifique et les contributions développées au sein de la thèse.

Tout d'abord, une introduction aux notions de mécanique de la **théorie des milieux continus** est faite au chapitre 1. Ce chapitre s'adresse à des non mécaniciens afin de leur donner les bases nécessaires à la compréhension de la partie contribution.

Le chapitre 2 parcourt l'ensemble des **travaux existants** en informatique graphique dont nous nous sommes inspirés pour réaliser ce travail. Il présente le concept des différentes techniques et discute de leurs avantages et inconvénients.

Dans la seconde partie présentant les contributions, le pipeline général de notre **technique de découpe** est présenté au chapitre 3. Nous introduisons la possibilité de prendre en compte les contacts et d'effectuer la découpe du modèle une fois un seuil de pression atteint.

Une vue générale de notre **modèle physique**, basé sur l'utilisation de repère, est présentée au début du chapitre 4. La suite de ce chapitre introduit nos fonctions de forme anisotropes. Elles permettent aussi d'encoder les propriétés hétérogènes de la matière.

Le chapitre 5 détaille la partie dynamique de ce **modèle physique**. Le nombre de nœuds employés dans une simulation peut être extrêmement réduit, le rendant très léger et rapide même sur des modèles à géométrie complexe. Son initialisation peut être automatique et il rend plus aisé les changements topologiques.

Enfin, les résultats obtenus et les travaux en cours sont listés au chapitre 6. La validation de la méthode est réalisée par comparaison avec la méthode des éléments finis.

Contexte scientifique

Chapitre 1

Introduction à la théorie des milieux continus

Contents

1.1	Discretisation et degrés de liberté	8
1.2	Vue d'ensemble	8
1.3	Déplacements	9
1.4	Tenseur des déformations	10
1.4.1	Le tenseur de Cauchy	11
1.4.2	Le tenseur de Green-Lagrange	11
1.5	Tenseur des contraintes	12
1.6	Loi de comportement	12
1.6.1	La loi élastique	13
1.6.2	La loi de Hooke	13
1.7	Matrice de rigidité	15
1.7.1	Formulation linéaire	15
1.7.2	Formulation non linéaire	16
1.8	Matrice de masse	16
1.9	Intégration dynamique	16
1.10	Conclusion	18

La mécanique des milieux continus est le domaine de la mécanique qui s'intéresse à la déformation des solides et à l'écoulement des fluides. Nous nous intéressons ici à la déformation des solides.

Si l'on regarde la matière à l'échelle nanoscopique, elle est discontinue, faite d'atomes. Mais à l'échelle macroscopique, un objet solide semble continu, c'est-à-dire que ses propriétés semblent varier progressivement, sans à-coups.

Le but de la dynamique des milieux continus est de mettre en relation les forces appliquées sur un objet et les déplacements des points constituant l'objet qui en résultent au cours du temps.

1.1 Discrétisation et degrés de liberté

L'étude de corps présentant des propriétés a priori continues (par morceaux) permet d'avoir recours aux outils mathématiques reposant sur les fonctions continues et dérivables. Dans la simulation informatique, la résolution de telles équations n'est réalisable que de manière discrète : une résolution analytique nécessiterait une définition analytique des géométries et serait complexe à résoudre. De ce fait, quelle que soit la méthode dérivant des milieux continus utilisée, un ensemble fini de primitives représentatives, nommées *nœuds*, est simulé. Dans la majorité des techniques actuelles, les nœuds sont des points. Des repères sont utilisés dans certaines techniques, incluant alors la rotation en plus de la translation. Ces nœuds sont composés de degrés de liberté (par exemple, les dimensions spatiales pour un point). L'ensemble de nœuds intégrés dans le temps doit être suffisamment restreint pour permettre une résolution dans un temps acceptable. Le déplacement du reste de la matière est ensuite défini en tous points de l'objet suivant une fonction de forme se basant sur le déplacement des nœuds intégrés.

Dans un cas simple d'éléments finis par exemple, un échantillonnage de la matière peut être effectué suivant une grille régulière. Les degrés de liberté sont les sommets de cette grille qui sont simulés dans le temps. Le déplacement de tout autre point de la matière est exprimé par une interpolation des positions des sommets de l'élément dans lequel se trouvent ces points. L'interpolation couramment utilisée dans cette méthode repose sur les coordonnées barycentriques du point dans l'élément.

1.2 Vue d'ensemble

A partir des positions des degrés de liberté au temps t et des forces appliquées sur le modèle, telles que les forces externes (gravité, collisions) et les forces internes (dues aux déformations), le but est de calculer les positions des degrés de liberté au temps $t + 1$.

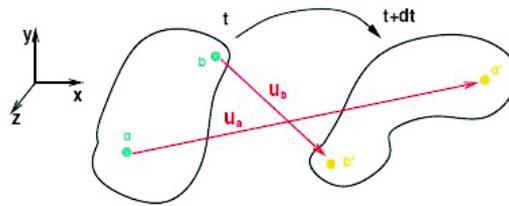
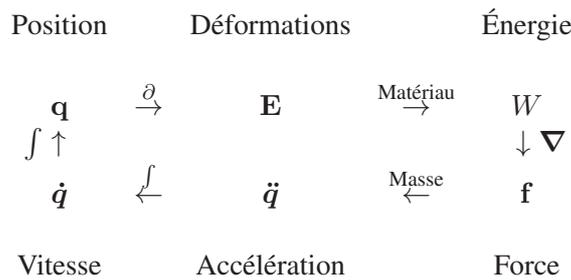


FIGURE 1.1: Champ de déplacement

Le diagramme suivant illustre le pipeline général :



A partir des positions actuelles des nœuds et à l'aide des fonctions de forme, le tenseur de déformation est déterminé en tous points de l'objet. Le tenseur des contraintes en est ensuite déduit en fonction des propriétés du matériau. Ce tenseur permet de calculer les forces internes dues aux déformations, appliquées sur les degrés de liberté. Les forces externes, qui peuvent être engendrées par la gravité, des champs de forces ou des collisions avec d'autres modèles sont reportés sur les degrés de liberté. A l'aide de la matrice de masse de l'objet, l'ensemble des forces, internes et externes, permet de calculer les accélérations de chaque degré de liberté. Ces accélérations sont ensuite intégrées deux fois dans le temps afin de déterminer les vitesses, puis les nouvelles positions au temps $t + 1$ des degrés de liberté.

Chacune de ces étapes est détaillée au cours de ce chapitre.

1.3 Déplacements

Dans un repère spatial, la position d'un point constituant le solide au repos peut être exprimée par un vecteur \mathbf{x}_0 . Après déplacement, ce même point se retrouve à une nouvelle position $\mathbf{x} = (x, y, z)$. Le vecteur

$$u(\mathbf{x}) = \mathbf{x} - \mathbf{x}_0 = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$$

correspond au déplacement du point \mathbf{x} et u caractérise le champ de déplacement des particules du solide. Dans le cas de solides rigides, le déplacement des particules d'un solide peut être exprimé à l'aide d'une translation et d'une rotation. Ainsi, au temps t , la position de chaque particule peut être déterminée par la formule $\mathbf{x}_t = \mathbf{t}_t + \mathbf{R}_t * \mathbf{x}_0$, où \mathbf{t} représente la translation du repère du solide et \mathbf{R} la rotation.

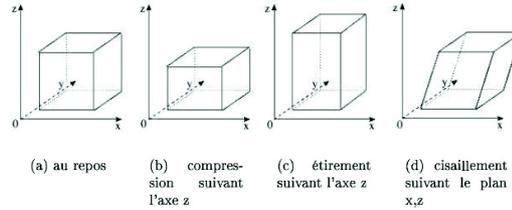


FIGURE 1.2: Déformations linéaires

Dans le cas de solides déformables, on échantillonne l'espace et des hypothèses simplificatrices sont faites sur la cinématique. Ces hypothèses dépendent de la formulation du champ de déplacement qui est différente selon les méthodes utilisées.

Par exemple, dans la méthode des éléments finis, on considère que les propriétés physiques varient continument d'un sommet d'un élément à l'autre. Cette variation dépend de la fonction d'interpolation présentée dans le chapitre 2.2.3 et peut être linéaire ou d'ordre supérieur. Ainsi, seuls les sommets des éléments ont un mouvement libre. Les autres points de l'objet dépendent entièrement de ceux-ci.

On peut ainsi se ramener à un nombre fini d'équations physiques permettant d'intégrer la position des degrés de liberté dans le temps.

1.4 Tenseur des déformations

Dans le cas de solides déformables, des déformations interviennent en conséquence du déplacement du solide sous l'action de forces. Ces déformations peuvent être exprimées par un champ de tenseurs des déformations, noté ε . Le tenseur des déformations, exprimable en une matrice symétrique 3×3 , sert à décrire l'état de déformation local résultant de contraintes (efforts internes). Les déformations sont de deux types : les dilatations (élongation/compression) et les cisaillements.

Le principe des tenseurs est simple : pour obtenir la position d'un point du solide après déformation, on s'intéresse à la première dérivée spatiale du champ de déplacement $u(x)$ défini au chapitre 1.3. On obtient

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

C'est la première partie, qui correspond à la déformation du solide, qui est contenue dans le tenseur des déformations. La seconde partie est une matrice antisymétrique et correspond à une rotation rigide si les déplacements u sont petits.

Nous présentons ici deux tenseurs des déformations qui sont couramment utilisés : le tenseur de Cauchy et le tenseur de Green-Lagrange.

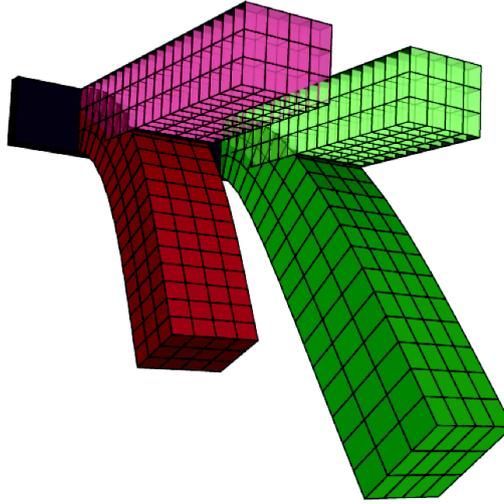


FIGURE 1.3: Les tenseurs des déformations de Cauchy et Green-Lagrange. (A gauche) le modèle non linéaire de Green-Lagrange conserve l'aspect global. (A droite) le modèle linéaire de Cauchy distord l'élément lors de grandes déformations.

1.4.1 Le tenseur de Cauchy

Le tenseur de Cauchy est de la forme :

$$\varepsilon_{Cauchy} = \frac{1}{2} (\nabla u + \nabla u^T)$$

ce qui correspond à la partie déformation de la formule précédente. L'avantage de ce tenseur est d'être invariant aux translations puisque l'on considère la dérivée du déplacement. Mais il ne l'est pas pour l'autre transformation rigide qu'est la rotation. Le comportement d'un objet animé à l'aide de ce tenseur reste réaliste tant que l'on se limite à de petits déplacements.

1.4.2 Le tenseur de Green-Lagrange

Pour pallier à ce problème, le tenseur de Green-Lagrange se place dans le repère local de l'objet. Les points du solide sont donnés dans le repère global mais la dérivée spatiale est faite suivant un repère local à l'objet. Si l'on désigne le système de coordonnées lié au repère du matériau par Ω , le tenseur de Green-Lagrange est défini par $\varepsilon_{Green-Lagrange} = \nabla_{\Omega} \mathbf{x} \nabla_{\Omega} \mathbf{x} - I_3$ avec $(\nabla_{\Omega} \mathbf{x})_{ij} = \left(\frac{\partial x_i}{\partial \Omega_j} \right)$, soit

$$(\varepsilon_{Green-Lagrange})_{ij} = \left(\frac{\partial u}{\partial \Omega_i} \cdot \frac{\partial u}{\partial \Omega_j} \right) - \delta_{ij}$$

où δ_{ij} est le symbole de Kronecker ($\delta_{ij} = 1$ si $i = j$ et 0 sinon).

On peut réécrire ce tenseur sous une forme équivalente à celle employée pour le tenseur de Cauchy :

$$\varepsilon_{Green-Lagrange} = \frac{1}{2} (\nabla u + \nabla u^T + \nabla u \nabla u^T)$$

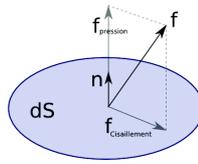


FIGURE 1.4: *Contrainte en réponse à une force*

On observe que la différence entre ces deux tenseurs est l'ajout de termes quadratiques : $\nabla u \nabla u^T$.

Ces termes rendent le tenseur de Green-Lagrange insensible aux rotations. Le tenseur de Cauchy engendre une erreur de calcul acceptable si la rotation est petite, mais devenant gênante plus la rotation augmente, comme illustré sur la barre de droite de la figure 1.3. Le tenseur de Green-Lagrange, grâce à l'ajout des termes croisés, évite cette erreur (barre de gauche de la figure 1.3). Il est donc plus précis quel que soit le mouvement de l'objet.

Un inconvénient de ce tenseur est que, suite à l'ajout de ces termes croisés (ou quadratiques), le passage du déplacement ($u(x)$) au tenseur de déformation ($\varepsilon_{Green-Lagrange}$), n'est plus linéaire.

1.5 Tenseur des contraintes

Maintenant que nous avons modélisé les déformations de l'objet, la prochaine étape est de représenter les contraintes dans l'objet. Une contrainte correspond à une force par unité de surface. En chaque point de l'objet, nous utilisons un tenseur des contraintes qui décrit la répartition des forces internes. A l'aide de ce tenseur et de la normale à la surface d'application, il est possible d'extraire la densité de force appliquée en un point de l'objet par :

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

Le tenseur des contraintes, noté $[\sigma]$, est également une matrice 3×3 symétrique. Comme le tenseur des déformations, il représente une approximation de ce qui se passe en chaque point du matériau. Il est défini selon la loi de comportement choisie. C'est cette loi qui va permettre de relier les contraintes aux déformations de l'objet, en fonction des propriétés du matériau.

1.6 Loi de comportement

La loi de comportement d'un matériau décrit la relation entre les contraintes internes de l'objet en fonction des changements d'états subis. L'étude de la résistance des matériaux classe des modèles de comportement pour les fluides et les corps solides. Il existe de nombreuses lois décrivant le comportement de fluides newtoniens ou non newtoniens, de composites mi-fluides mi-structure (poroélasticité, comportement pâteux, viscoélasticité...). Nous nous intéressons ici aux comportements élastiques et plastiques, car ils représentent la majorité des comportements des solides déformables.

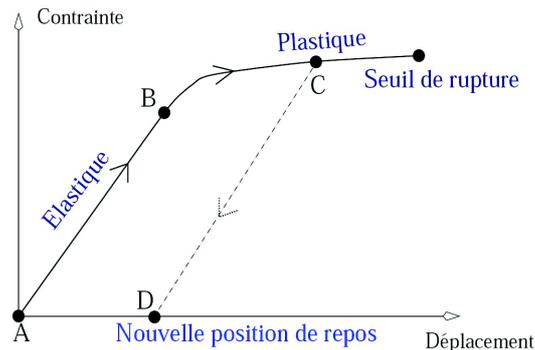


FIGURE 1.5: Loi de comportement plastique

1.6.1 La loi élastique

Pour de faibles déformations, les objets déformables sont souvent représentés par une loi élastique, car elle correspond à un grand nombre de matériaux réels. Élastique indique qu'il y a mémoire de forme, et que l'état des déformations ne dépend que de l'état présent des contraintes. C'est à dire que lorsqu'un objet est déformé et qu'aucune force ne lui est plus appliquée, il reviendra toujours à son état d'origine et ce, quelles que soient les déformations précédentes. Un point d'un objet élastique qui a été déplacé, reviendra à sa position d'origine en parcourant le chemin inverse (à une transformation rigide près).

Pour de petites déformations, cette loi de comportement peut être simplifiée par l'élasticité linéaire et être ainsi représentée par une fonction linéaire. Les contraintes sont alors linéaires aux déformations. Au delà de 20%, on considère que les déformations sont grandes et le comportement n'est plus linéaire. Il est alors souvent modélisé à l'aide d'une fonction linéaire par morceaux comme le montre la figure 1.5.

Entre A et B, le comportement est élastique. Lorsque les déformations dépassent un certain seuil C, le comportement devient plastique. Relâcher la contrainte une fois le seuil C dépassé fait alors revenir le matériau à une nouvelle position d'équilibre D, selon une pente égale à celle de son comportement élastique.

Au delà d'une contrainte trop importante, la rupture survient lorsqu'un défaut, induit par la déformation, devient critique et se propage. La ductilité du matériau désigne alors sa capacité à se déformer sans se rompre, c'est à dire sa capacité à résister à cette propagation.

La loi élastique utilisée par la suite est une approximation linéaire :

$$\{\sigma\}_{6 \times 1} = \mathbf{D}_{6 \times 6} \{\varepsilon\}_{6 \times 1}$$

Les deux tenseurs σ et ε sont exprimés sous formes vectorielles à l'aide de leurs 6 termes indépendants. La matrice D comporte donc 36 termes. La linéarité provient du fait que l'on est en faible déformation, on peut donc faire une approximation linéaire de la loi réelle (développement limité au premier ordre).

1.6.2 La loi de Hooke

Cette loi de comportement linéaire introduite précédemment est trop générale pour la plupart des simulations. En effet, la matrice D comporte 36 termes qui approximent le

comportement de tous les matériaux réels (approximés linéaires). Si l'on considère que le matériau est isotrope, c'est à dire que ses propriétés sont identiques dans toutes les directions (et c'est très souvent le cas excepté pour des matériaux fibreux tels que les muscles ou le bois), les 36 termes de la matrice \mathbf{D} peuvent être réduits à 2 à l'aide de symétries. Ces deux constantes sont appelées les coefficients de Lamé et sont notées λ et μ . La loi de comportement s'écrit alors :

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

avec I la matrice identité et $\text{tr}(X)$ la trace de X .

On peut donc mettre cette relation sous la forme matricielle :

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

où σ et ε sont des vecteurs contenant les composantes indépendantes des tenseurs de dé-

formations et de contraintes. $\{\varepsilon\} = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{pmatrix}$ avec ε_i étant la compression/élongation de

l'axe i , γ_{ij} le cisaillement entre les axes i et j , et où la matrice \mathbf{D} correspond à la matrice de rigidité du matériau pour les matériaux isotropes. \mathbf{D} est de la forme :

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

Les coefficients de Lamé peuvent être déterminés par des séries de tests sur des matériaux réels. De manière pratique, les deux valeurs mesurées pendant ces tests sont les modules de Young E (homogène à une pression) et de Poisson (sans unité). Ces valeurs sont liées aux coefficients de Lamé par les équations :

$$\lambda = \frac{\nu E}{(1 - 2\nu)(1 + \nu)}, \quad \mu = \frac{E}{2(1 + \nu)}$$

ou encore

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

Le module de Young correspond à la rigidité du matériau, exprimée en Pa, et celui de Poisson ν à sa compressibilité, comme illustré sur la figure 1.6.

Il est à noter que le système est inversible. Il est donc possible de déduire les déformations à partir des contraintes :

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

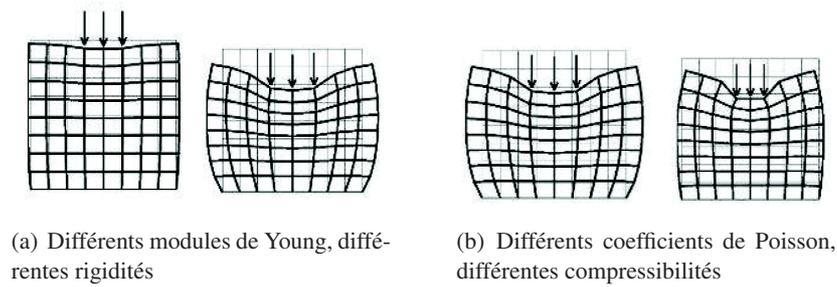


FIGURE 1.6: Le module d'Young et le coefficient de Poisson

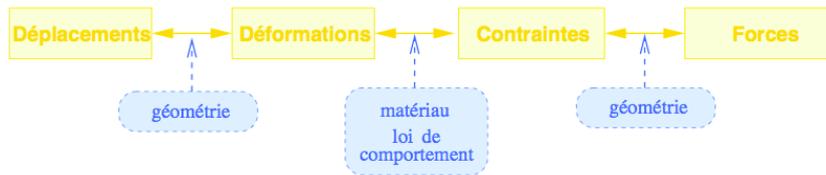


FIGURE 1.7: Relation entre déplacements, déformations, contraintes et forces, en fonction de la géométrie et des propriétés du matériau.

Dans le cas d'un matériau anisotrope, on définit la contrainte et la déformation localement par un tenseur 3×3 , le tenseur des contraintes $[\sigma_{ij}]$ et le tenseur des déformations $[\varepsilon_{ij}]$. Le comportement élastique du matériau est alors modélisé par un tenseur d'ordre 4 $[C_{ijkl}]$ contenant 81 coefficients élastiques. Le nombre de coefficients indépendants est réduit à 21 en tenant compte de la symétrie des tenseurs de contraintes et de déformations. Du fait de ces propriétés de symétrie, le tenseur C_{ijkl} peut être représenté sous la forme d'une matrice 6×6 , où les directions représentent les directions de la déformation.

1.7 Matrice de rigidité

Les forces peuvent être trouvées à partir des déplacements. Les différentes étapes précédemment présentées peuvent être résumées par la figure 1.7. Le passage des forces aux contraintes et des déformations aux déplacements dépend des tenseurs choisis. Le passage des contraintes aux déformations dépend de la loi de comportement.

1.7.1 Formulation linéaire

En utilisant les tenseurs de Cauchy, l'expression de la force en fonction des déplacements est linéaire lors de chaque étape. Dans ce cas là, il suffit de résoudre le système matriciel linéaire :

$$\mathbf{f} = \mathbf{K}\mathbf{u} = \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{u}$$

où \mathbf{K} est la matrice de rigidité de l'objet.

Cette formulation linéaire est limitative mais permet un calcul très rapide. Lors de faibles déformations, cette approximation procure des résultats tout à fait acceptables.

1.7.2 Formulation non linéaire

Le modèle non linéaire de Saint Venant-Kirchhoff est le modèle non linéaire le plus simple. Il utilise le tenseur de déformation de Green-Lagrange, invariant aux rotations, et la loi de comportement de Hooke. Il est très apprécié dans les domaines où la rapidité de la simulation prévaut mais où le modèle est soumis à de grands déplacements. Il présente un excellent compromis entre temps de calcul et précision.

D'autres modèles existent mais ne sont pas présentés ici car leurs calculs plus coûteux les rendent moins appropriés aux simulations en temps réel.

1.8 Matrice de masse

Une fois les forces internes calculées, celles-ci sont sommées aux forces externes appliquées sur le modèle. La matrice de masse met en relation les forces et les accélérations en fonction de la répartition de la matière dans l'objet. Sa taille dépend du nombre de nœuds présents dans la simulation. Elle est construite par assemblage des blocs \mathbf{M}_{ij} relatifs aux primitives i et j :

$$\mathbf{M}_{ij} = \int_{\mathcal{V}} \rho \frac{\partial \mathbf{u}^T}{\partial \mathbf{q}_i} \frac{\partial \mathbf{u}}{\partial \mathbf{q}_j} = \int_{\mathcal{V}} \rho \mathbf{J}_i^T \mathbf{J}_j d\mathcal{V} \quad (1.1)$$

où \mathbf{J}_i est la matrice Jacobienne de la primitive i :

$$\mathbf{J}_i = \frac{\partial \mathbf{u}}{\partial \mathbf{q}_i} \quad (1.2)$$

La matrice de masse est donc construite à partir de la Jacobienne qui encode la relation entre le point du modèle et le nœud qui l'influence. L'intégrale volumique est réalisée de manière discrète sur le modèle par échantillonnage. La Jacobienne est calculée pour chaque voxel influencé par un nœud et cette contribution, pondérée par la masse volumique du voxel, est ajoutée au bloc correspondant.

Les blocs de la matrice de masse encodent la distribution de matière (géométrie de l'objet) en rapport avec les nœuds. En théorie, la matrice d'inertie devrait donc être recalculée à chaque pas de temps, afin de tenir compte de la géométrie actuelle de l'objet. Mais l'intégrale volumique nécessite le parcours de l'ensemble des échantillons et cette opération est coûteuse. Pour accélérer le temps de calcul, la masse peut être pré-calculée en position non déformée de l'objet. Ceci n'est valable que pour de petits déplacements et déformations. Dans le cas où les déformations restent faibles mais où les déplacements sont importants, il est possible de pré-calculer la masse et de transformer chaque bloc de celle-ci par la rotation courante des degrés de liberté correspondants. Cette approximation couramment réalisée réduit nettement le temps de simulation sans nuire aux résultats.

1.9 Intégration dynamique

Dans le cas de simulations interactives, on ne cherche pas explicitement un état d'équilibre, mais plutôt l'évolution du système dans un avenir proche. En effet, puisque l'utilisateur interagit avec la scène, les forces évoluent de manière imprévisible. Il est donc nécessaire d'intégrer le temps et ainsi d'aborder le problème en dynamique.

Pour cela, il faut résoudre une équation différentielle du second ordre de la forme :

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}$$

où \mathbf{M} est la matrice de masse, \mathbf{C} la matrice d'amortissement, \mathbf{K} la matrice de rigidité de l'objet dont \mathbf{u} représente les déplacements et \mathbf{f} les forces qui lui sont appliquées.

Les composantes de ces matrices dépendent de la géométrie de l'objet et doivent donc être recalculées après une déformation mais une grande partie des matrices peut être pré-calculée.

Le calcul des matrices \mathbf{M} et \mathbf{K} est décrit aux chapitres 1.8 et 1.7. Pour le calcul de la matrice d'amortissement \mathbf{C} , le plus simple est d'utiliser l'amortissement de Rayleigh. Dans ce cas :

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$$

où α et β sont les coefficients de Rayleigh.

La résolution de cette équation différentielle partielle se fait à chaque pas de temps en intégrant le temps par la formule :

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

où les degrés de liberté \mathbf{q} ainsi que leurs dérivées $\dot{\mathbf{q}}$ et $\ddot{\mathbf{q}}$ sont connus à chaque pas de temps et où Δt est un pas de temps.

On ne sait pas calculer analytiquement cette intégrale mais il est possible de l'approcher numériquement. Pour cela, quantité de méthodes d'intégration ont été développées, qui diffèrent par leur précision, leur temps de calcul et leur stabilité. On peut tout de même scinder ces méthodes en deux familles :

- Les méthodes explicites : qui sont plus simples mais moins stables. Le calcul des dérivées secondes est effectué à partir des dérivées premières et secondes de l'état actuel. Les dérivées premières et l'état du pas de temps suivant sont calculés à partir de cette dérivée seconde. Généralement, ces techniques ajoutent de l'énergie au système qui a tendance à diverger. Pour conserver la stabilité de la simulation, il est donc nécessaire de choisir un petit pas de temps. Par exemple, la technique de résolution d'« Euler explicite », qui suppose que la vitesse ne varie pas du début à la fin du pas de temps, est de la forme :

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

- Les méthodes implicites : qui sont plus stables mais dont le coût en terme de temps est plus important. Le calcul des dérivées secondes est effectué à partir des dérivées premières et secondes de manière itérative. Chaque itération raffine les valeurs des dérivées secondes jusqu'à obtenir un état stable. Le choix de ce critère de stabilité détermine le compromis précision/temps de calcul. Les dérivées du prochain pas de temps sont ensuite prises en compte pour l'intégration temporelle du pas de temps actuel. Ce qui assure la stabilité de ces méthodes et permet l'utilisation de grands pas

de temps. La convergence vers la position d'équilibre est néanmoins plus longue. Par exemple, la résolution d'« Euler semi-implicite » est de la forme :

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

équivalent au système d'équation (avec amortissement de Rayleigh [BW98]) :

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

La méthode Euler implicite sera celle utilisée par la suite car elle offre un meilleur compromis stabilité/vitesse/précision d'après [BW98].

1.10 Conclusion

La théorie des milieux continus permet de simuler le comportement de fluides ou de solides en fonction de leurs propriétés matérielles. Différents modèles informatiques peuvent être élaborés à partir de cette théorie, axés sur la vitesse de simulation ou la précision des résultats. Certains choix sont donc faits concernant :

- les approximations du modèle, telles que la linéarisation des tenseurs ou de la matrice de rigidité ou les optimisations de la matrice de masses, pour celles présentées ici.
- l'échantillonnage de la matière, dans la mesure où la théorie décrit des milieux continus alors que l'information est traitée de manière discrète par les ordinateurs.
- les nœuds (ordre : points, repères...) ainsi que les fonctions de forme.

Le chapitre suivant présente les techniques existantes dérivées de cette théorie afin de dégager les notions importantes requises pour un modèle de simulation par informatique et les difficultés et avantages de chacun de ces modèles.

Chapitre 2

Modèles déformables discrets

Contents

2.1	 Systèmes masses-ressorts	20
2.1.1	Force de rappel	20
2.1.2	Force d'amortissement	20
2.1.3	Force extérieure	21
2.1.4	Discussion	21
2.2	 Éléments finis	22
2.2.1	Principe	22
2.2.2	Maillage	22
2.2.3	Fonctions d'interpolation	22
2.2.4	Cages de déformation	23
2.2.5	Multirésolution	24
2.2.6	Adaptativité	25
2.2.7	Conclusion	26
2.3	 Méthodes particulières (alias Meshless)	26
2.3.1	Principe	26
2.3.2	Techniques basées sur les déplacements	26
2.3.3	Techniques basées sur l'énergie de déformation	27
2.3.4	Conclusion	28
2.4	 Enrobage (alias Skinning)	28
2.4.1	Principe	29
2.4.2	La pondération	30
2.4.3	Interpolation de transformations rigides	30
2.4.4	Conclusion	32
2.5	 Les fonctions de forme	32
2.5.1	Interpolation dans un élément	32
2.5.2	Interpolation sans élément	34
2.6	 Simulation de découpe	38
2.6.1	Modèles physiques maillés	38
2.6.2	Modèles meshless	40
2.7	 Conclusion	41

La simulation de modèles déformables au comportement physiquement réaliste attire une attention continue dans le domaine de l'informatique graphique, depuis le travail initial de Terzopoulos [TPBF87]. Le lecteur peut se reporter à l'excellent état de l'art de [NMK⁺05] pour une introduction à ce domaine.

Les techniques dynamiques les plus courantes du domaine de la simulation temps réel sont présentées ici. Leurs avantages et inconvénients sont discutés en terme de compromis entre précision et temps de calcul et de gestion des changements topologiques pour les applications destinées à la découpe. La technique cinématique du skinning est aussi présentée pour certains de ses avantages par rapport aux techniques dynamiques. Enfin, le mélange de primitives dynamiques et l'interpolation sur leur domaine sont discutés en fin de chapitre.

2.1 Systèmes masses-ressorts

Les techniques masses-ressorts ont vu le jour à la fin des années 80, suite au travail de [TPBF87]. Les degrés de liberté utilisés sont des points auxquels sont attribués des masses ponctuelles. Ceux-ci sont reliés par des ressorts qui encodent les propriétés internes du matériau (force de rappel et d'amortissement).

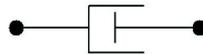
2.1.1 Force de rappel



La force de rappel sert à simuler le comportement élastique de l'objet. Elle tend à ramener les deux extrémités i et j d'un ressort à la même distance. Cette force de rappel, pour un ressort d'une longueur initiale l_0 , est proportionnelle à son allongement relatif et à sa raideur k dans la direction \mathbf{j}_i :

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

2.1.2 Force d'amortissement



La force d'amortissement sert à réduire le mouvement relatif des deux extrémités i et j du ressort. Elle est proportionnelle à la viscosité ν et à sa vitesse d'allongement dans la direction \mathbf{j}_i :

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

2.1.3 Force extérieure

Les forces appliquées sur un point du modèle autre que les degrés de liberté peuvent être réparties sur ces derniers à l'aide de coordonnées barycentriques. Les forces extérieures appliquées en chaque sommet sont ensuite sommées pour déterminer la force extérieure globale F_{ext} utilisée lors de la résolution de l'équation différentielle :

$$M\ddot{u} + C\dot{u} + Ku = F_{ext}$$

2.1.4 Discussion

Les systèmes masses-ressorts sont simples, rapides et polyvalents. Ils sont donc très appréciés dans le domaine de la synthèse d'images et ils l'ont aussi été pour la simulation. Notamment la simulation chirurgicale où les changements de topologie des objets - lors d'une découpe par exemple - sont très facile à prendre en compte sur un réseau de ressorts.

Toutefois, à cause de la simplicité des lois physiques et de sa structure, ce modèle s'avère limité et ne peut pas simuler fidèlement les milieux volumiques. Il arrive que l'objet trouve une position d'équilibre différente de celle d'origine, physiquement impossible. Cette situation survient, par exemple, lorsque le modèle est comprimé. Physiquement, si l'on suppose que la force entraînant la compression est infinie, la force de retour le serait aussi car, dans la réalité, un matériau devient incompressible à partir d'un certain point. Dans la simulation informatique, il arrive que l'objet comprimé s'inverse, comme l'illustre la figure 2.1. Cela provient du fait que le solveur travaille avec un pas de temps discret et que, dans le cas des systèmes masses-ressorts, les ressorts ne contiennent qu'une information de longueur et ne peuvent donc pas prévenir le retournement de l'objet. Une position d'équilibre respectant la longueur des ressorts est trouvée, différente de la position initiale.

Du fait que les forces engendrées par les ressorts se basent uniquement sur la distances entre les points, ce modèle ne peut pas non plus garantir la conservation du volume. Des améliorations de ce modèle ont été faites dans ce sens. Ainsi par exemple, [HJCW06] garantissent la conservation du volume global de l'objet pour l'animation de muscles à l'aide de système masses-ressorts. La conservation de volume de chaque élément n'est pas garantie car elle demande la résolution d'un système linéaire dont la taille augmente linéairement avec le nombre d'éléments. C'est pourquoi uniquement le volume global de l'objet est préservé, afin d'obtenir des résultats temps réels.

Un dernier inconvénient de cette méthode est qu'elle est difficilement paramétrable à partir d'une scène réelle. En effet, nous avons vu que les expériences menées pour trouver les propriétés physiques d'un matériau définissent aisément les modules de Young et de Poisson. Les constantes de raideur des ressorts d'un système à animer sont difficiles à calculer à partir de ces paramètres.

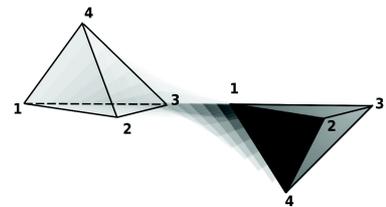


FIGURE 2.1: Nouvelle position d'équilibre d'un tétraèdre, à l'envers.

2.2 Éléments finis

Comme nous l'avons vu dans le chapitre précédent, simuler une scène revient à trouver une solution à une équation différentielle, ce qui n'est pratiquement jamais réalisable de manière analytique. La méthode des éléments finis permet d'approcher numériquement cette solution. Elle est utilisée pour représenter le comportement dynamique de certains systèmes physiques - comme par exemple les systèmes mécaniques, thermodynamiques ou acoustiques - ou encore pour modéliser le comportement d'un fluide.

De manière générale, les seules conditions de cette méthode est que l'équation différentielle dont on cherche la solution soit continue avec des conditions aux bords (qui assurent l'existence et l'unicité de la solution).

2.2.1 Principe

Pour résoudre numériquement une équation différentielle partielle (EDP), la méthode des éléments finis nécessite une transformation du problème à résoudre. Cette phase correspond à la mise sous forme variationnelle du problème d'EDP. Le problème initial est de dimension infinie et la méthode des éléments finis consiste à modéliser un problème analogue en dimension finie, à partir d'une discrétisation du domaine où est définie l'EDP. De manière simplifiée, l'idée est de découper l'objet en un nombre fini d'éléments, dont les sommets sont appelés nœuds. L'équation d'équilibre du système doit être vérifiée sur chacun des nœuds. A ce système d'équations s'ajoute celles induites par les contraintes et les interactions entre les nœuds. La résolution de ce système permet d'obtenir une approximation des propriétés physiques de l'objet.

La qualité de la solution dépend du maillage (nombre et dimension des éléments), du choix de la fonction d'interpolation et des conditions de continuité que l'on impose aux frontières des sous domaines (C^0 , C^1 , ...).

2.2.2 Maillage

Il existe plusieurs façons de mailler un objet. Le maillage peut être linéique, surfacique ou volumique selon la dimension de l'objet. Habituellement, on choisit un maillage carré ou triangulaire si le maillage est surfacique, tétraédrique ou hexaédrique pour un maillage volumique.

Il est aussi possible de découper l'objet en utilisant différents maillages (en maillant une partie avec des triangles et une autre avec des quadrilatères mais des complications apparaissent ensuite). Il n'est pas non plus nécessaire que le maillage soit régulier et de meilleurs résultats sont obtenus en resserrant le maillage autour des zones d'intérêt. Cependant, il ne faut pas rendre le maillage trop irrégulier.

Enfin, plus le maillage est resserré, plus les résultats sont exacts mais coûteux.

2.2.3 Fonctions d'interpolation

Le principe de la méthode des éléments finis est d'interpoler les propriétés physiques de chacun des sommets pour qu'elles varient continument dans l'espace. Plusieurs choix

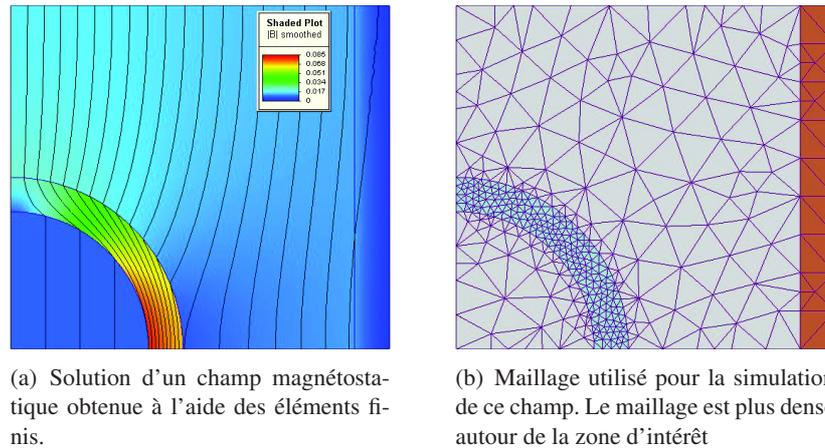


FIGURE 2.2: Un exemple de modélisation effectuée par éléments finis.

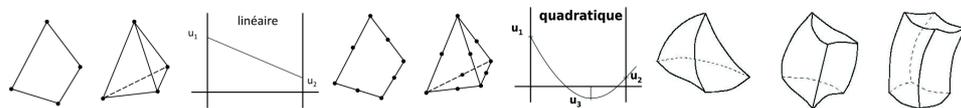


FIGURE 2.3: Courbes d'interpolation linéaire et quadratique. Cette dernière nécessite plus de nœuds et des déformations plus complexes.

sont alors possibles selon le degré de continuité désiré. La fonction de forme la plus simple, communément utilisée, interpole linéairement les valeurs d'un sommet à l'autre, à l'aide de coordonnées barycentriques.

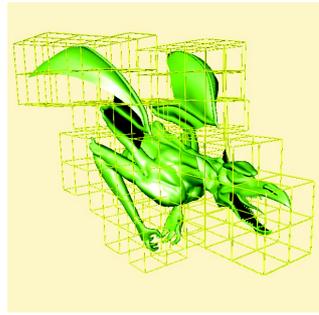
Une autre solution couramment utilisée consiste à créer une base de fonctions d'interpolation sur une maille topologiquement équivalente à celle choisie, puis de calculer les propriétés physiques des sommets, et de transformer enfin cette base pour qu'elle devienne une base de fonctions d'interpolation sur les mailles réelles. Cette méthode demande moins de calcul.

Il est aussi possible de choisir une interpolation quadratique qui nécessite d'ajouter un sommet entre chacun des sommets initialement définis. Plus le degré d'interpolation est élevé, plus l'élément peut se déformer, moyennant une résolution de l'EDP plus longue.

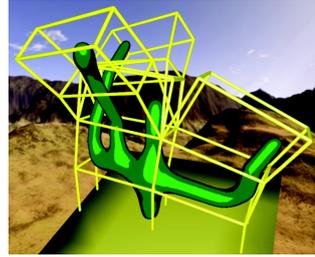
Les fonctions de forme utilisées dans les différentes techniques de simulation d'objets déformables sont plus largement discutées en 2.5.

2.2.4 Cages de déformation

Pour simuler un modèle, il est donc nécessaire de produire un maillage volumique équivalent. Cette tâche préliminaire est difficilement automatisable et détermine la stabilité et la précision des résultats. Les cages de déformation (ou "Free From Deformation" - FFD) ont été introduites par [Bar84]. Le modèle à simuler est plongé dans une grille régulière d'hexaèdres. La simulation est effectuée sur les sommets de cette grille, chaque hexaèdre définissant un élément fini. Les sommets du modèle initial sont associés à ceux



(a) Modèle simulé par cage de déformation, comprenant des parties surfaciques (ailes) et volumiques.



(b) Cage de déformation avec nœuds virtuels : la topologie de l'objet est respectée.

FIGURE 2.4: *Simulation par cages de déformation*

de la grille par coordonnées barycentriques afin de déplacer les sommets du modèle initial en fonction des déformations de la grille. L'utilisation d'une grille régulière facilite la génération du maillage et évite les instabilités numériques de celui-ci dues aux éléments mal conditionnés. De plus, elle permet le découplage du modèle simulé et du modèle physique, permettant la simulation d'un objet détaillé par une grille grossière. Un autre avantage est que la topologie du modèle initial est sans importance : celui-ci peut contenir des parties volumiques ou surfaciques, tel que les ailes du dragon de la figure 2.4(a), tous simulés de manière volumique par les éléments.

Néanmoins, la structure régulière de la grille ne dépend pas des propriétés de la matière et certaines parties rigides peuvent chevaucher plusieurs éléments. Ce qui entraîne un couplage fort dans la matrice de raideur entre ces éléments et peut mener à des instabilités numériques ; la matrice permettant de résoudre l'équation différentielle devenant singulière. Ce problème est en effet commun à toutes les techniques dont l'échantillonnage ne se base pas sur les propriétés de la matière et il a de grandes chances de survenir lors de l'utilisation de grille régulière. Certains solveurs numériques remédient à cet inconvénient moyennant un temps de calcul plus importants, inapproprié au calcul en temps réel.

2.2.5 Multirésolution

La technique des cages de déformation à l'aide d'une grille régulière a aussi simplifié la mise en place d'approches multirésolutions, réduisant ainsi le nombre d'éléments simulés. [NPF06] propose ainsi de construire une grille d'éléments finis à plusieurs niveaux de résolution. Le niveau le plus fin est une discrétisation de l'objet qui fournit les propriétés physique de l'objet (masse volumique, module de Young, coefficient de Poisson, etc). Dans le domaine de la simulation chirurgicale, cette discrétisation peut être obtenue suite à une acquisition d'IRM par exemple. Des niveaux plus grossiers d'éléments finis sont alors construits sur les premiers par assemblage. Afin de prendre en compte correctement la répartition de matière au sein des éléments, les matrices de masses et de raideurs des éléments plus grossiers sont assemblées suivant les matrices des blocs plus fins. Plus de détails sont fournis dans [NPF06]. Le calcul des coordonnées barycentriques des sommets

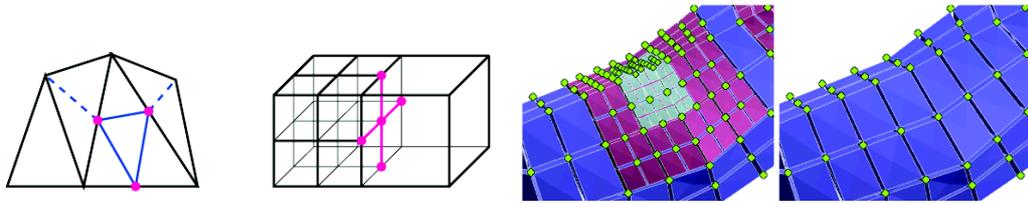


FIGURE 2.5: *A gauche : jonctions en T apparaissant lors de la subdivision d'un élément. A droite : raffinement des fonctions de base dans [GKS02].*

plongés dans la grille repose sur la répartition et les propriétés de la matière dans les éléments. Ainsi, la présence de parties plus ou moins rigides dans le même élément grossier est correctement prise en compte.

Néanmoins, cette méthode présente quelques difficultés à simuler des géométries complexes à l'aide d'éléments très grossiers ; Lorsque le niveau de multirésolution est trop élevé, la topologie de la grille grossière ne correspond plus nécessairement à la topologie de l'objet initial. L'utilisation de nœuds virtuels [NKJF09b] permet l'animation de maillages à géométrie complexe à l'aide de grilles grossières, en respectant leur topologie, comme illustré sur la figure 2.4(b).

La résolution de l'équation différentielle par le solveur est coûteuse. Plus le nombre de degrés de liberté est important, plus le système d'équations est important et la simulation lente. Afin de disposer de peu de nœuds, les fonctions de forme des éléments, encodées par les matrices assemblées, ont été complexifiées. Ainsi, le découplage des modèles visuels et physiques et l'élaboration de meilleures fonctions de formes permettent de simuler des scènes volumineuses et détaillées en temps réel.

2.2.6 Adaptativité

Toujours dans l'optique d'alléger les simulations, des travaux sur l'adaptativité du modèle physique ont vu le jour. L'idée est de débiter la simulation de l'objet avec un modèle physique grossier. Le nombre de degrés de liberté varie au cours de celle-ci en fonction de critères afin de raffiner les résultats dans les zones d'intérêt.

Des approches adaptatives ont été proposées sur les modèles physiques multirésolutions et FFD, disposant d'une structure régulière. Différents critères existent, privilégiant précision ou temps de calcul. La plupart des critères reposent sur la déformation du modèle (courbure, tenseur de déformation ou de contrainte [WDGT01, DDCB01]). Ces méthodes affinent les résultats au cours de la simulation, quelque soit le temps d'exécution. Mais d'autres méthodes où le temps réel doit être garanti, dites en temps critique, répartissent les degrés de liberté insérés de manière à améliorer la précision de la simulation tout en respectant un temps d'exécution fixé ([DMG05]). On retrouve aussi ce type de critère en détection de collisions ([DO00, KZ03, MO05]).

Les méthodes classiques d'éléments finis furent aussi étendus par des algorithmes adaptatifs. La plupart de ces techniques ([Bey00, RI97]) subdivisent l'élément par insertion de nœuds sur les arêtes des éléments existants. Ce qui entraîne la création de jonctions en T , voir figure 2.5, qui posent des problèmes de connectivité lors de la résolution de l'équation différentielle, rendant deux éléments voisins incompatibles. Dans [GKS02], Grinspun

règle ce problème en insérant des degrés de liberté à l'intérieur de l'élément (mais pas sur les frontières) et en remplaçant la fonction de base existante par plusieurs fonctions recouvrantes. Ce calcul reste local à l'élément, et conserve des frontières compatibles. D'autres méthodes [WDGT01] remaillent localement le modèle physique, ce qui s'avère être une opération complexe et coûteuse. De plus, des éléments mal conditionnés peuvent apparaître, entraînant des instabilités, et il est impossible de tirer parti d'éventuels pré-calculs.

2.2.7 Conclusion

La méthode des éléments finis dispose de nombreuses variantes qui en font une méthode robuste, physiquement précise et elle a été largement validée au cours de ces dernières années. Depuis [CDC⁺96, DCA99] et avec l'amélioration des ressources matérielles, cette méthode s'exécute en temps réel sur des scènes relativement complexes.

Néanmoins, la mise en place des simulations ainsi que toutes les améliorations proposées, telles que les changements topologiques ou l'adaptativité, se heurtent à la difficulté de devoir traiter un maillage. Celui-ci devant être de qualité afin de garantir de bons résultats et la stabilité de la méthode.

De nouvelles approches, dites meshless et introduites dans la section suivante, sont récemment apparues et présentent l'avantage de ne nécessiter aucune connectivité entre les degrés de liberté.

2.3 Méthodes particulières (alias Meshless)

Les méthodes sans maillage dites "meshless" utilisent des points de contrôle non structurés comme degrés de liberté. Dans le domaine graphique, les méthodes meshless ont initialement été introduites pour la simulation de fluides et ont ensuite été étendues à la mécanique des solides [MKN⁺04, GP07]. Cette technique suscite de plus en plus d'intérêt dans le milieu de la synthèse d'images temps réel. Dans ce chapitre, le principe de cette méthode ainsi que les cas les plus généraux sont présentés. Pour plus de détails, le lecteur peut se reporter à l'excellent rapport de [FM03].

2.3.1 Principe

Cette méthode est à base de particules. L'objet à animer est tout d'abord échantillonné par un nuage de points auxquels on attribue une masse. Chaque point de contrôle a une influence donnée qui décroît généralement suivant la distance à celui-ci. L'intégration est ensuite effectuée de manière à minimiser une fonction d'erreur basée sur le déplacement des points ou l'énergie de déformation. Ainsi, aucune information de connectivité entre les points n'est nécessaire. Les méthodes d'approximation ou d'interpolation classiques ont été étudiées pour la simulation physique comme les fonctions de Shepard, les fonctions à bases radiales ou les moindres carrés (voir [FM03] et le chapitre 2.5.2 pour plus de détails).

2.3.2 Techniques basées sur les déplacements

Dans les techniques meshless, certaines méthodes développées pour les jeux vidéos se concentrent uniquement sur la vitesse d'exécution. [MHTG05] présente une technique où

l'objet simulé adopte un comportement visuellement plausible en minimisant la déformation mais qui ne repose pas sur la physique.

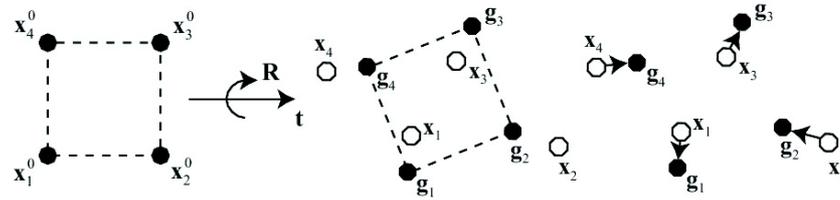


FIGURE 2.6: Pipeline d'une technique meshless basée sur les déplacements.

L'image 2.6 illustre le pipeline de la méthode.

A gauche : La position initiale x_i^0 de chaque points est enregistrée.

Au milieu : Les forces externes appliquées sur l'objet, dues à la gravité ou à une réponse à des collisions, entraînent un déplacement des particules en x_i . La transformation rigide optimale (couple translation / rotation) est déterminée. La translation est calculée suivant le déplacement du centre de masse des points x_i . Les vecteurs $\mathbf{q}_i = \mathbf{x}_i - \mathbf{x}_{\text{cm}}$ et $\mathbf{p}_i = \mathbf{x}_i - \mathbf{x}_{\text{cm}}$ (position des points relatifs à leur centre de masse) permettent de calculer la transformation affine \mathbf{A} minimisant $\sum_i m_i (\mathbf{A}\mathbf{q}_i - \mathbf{p}_i)^2$ (se reporter à [MHTG05] pour plus de détails). La rotation est ensuite déduite par décomposition polaire de la matrice affine \mathbf{A} .

A droite : Les points g_i ainsi obtenus sont utilisés pour engendrer des forces corrigeant le déplacement des points x_i . La pondération plus ou moins importante de ces forces permet de varier d'un comportement rigide à élastique.

Pour des déformations plus complexes, une combinaison de la matrice affine \mathbf{A} et de la rotation est utilisée (au lieu d'une simple rotation) pour corriger les points, produisant des déformations affines de l'objet. Une troisième version permet de calculer la déformation quadratique formée par les points g_i lors du calcul de \mathbf{A} (se reporter à l'article pour plus de détails). Cette matrice peut être combinée avec la matrice de rotation pour obtenir des déformations quadratiques.

Cette technique permet une grande rapidité d'exécution dans le cas rigide et affine. Les déformations quadratiques demandent tout de même plus de calculs mais certaines parties peuvent être pré-calculées. Néanmoins, elle n'est pas capable de modéliser des matériaux réels, étant basées sur la géométrie et les résultats ne résultent pas d'un calcul de dynamique physique.

2.3.3 Techniques basées sur l'énergie de déformation

Une approche basée sur la minimisation d'une énergie (incluant l'énergie de déformation) est présenté par [AOW⁺08]. Selon le schéma classique, un champ de déplacement est choisi. La dérivation de celui-ci permet de calculer la matrice de masse et la matrice de raideur, à l'aide des propriétés de la matière. Le déplacement des points au cours de la simulation permet d'obtenir l'énergie de déformation pour calculer les forces internes. Ces forces sont ensuite prises en compte dans le bilan des forces avant l'intégration temporelle pour déterminer les positions du prochain pas de temps. [AOW⁺08] ajoute à ce schéma des contraintes (de positionnement, de raideur et de préservation de volume) et définit donc des

énergies supplémentaires.

Le modèle est échantillonné par un nuage de N points x_i . Le champ de déformation est défini par

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^N \Phi_i(\mathbf{x}) \mathbf{u}_i$$

$\Phi_i(x)$ étant des fonctions de forme associées aux points x_i , et \mathbf{u}_i des vecteurs de déplacement des points x_i . Certains points x_i du modèle sont fixés par l'utilisateur (de manière interactive ou prédéfini temporellement). Pour des facilités de notations, soit $\mathbf{f}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(x)$ un mapping couplant les positions initiales des points et leur déplacement. Une énergie globale de déformation est définie suivant :

$$\mathbf{E} = \alpha_1 \mathbf{E}_{positions} + \alpha_2 \mathbf{E}_{raideur} + \alpha_3 \mathbf{E}_{volume}$$

où α_1 , α_2 et α_3 modulent la contribution des contraintes. Chacune des trois énergies est définie de manière classique (distance au point de contraintes ou dérivation du champ de déformation) à partir de $\mathbf{f}(\mathbf{x})$. La minimisation de cette énergie, par résolution d'un système non linéaire, permet de déterminer les déplacements \mathbf{u}_i et ainsi les nouvelles positions, en respect des contraintes imposées.

2.3.4 Conclusion

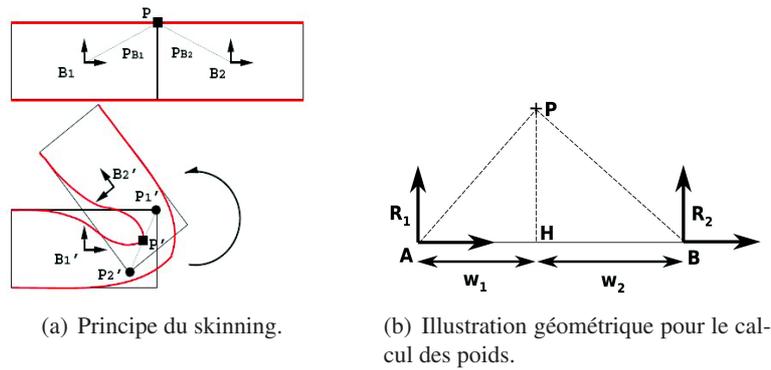
Les deux intérêts majeurs de cette technique sont la rapidité et qu'elle ne nécessite aucune information de connectivité, la rendant ainsi attractive pour la gestion des changements topologiques.

Les techniques basées sur les déplacements, extrêmement rapide, entraînent des résultats suffisants pour l'animation mais trop imprécis pour la simulation. Celles basées sur l'énergie de déformation présentent des résultats physiques visuellement plausibles mais sont incapables de modéliser des matériaux complexes et les matériaux actuellement traités sont supposés uniformes.

Malgré l'ajout de flexibilité procurée par l'absence d'éléments, ces techniques nécessitent que les régions d'influence des fonctions d'approximation des points de contrôle se recouvrent. En particulier, l'approximation des rotations requiert l'influence d'au moins quatre points non coplanaires. [MKB⁺10] atténuent ce problème grâce à une méthode des moindres carrés étendue à l'utilisation de repères au lieu de points. Ceux-ci sont utilisés comme nœuds dynamiques mais aussi pour l'évaluation de la déformation pour le calcul des forces internes. Ils se focalisent essentiellement sur une nouvelle mesure de différentiation de la déformation, appelée *élaston*, combinée à des fonctions de poids à bases radiales et un échantillonnage dense. Néanmoins, le nombre élevé d'élastons requis pour la simulation de la scène entraîne un temps de calcul important.

2.4 Enrobage (alias Skinning)

Les fonctions d'approximation des techniques meshless estiment un champ de déformation continu aux positions influencées par au moins quatre points de contrôle tandis que les fonctions utilisées avec les éléments finis interpolent la déformation au sein de l'élément. Une troisième approche, inexploitée jusqu'à présent dans les simulations physiques,



(a) Principe du skinning.

(b) Illustration géométrique pour le calcul des poids.

FIGURE 2.7: *Positionnement d'une surface par skinning.*

consiste à directement interpoler les transformations rigides. En animation, cette technique est connue sous le nom de skinning, dans laquelle l'objet déformable est contrôlé par un mélange de transformations rigides des os [MTLT88]. Elle est souvent utilisée pour l'animation de personnages, dans les jeux vidéos et les effets spéciaux, appréciée pour son aspect intuitif.

2.4.1 Principe

Le but de cette technique est de définir l'animation d'un personnage en manipulant son squelette ; c'est à dire d'exprimer la position d'un maillage surfacique par un ensemble de degrés de liberté comprenant la position et l'orientation, couramment représentés par des repères.

La position de 0p , un point de la surface exprimé dans le repère global, est fonction d'une combinaison linéaire des différents degrés de liberté qui l'influencent :

$${}^0p = \sum_i \omega_i \mathbf{A}_i {}^i p = \sum_i \omega_i \mathbf{A}_i \bar{\mathbf{A}}_i^{-1} {}^0 \bar{p}$$

où ω_i est un poids (i.e. l'influence du repère i associé à \mathbf{p}), \mathbf{A}_i la transformation rigide du repère i , et $\bar{\mathbf{A}}_i$ la transformation initiale du repère. Le point \mathbf{p} est donc exprimé par une matrice correspondant au mélange des transformations relatives des os influents.

Un exemple de transformation est illustré sur la figure 2.7(a). Les poids du point \mathbf{P} sont d'un demi pour chacun des repères $B1$ et $B2$ et varient continuellement pour les autres points de la surface. Une déformation continue est ainsi obtenue lorsque les degrés de liberté sont déplacés.

Cependant, on observe sur cette figure un pincement du modèle et une réduction du volume du maillage. La formulation du skinning décrite ici présente des problèmes lors de grandes déformations : lors de torsions ou de flexions importantes, des artefacts apparaissent (aussi illustré sur les images de gauche de la figure 2.9). Ceci est dû au fait que le mélange linéaire de matrices de transformations n'entraîne pas une transformation continue d'un état à l'autre. Ce qui s'avère problématique en pratique car de telles déformations surviennent couramment à chaque articulation d'un personnage. Dans l'industrie, les graphistes compensent ces artefacts en modifiant manuellement les poids d'influence

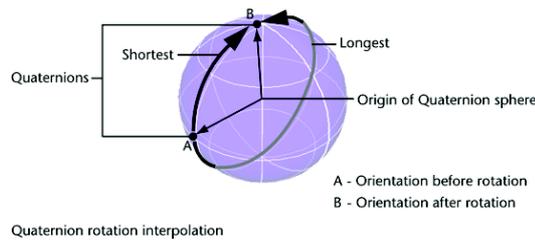


FIGURE 2.8: Espace de rotation unitaire et encodage d'une rotation par un quaternion.

afin d'obtenir une déformation correcte, ce qui demande un travail important. Des solutions portant sur le mélange des transformations rigides ont été proposées pour pallier à ce problème et sont présentées à la section 2.4.3.

2.4.2 La pondération

La pondération détermine l'influence des os sur la peau. Une façon naïve pour fixer les poids ω_i serait de les initialiser à une valeur uniforme, comme dans l'exemple ci-dessus : $\omega_i = \frac{1}{N}$ où N est le nombre de degrés de liberté qui expriment le point. Cette pondération entraîne des discontinuités lors de changement des zones de recouvrement de l'influence des os.

Les pondérations classiques reposent ainsi sur la distance du point à l'os : les poids sont importants pour les points proches de l'os et décroissent jusqu'à zéro, ce qui assure la continuité des déformations lors de changements des zones de recouvrement. Parmi les pondérations classiques, on trouve (voir figure 2.7(b)) :

- l'abscisse curviligne où les poids sont interpolés linéairement d'un degré de liberté à l'autre :

$$\omega_i = \|\mathbf{AH}\| = \frac{\|\mathbf{AB}\| \|\mathbf{AP}\|}{\|\mathbf{AB}\|^2}$$

- des pondérations basées sur l'inverse de la distance :

$$\omega_i = \frac{1}{\|\mathbf{AP}\|}, \text{ ou } \omega_i = \frac{1}{\|\mathbf{AP}\|^2}$$

- des interpolations splines (Bézier, hermite...) sur un nombre plus important de repères présentant des continuités d'ordre supérieur.

Une discussion plus large sur le choix de pondération se trouve à la section 2.5.

2.4.3 Interpolation de transformations rigides

Comme vu précédemment dans la section 2.4, l'interpolation linéaire de matrices représentant des transformations rigides ne permet pas d'obtenir une interpolation lisse du premier état au second mais un comportement chaotique. Si l'on considère l'espace de rotation unitaire, représenté par une sphère sur la figure 2.8, interpoler linéairement les termes

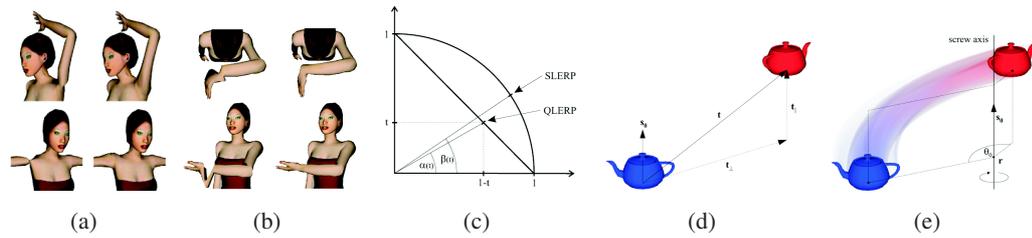


FIGURE 2.9: Comparaison du ScLERP avec un mélange linéaire (a). Comparaison du ScLERP avec le mélange logarithmique (b). Interpolation de transformations rigides linéaire (c), puis par décomposition (d), enfin par les ScLERP (e).

des matrices de rotation revient à passer par la droite coupant la sphère d'un point à l'autre. Les matrices résultantes n'encodent pas des transformations rigides et l'interpolation ne se fait pas à vitesse constante, voir la figure 2.9(c).

De nombreuses solutions ont été proposées ces dernières années pour pallier à ce problème. [Ale02] proposait d'interpoler des matrices logarithmiques. Cette méthode permettait d'obtenir de bons résultats lors de faibles rotations pour un coût modique : l'espace logarithmique étant proche de l'espace sphérique pour de petits déplacements. Néanmoins, cette technique mène à des aberrations lors de grandes déformations.

[Kea05] et [Hej04] ont décidé de décomposer la matrice de transformation en un couple (translation, quaternion) pour ensuite les mélanger, comme illustré sur la figure 2.9(d). D'autres techniques similaires reposant sur la décomposition des matrices de transformations en différentes transformations rigides ont été avancées. Mais dans toutes ces techniques, la décomposition était arbitraire et l'interpolation des translations et des rotations se trouvaient découplées.

L'interpolation linéaire de transformations rigides n'est pas parfaite car ce n'est pas une opération intrinsèque au groupe de rotation $SO(3)$. La solution exacte théorique est donnée par la méthode dite SLERP (Spherical Linear Interpolation) [Sho85], qui permet d'interpoler linéairement deux rotations encodées par deux quaternions unitaires.

Dans [KCZO07], Kavan étend cette formulation au ScLERP (Screw Linear Interpolation), qui permet d'interpoler linéairement plusieurs transformations rigides à l'aide des quaternions duaux. Un descriptif plus complet est publié dans [KCZO08]. Les quaternions duaux sont des paires de quaternions munis d'une algèbre spéciale qui encodent des transformations rigides lorsqu'ils sont normalisés. La plupart des propriétés des quaternions se retrouvent dans les quaternions duaux. L'interpolation obtenue est un mouvement de vis (ou "Screw Motion") qui présente trois propriétés intéressantes : l'interpolation passe par le plus court chemin, avec des déplacements unitaires en translation et en rotation constants, comme sur la figure 2.9(e).

L'interpolation se fait donc à vitesse constante, contrairement au mélange linéaire. Elle présente aussi l'avantage de coupler translation et rotation contrairement à la décomposition. De plus, cette technique ne présente pas les artefacts des mélanges linéaires ou logarithmiques, illustrés sur les figures 2.9(a) et 2.9(b). Enfin, elle s'avère rapide et règle le problème de mélange linéaire de transformations rigides.

2.4.4 Conclusion

L'attrait du skinning est dû à l'utilisation de repères comme degrés de liberté, qui procurent une manipulation intuitive de l'objet.

Son calcul est simple et très rapide. Néanmoins, les résultats varient fortement en fonction de la pondération choisie. Les pondérations automatiques existantes sont nombreuses et aucune ne prime sur les autres de manière générale.

Son utilisation pour l'animation cinématique est très courante mais elle n'a pas encore été utilisée pour des techniques dynamiques.

2.5 Les fonctions de forme

Toutes les approches de simulation physique par informatique discrétisent l'objet à simuler à l'aide de degrés de liberté. Les déplacements en tous points de l'objet sont ensuite déduites de cet ensemble par des fonctions de formes. Il est intéressant de comparer les fonctions de formes des méthodes existantes dans un même chapitre afin de comprendre quels sont les avantages et les contraintes de chacune.

Les fonctions d'interpolation varient en fonction des nœuds utilisés. Dans les différentes méthodes présentées précédemment, les nœuds peuvent être des points (dérivée spatiale d'ordre 0), des repères rigides (d'ordre 1) ou déformables (d'ordre 1 ou supérieur quelconque), et peuvent posséder des informations de connectivité. L'ordre des degrés de liberté permet de définir le champ de déplacement à l'aide de fonctions de forme polynômiales d'ordre plus ou moins élevé, permettant des déformations plus ou moins complexes de la matière.

2.5.1 Interpolation dans un élément

Nous nous intéressons ici aux interpolations effectuées dans un domaine borné. C'est à dire celles utilisées pour les modèles masses-ressorts ou les éléments finis.

Purement linéaire

Prenons le cas simple de la simulation d'une barre homogène par un élément fini. Les positions des degrés de liberté sont calculés par résolution de l'équation différentielle. Lors du déplacement d'une extrémité de la barre, il est attendu que l'extension de la matière dans l'élément soit uniforme. Ainsi, une propriété d'interpolation importante est la linéarité.

Les coordonnées barycentriques permettent d'obtenir une interpolation linéaire dans des éléments en trois dimensions. Les trois arêtes partant d'un des sommets d'un élément tétraédrique définissent une base pour paramétrer le déplacement de la matière dans celui-ci en fonction du déplacement des sommets de manière linéaire. Il en va de même pour les éléments hexaédriques, pour lesquels la base est définie par les huit sommets. Cette technique est la plus couramment utilisée dans les éléments finis de type tétraédriques ou hexaédriques.

Un inconvénient de cette méthode est que la continuité entre les éléments est C^0 . L'interpolation au sein de l'élément est linéaire mais des discontinuités apparaissent à la frontière de ceux-ci. La linéarité et la continuité sont deux propriétés souhaitées pour la simu-

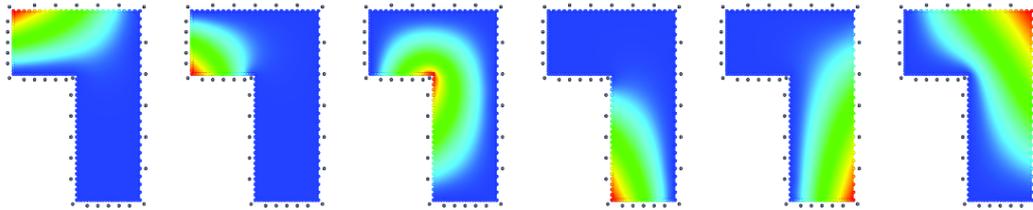


FIGURE 2.10: Fonctions de forme harmoniques de chaque degré de liberté, définies sur un élément fini disposant de six sommets.

lation des objets mais ces deux propriétés sont incompatibles en trois dimensions. Ceci étant, une continuité C^0 est acceptable si la résolution du maillage physique est suffisamment grande.

Un compromis entre linéarité et continuité

Les coordonnées harmoniques sont introduites pour une technique cinématique de skinning, dans [JMD⁺07]. Une fonction de forme $N_i : e \rightarrow \mathbb{R}$ est harmonique si son laplacien est minime en e , ses valeurs étant déterminées par les conditions au bord de Dirichlet en ∂e :

$$\begin{aligned} \Delta N_i(x) &= 0, \quad \forall x \in e \\ N_i^e(x) &= b_i(x), \quad \forall x \in \partial e \end{aligned} \quad (2.1)$$

Cette définition généralise les coordonnées barycentriques et permet d'obtenir une fonction de forme la plus linéaire possible quel que soit le nombre de degrés de liberté.

La valeur maximale est fixé à 1 pour le degré de liberté considéré et la valeur minimale de 0 est attribuée aux autres degrés de liberté. Le système d'équation est ensuite résolu pour calculer les valeurs dans le domaine. En procédant ainsi pour chaque degré de liberté, on obtient les influences représentées sur la figure 2.10 par les couleurs rouge (100%) et bleu (0%). Cette technique assure une pondération continue et la partition de l'unité : les valeurs varient de 0 à 1 dans tout le modèle et la somme des valeurs associées aux différents degrés de liberté en un point est égale à 1.

Dans [MKB⁺08], l'utilisation de cette technique permet de manipuler des éléments finis de forme quelconque. Ainsi, l'élément fini correspond grossièrement à la géométrie de l'objet, sans se soucier des problèmes de continuité entre les éléments. Toutefois, il peut s'avérer utile de définir plusieurs éléments pour améliorer les résultats. Dans ce cas, la continuité entre éléments est C^0 . Pour assurer la continuité entre les différents éléments du maillage, les conditions aux bords sont tout d'abord évaluées par interpolation linéaire le long des arêtes de l'élément, puis les valeurs sont déterminées au sein de l'élément par résolution du système d'équations (2.1), comme dans [JMD⁺07]. L'interpolation n'est pas linéaire mais présente un bon compromis entre continuité et linéarité.

Vers une meilleure continuité

D'autres techniques permettent une continuité entre les éléments encore meilleure. Ainsi, dans [KMBG08], Kauffman et al. utilisent des fonctions de formes discontinues de

Galerkin sur les éléments finis. Le domaine Ω est partitionné en un ensemble d'éléments $K \in \Gamma$.

Dans la version continue présentée antérieurement, il est possible de définir sur cet ensemble des fonctions de forme continues $\{N_1, \dots, N_n\}$ pour approximer le champ de déplacement u tel que :

$$u(x) \approx \sum_{i=1}^n u_i N_i(x)$$

où la fonction de forme N_i , pour une formulation faible à m dérivées partielles, est choisie dans l'espace de Sobolev $H^m(\Omega)$ (voir [HH00]). Cela restreint les fonctions de forme à être *conformes*, c'est à dire à être C^m continues au sein d'un élément et C^{m-1} continues à la frontière des éléments.

Dans la version discontinue, des fonctions de formes sont associées indépendamment à chaque élément pour assurer le calcul des forces internes. Lors d'une seconde passe, des contraintes faibles sont prises en compte pour lier les éléments, et les fonctions de formes sont modifiées pour assurer une meilleure continuité entre ceux-ci. Les modèles de collision et visuels sont mis en relation avec le modèle physique par la méthode des moindres carrés, présentée au prochain paragraphe.

Dans cette technique, le modèle physique présente une excellente continuité. Par contre, cette approximation du champ de déplacement procure des résultats visuellement plaisants mais physiquement moins précis.

2.5.2 Interpolation sans élément

Les méthodes particulières, alias *meshless* ou *meshfree* apparues ensuite, se prêtent à l'animation.

[FM03] classifient et comparent de nombreuses variantes de ces méthodes. La formulation du champ de déplacement est plus libre dans la mesure où les propriétés physiques du matériau sont interpolées à partir des degrés de liberté, sans informations de connectivité.

Cette absence de connectivité permet de simuler tous types de modèles sans pré-calcul de maillage. Elle permet aussi des changements adaptatifs plus simples, le remaillage adaptatif étant une opération complexe et coûteuse.

Dans la plupart des méthodes, une région d'influence est attribuée à chaque degré de liberté. Le déplacement d'un point de la matière est alors déterminé par une somme pondérée des propriétés des degrés de liberté. De nombreuses formulations ont été proposées, permettant différents compromis entre linéarité et continuité. Contrairement aux méthodes reposant sur un maillage, l'élaboration de fonctions de forme continues (même C^1) est plus complexe [LLH⁺04]. Il est néanmoins plus simple de modéliser des fonctions de formes discontinues, utiles lors de simulations de fractures ou de matériaux aux propriétés différentes, entre autres.

En pratique, la convergence d'une simulation à l'aide de ces méthodes est considérablement plus rapide que celles basées sur des maillages [LL96]. Même si cela n'est pas, à ce jour, vérifié en théorie [LL96]. En revanche, le temps de calcul nécessaire est plus important. Les fonctions de forme des méthodes *meshless* sont d'une nature plus complexe que celles des méthodes basées sur des maillages, assimilables à des fonctions polynômiales. En conséquence, le nombre de points d'intégration requis pour une évaluation suffisam-

ment précise de l'intégrale sous forme faible est plus important. Ce critère dépend bien sûr de la formulation meshless choisie. La plupart des autres étapes coûteuses sont effectuées à l'initialisation et ne pénalisent donc pas le temps d'exécution, tels que le calcul des fonctions de forme (pour être représentatives de leur domaine d'influence) ou des propriétés des points d'intégration (afin de déterminer leurs dérivées spatiales). Lors de changements adaptatifs, ces opérations sont à réaliser en cours d'exécution. Mais, ces calculs sont hautement parallélisables et peuvent être effectués sur le GPU, et/ou localement. Enfin, le système d'équations à résoudre est généralement plus large que pour les méthodes basées sur des maillages[BKO⁺96].

Comme dit précédemment, de nombreuses variantes des méthodes meshless existent. Le concept d'interpolation/approximation des méthodes les plus répandues sont brièvement présentées ici : par fonctions de Shepard et par les moindres carrés. Ces deux méthodes sont assez proches. Elles permettent d'élaborer une fonction de forme polynômiale $u(x)$ permettant d'approximer/interpoler un ensemble de valeurs scalaires u_i associées à un nuage de n points x_i dans \mathbf{R}^d .

Fonctions à base radiale (RBF) :

Mais avant de parler de ces interpolations, voici une rapide introduction aux fonctions à base radiale, souvent utilisées dans les différentes interpolations des techniques meshless. Une fonction à base radiale est une fonction à valeur réelle dont la valeur dépend seulement de la distance à l'origine, tel que : $\phi(x) = \phi(\|x\|)$. Ou alternativement, de la distance à un autre point c , appelé centre, tel que : $\phi(x, c) = \phi(\|x - c\|)$. Toute fonction vérifiant cette propriété est une fonction à base radiale. Par exemple, les fonctions gaussiennes ($e^{(-\epsilon x)^2}$), multi-quadriques ($\sqrt{1 + (\epsilon x)^2}$), quadrique inverses ($\frac{1}{1 + (\epsilon x)^2}$), splines polyharmoniques ($x^k, k = 1, 3, 5, \dots$; $x^k \ln(x), k = 2, 4, 6, \dots$) sont des fonctions à base radiale. La norme habituellement utilisée est la distance euclidienne, bien que d'autres fonctions de distance soient possibles.

Ces fonctions sont appréciées dans les techniques d'interpolation/approximation meshless car elles présentent une diffusion de l'influence sphérique et isotrope.

Fonctions de Shepard :

La fonction de Shepard est un modèle interpolant de la forme :

$$u(x) = \sum_{i=1}^n w_i u_i$$

où les w_i sont les fonctions de poids associées à chaque point x_i . La forme classique des fonctions de poids est :

$$w_i = \frac{d(x, x_i)_i^{-p}}{\sum_{j=1}^n d(x, x_j)_j^{-p}}$$

où p est un nombre réel positif arbitraire (typiquement égal à 2) et $d(x, x_i)$ la distance (par défaut euclidienne) entre x_i et le point interpolé. Les plus grandes valeurs de p donnent une influence plus grande aux valeurs les plus proches du point interpolé. Pour $0 < p < 1$, $F(x)$ donne des pics lissés au-dessus du point interpolé x_i , alors que pour $p > 1$ le

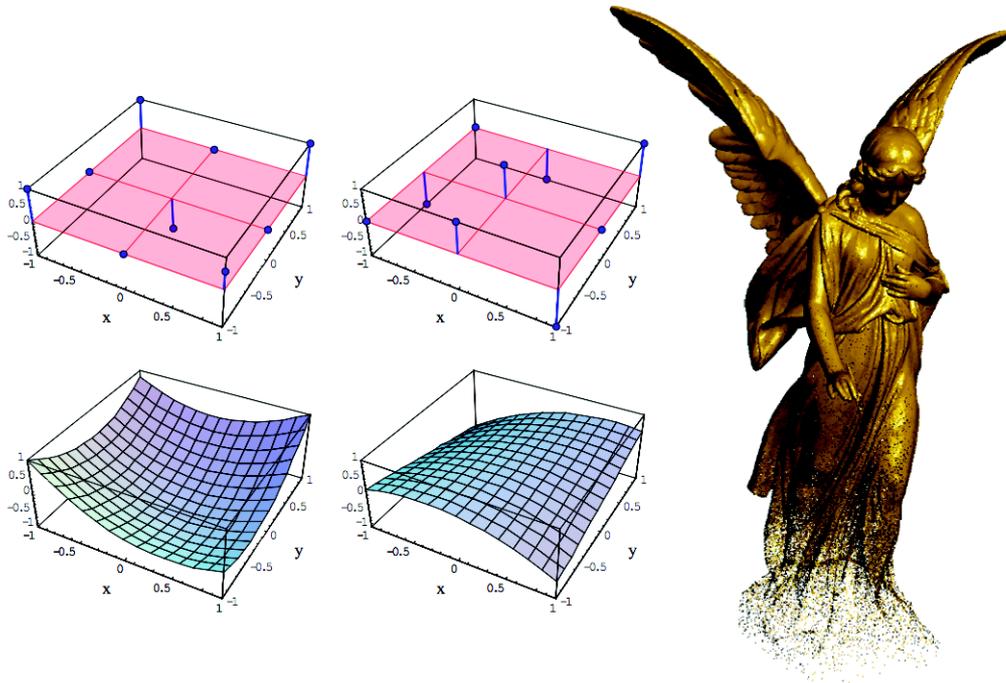


FIGURE 2.11: [Nea04] A gauche : approximation d'un champ scalaire 2D par moindres carrés avec une base polynomiale quadratique. A droite : Reconstruction de surface à partir d'un nuage de points à densité croissante (de bas en haut) par GMLS.

pic devient plus pointu. Le choix de p est donc une fonction du degré de lissage désiré pour l'interpolation, de la densité et la distribution des échantillons interpolés, et de la distance maximale au delà de laquelle un échantillon individuel peut influencer les points environnants.

Les moindres carrés (LS, WLS, MLS, GMLS) :

Il existe des versions plus ou moins simples de cette méthode : les Least Square (LS), Weighted Least Square (WLS), Moving Least Square (MLS) et Generalized Moving Least Square (GMLS). Ces versions reposent sur le même concept mais présentent une flexibilité croissante. [Nea04] présente de manière synthétique les trois premières versions.

LS : La fonction de forme, appartenant à Π_m^d , l'espace des polynômes de degrés m et de dimensions spatiales d , est calculée par la formule :

$$u(x) = b(x)^T c$$

où $b(x) = [b_1(x), \dots, b_k(x)]^T$ est la base polynomiale sous forme vectorielle et c les coefficients polynomiaux, déterminés par la minimisation de l'erreur :

$$\min_{u \in \Pi_m^d} \sum_{i=1}^n \|u(x_i) - u_i\|^2$$

Pour plus de détails sur le calcul des coefficients, se reporter à [Nea04]. La représentation du polynôme est ici matricielle. La base $b(x)$ choisie dépend du degré m et de la dimension spatiale d . Pour $m = 2$ et $d = 2$, $b(x) = [1, x, y, x^2, y^2, xy]$. Pour $m = 1$ et $d = 3$, $b(x) = [1, x, y, z]$. En général, le nombre k d'éléments dans $b(x)$ est $k = \frac{(d+m)!}{m!d!}$ [FM03]. Cette méthode permet d'établir une fonction de forme qui approxime le nuage de point de manière globale. Ce qui peut s'avérer imprécis sur un domaine large.

WLS : Les moindres carrés pondérés permettent d'obtenir une meilleure approximation du domaine en minimisant l'erreur non pas de manière globale mais sur plusieurs points $\bar{x} \in \Omega$:

$$\min_{u \in \Pi_m^d} \sum_{i=1}^n \theta(\|\bar{x} - x_i\|) \|u(x_i) - u_i\|^2$$

Une fonction de forme est alors associée à chaque \bar{x} , comme précédemment :

$$u_{\bar{x}}(x) = b(x)^T c(\bar{x})$$

Une influence est attribuée à chacune de ces fonctions (déterminée par les poids $\theta(d)$, fonctions de la distance de x_i à \bar{x}), utilisée lors de la minimisation de chaque fonction de forme, et permettant de mélanger continument celles-ci. Il est nécessaire que le support de l'ensemble des fonctions de poids θ_j , centrées en \bar{x}_j recouvrent Ω . Une manière propre de procéder est de construire une partition de l'unité :

$$\varphi_j(x) = \frac{\theta_j(x)}{\sum_{k=1}^n \theta_k(x)}$$

Ainsi, $\sum_{j=1}^n \varphi_j(x) \equiv 1$ partout dans Ω . Et la fonction de forme globale est exprimée suivant :

$$u(x) = \sum_{j=1}^n \varphi_j(x) b(x) c(\bar{x}_j)$$

Les fonctions de poids étant généralement basées sur la distance à \bar{x} , les fonctions de forme des WLS sont plus proches des u_i .

MLS : L'approximation et l'interpolation par les moindres carrés (MLS), introduits par [LS81], est encore plus précise que les précédentes versions. Au lieu de construire des fonctions de forme optimisées pour un ensemble de points $\bar{x} \in \Omega$, celles-ci sont évaluées en tout point de l'espace. L'idée est de commencer avec une formulation par moindres carrés pondérés (WLS) sur un point arbitraire, et de *déplacer* ce point sur l'ensemble du domaine. La fonction de forme globale $u(x)$, obtenue à partir d'un ensemble de fonctions locales, est :

$$u(x) = u_x(x), \min_{u \in \Pi_m^d} \sum_{i=1}^n \theta(\|\bar{x} - x_i\|) \|u(x_i) - u_i\|^2$$

continument dérivable si et seulement si la fonction de poids est continument dérivable [Lev98]. La fonction de poids choisie permet de choisir la nature interpolante ou approximante de la méthode. En effet, en fixant une influence infinie au point x_i considéré décroissante jusqu'à zéro pour les x_i voisins, la fonction de forme obtenue est interpolante.

GMLS : Les GMLS (Generalized Moving Least Squares) ont été introduit par [ACK99]. La méthode des MLS est étendue à des degrés de liberté d'ordre supérieur (ordre spatial au moins égal à 2). Concrètement, lorsque l'ordre spatial est égal à 2, la primitive possède une dérivée première et peut être représentée par un repère, où chaque axe décrit une tangente au point. Ces dérivées sont incorporées à la fonction de forme à l'aide de l'interpolation d'Hermite.

Ce qui permet d'obtenir une meilleure continuité et garantit le calcul des poids. Le calcul des coefficients requiert l'inversion d'une matrice (voir [Nea04] pour plus de détails). Pour cela, quatre points non coplanaires sont nécessaires et suffisants pour éviter les singularités (impliquant la non inversion) de cette matrice. Avec l'utilisation de degrés de liberté d'ordre plus élevé, le calcul des coefficients est garanti puisqu'un repère (ordre 2) est équivalent à quatre points.

[MKB⁺10] utilisent les GMLS dans une technique d'animation à l'aide de repères. Ils introduisent aussi les élastons, qui assurent l'évaluation de la déformation dans le modèle. Ce modèle leur permet de simuler correctement des objets volumiques, surfaciques et linéiques avec un nombre restreint de degrés de liberté et assure notamment la prise en compte de la torsion entre deux primitives. Évaluer celle-ci dans une technique à base de points nécessiterait au moins huit points, pour calculer la différence de rotation de la torsion.

2.6 Simulation de découpe

La découpe d'un objet se traduit par la suppression de la relation de voisinage entre les éléments de la matière qui le constituent. Cette opération peut survenir suite à une incision du modèle par un objet tranchant ou lors d'une fracture.

Dans le cas d'une découpe à l'aide d'un outil tranchant, le chemin suivi par celui-ci est généralement calculé sur le pas de temps en cours afin de déterminer le profil de découpe [GCMS00]. Ceci afin de prendre en compte des pas de temps longs. Si le pas de temps est suffisamment petit, le profil de découpe peut être basé sur les collisions en cours. Dans le cas d'une fracture, la découpe est causée par des efforts mécaniques trop importants et peut survenir à plusieurs endroits du modèle [Gdo05, MMDJ01]. D'autres techniques intermédiaires allient une découpe de l'utilisateur entraînant une fracture progressive de l'objet [Nie03].

Dans tous les cas, la découpe implique des changements topologiques sur les modèles visuel, physique et de collision. Ces actions ont des répercussions différentes sur les modèles et la manière de procéder dépend de leur nature. Suivant l'utilisation de modèles avec/sans maillage, l'application de la découpe est plus ou moins complexe.

2.6.1 Modèles physiques maillés

Les méthodes de simulation les plus courantes utilisent un maillage, notamment les éléments finis. La découpe du modèle simulé se traduit par une modification du maillage du modèle physique. Or, cette opération est complexe car il est nécessaire de garantir la qualité du maillage pour conserver la stabilité et un bon comportement ; il ne doit pas apparaître d'éléments dégénérés et les éléments voisins ne doivent trop varier en taille. Recalculer un nouveau maillage pour la nouvelle topologie serait trop coûteux et plusieurs solutions

ont été envisagées basées sur les éléments (par suppression, subdivision, séparation ou duplication) ou les fonctions de base.

Ainsi, la modification peut être effectuée en supprimant les éléments intersectant l'instrument coupant [CDA00]. Suivant la technique d'animation utilisée, il suffit de supprimer un ressort ou un élément sur lequel reposent les calculs mécaniques. Une attention doit être portée sur la cohérence du maillage [FDA02] afin de conserver un maillage manifold. Cette opération est la plus simple et s'avère rapide. De plus, la suppression de nœuds de simulation accélère la vitesse de simulation. En revanche, elle provoque des artefacts visuels relatifs à la finesse du maillage initial et une disparition de matière importante modifie le comportement de l'objet simulé.

La technique de séparation de faces consiste à séparer les éléments. L'ensemble des faces les plus proches de la trajectoire de découpe est déterminé. Une nouvelle face est ensuite insérée sur chacun des éléments séparés. Afin de découper au plus proche de la trajectoire, les nœuds sont déplacés le long de la trajectoire [NvdS01]. Le nombre de nœuds de simulation reste constant, ce qui maintient les performances, et la quantité de matière est conservée. Bien qu'aucun élément ne soit créé, le déplacement des nœuds de simulation engendre des éléments dégénérés.

D'autres techniques subdivisent les éléments en intersection avec l'objet tranchant. [MK00] propose différents motifs de découpe pour des triangles et des tétraèdres. Des nœuds sont donc ajoutés au fil de la découpe, ralentissant le temps de simulation. Pour éviter que leur nombre ne devienne trop rapidement important, [BMG99] effectue la subdivision de l'élément une fois que l'objet coupant n'est plus en collision avec celui-ci. Concernant la fracture, [OH99] propose une méthode par subdivision d'éléments. Ces approches restent temps réel et la découpe est belle. Son principal défaut est la création de petits éléments mal conditionnés, impropres à la simulation par éléments finis, causant des instabilités numériques, néanmoins envisageable à l'aide de modèle masse-ressorts ou éléments finis inversibles [ITF06]. [WBG07] utilisent une approche basée sur des éléments finis convexes arbitraires. Bien qu'ils remaillent les éléments finis suite à une découpe, les nouveaux éléments créés ne sont pas subdivisés en tétraèdres, réduisant ainsi la création potentielle d'éléments mal conditionnés, sans pour autant l'éviter complètement.

Afin d'éviter la création d'éléments dégénérés, d'autres ont adopté la duplication d'éléments. Contrairement aux méthodes précédentes, l'algorithme de nœuds virtuels introduit par [MBF04] ne nécessite pas que la découpe soit alignée avec les bords des éléments. Au lieu de subdiviser un élément découpé, des copies de celui-ci sont créées. La matière du matériau contenu dans l'élément initial est fragmentée et répartie dans les différentes copies. Dans l'algorithme original, le nombre de coupures par élément est limitée de manière à ce que chaque nouvel élément contienne un nœud du maillage d'élément fini initial. Cette limitation est résolue par [SDF07], permettant ainsi des découpes arbitraires sur des éléments tétraédriques ; bien qu'ils ne mentionnent pas la manière dont les matrices de masses étaient mises à jour. Cette approche permet ainsi de créer peu d'éléments au cours de la découpe, ce qui ralentit peu le temps de calcul, et garantit la stabilité des éléments en évitant les éléments dégénérés.

D'autres méthodes modifient la fonction de base de l'élément pour procéder à la découpe de la matière. La méthode d'éléments finis étendus (XFEM, "extended finite elements method" [SMMB00]), introduite par [BB99] pour la simulation de fractures sur des structures mécaniques, permet de modéliser différentes régions discontinues dans un

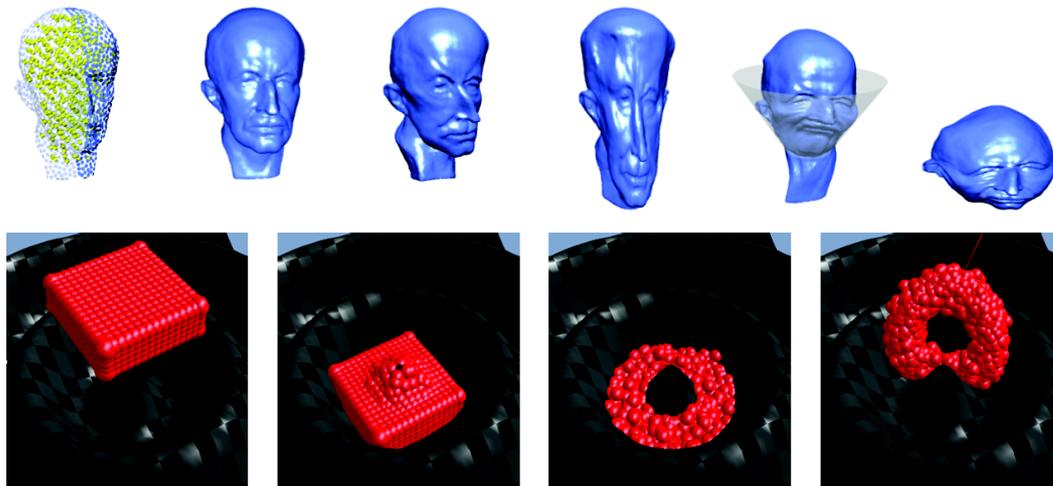


FIGURE 2.12: Simulation de découpe de [MKN⁺04] une déformation élastique/plastique et des changements topologiques à l'aide d'un échantillonnage dense.

maillage d'éléments finis par l'ajout de discontinuités dans une sous région. Les XFEM ne requièrent pas l'alignement de la coupure avec le maillage d'éléments finis et ont été utilisées par [JK09] pour des simulations chirurgicales où une seule coupure par élément était permise. Le nombre de noeuds reste constant et la forme des éléments n'est pas modifiée. Ce qui en fait une technique appréciée pour sa rapidité et sa stabilité.

2.6.2 Modèles meshless

Les méthodes meshless se prêtent mieux aux opérations de découpe. L'absence de maillage rend les changements topologiques plus simples (aucune opération de remaillage) et plus stable ; l'instabilité des éléments finis étant souvent provoqué par la qualité du maillage.

[MKN⁺04] présente une méthode meshless dont les noeuds sont des points, le champ de déplacement est défini par moindres carrés et les forces internes sont basées sur l'énergie de déformation, permettant un comportement élastique et plastique. Un graphe de relations de voisinages entre les noeuds est calculé à l'initialisation. Lors de la modification de la position de référence (utilisée pour le comportement plastique), le graphe est mis à jour. La découpe ainsi que la fracture se traduit par la suppression d'une arête dans le graphe de relation. Néanmoins, cette méthode nécessite un échantillonnage dense pour le rendu par surface implicite et la physique ; les fonctions de forme trop simples ne séparant pas les noeuds de simulation de la matière. [WSG05] pallie à cet inconvénient dans une approche comparable. Le modèle est échantillonné par un nuage de points dense à sa surface. Un graphe de voisinage est alors calculé sur un ensemble représentatif de ces points de manière à ce qu'il approxime correctement la courbure de la surface en plus d'encoder les relations de voisinage ; les sommets du graphe devenant les noeuds de simulation. L'échantillonnage des noeuds est ainsi moins dense mais cette technique ne s'adresse qu'à la simulation d'objet surfacique.

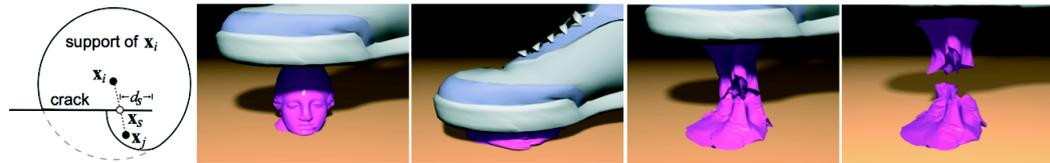


FIGURE 2.13: [PKA⁺05] Modification du noyau suivant la visibilité permettant la déchirure d'objets.

D'autres méthodes effectuent la découpe grâce au noyau des fonctions de forme. L'utilisation de la distance géodésique plutôt qu'euclidienne dans leur calcul produit un meilleur résultat mais il est plus long. [PKA⁺05] modifie l'influence du noyau en fonction de la *visibilité* entre les nœuds (Fig 2.13). La distance est modifiée suivant le bord de l'objet par un facteur de *transparence* lors d'une fracture, ce qui permet d'approcher la distance géodésique par un calcul rapide.

2.7 Conclusion

Nous nous sommes inspirés de l'ensemble des techniques présentées au cours de ce chapitre pour élaborer notre méthode de simulation dynamique et de découpe. Tout d'abord, voici un rappel des buts de notre méthode :

- La qualité : la rapidité d'exécution, la précision du comportement et la robustesse doivent être au moins égales aux méthodes existantes.
- La simplicité d'utilisation : par la possibilité d'une automatisation et d'une édition du comportement à la manière du skinning.
- La découpe : par une formulation facilitant les changements topologiques.
- La simulation de matériaux réels : par la prise en compte de matériaux à propriétés hétérogènes.

Les deux principaux aspects déterminants sont :

- l'échantillonnage de l'espace.
- les fonctions de forme, soit la formulation du champ de déplacement reposant sur ces échantillons.

Au vu des avantages et inconvénients des méthodes existantes dressés dans ce chapitre, voici les choix que nous avons fait pour la définition de notre modèle de simulation.

Concernant la **découpe**, nous souhaitons conserver la possibilité de manipuler l'objet simulé avec les outils tranchants en conservant la **réponse aux contacts**. Elle doit survenir lorsque la **pression** appliqué par l'instrument devient **trop importante**.

L'**échantillonnage** des nœuds du modèle physique doit pouvoir être réglable, voire réduit à un très **faible nombre de nœuds**. Le nombre de nœuds présents dans la simulation détermine la précision du comportement mais aussi la vitesse d'exécution, car il définit le nombre d'équations à résoudre dans le système. L'amélioration des fonctions de formes entre les nœuds (2.2.5) permet de réduire ce nombre.

Il doit mener à un **bon conditionnement des matrices**. Une stratégie d'échantillonnage régulier provoque un mauvais conditionnement des matrices, ce qui entraîne des in-

stabilités numériques (2.2.4) lorsque la raideur entre deux nœuds est trop importante. Elle doit donc reposer sur les propriétés matérielle pour éviter ces instabilités.

Il est préférable de **s'abstraire du maillage**. En effet, le remaillage est une opération laborieuse et coûteuse (2.2.6). La définition des fonctions de formes est aussi moins restrictive (2.5). Une attention doit être portée à la définition des conditions aux bords du domaine, qui sont plus libres.

Les **nœuds** doivent être **de plus haut degré**, par l'utilisation de repères au lieu de points. Cela permet de réduire le nombre de nœuds utilisés (jusqu'à un unique repère pour un solide rigide). Cela permet aussi de définir des fonctions de formes plus complexes, à la manière de skinning (2.4) ou du GMLS (2.5.2). Contrairement au skinning qui utilise des repères rigides pour n'encoder que la rotation en plus de la translation, les nœuds peuvent être **déformables** afin d'ajouter les déformations affines, quadratiques ou supérieures en ces points. Les nœuds de plus haut degré permettent encore de pallier au problème de couverture des influences lors du calcul des fonctions de forme (2.5.2).

La formulation de **fonctions de forme plus flexibles** présentent deux avantages. Dans le cadre de l'animation, cela permet de profiter des avantages du skinning sur une méthode dynamique. Pouvoir fixer et **modifier les poids simplement** pour changer l'influence des nœuds, serait très apprécié par les designers (2.4). Pour des utilisations plus réalistes, les propriétés des **matériaux hétérogènes** (dont anisotropes) peuvent être **encodées dans les poids** des fonctions de forme.

Simulation de découpe et déformation par repères déformables

Pour faciliter la lecture, nous rappelons les grandes lignes de cette partie décrivant les contributions.

Dans le chapitre 3, nous présentons notre modèle de découpe. Le modèle physique employé dans cette méthode repose sur des éléments finis et présente l'avantage de simuler rapidement des scènes de taille importante. Néanmoins, il est souhaitable de s'abstraire du maillage comme vu dans l'état de l'art et les éléments comprenant peu de matière et la répartition régulière des nœuds entraîne certains problèmes numériques. Le modèle meshless présenté dans les chapitres 4 et 5 apparaît plus avantageux pour des simulations de découpe. Dans le chapitre 4, une vue d'ensemble est donnée suivie de la définition des fonctions de forme. Le chapitre 5 décrit les différentes étapes de la résolution dynamique. Le chapitre 6 présente les résultats obtenus sur ces méthodes et les travaux en cours.

Les deux méthodes ne sont actuellement pas unifiées en terme d'implémentation. L'algorithme 1 résume l'ensemble des étapes de simulation pour une meilleure vue d'ensemble d'une version unifiée.

Données : Carte de voxels décrivant les propriétés matérielles	
Initialisation :	
Génération des modèles triangulaires ;	// sec. 3.2.3
Distribution des nœuds de contrôle ;	// sec. 4.3
Calcul des fonctions de forme ;	// sec. 4.2
Calcul de la matrice de masse ;	// sec. 5.4
Génération des points d'intégration ;	// sec. 5.3.2
Calcul des poids des sommets de la surface ;	// sec. 5.6
Initialisation des mappings topologiques ;	// sec. 3.2.3
Simulation :	
Opérations de découpe :	
Analyse de la rasterisation des intersections avec d'autres modèles ;	// sec. 3.3.1
pour chaque <i>modèle tranchant en collision</i> faire	
Suppression des voxels intersectés ;	// sec. 3.3.2
Mise à jour du modèle physique ;	// sec. 3.2.2
Mise à jour des modèles triangulaires ;	// sec. 3.3.3
fin	
Résolution dynamique :	
pour <i>Accumulation des forces de chaque point d'intégration</i> faire	// eq. 5.18
Calcul du gradient de déformation F (et ses dérivées spatiales);	
Calcul de ϵ (et ses dérivées) à partir de F en fonction d'une mesure de déformation donnée;	
Calcul de σ (et ses dérivées) à partir de ϵ en fonction des propriétés matérielles;	
Ajout des forces intégrées sur chaque primitive influente;	
fin	
Forces externes et gestion des collisions ;	// sec. 5.6.1
pour chaque <i>itération du solveur</i> faire	// eq. 5.5
Accumulation du changement de force de chaque point d'intégration ;	// eq. 5.19
fin	
Mise à jour des déplacements du modèle visuel ;	// sec. 5.6.2

Algorithme 1: Étapes de simulation.

Chapitre 3

Simulation de découpe

Contents

3.1	Présentation de la méthode	48
3.2	Modèles physique, visuel et de collision	49
3.2.1	Topologies des modèles	49
3.2.2	Modèle physique	50
3.2.3	Modèles triangulaires	51
3.3	Processus de découpe	53
3.3.1	Rastérisation de la scène	53
3.3.2	Propagation dans les hexaèdres	53
3.3.3	Mise à jour du maillage triangulaire	54
3.4	Réponse au contact	54
3.5	Rendu	55

3.1 Présentation de la méthode

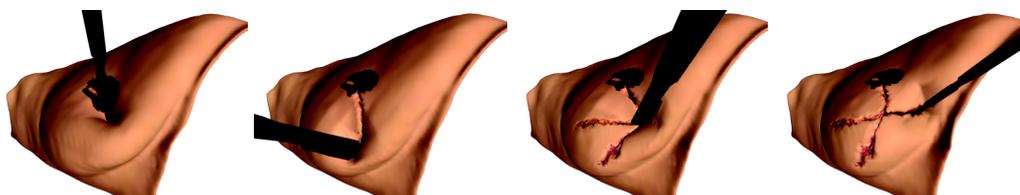


FIGURE 3.1: Réponse au contact avant et pendant la découpe exploitant les données médicales.

La technique de découpe à but chirurgical spécifique au patient présentée dans ce chapitre s'exécute en temps réel, avec une réponse aux collisions pendant la découpe et exploite les données médicales pour un rendu réaliste.

A notre connaissance, la réponse au contact dans les simulations de découpe est un thème peu exploré. Excepté [MC97], les techniques de découpe actuelles effectuent une découpe pendant laquelle aucune interaction n'intervient avec les modèles coupants, telle une découpe au laser. Pourtant, la réaction suite à une collision des modèles lors de la simulation enrichit grandement l'immersion de l'utilisateur dans les simulations médicales, notamment lors de l'utilisation d'instruments haptiques à retour de forces. C'est pourquoi il nous a semblé important d'inclure cette fonctionnalité dans notre méthode.

Le développement d'une telle technique inclut de nombreux aspects : la génération de modèles physique, de collision et de rendu, à partir des données médicales du patient, leur mise en relation et leur modification au cours de la simulation lorsque les opérations de découpe interviennent.

Le modèle présenté dans ce chapitre est présenté dans la publication [JBB⁺10]. Les trois modèles (physique, de collision et visuel) sont indépendants et basés sur les données volumiques de l'objet. La résolution dynamique est assurée par des éléments finis hexaédriques hiérarchiques. La collision et le rendu sont modélisés par des maillages triangulaires, générés à l'aide de l'algorithme du *marching cubes* appliqué sur les voxels. De cette manière, nous évitons les problèmes de couplage entre ceux-ci. De plus, cette séparation permet de conserver la stabilité physique lors de changements topologiques, engendrés par la découpe du modèle de collision. En terme d'implémentation, leur développement est facilité et leur mise en relation (physique et topologique) est effectuée par des composants appelés *mappings*.

La découpe est réalisée par suppression de matière au niveau des voxels. Voici le pipeline général de la simulation, résumé dans l'algorithme 2. Au début de chaque pas de temps, les volumes d'intersection des objets en collision sont calculés. Dans le cas d'une collision entre un objet tranchant et le modèle triangulaire généré, ces données sont utilisées pour déterminer l'ensemble des voxels à supprimer de la simulation. Les éléments finis ainsi que les maillages triangulaires du modèle découpé sont mis à jour localement. Le pipeline classique est ensuite exécuté pour calculer les propriétés physiques du prochain pas de temps.

Dans notre approche de simulation, la découpe et les réponses au contact sont deux aspects indépendants. Nous abordons dans cette section le principe de notre simulation

Données : Carte de voxels décrivant les densités et de segmentation de la matière.	
Initialisation :	
Génération du modèle hexaédrique ;	// sec. 3.2.2
Génération des modèles triangulaires ;	// sec. 3.2.3
Simulation :	
Analyse de la rasterisation des intersections avec d'autres modèles ;	// sec. 3.3.1
pour chaque modèle tranchant en collision : faire	
Suppression des voxels intersectés ;	// sec. 3.3.2
Mise à jour du modèle physique ;	// sec. 3.2.2
Mise à jour des modèles triangulaires ;	// sec. 3.3.3
fin	
Résolution dynamique du modèle physique;	

Algorithme 2: Simulation de découpe.

physique, comment est traitée la découpe au laser, puis la réponse au contact, et enfin comment peuvent être exploitées les données médicales pour apporter une information visuelle supplémentaire quant à la nature de l'objet découpé.

3.2 Modèles physique, visuel et de collision

3.2.1 Topologies des modèles

La définition des topologies des modèles hexaédriques et triangulaires et leur modification reposent sur le travail de [AD08], dont l'organisation et l'opération de suppression sont résumées ici pour la compréhension des sections suivantes. En terme d'implémentation, une hiérarchie de composants décrit la structure du maillage ; chaque composant se basant sur les éléments du composant inférieur :

- **Sommets** : vecteur de coordonnées spatiales des sommets.
- **Arêtes** : vecteur de couples d'indices de sommets.
- **Triangles** : vecteur de triangles défini par trois indices d'arêtes.
- **Hexaèdres** : vecteur d'hexaèdres défini par quatre indices d'arêtes.

Les changements topologiques sont effectués par propagation hiérarchique sur ces structures. Chaque composant supprimant les éléments le concernant et passant l'information au composant suivant.

Afin d'optimiser les accès en mémoire et le temps d'exécution, chaque composant décrit son ensemble d'éléments par un vecteur contigu en mémoire pour un accès linéaire. Supprimer un élément au milieu du vecteur entraînerait le déplacement des éléments suivants en mémoire, ce qui est coûteux. Pour éviter cela, l'élément à supprimer est remplacé par le dernier élément de la liste avant de supprimer celui-ci en mémoire. Les indices des éléments déplacés sont alors modifiés dans les composants y faisant référence.

Le processus de modifications topologiques se décompose ainsi en deux étapes : l'annonce d'un changement permettant aux composants de modifier leurs références, puis la modification effective en mémoire.

3.2.2 Modèle physique

Après des années de progrès constants, le maillage ainsi que le remaillage reste un problème difficile, voir e.g. [TWAD09]. Alors que des géométries détaillées peuvent être plongées dans des cellules grossières pour réduire la complexité, un matériau détaillé requiert un nombre important de cellules, ou des fonctions de forme sophistiquées pré-calculées par analyse statique lors de l'initialisation [NKJF09a].

Notre technique de simulation est relativement proche des éléments finis étendus et des nœuds virtuels. A cela s'ajoute une approche hiérarchique permettant la simulation d'éléments finis hexaédriques à raideurs non uniformes, initialement présentée par [NPF06], étendus de manière à respecter la topologie et prendre en compte ses changements pendant la simulation.

Le niveau de détail le plus fin est donné par la résolution des images médicales. Un nombre arbitraire de niveaux de détail plus grossiers est récursivement généré en doublant la taille des voxels pour chaque niveau. La connectivité mécanique entre les éléments est calculée sur la résolution fine. Deux éléments adjacents sont connectés au niveau $n + 1$ si de la matière est présente sur leurs bords au niveau n . Lors de la construction du niveau grossier, un élément fini est créé pour chaque composante connexe dans cette cellule. Le nombre d'éléments superposés par cellule n'est pas limité. Les éléments superposés pour un niveau donné décrivent le même volume, mais la différence de comportement provient de leurs matrices de masse et de raideur, qui traduisent l'occupation de la matière dans chaque élément. La détection de collision reposant sur les modèles triangulaires qu'ils animent, aucune collision n'est détectée entre les éléments superposés au début de la simulation. Elles peuvent ensuite survenir dans le cas où les modèles triangulaires plongés se rencontrent durant la simulation. La figure 3.2 illustre la comparaison entre l'approche originale et cette méthode.

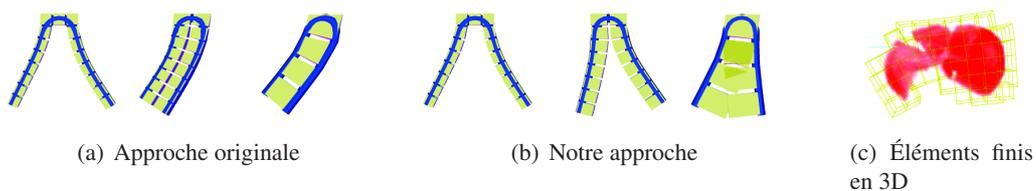


FIGURE 3.2: Principe et exemple de notre approche d'éléments finis étendus et hiérarchiques.

Le calcul des matrices de masse et de raideur pour chaque élément grossier est effectué par condensation de matrices uniformes des voxels qu'il contient, selon le principe introduit dans [NPF06]. Elles sont calculées en utilisant une condensation récursive de matrices suivant un schéma d'octree ($8 \rightarrow 1$): $A_{(l+1)} = A_{(l+1)} + H_i^T A_{(l)} H_i$ où l'indice l dénote le niveau, A peut être la matrice de masse ou de raideur, et H est une des huit matrices d'interpolation, qui reposent sur l'expression barycentrique des sommets d'un cube dans un cube englobant (données en annexe de [NPF06]).

Relation avec les maillages triangulaires

Une fois les trois modèles générés, ils sont mis en relation par coordonnées barycentriques. Cette mise en relation permet de mettre à jour les modèles triangulaires en fonction des déplacements des nœuds du modèle physique, et de faire remonter les forces appliquées au modèle de collision sur les nœuds de contrôle.

Relation avec les voxels

Un mapping topologique assure la cohérence entre le modèle physique et les voxels. Lors de la suppression de voxels, des changements topologiques peuvent survenir sur les éléments finis (au niveau grossier).

- Si le voxel supprimé était le dernier présent dans l'élément, il est supprimé.
- Si la suppression de voxels entraîne la création d'une composante indépendante ou plus, de nouveaux éléments sont créés pour chaque composante connexe dans cette région ; leurs nœuds sont connectés aux éléments voisins en fonction de la connectivité au niveau des voxels.
- Si les voxels supprimés étaient les derniers voxels à assurer la connexion avec l'élément voisin, leurs nœuds sont déconnectés.
- Dans les autres cas, aucun changement topologique n'est effectué sur les éléments finis.

Une fois les changements topologiques traités, la mise à jour des matrices de masse et de raideur est effectuée directement du niveau fin au niveau le plus grossier selon la formule : $A_{(L)} = A_{(L)} - H_{(L)i}^T A_{(0)} H_{(L)i}$ où l'indice (L) correspond au niveau le plus grossier et les matrices d'interpolation $H_{(L)i}$, des éléments les plus fins aux plus grossiers, sont pré-calculées.

3.2.3 Modèles triangulaires

Les modèles triangulaires utilisés pour le rendu et la détection de collisions sont générés à l'aide de l'algorithme du marching cubes à partir de la carte de densités. Alors que le modèle de collision est utilisé tel quel, plusieurs passes de lissages (laplacien des normales égal à zéro) sont appliquées au modèle visuel afin d'atténuer l'effet de marche d'escalier que l'algorithme du marching cubes entraîne. Néanmoins, les modifications du maillage triangulaire lors des opérations de découpe reposant sur les coordonnées des triangles, la position des sommets avant lissage est conservé pour raccorder les nouvelles parties de maillages.

Relation avec les voxels

Lors d'une collision avec un objet tranchant, l'ensemble des triangles en collision permet de déterminer les voxels à supprimer de la simulation. Cette suppression entraîne alors le remaillage local des maillages triangulaires. Ainsi, le mapping topologique doit conserver la relation entre les voxels et les triangles.

La grille utilisée par l'algorithme du marching cubes est définie par les voxels. Chaque cube est composé de huit sommets, qui sont des centres de voxels. Il est possible d'associer à un triangle l'ensemble des voxels qui définissent son cube et à un voxel l'ensemble des

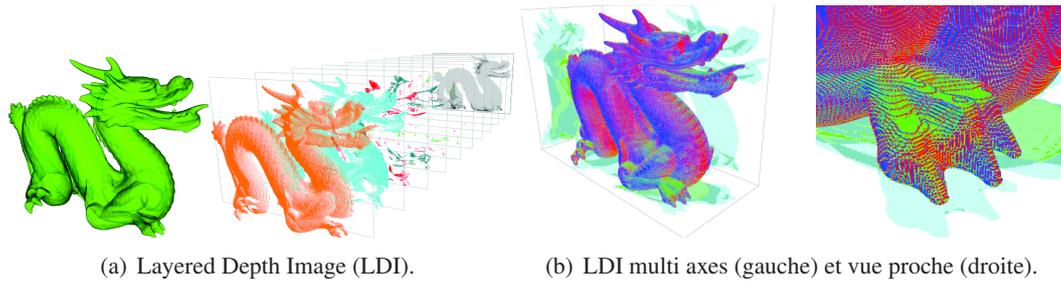


FIGURE 3.3: (a) Une image de textures de profondeurs (LDI), représentée en 2.5 dimensions, contient la valeur de profondeur de toutes les couches présentes sur chaque pixel. (b) Combiner les LDIs suivant les trois axes fournit une représentation volumique de l'objet.

triangles contenus dans les cubes définis par ce voxel. Ces relations sont enregistrées et permettent un accès direct entre les deux topologies, évitant de parcourir l'ensemble des éléments.

Dans notre première implémentation, deux structures étaient utilisées : la première listait l'ensemble des voxels reliés à chaque triangle tandis que la seconde encodait l'ensemble des triangles connectés à chaque voxel ; liant les indices d'une topologie à l'autre. Lors de la suppression d'un ensemble de voxels et de triangles, opération survenant pendant la découpe, l'algorithme de mise à jour de ces deux structures était d'une complexité trop élevée, du fait des références croisées des deux structures. En effet, l'accès des références d'une topologie à l'autre étant assurée par ces structures, leur modification croisée nécessitait des copies temporaires des éléments supprimés de la première liste afin de réorganiser la seconde.

En scindant chacune des deux structures en deux par l'intermédiaire de l'indice spatial des voxels dans la grille régulière non déformée, les structures deviennent indépendantes et le temps d'exécution de notre scène test a été accéléré par 10. Quatre structures sont donc actuellement utilisées, permettant l'accès direct entre :

- l'indice des voxels de la grille régulière à l'indice topologique.
- l'indice des voxels de la topologie à l'indice dans la grille régulière.
- l'ensemble des triangles (indice topologique) liés à un voxel (indice dans la grille régulière).
- l'ensemble des voxels (indice dans la grille régulière) définissant un triangle (indice topologique).

Pour des raisons pratiques, l'indice d'un voxel dans la grille régulière est ramené à une dimension suivant $i = x \times \text{tailleY} \times \text{tailleZ} + y \times \text{tailleZ} + z$, où x , y et z sont les coordonnées du voxel dans la grille et tailleX , tailleY et tailleZ la taille du tableau.

3.3 Processus de découpe

3.3.1 Rastérisation de la scène

Pour déterminer l'ensemble des hexaèdres intersectant un modèle coupant, nous avons utilisé et étendu la technique des LDI [FBAF08]. Cette technique permet de rastériser des modèles triangulaires sur GPUs.

Des textures de profondeur, visibles sur la figure 3.3(a), sont générées suivant les trois axes (par depth peeling) et rapatriées vers le CPU. Dans l'exemple de la figure 3.3(b), l'analyse de ces textures permet de reconstruire le modèle sous une forme discrète. En pratique, cette technique destinée à la détection de collision, permet de déterminer de manière efficace les volumes d'intersections entre les modèles pour calculer les forces de répulsion à leur appliquer. Une fois les textures de profondeur générées, nous les exploitons pour réaliser les opérations de découpe avant d'exécuter la détection de collision et le pipeline dynamique classique.

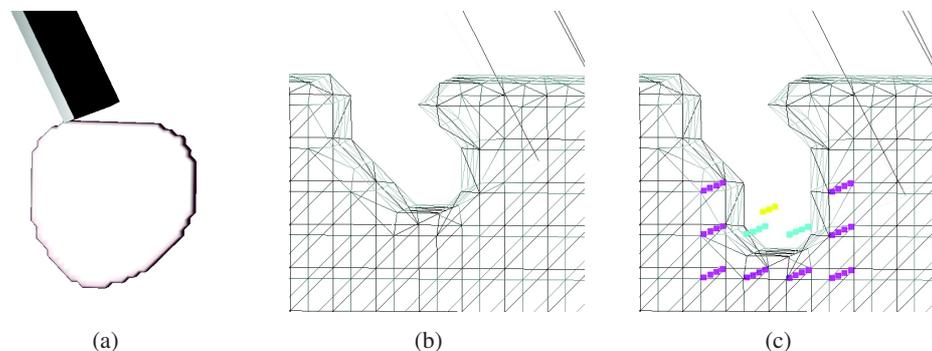


FIGURE 3.4: Propagation sur les voxels à supprimer. (a) Maillage coupé par une lame. (b) En gros plan, à l'instant t . (c) A $t + 1$, nous calculons l'ensemble des triangles en collision (jaune), des voxels à l'intérieur (cyan) et à l'extérieur (magenta) du volume d'intersection.

3.3.2 Propagation dans les hexaèdres

Suite à cet algorithme, nous disposons de l'ensemble des volumes d'intersection entre le maillage triangulaire de collision et les objets présents dans la simulation. Ces volumes sont définis par deux sous ensembles de maillages triangulaires et nous disposons de l'ensemble des triangles en collision avec le modèle coupant (points jaunes pour le centre des triangles sur la figure 3.4(c)). A partir de ceux-ci, et grâce aux structures de relation entre triangles et voxels, nous déterminons l'ensemble des voxels en relation directe (points cyans sur la figure 3.4(c)). Ces voxels définissent les triangles lors de la génération du marching cubes ; ils sont en surface de l'objet et les indices obtenus correspondent aux voxels dans l'espace non déformé.

Pour étendre cet ensemble à tous les voxels présents dans le volume d'intersection, une propagation est alors lancée. La position du centre de chaque voxel testé est mise à jour en espace déformé par coordonnées barycentriques et la propagation continue s'il est inclus

dans le volume d'intersection. Elle s'arrête sur les voxels extérieurs au volume, en magenta sur la figure 3.4(c).

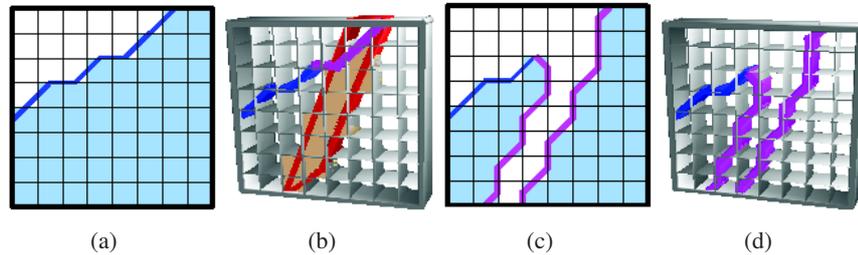


FIGURE 3.5: Mise à jour du maillage triangulaire (a) le maillage initial au repos (bleu). (b) les hexaèdres en collision (beige) sont listés en espace déformé. (c) mise à jour du maillage (violet) au repos. (d) nouveau maillage en espace déformé.

3.3.3 Mise à jour du maillage triangulaire

Une fois déterminée la liste des voxels intersectant le modèle coupant, ceux-ci sont supprimés de la topologie. L'ensemble des triangles en relation avec ces voxels sont supprimés (figure 3.5(b)). L'ajout du nouveau patch triangulaire est illustré sur la figure 3.5(c). Ce travail s'effectue dans l'espace non déformé, pour bénéficier d'une grille régulière et calculer efficacement les indices de relation à partir des coordonnées. On déduit de la liste des voxels supprimés l'ensemble minimal des cubes, définis par ces voxels, à parcourir avec l'algorithme du marching cubes pour calculer le nouveau patch, ajouté à la topologie triangulaire. La fusion avec l'ancien maillage est effectuée en fournissant au marching cubes la liste des sommets composant la bordure du patch précédemment supprimé. Ainsi, les nouveaux triangles créés utilisent les sommets de l'ancien maillage et le nouveau patch triangulaire est ajouté à la topologie. Les coordonnées barycentriques des nouveaux points par rapport aux hexaèdres sont calculées dans l'espace non déformé. Elles sont ensuite utilisées pour positionner les points en fonction de la déformation actuelle des hexaèdres.

3.4 Réponse au contact

La découpe intervient lorsqu'un seuil de pression donné est dépassé. La pression s'exprime sous la forme :

$$\mathbf{F} \cdot \mathbf{n} = p \, dS$$

où p est la pression définie par une force \mathbf{F} appliquée sur une surface élémentaire dS de normale \mathbf{n} , comme illustrée sur la figure 3.6(a).

Pour une surface et une force données, la pression dans les objets est constante. Mais la force appliquée étant déterminée par le volume de collision entre les modèles, une représentation intuitive de la relation entre volume de collision et pression peut être faite comme illustré sur la figure 3.6.

La diffusion de la pression au sein d'un volume d'intersection peut être représentée comme sur la figure 3.6(b). Les contraintes de chacun des points dans le volume d'inter-

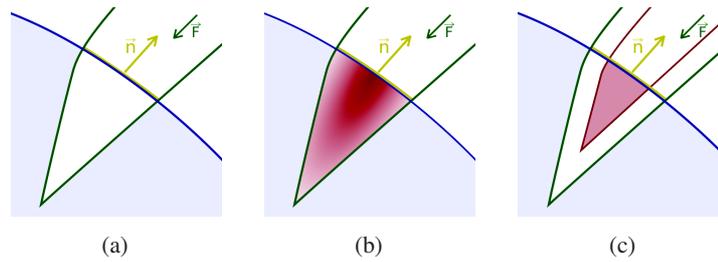


FIGURE 3.6: Réponse au contact : l'objet découpeable (bleu) est en collision avec une lame (verte). (a) Le vecteur n représente la normale de la surface de collision (jaune) et F la force appliquée par la lame. (b) Visualisation de la pression dans le volume de collision. (c) Modèle coupant (rouge) à l'intérieur du modèle de collision (vert); isosurface d'un seuil de pression.

section dépendent de la distance au bord de ce volume. Plus un point est à l'intérieur du volume, plus la pression subie est importante. Une isosurface peut être générée pour une pression donnée, comme illustré sur la figure 3.6(c). Cette isosurface correspond à la géométrie du modèle coupant, décalée d'une certaine distance qui dépend du seuil de pression choisi.

La découpe avec réponse au contact est modélisée à l'aide d'un second modèle à l'intérieur du modèle coupant. Le premier modèle (en vert sur la figure 3.6(c)) sert à calculer les forces de répulsion à appliquer aux modèles en collision tandis que le second (en rouge), interne au premier, supprime la matière des modèles qu'il intersecte.

3.5 Rendu

Pour améliorer le maillage triangulaire généré par marching cubes, un algorithme de lissage est appliqué. Une position de référence non lissée est conservée pour l'opération de fusion des maillages lors du remaillage local en espace non déformé. Lorsqu'une partie du maillage est mise à jour, l'algorithme de lissage est appliqué sur le nouveau patch en fixant les sommets de la bordure pour assurer la jointure avec le maillage existant.

Les informations médicales sont utilisées pour plaquer des textures sur le maillage. Lors de la création ou de la mise à jour du maillage par application du marching cubes, l'identifiant de segmentation du plus proche voxel est associé à chaque sommet créé. Chaque identifiant possède une texture et des paramètres de rendu correspondant au matériau identifié. Lors du rendu, cet identifiant est utilisé dans un fragment shader pour plaquer la texture associée à ce sommet. L'utilisation des identifiants de segmentation permet ainsi de texturer le maillage au fil de la découpe sans avoir recours à une texture en trois dimensions, comme illustré sur la figure 3.1.

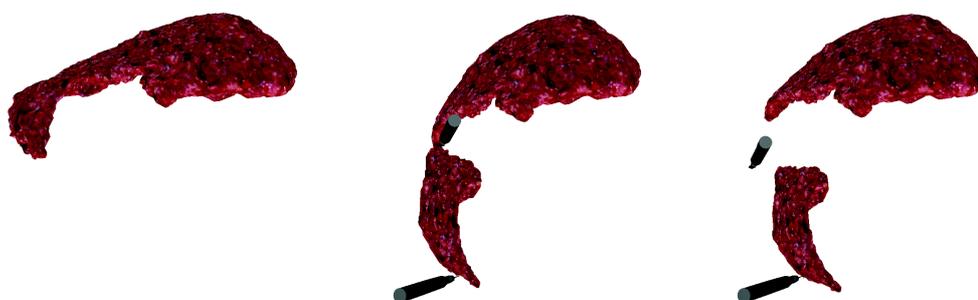


FIGURE 3.7: *Simulation chirurgicale d'un foie. Centre et droite : le foie est déformé et coupé interactivement.*

Chapitre 4

Fonctions de forme relatives aux matériaux et Echantillonnage

Contents

4.1	Présentation du modèle	58
4.1.1	Rappel lexical	59
4.1.2	Choix d'échantillonnage et du modèle interpolant	59
4.2	Fonctions de forme relatives aux matériaux	62
4.2.1	Première approche : les matériaux homogènes	62
4.2.2	Fonctions de forme relatives au matériau	64
4.3	Distribution des nœuds	69

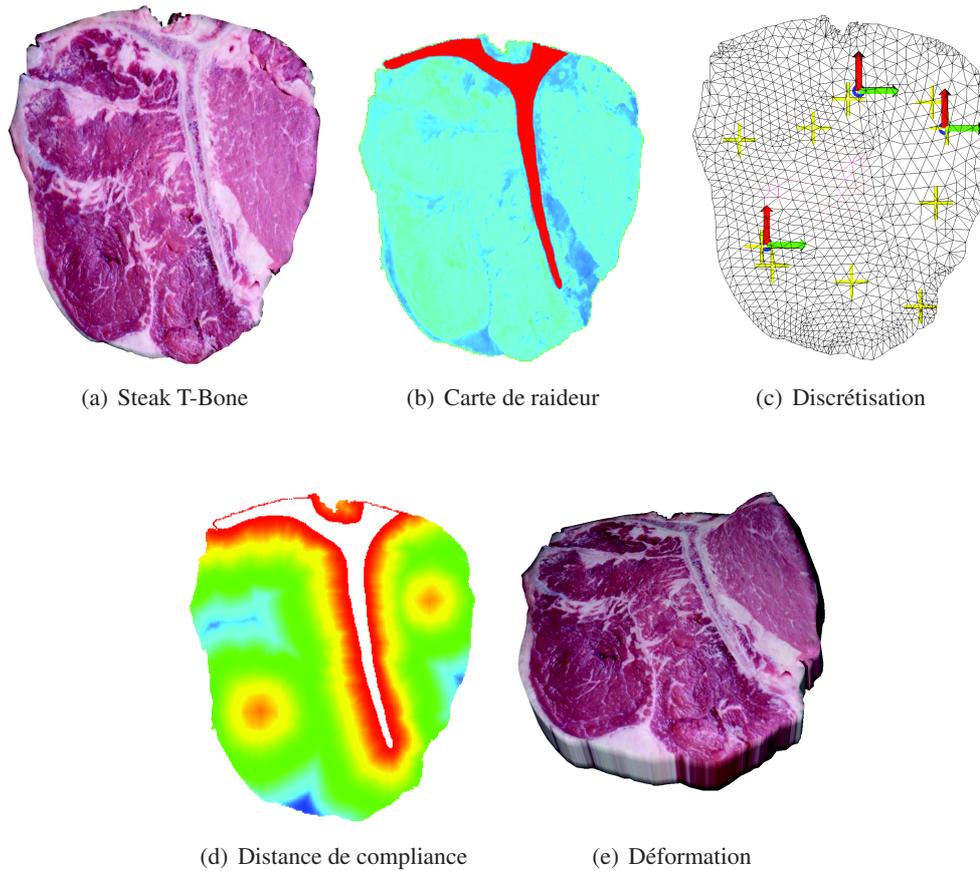


FIGURE 4.1: Simulation d'un steak T-Bone (a) comprenant un os rigide et des tissus déformables à raideurs hétérogènes (b). Notre méthode permet une exécution à 500 Hz sur un pc ordinaire avec trois repères et dix points d'intégration (c). Les repères sont automatiquement placés en fonction d'une nouvelle distance reposant sur la souplesse du matériau (d). Lorsqu'une partie du steak est déformée (e), l'os reste rigide et les deux parties contenant de la chair sont correctement découplées.

4.1 Présentation du modèle

Les modèles déformés de manière physique sont devenus très répandus en informatique graphique. Ils permettent des applications interactives tels que les jeux vidéos et les simulateurs dans lesquels les entrées utilisateurs influent sur la scène pour obtenir des déformations réalistes en temps réel. Ils sont aussi utilisés dans les films pour générer de manière efficace des déformations complexes difficile à réaliser à l'aide des techniques cinématiques courantes. Les éléments finis sont devenus populaires depuis quelques années pour calculer les déformations d'objets viscoélastiques. Plus récemment, les modèles meshless, dont la simulation est visuellement réaliste mais non physique, ont été proposés pour modéliser des comportements complexes tels que les mélanges ou les déchirures.

La majorité des travaux se sont focalisés sur l'animation d'objets composés de matériaux homogènes isotropes. Néanmoins, la plupart des objets réels, tel que le steak T-

Bone de la figure 4.1, sont composés de matière hétérogène. La simulation d'objets aussi complexes à l'aide des techniques de simulations précédentes requiert une discrétisation spatiale à haute résolution pour résoudre les variations des propriétés du matériau. En conséquence, la simulation réaliste de tels objets est restée impossible dans les applications interactives. Avec ce modèle de simulation par repères déformables, nous traitons cette problématique et présentons une nouvelle approche de simulation pour les objets hétérogènes, aux raideurs variables. Ce chapitre décrit l'encodage des propriétés du matériau dans les fonctions de forme et le chapitre 5 le pipeline dynamique de la méthode. Nous proposons une implémentation meshless dans laquelle les nœuds utilisés sont des repères et nous montrons qu'il est possible de simuler des objets hétérogènes complexes avec un nombre réduit de points d'échantillonnage à l'aide de nouvelles fonctions de formes basées sur les propriétés matérielles.

4.1.1 Rappel lexical

Dans les simulations numériques d'objets continûment déformables, le déplacement est habituellement échantillonné sur un nombre discret de primitives, appelés *nœuds*, composés des degrés de liberté indépendants. Dans le cas des éléments finis, les nœuds sont des points composés des degré de liberté x, y, z en trois dimensions. Nous utilisons des nœuds de contrôle de différent types : les points couramment utilisés mais aussi des repères rigides ou déformables par une transformation affine ou quadratique. Par simplicité de rédaction, nous nommons ces nœuds *points*, *repères rigides*, *repères affines* et *repères quadratiques*.

Aux nœuds sont associés des *noyaux* ou des *fonctions de formes*. Les fonctions de forme sont similaires aux noyaux mais celles-ci sont normalisées pour obtenir une partition de l'unité du domaine. Elles définissent le *champ de déplacement* mettant en relation les nœuds et les points matériels du solide. Ainsi par exemple, le déplacement d'un objet continu est calculé par résolution de l'*équation différentielle* sur l'ensemble discret des nœuds, puis interpolé en tous points de l'objet à l'aide des *fonctions de formes*.

Pour la méthode des éléments finis, ces fonctions sont définies dans chaque cellule du maillage sur les points de contrôles (typiquement par interpolation barycentrique), tandis que pour les méthodes meshless, elles sont définies dans un voisinage sphérique à chaque point de contrôle. Le champ de déplacement est un mélange des fonctions de forme associées aux nœuds encodant leur influence sur la matière environnante.

Afin de réaliser les intégrations spatiales, l'objet est échantillonné dans notre cas suivant une grille régulière à résolution arbitraire. Les voxels de cette grille sont nommés *échantillons* ou *points d'échantillonnage*. A ne pas confondre avec les *points d'intégration* relatifs à notre méthode. Ces derniers sont définis d'après les échantillons par une approximation aux moindres carrés dans le but de réduire le nombre d'échantillons à intégrer pendant la simulation.

4.1.2 Choix d'échantillonnage et du modèle interpolant

Dans les méthodes traditionnelles, les fonctions de forme sont définies géométriquement de manière à atteindre un certain degré de localité et de continuité, indépendamment du matériau. Les déformations résultantes sont homogènes entre les nœuds, ce qui cor-

répond à une raideur homogène. Simuler avec précision des objets hétérogènes requiert des nœuds à la frontière des différents matériaux, mais cela entraîne les erreurs connues de segmentation et de maillage. La simulation d'un matériau comprenant un nombre restreint de discontinuités matérielles est acceptable. Mais la simulation d'un matériau aux frontières multiples, géométriquement détaillées, requiert un échantillonnage mécanique dense et s'avère non interactive. De plus, un échantillonnage dense crée des instabilités numériques, spécialement dans le cas de matériaux raides. Notre but est de modéliser de manière efficace des objets hétérogènes complexes à l'aide d'un nombre réduit de nœuds.

Dans la méthode des éléments finis, les valeurs de la fonction de forme sont généralement définies par les coordonnées barycentriques (généralisées) du point $\bar{\mathbf{p}}$ dans l'élément qui le contient. Il est facile de montrer qu'une interpolation linéaire (resp. quadratique) dans un tétraèdre est équivalente au skinning par mélange linéaire avec un repère affine (resp. quadratique). Étant donné que les fonctions de forme ne sont pas bornées par des éléments, le skinning par mélange linéaire présente une interpolation globale, préférable lors de l'utilisation de repères épars.

Dans les méthodes meshless, les fonctions de forme sont calculées à partir des fonctions noyau $\phi_i(\|\bar{\mathbf{p}} - \bar{\mathbf{x}}_i\|)$. Pour les modèles paramétriques (e.g., les splines), les fonctions de forme sont généralement des fonctions polynomiales des coordonnées matérielles. Cependant, les fonctions de forme non linéaires peuvent entraîner des artefacts lors de la modélisation de matériaux homogènes par des primitives espacées : dans le cas de pures extensions, le gradient de déformation $F = d\mathbf{p}/d\bar{\mathbf{p}}$ souhaité est uniforme, ce qui survient seulement si les fonctions de forme sont linéaires dans $\bar{\mathbf{p}}$ (i.e., $dw_i/d\bar{\mathbf{p}}$ est uniforme). Contrairement au skinning par mélange linéaire (eq.5.1), la méthode des moindres carrés (MLS) [FM03] approxime le déplacement par $\mathbf{u} = \mathbf{A}(\bar{\mathbf{p}})\bar{\mathbf{p}}^* - \bar{\mathbf{p}}$, où la matrice $3 \times d$ résultante du mélange $\mathbf{A}(\bar{\mathbf{p}})$ est calculée lors de l'initialisation par minimisation de l'erreur d'interpolation à l'endroit des primitives. Pour un point \mathbf{p} , l'erreur est localement pondérée à l'aide d'une fonction noyau : $e = \sum_i \phi_i(\|\bar{\mathbf{p}} - \bar{\mathbf{x}}_i\|) \|\mathbf{A}(\bar{\mathbf{p}})\bar{\mathbf{x}}_i^* - \mathbf{A}_i\bar{\mathbf{x}}_i^*\|^2$. La convergence des moindres carrés mène à la solution la plus proche :

$$\begin{aligned}
 \mathbf{u} &= \sum_i \phi_i(\|\bar{\mathbf{p}} - \bar{\mathbf{x}}_i\|) \mathbf{A}_i \bar{\mathbf{x}}_i^* \bar{\mathbf{x}}_i^{*T} \mathbf{G}^{-1} \bar{\mathbf{p}}^* - \bar{\mathbf{p}} \\
 \mathbf{G} &= \sum_i \phi_i(\|\bar{\mathbf{p}} - \bar{\mathbf{x}}_i\|) \bar{\mathbf{x}}_i^* \bar{\mathbf{x}}_i^{*T}
 \end{aligned} \tag{4.1}$$

Un inconvénient des MLS est que \mathbf{G} est inversible seulement si le point est pondéré par au moins d points de contrôles non coplanaires. Pour remédier à cela, les moindres carrés généralisés (GMLS) ont été développés [FM03, MKB⁺10]. Contrairement au skinning par mélange linéaire, l'uniformité de la déformation n'est pas assurée avec les GMLS, comme F est non linéaire dans $\bar{\mathbf{p}}$ de par la fonction rationnelle \mathbf{G}^{-1} . De plus, l'utilisation des coordonnées barycentriques comme fonctions noyau maximise la linéarité. Les GMLS, quant à elles, procurent des interpolations linéaires comme sur la figure 4.2, au lieu de la flexion continue et naturelle obtenue par skinning. Ce qui peut s'avérer suffisant avec un échantillonnage dense, puisque la flexion peut être approximée par une séquence de transformations affines. Dans le cas d'un échantillonnage faible, la possibilité de simuler toutes les déformations naturelles entre deux nœuds est fortement souhaitée, ce qui fait du skinning un meilleur choix.

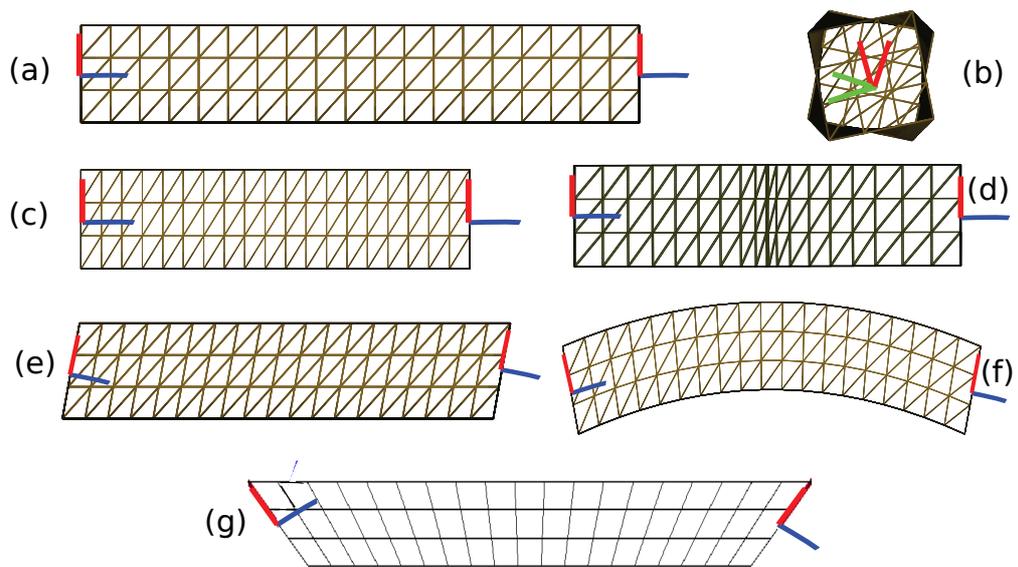


FIGURE 4.2: Modes de déformation. (a) Position au repos. (b) Torsion. (c), (d) Compression avec des fonctions de forme linéaires (resp. non linéaires). (e) Cisaillement. (f) Flexion obtenue par skinning. (g) ... contrairement aux GMLS avec des poids linéaires.

Un point très intéressant de [MKB⁺10] est la nouvelle approximation affine (au premier ordre) de la déformation, appelée élaston. À l'inverse de la valeur scalaire (degré zéro) de la déformation habituellement utilisée, cela permet à chaque point d'intégration de prendre en compte la flexion et la torsion en plus des modes d'étirement et de cisaillement classiques. Ainsi, un seul nœud dans le voisinage suffit à calculer un déplacement local, comme illustré sur la figure 4.6(c). Ces améliorations par rapport aux méthodes précédentes suppriment toutes les contraintes au voisinage du nœud et permettent la simulation d'objets à topologies arbitraires à l'aide d'une seule technique unifiée. Cependant, un échantillonnage dense des objets est nécessaire, entraînant des temps de calcul importants, et la question d'une plus faible densité n'est pas abordée. Modéliser un steak T-Bone comme celui représenté sur la figure 4.6(d) requiert un nombre important de nœuds à faible influence pour éviter que les nœuds présents d'un côté de l'os n'influencent pas les nœuds de l'autre côté. De plus, la simulation des parties rigides requiert des forces d'interaction importantes qui peuvent générer des artefacts numériques dûs au faible conditionnement.

En combinant le skinning et les élastons, il est possible d'obtenir une simulation de solides déformables pratiquement physique et visuellement plaisante. Contrairement à la technique des élastons qui est dense, nous nous concentrons sur un calcul rapide et une simplicité de modélisation, reposant sur un échantillonnage des nœuds faible et des fonctions de forme arbitraires. Le choix d'une méthode meshless est renforcé par sa facilité à gérer les changements topologiques. Nous prévoyons d'explorer l'aspect adaptatif de cette méthode et de l'unifier avec le modèle de découpe du chapitre précédent.

La création d'une technique meshless dont la formulation du champ de déplacement s'apparente au skinning nécessite d'inclure des degrés de liberté de rotation dans les nœuds.

Les repères rigides représentent des nœuds comprenant la translation et la rotation. Un repère rigide est équivalent à un élément fini tétraédrique en terme de degrés de liberté. L'utilisation de repères déformables (affines ou quadratiques) permet de définir le tenseur de déformation sur la primitive, à la manière des élastons. Les déformations obtenues à l'aide d'un repère affine (respectivement quadratique) sont équivalentes à celles d'un élément fini tétraédrique muni d'une interpolation barycentrique linéaire (respectivement quadratique). Des déformations d'ordres supérieurs sont possibles mais, au même titre que pour l'interpolation des éléments finis, le gain en déformation est minime pour des calculs devenant excessifs.

Les déplacements des nœuds (repères rigides, affines ou quadratiques) sont combinés localement suivant leurs fonctions de forme associées, définies par les poids w_i :

$$\mathbf{u} = \sum_i w_i(\mathbf{p}) \mathbf{A}_i \bar{\mathbf{p}}^* - \bar{\mathbf{p}}, \quad (4.2)$$

où \mathbf{A}_i correspond à la transformation du repère i . Les poids constituent une partition de l'unité ($\sum w_i(\mathbf{p}) = 1$) et peuvent être calculés par normalisation des fonctions noyaux. Pour imposer des conditions aux bords de Dirichlet, il est pratique de disposer des fonctions d'interpolation à $\bar{\mathbf{x}}_i$, position initiale de l'origine du nœud q_i dans l'espace 3D : $w_i(\bar{\mathbf{x}}_i) = 1$ et $w_j(\bar{\mathbf{x}}_i) = 0, \forall j \neq i$.

L'équation 4.1 ressemble à l'équation de mélange 4.2, excepté que les positions initiales $\bar{\mathbf{p}}^*$ sont altérées par une matrice constante de manière à maximiser la qualité de l'interpolation. Pour pallier aux artefacts possibles du mélange linéaire, une version utilisant un mélange par quaternions duaux est aussi présentée, restreinte à l'utilisation de repères rigides. Les parties rigides sont attachées à un unique repère, tandis que les parties déformables subissent les influences de différents repères. Les variations des propriétés matérielles sont encodées dans les fonctions de forme, ce qui permet un meilleur conditionnement numérique que les approches traditionnelles. Finalement, étant donné un objet déformable à simuler et un nombre de points de contrôle correspondant au temps de calcul attendu, des critères d'optimisation peuvent être utilisés pour calculer, à l'initialisation, une discrétisation de l'objet et les fonctions de forme associées, de manière à obtenir un bon réalisme.

4.2 Fonctions de forme relatives aux matériaux

Notre approche est basée sur une simple observation : les points connectés par des matériaux raides se déplacent de manière similaires comparés à ceux d'un matériau plus souple. Le cas limite est le solide rigide, où tous les points se déplacent suivant un repère rigide. Nous montrons qu'une définition des fonctions noyaux à partir des propriétés matérielles associées aux nœuds permet l'unification des mécaniques rigides et déformables.

4.2.1 Première approche : les matériaux homogènes

Les fonctions de poids encodent les influences des nœuds et doivent être définies à chaque point d'échantillonnage. Dans la figure 4.3, un objet en forme de U est contrôlé par deux repères q_0 et q_1 . La couleur varie selon les poids relatifs à q_0 , calculés selon différentes

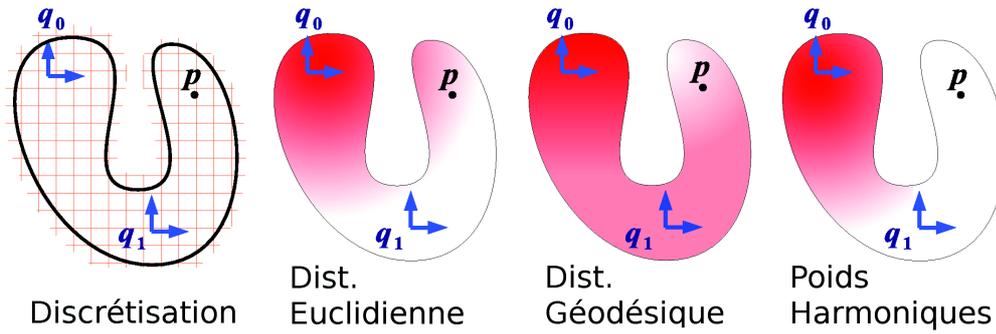


FIGURE 4.3: Gauche : discrétisation pour le calcul de intégrales volumiques. Droite : les poids associés au repère q_0 en utilisant différentes distances (rouge=poids maximal et blanc=poids minimal)

méthodes. La distance euclidienne ne parvient pas à capturer la géométrie de l'objet et des parties éloignées dans l'objet mais proches dans l'espace se déplacent ensemble. Bien que la distance géodésique soit plus adaptée, des artefacts peuvent survenir loin de tous repères ; le point \bar{p} se déplace suivant le repère q_0 même si le repère q_1 reste fixe, contrairement à ce que nous souhaiterions.

Pour résoudre cela, nous utilisons des fonctions de poids harmoniques [JMD⁺07] calculées à l'initialisation. Lors du calcul de la fonction de forme j , la valeur maximale est attribuée en \bar{x}_i , position initiale de l'origine du repère q_i dans l'espace 3D et 0 pour tout autres repères : $w_i(\bar{x}_i) = 1$ et $w_j(\bar{x}_i) = 0, \forall j \neq i$. Pour faciliter la résolution numérique de l'équation harmonique, nos échantillons sont alignés suivant une grille de voxels, et nous appliquons itérativement un masque Gaussien jusqu'à convergence. Les bords sont présumés dissipatifs, et l'application du noyau Gaussien entraîne une extrapolation des poids constante.

Dans notre exemple, cette solution permet la distribution des poids voulue, où \bar{p} est seulement influencé par le repère 1. Les gradients des poids, qui sont requis à la formulation du tenseur de déformation, sont approximés par différences finies d'après la carte des poids.

Dans le cas de l'utilisation du mélange par quaternions duaux, nos poids ne nécessitent pas que leur somme soit unitaire grâce à la normalisation intrinsèque de l'équation (5.3), et peuvent être négatifs, contrairement aux autres méthodes meshless. Lorsqu'ils sont normalisés, ils sont analogues aux fonctions de forme utilisées dans les éléments finis. De plus, en fixant des poids maximum proches des repères (pour éviter un lissage trop important par le noyau Gaussien), le champ de déformation devient interpolant et pas seulement approximant, ce qui s'avère pratique pour contraindre ou interagir avec le modèle à l'aide de contraintes de Dirichlet. Ce qui n'est pas le cas avec des méthodes meshless utilisant les moindres carrés d' [AOW⁺08, MKB⁺10], où des termes de pénalités doivent être ajoutés.

Ainsi, il est trivial d'attacher rigidement une partie d'un objet à un certain repère, en contraignant les poids des autres repères à zéro, comme pour la carapace de la tortue (figure 4.4). Les transitions vers les parties déformables sont naturellement obtenues grâce à

la continuité des fonctions harmoniques. Cet aspect très apprécié du skinning pour l'animation cinématique, est plus difficile à obtenir avec des éléments finis ou des approches basées sur des particules car les points n'encodent pas les rotations. Comme vu dans l'état de l'art, [NKJF09a] encode la raideur du matériau dans des éléments à résolution grossière en utilisant des fonctions de forme suite à une analyse statique au niveau fin, mais ils utilisent des fonctions de forme et des propriétés de matériau basiques au niveau fin. Cette dernière fonctionnalité donne l'intuition de la manière d'encoder les raideurs dans les poids pour la simulation de matériaux à propriétés non homogènes.

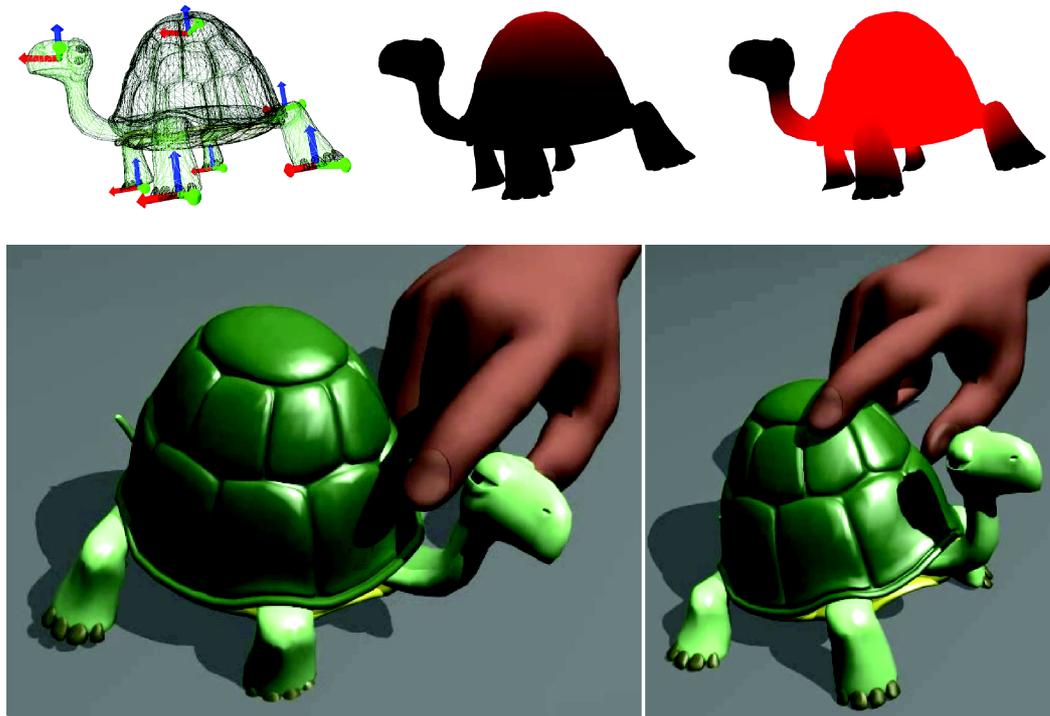


FIGURE 4.4: Modélisation de raideur par les fonctions de poids. En haut : (Gauche) Les repères de déformation. (Milieu) Les poids de la carapace sont automatiquement calculés. (Droite) Les poids sont fixés de manière à modéliser une carapace rigide. En bas : La tortue avec une carapace rigide.

4.2.2 Fonctions de forme relatives au matériau

Construire des modèles physiques à faible densité de repères requiert non seulement des fonctions de déformations appropriées comme discuté en 4.1.2, mais aussi des fonctions de forme anisotropiques pour correctement représenter les matériaux hétérogènes.

Distance de compliance

Considérons la déformation d'une barre hétérogène en une dimension, comme sur la figure 4.5, où chaque point p est paramétré par une coordonnée matérielle x . Soit les points

extrêmes \mathbf{p}_0 et \mathbf{p}_1 les points entre lesquels le champ de déplacement interpole les positions. En tout point, le déplacement est une somme pondérée du déplacement des points : $\mathbf{u}(x) = w_0(x)\mathbf{u}_0 + w_1(x)\mathbf{u}_1$. Si la barre est hétérogène, la déformation n'est pas uniforme et dépend de la raideur locale, comme illustré sur la figure 4.5(b).

Nous qualifions une fonction de forme d'*idéale* si elle encode le déplacement exact au sein de la barre étant donnés les déplacements des points extrémités, comme lors d'une analyse statique. Choisir la solution statique comme référence est arbitraire, puisque les effets d'inertie jouent un rôle dans les simulations dynamiques. Cependant, le calcul des positions internes à partir des positions au bord est un problème sous conditionné en dynamique, puisque la solution dépend des vitesses et du pas de temps. De plus, en graphique, notre perception du réalisme est plus précise pour les scènes statiques que pour les objets en mouvement. Utiliser une solution statique comme fonction de forme prend sens de ce point de vue, et encode plus d'informations qu'une fonction de forme purement géométrique.

Il est possible de déduire les fonctions de forme idéales en calculant la solution statique $\mathbf{u}(x)$ correspondante à la force de compression \mathbf{f} appliquée aux extrémités. Par simplicité, nous supposons que la barre a une section unitaire. En tout point, la compression locale est $\epsilon = \frac{d\mathbf{u}}{dx} = \mathbf{f}/E = \mathbf{f}c$, où E est le module d'Young, et son inverse c est la *compliance* du matériau (anglicisme utilisé en mécanique pour désigner la souplesse ou la flexibilité du matériau). Résoudre cette équation différentielle donne : $\mathbf{u}(x) = \mathbf{u}(x_0) + \int_{x_0}^x \mathbf{f}c \, dx$, et comme la force est constante tout au long de la barre, la fonction de forme w_0 illustrée sur la figure 4.5(d) est exactement :

$$w_0(x) = \frac{\mathbf{u}(x) - \mathbf{u}(x_1)}{\mathbf{u}(x_0) - \mathbf{u}(x_1)} = \frac{\int_x^{x_1} c \, dx}{\int_{x_0}^{x_1} c \, dx} \quad (4.3)$$

Soit la distance de compliance entre les points \mathbf{a} et \mathbf{b} définie telle que $d_c(\mathbf{a}, \mathbf{b}) = \int_{x_a}^{x_b} c \, |dx|$. La pente de la fonction de forme idéale est : $\frac{dw_0}{dx} = -c/d_c(\mathbf{p}_0, \mathbf{p}_1)$. Ce qui est proportionnel à la compliance locale c et à l'inverse de la distance de compliance entre les points extrémités. La fonction de forme est ainsi une fonction affine de la distance de compliance, comme illustré sur la figure 4.5(e), et elle peut être calculée sans avoir recours à la solution d'une équation statique.

Extension aux dimensions deux et trois

Nous avons montré dans la section précédente que le calcul des fonctions de forme idéales pour des objets 1D, sans l'aide d'analyses statiques comme dans [NKJF09a], est facilement réalisable par l'utilisation de distances de compliance.

Soit n le nombre de points dont nous voulons calculer le déplacement exact pour l'encoder dans les fonctions de forme. En dimension une, la solution statique comme le champ de distance peut être calculé en temps linéaire. En dimension deux, calculer la solution statique pour n points indépendants requiert la solution d'un système de $2n \times 2n$ équations. Le pire cas de la solution en terme de complexité est de $O(n^3)$, et les solveurs creux directs peuvent converger entre $O(n^{1.5})$ et $O(n^2)$ en pratique. En contrepartie, le calcul d'une approximation d'un champ de distance sur une grille de voxel est de $O(n \log n)$, ce qui est beaucoup plus rapide, mais ne garantit pas une solution exacte comme en dimension une. La raison est qu'il existe une infinité de chemins d'un point à l'autre pour propager

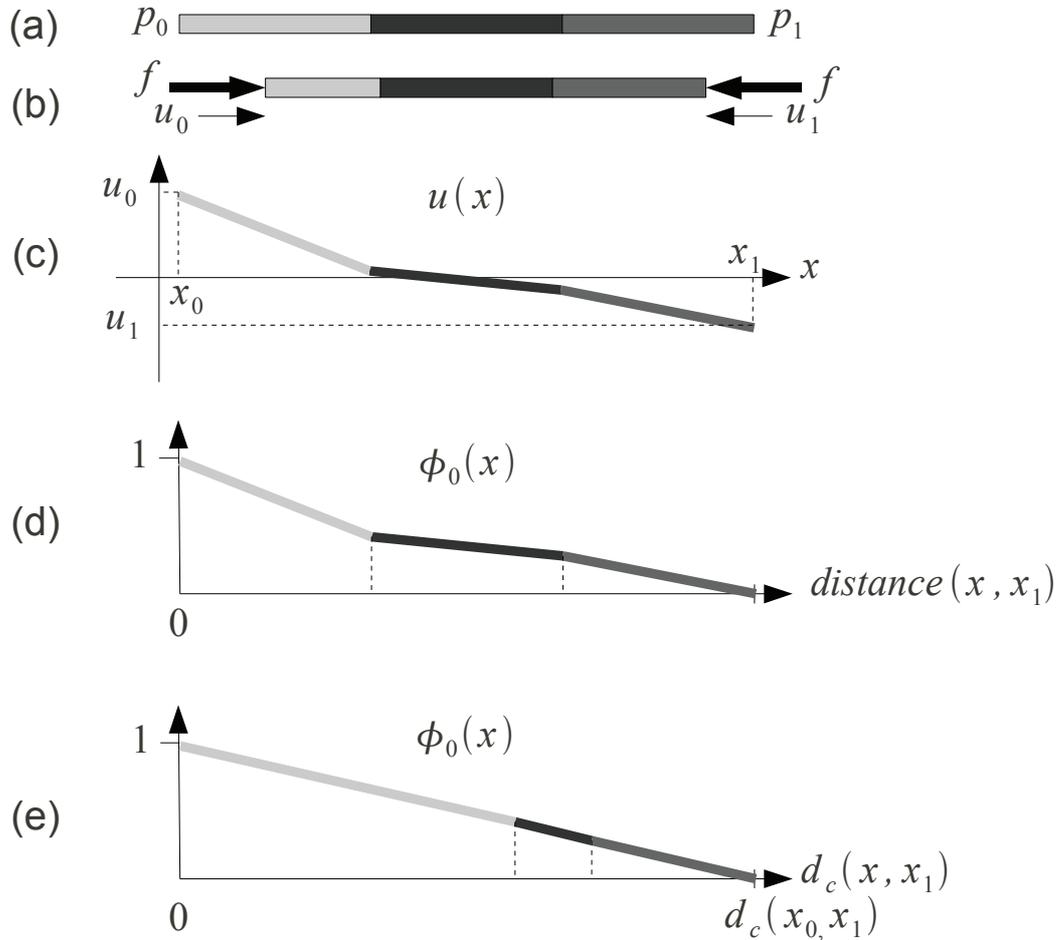


FIGURE 4.5: Fonction de forme basée sur les distances de compliance. (a) Une barre composée de 3 différents matériaux, en configuration initiale, aux raideurs proportionnelles au contraste. (b) La barre comprimée par une force externe. (c) Le déplacement le long de la barre. (d) La fonction de forme idéale w_0 pour encoder la raideur du matériau. (e) La même, fonction de la distance de compliance.

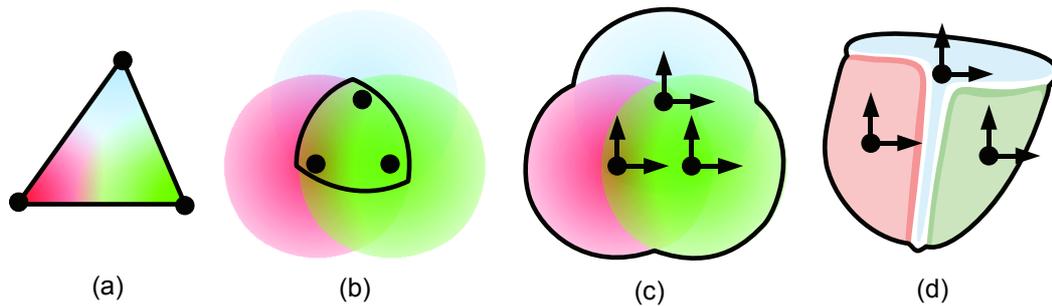


FIGURE 4.6: Comparaison de fonctions de forme. La ligne noire délimite l'aire dans laquelle la fonction de forme est définie, à partir des positions des nœuds (cercles noirs) et des fonctions associées (aires colorées). (a) Élément fini, (b) Simulation par points, (c) Simulation par repères avec des noyaux par fonctions à bases radiales, (d) Simulation par repères avec nos noyaux basés sur les propriétés matérielles.

les forces. De ce fait, les contraintes ne sont pas uniformes et ne peuvent pas être sorties de l'intégrale et simplifiées comme dans l'équation 4.3. Une autre différence avec la dimension une est le nombre de modes de déformation. Les objets de plus hautes dimensions font preuve d'étirement et de cisaillement, et nous ne pouvons pas nous attendre à ce que le taux de déplacement entre deux points soit le même dans chaque mode. Du fait qu'une valeur scalaire ne peut pas encoder différents taux, il n'existe pas de fonction de forme idéale dans les dimensions supérieures à une. Une autre limite de cette technique est que la distance de compliance est la longueur du plus court chemin (soit le plus raide) d'un point à un autre, indépendamment des autres chemins. De plus, les matériaux anisotropes ne sont pas modélisables avec des valeurs de raideur scalaires. Néanmoins, les distances de compliance permettent le calcul de fonctions de forme adaptées à la simulation de modèle complexes avec peu de repères, comme nous le montrons ensuite.

Fonctions noyaux de Voronoï

Dans les méthodes meshless présentées dans la section 2.5.2, chaque nœud est associé à une fonction noyau qui définit son influence dans l'espace. Une large variété de fonctions noyaux ont été proposées dans la littérature, le plus souvent basées sur des supports sphériques, ellipsoïdal ou parallélépipédique. Notre conception part de cette idée, et est guidée par un ensemble de propriétés que nous considérons désirables pour la simulation de modèles déformables par un faible nombre de nœuds (la simplicité, les déformations uniformes, le choix entre interpolation/extrapolation, la possibilité d'utilisation de matrices creuses, une intégration spatiale plus efficace, etc).

Pour simuler correctement l'exemple de la figure 4.6d, il est nécessaire de restreindre le recouvrement des noyaux, pour éviter que le nœud de gauche n'influence la chair présente à la droite de l'os. Les valeurs du noyau doivent ainsi être contraintes, tout en les gardant aussi lisses que possible. En particulier, nous favorisons les fonctions de forme aussi linéaires que possible dans l'espace de compliance, de manière à reproduire la solution théorique en pure extension. Les valeurs d'un noyau ne doivent pas devenir nulles

avant d'avoir atteint le nœud voisin, autrement il y a aurait une partie rigide autour de chaque nœud. Avec un nombre suffisant de fonctions à bases radiales (RBF), toutes les conditions au bord seraient respectées [Pow90]. Malheureusement, les fonctions de forme calculées par RBF sont généralement globales et décroissent avec la distance, produisant des déformations non réalistes. Les RBF locales ont un support compact isotrope, et sont ainsi seulement approximantes.

Comme il n'existe pas de solution analytique générale qui satisfasse toutes les propriétés souhaitées, nous calculons numériquement une approximation discrète de la solution sur l'objet voxelisé. Une méthode possible est de résoudre une équation de chaleur à chaque voxel, en imposant les températures aux voxels contenant les valeurs fixes. Ce qui ne permet pas d'encoder les raideurs dans les fonctions noyau, car la diffusion de chaleur converge vers le même équilibre statique, indépendamment du matériau. Une diffusion anisotrope permettrait de prendre en compte le matériau, en associant la raideur à la conductivité de la chaleur. Cependant, la convergence de ces calculs est lente. Plus important, la diffusion tend à créer des fonctions constantes sur les supports non bornés, à moins que la valeur ne soit contrainte à être nulle sur une enveloppe autour du nœud. Alors qu'une contrainte de ce type est naturelle dans les éléments finis [MKB⁺08] où des nœuds sont présents sur les bords, elle n'est pas adaptée à une technique meshless, où les fonctions noyaux sont basées sur la distance uniquement. Une solution possible serait de placer des nœuds à l'extérieur du modèle de manière à ce que tous les points de la matière soit entourés de repères. Néanmoins, lorsqu'un repère à l'extérieur du modèle influence peu de matière, le déplacement des points compris sur le bord de l'objet entraîne un grand déplacement de ce repère. La force interne engendrée est alors très importante (contrebalancée par la matrice d'inertie dont les termes sont faibles relativement à ce repère), ce qui entraîne des instabilités numériques. Nous pensons qu'il est préférable de conserver l'échantillonnage des repères au sein de la matière.

Une partition de Voronoï du volume permet de calculer simplement les noyaux avec des supports compacts, des valeurs imposées et une décroissance linéaire. Un champ de distance est calculé sur la grille de voxel par l'algorithme de Dijkstra à partir de chacun des repères. Comme un point à la frontière de deux cellules est à égale distance d des deux nœuds, la valeur des deux noyaux est de 0.5 en ce point. Chaque fonction noyau est ensuite interpolée dans la cellule, depuis la frontière jusqu'au nœud, en mettant la distance à l'échelle par $0.5/d$. Pour étendre la fonction de distance à la cellule voisine (au delà de 0.5), nous générons l'isosurface comprise entre les nœuds et l'isosurface précédemment calculée. Et ce récursivement autant que nécessaire ; générant ainsi les isosurfaces 1/2, 1/4 & 3/4, 1/8 & 3/8 & 5/8 & 7/8, etc. Les valeurs des noyaux peuvent être interpolées entre ces isosurfaces : par exemple, la valeur en P_1 sur la figure 4.7 est $(\frac{1}{2}d_{3/4} + \frac{3}{4}d_{1/2}) / (d_{3/4} + d_{1/2})$, où d_i est la distance à l'isosurface i de laquelle le point est le plus proche (selon la carte des distances de Dijkstra). Pour calculer les valeurs entre la dernière isosurface et zéro (au nœud voisin), nous appliquons un cas particulier, comme pour P_2 qui ne devrait pas être influencé par le repère. Dans ce cas, nous extrapolons linéairement la fonction noyau : $(-\frac{1}{2}d_{1/4} + \frac{1}{4}d_{1/2}) / (d_{1/4} + d_{1/2})$. Les fonctions noyaux sont des cônes généralisés, de valeur 1 au nœud et décroissantes jusqu'aux voisins. Comme les points peuvent être dans le domaine de plus de deux noyaux, une normalisation est nécessaire pour obtenir une partition de l'unité. De ce fait, la linéarité des fonctions de forme n'est pas parfaite, mais les fonctions sont souvent proches du linéaire comme montré dans la vidéo associée à

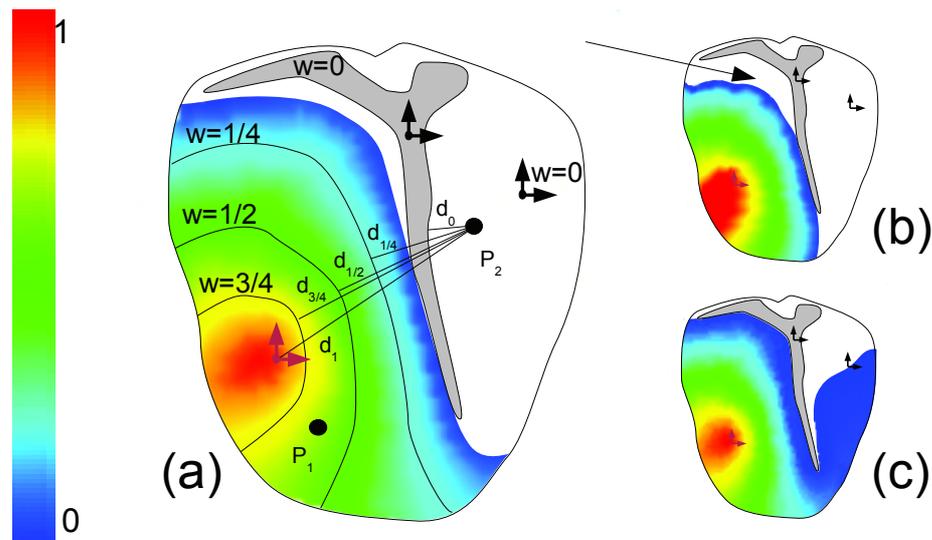


FIGURE 4.7: Carte de couleur de la fonction de forme normalisée correspondante au nœud rouge, calculée avec deux subdivisions de Voronoï (gauche), une subdivision (haut droite) et cinq subdivisions (bas droite).

[GBFP11] consultable à l'adresse <http://hal.inria.fr/docs/00/61/55/07/VIDEO/video.avi>.

Cette technique est facilement généralisable aux distances de complianc. Les formes correspondantes de cellules ne sont pas forcément convexes dans l'espace euclidien, ce qui leur permet de résoudre des distributions matérielles complexes, comme illustré sur le champ de distance de la figure 4.1(d). Cependant, cela ne garantit pas d'atteindre toutes les régions attendues (voir la flèche de la figure 4.7b), parce que le support a atteint le nœud voisin (l'os) avant de recouvrir l'ensemble de la région d'influence. Propager d'avantage pourrait dépasser l'os, qui a une épaisseur nulle du point de vue de la distance de complianc. Pour pallier à ce problème, il est possible d'utiliser plus de nœuds, de calculer de nouvelles isosurfaces de la fonction noyau, entre la surface actuelle (à 0.5) et l'os, comme celle à 0.75 illustrée sur la figure 4.7a, et extrapoler à partir de là. Ce procédé peut être répété, mais la propagation peut alors atteindre des régions inattendues, comme montré sur la figure 4.7c. La conception de noyaux plus précis est laissé en travail futur.

4.3 Distribution des nœuds

Les calculs des Voronoï procurent une façon naturelle de distribuer uniformément les nœuds dans l'espace des distances de complianc. Nous appliquons un échantillonnage standard par insertion de repères suivant le point le plus éloigné des repères existants suivi d'une relaxation de Lloyd (repositionnement itératif des nœuds au centre des régions de Voronoï), comme le font [AOW⁺08, MKB⁺10].

Il résulte de l'échantillonnage uniforme par distance de complianc une plus faible densité de repère dans les régions raides, comme illustré sur la figure 4.8. Ce qui entraîne un

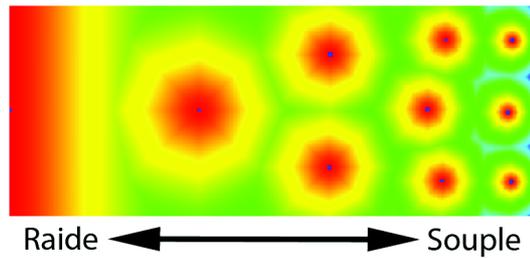


FIGURE 4.8: Une barre à raideur linéairement décroissante est simulée. Les nœuds sont distribués à l'initialisation suivant une relaxation de Lloyd basée sur la distance de compliance. Plus de nœuds sont présents dans les parties souples où les déformations sont plus importantes.

comportement raisonnable, étant donné qu'il survient moins de déformation dans ces régions. Comme une zone rigide de l'objet correspond à un point dans l'espace des distances de compliance, tous ces points dans l'espace cartésien ont les mêmes valeurs de fonctions de forme. Ce qui implique un déplacement rigide, même si ces points ne sont pas associés à un seul nœud.

Dans le cas du blending linéaire, cela pourrait entraîner une compression en cas de grandes déformations, dû aux artefacts connus du skinning. Ce problème serait atténué avec l'utilisation du mélange par quaternions duaux [KCZO07] mais au prix de calculs mécaniques plus importants, dûs à la normalisation. Bien que nous n'ayons pas remarqué cet artefact, il peut être facilement évité en initialisant les premiers nœuds dans les parties rigides et en les contraignant à ces emplacements durant la relaxation de Lloyd appliquée aux repères restant à insérer.

Chapitre 5

Dynamique par repères déformables

Contents

5.1	Algorithme de simulation	72
5.2	Champ de déplacement	73
5.3	Intégration temporelle & spatiale	74
5.3.1	Équations différentielles	74
5.3.2	Intégration spatiale	75
5.4	Masse	77
5.4.1	Pré-calcul	78
5.5	Déformations & Forces internes	78
5.5.1	Déformation	78
5.5.2	Énergie de déformation et effort	79
5.5.3	Forces internes et matrice de raideur	79
5.5.4	Optimisation	80
5.5.5	Pré-calculs	82
5.6	Surfaces de contacts et de rendu	83
5.6.1	Report des forces externes	83
5.6.2	Mise à jour du modèle visuel	84

5.1 Algorithme de simulation

La simulation physique est effectuée sur un nombre discret de nœuds, pouvant être des repères rigides, affines, ou quadratiques. Le champ de déplacement, dont la formulation s'apparente au skinning, interpole continument les quantités physiques dans le matériau entre les repères à l'aide des fonctions de forme présentées au chapitre précédent. Le modèle dynamique, qui découle d'une méthodologie classique de la mécanique des milieux continus, est similaire aux méthodes meshless.

Données : Carte de voxels décrivant les propriétés matérielles	
Initialisation :	
Distribution des nœuds de contrôle ;	// sec. 4.3
Calcul des fonctions de forme ;	// sec. 4.2
Calcul de la matrice de masse ;	// sec. 5.4
Génération des points d'intégration ;	// sec. 5.3.2
Calcul des poids des sommets de la surface ;	// sec. 5.6
Simulation :	
pour Accumulation des forces de chaque point d'intégration faire	// eq. 5.18
Calcul du gradient de déformation F (et ses dérivées spatiales);	
Calcul de ϵ (et ses dérivées) à partir de F en fonction d'une mesure de déformation donnée;	
Calcul de σ (et ses dérivées) à partir de ϵ en fonction des propriétés matérielles;	
Ajout des forces intégrées sur chaque primitive influente;	
fin	
Forces externes et gestion des collisions ;	// sec. 5.6.1
pour chaque itération du solveur faire	// eq. 5.5
Accumulation du changement de force de chaque point d'intégration ;	// eq. 5.19
fin	
Mise à jour du modèle visuel ;	// sec. 5.6.2

Algorithme 3: Calculs du modèle déformable.

L'algorithme de notre méthode est résumé dans l'algorithme 3. Notre méthode nécessite une carte volumique de densité en entrée et les paramètres du matériau (généralement le module d'Young et le coefficient de Poisson) ainsi que le nombre de repères désiré. A partir de ces données, nous calculons la distribution des nœuds et les données nécessaires à une exécution rapide de la boucle de simulation.

Concernant la boucle de simulation, les tenseurs de déformation et d'effort sont calculés de manière discrète à partir des déplacements des nœuds, à l'aide de dérivées spatiales sur des échantillons, régulièrement répartis dans la matière. Les propriétés du matériau sont utilisées pour calculer l'énergie interne relativement au tenseur de déplacement et les forces externes sont transférées sur les degrés de liberté grâce à l'expression de la Jacobienne. Les accélérations sont obtenues à partir de la somme de ces forces à l'aide de la masse. Puis, celles-ci sont enfin intégrées dans le temps pour déterminer les vitesses et les positions au prochain pas de temps.

Les étapes relatives à la dynamique du modèle sont détaillées au fil de ce chapitre. Les quantités dérivées sont définies de manière analytique pour tout point de l'espace, excepté pour les poids et leurs gradients qui sont estimés numériquement, comme développé dans la section 4.2.

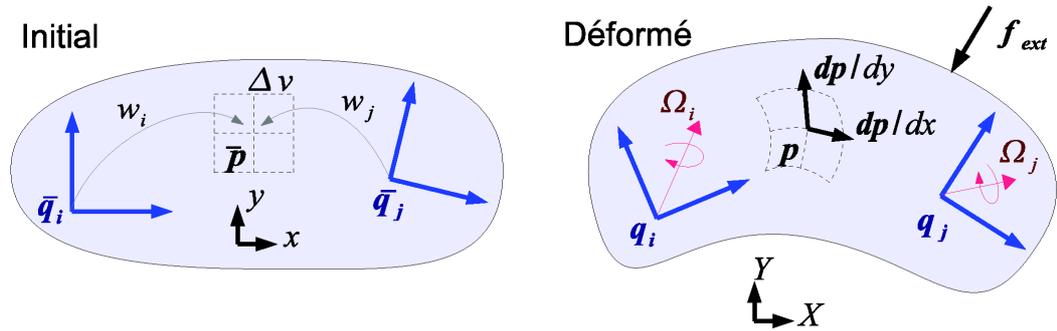


FIGURE 5.1: Un objet déformé par skinning à l'aide de deux repères.

5.2 Champ de déplacement

Pour modéliser des objets déformables avec un faible nombre de repères, il est nécessaire de définir des fonctions de déformations naturelles et pratiques. Le champ de déplacement utilisé est une généralisation du skinning, créé pour déformer une surface lisse à l'aide d'un squelette articulé. Deux formulations du skinning ont été retenues : par mélange linéaire (Skeleton Subspace Deformation - SSD) [MTLT88] et par quaternions duaux [KCZO07, KCZO08]. Comme vu au chapitre 2.4.3, le skinning par mélange linéaire est rapide mais peut produire certains artefacts. À l'inverse, le skinning par quaternions duaux présente des meilleures propriétés d'interpolations pour un coût plus élevé. Dans les deux cas, le pipeline de la dynamique est identique et les dérivations suivantes conviennent quelque soit le degré des nœuds de contrôle utilisés : repères rigides, affines ou quadratiques.

Les deux versions du champ de déplacement ont été implémentées : le mélange linéaire avec des repères rigides, affines et quadratiques, et le mélange par quaternions duaux avec des repères rigides. Pour plus de clarté, le champ de déplacement par mélange linéaire est utilisé dans les calculs par la suite. Certaines décompositions des dérivations sont données pour le mélange par quaternions duaux. Les formules utiles aux calculs relatifs au mélange par quaternion duaux sont consultables en annexe 3.

Les déplacements des nœuds de contrôle q_i sont combinés localement selon leur fonction de forme w_i , aussi appelés *poids*. Soit $\bar{\mathbf{p}}$ et \mathbf{p} les positions dans la configuration initiale non déformée et courante, et $\mathbf{u} = (\mathbf{p} - \bar{\mathbf{p}})$ le déplacement correspondant, exprimé tel que :

Cas linéaire :

$$\mathbf{u} = \sum_i w_i(\mathbf{p}) \mathbf{A}_i \bar{\mathbf{p}}^* - \bar{\mathbf{p}}, \quad (5.1)$$

où l'opérateur $(\cdot)^*$ dénote un vecteur de polynômes de dimension d et $\mathbf{A}_i(q_i)$ une matrice $3 \times d$. Les positions de référence, dans la configuration non déformée sont notés à l'aide de l'opérateur $(\bar{\cdot})$. La matrice \mathbf{A}_i représente la transformation de la primitive de l'état initial à l'état courant. Celle-ci est calculée à partir des degrés de liberté indépendants : par exemple, les 12 degrés de liberté d'une primitive affine sont traités sous la forme d'une matrice 3×4 , alors que les 6 degrés de liberté d'une primitive rigide sont converti en une matrice suivant la formule de Rodrigues. Pour les primitives points, rigides ou affines, et quadratiques,

nous utilisons respectivement des bases polynomiales complètes d'ordre $n = 0$, $n = 1$, $n = 2$, notées $(\cdot)^n$. En 3D, $d = (n + 1)(n + 2)(n + 3)/6$ et les trois premières bases sont : $\mathbf{p}^0 = [1]$, $\mathbf{p}^1 = [1, x, y, z]^T$, $\mathbf{p}^2 = [1, x, y, z, x^2, y^2, z^2, xy, yz, zx]^T$.

Cas des quaternions duaux :

$$\mathbf{u} = (\mathbf{R}\bar{\mathbf{p}} + \mathbf{t}) - \bar{\mathbf{p}}, \quad (5.2)$$

où la matrice de rotation \mathbf{R} et le vecteur de translation \mathbf{t} sont calculés comme une somme pondérée des déplacements des repères influençant le point dans la configuration initiale $\bar{\mathbf{p}}$.

Les déplacements d'un mélange de repères sont calculés comme une somme pondérée de quaternions duaux :

$$\mathbf{b}' = \frac{\mathbf{b}}{\|\mathbf{b}\|} = \frac{\sum w_i \bar{\mathbf{T}}_i \mathbf{q}_i}{\|\sum w_i \bar{\mathbf{T}}_i \mathbf{q}_i\|} \quad (5.3)$$

où la matrice $\bar{\mathbf{T}}_i$ encode le produit de quaternion duaux avec $\bar{\mathbf{q}}_i^{-1}$, l'inverse de la transformation du repère dans la position de départ. Ce produit représente la transformation appliquée à la configuration initiale, et est utilisée comme un paramètre cinématique dans [KCZO08].

Le déplacement mélangé $\{\mathbf{R}, \mathbf{t}\}$ de l'équation (5.2) est extrait du quaternion dual \mathbf{b}' . Comme les poids w_i sont fonctions de $\bar{\mathbf{p}}$, il doit être calculé à chaque sommet de la géométrie déformée.

La figure 5.1 illustre la déformation d'un objet simulé par skinning avec deux repères et les dérivations spatiales locales, correspondantes au tenseur de déformation au premier ordre en un point \mathbf{p} , initialement orthogonales.

5.3 Intégration temporelle & spatiale

5.3.1 Équations différentielles

Les modèles mécaniques lagrangiens obéissent aux équations différentielles classiques suivantes, en coordonnées généralisées :

$$\mathbf{M}\ddot{\mathbf{q}} - \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{f}_{ext}(\mathbf{q}, \dot{\mathbf{q}}) \quad (5.4)$$

où \mathbf{M} est la matrice de masse, \mathbf{q} sont les positions, $\dot{\mathbf{q}}$ les vitesses, $\ddot{\mathbf{q}}$ les accélérations, \mathbf{f} les forces internes, \mathbf{f}_{ext} les forces externes et d'inertie.

De nombreuses méthodes ont été proposées pour résoudre ces EDP. Considérons l'intégration implicite d'Euler sans perte de généralité (e.g. [BW98]), qui calcule la mise à jour des vitesses suivant l'équation :

$$(\mathbf{M} - h\mathbf{C} - h^2\mathbf{K}) \delta\dot{\mathbf{q}} = h(\mathbf{f}_{ext} + h\mathbf{K}\dot{\mathbf{q}}) \quad (5.5)$$

où h est le pas de temps, $\mathbf{K} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ est la matrice de raideur, et $\mathbf{C} = \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$ la matrice d'amortissement, souvent représentée par l'assomption de Rayleigh $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$.

Les positions des repères sont mises à jour à partir des vitesses suivant les équations :

$$\begin{aligned} \mathbf{q}_0^i(t+h) &= \mathbf{q}_0^i(t) + \frac{h}{2}(\boldsymbol{\omega}_{i\wedge} \circ \mathbf{q}_0^i(t)) \\ \mathbf{t}_i(t+h) &= \mathbf{t}_i(t) + h\dot{\mathbf{t}}_i, \end{aligned} \quad (5.6)$$

suivi par une re-normalisation de la partie quaternion.

5.3.2 Intégration spatiale

Pour appliquer l'équation dynamique (5.4), il est nécessaire d'intégrer certaines quantités sur le volume de l'objet. Dans les éléments finis, les règles de quadrature permettent généralement de calculer les intégrales dans chaque élément comme la somme pondérée d'un petit nombre de valeurs évaluées à des points paramétriques précis. Cela n'est pas possible dans notre technique où les fonctions de forme sont plus libres : les influences des nœuds ne se recouvrent pas de façon régulière, tous points de la matière étant influencé par un nombre changeant de nœuds.

Nous calculons l'intégrale d'une fonction f en discrétisant régulièrement le volume à l'intérieur de la boîte englobante de l'objet non déformé par :

$$\int_{\mathcal{V}} f(\bar{\mathbf{p}})d\mathcal{V} \simeq \sum_k f(\bar{\mathbf{p}}_k)\Delta v, \quad (5.7)$$

où le volume Δv correspond à la résolution d'échantillonnage, et où la fonction est nulle à l'extérieur de l'objet (voir la figure 4.3 p.63 de gauche et la figure 5.2). Cela nous permet de modéliser des objets complexes aux raideurs et aux masses non-uniformes. Les valeurs de chaque voxel étant acquises à partir d'une texture 3d de propriétés. Celle-ci peut être générée (voir l'exemple 6.2.2) ou fournie par l'utilisateur. A chaque pas de temps, nous calculons les termes du système d'équation différentielle (eq : 5.4) en sommant les quantités locales définies dans les sections 5.4-5.6.1 sur tous les échantillons.

Points d'intégration

Il est préférable de disposer de nombreux échantillons afin de bien retranscrire les variations des propriétés de la matière dans les fonctions de forme. Néanmoins, les intégrales volumiques réalisées sur les échantillons sont coûteuses. Pour améliorer ce temps de calcul, nous avons réduits leur nombre par l'utilisation de fonctions d'évaluations, à la manière de [AKJ08]. Ces fonctions d'évaluation, appelés *points d'intégration*, nous permettent d'évaluer certaines quantités physiques des échantillons (dans notre cas la déformation et ses possibles dérivées spatiales en ce point) et sont illustrées sur l'exemple du steak T-Bone de la figure 4.1(c) p.58 par les 10 croix jaunes. Dans l'exemple précédent de la figure 5.2, l'ensemble la centaine d'échantillons sont réduits à quelques points d'intégration. Des déformations telles que l'étirement, le cisaillement, les flexions et les torsions, peuvent être obtenues avec seulement deux repères et un point d'intégration. Cette optimisation a été implémentée avec la formulation linéaire du champ de déplacement.

La précision de ce procédé, appelé *cubature*, est décrit selon son ordre, et les fonctions polynômiales de degrés inférieurs peuvent être intégrés de manière exacte. Comme

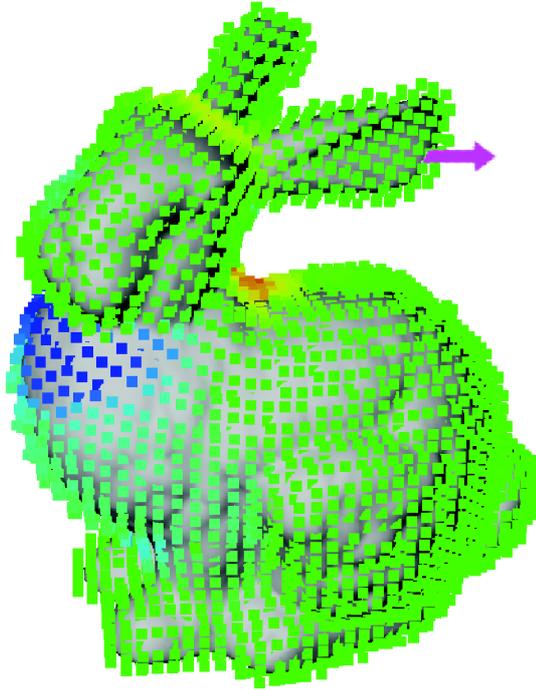


FIGURE 5.2: Des points d'échantillonnage dans une configuration déformée dont le tenseur de déformation est coloré selon une échelle hue (exemple haute résolution : 15000 échantillons, 2 repères)

discuté dans la section 4.2.2, des fonctions de forme linéaires sont souhaitables lors d'un nombre réduit de nœuds, de part leur capacité à générer des étirements uniformes. La table suivante résume les degrés des différentes quantités obtenues avec des fonctions de forme linéaires pour différentes primitives et mesures de déformations. Les méthodes de cubature

Nœud	Mesure de déformation	\mathbf{u}	\mathbf{M}	F	ϵ, σ	\mathbf{f}
Affine/Rigide	Corotationnel	2	4	1	1	2
Affine/Rigide	Green-Lagrange	2	4	1	2	4
Quadratique	Corotationnel	3	6	2	2	4
Quadratique	Green-Lagrange	3	6	2	4	8

TABLE 5.1: Degré polynômial obtenu avec des fonctions de forme linéaires.

telles que la règle du point médian (ordre 1), la règle de Simpson (ordre 3) ou la cubature de Gauss-Legendre (ordre 5) requièrent de nombreux points d'évaluation pour être précis. Les points d'évaluation les plus représentatifs peuvent être estimés comme dans [AKJ08], mais cela requiert une analyse statique intensive à l'initialisation. Heureusement, les déplacements résultants du skinning par mélange linéaire peuvent être facilement dérivés et toutes les quantités peuvent être intégrées explicitement dans les régions aux poids linéaires. Dans une région \mathcal{V}_e centrée sur $\bar{\mathbf{p}}$ et contenant les points $\bar{\mathbf{p}} + d\bar{\mathbf{p}}$, une quantité

intégrée $\int_{\mathcal{V}_e} v = \mathbf{v}^T \int_{\mathcal{V}_e} d\bar{\mathbf{p}}^n$ où \mathbf{v} est un vecteur contenant la quantité v et ses dérivés spatiales au degré n , et $\int_{\mathcal{V}_e} d\bar{\mathbf{p}}^n$ est la base polynômiale intégrée d'ordre n sur la région. Ce dernier terme peut être estimé à l'initialisation grâce à la carte de voxels. Cette intégration est exacte si n est le degré polynômial de v . Cette formulation généralise le concept des élastons [MKB⁺10], où les quantités d'ordre $n = 2$ sont implicitement intégrées dans des régions cubiques. Lorsque $n = 0$, le schéma d'intégration est équivalent à la règle du point médian : $\int_{\mathcal{V}_e} v \approx v\mathcal{V}_e$.

La méthode présentée dans la section 4.2.2 génère des fonctions de forme aussi linéaire que possible. Cependant, les gradients sont discontinus aux bords des régions d'influence. Le volume est par conséquent partitionné en régions influencées par le même ensemble de nœuds, et un point d'intégration \mathcal{V}_e est placé dans chacune de ces régions. Pour accroître la précision, les régions restantes sont récursivement subdivisées suivant le nombre de points d'intégration fixés par l'utilisateur. Le critère de subdivision est basé sur l'erreur engendrée lors de l'approximation des poids des voxels par les moindres carrés en une fonction linéaire. Pour plus de précision, les changements de raideur peuvent aussi être considérés dans l'évaluation de l'erreur. Une mesure de celle-ci, obtenue sur la scène du steak T-Bone, est illustrée sur la figure 5.3.

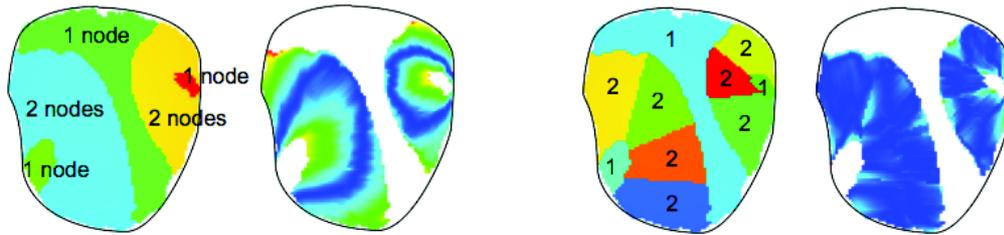


FIGURE 5.3: (A gauche) Les zones d'influences des nœuds et l'erreur d'approximation associée pour deux points d'intégration. (A droite) Des points d'intégrations sont insérés pour minimiser l'erreur.

5.4 Masse

La matrice de masse généralisée est construite par assemblage des blocs \mathbf{M}_{ij} suivant la méthode du chapitre 1.8.

Cas des quaternions duaux : Soit ∇_i un opérateur de dérivation tel qu'il dénote la différentiation par rapport aux d degrés de liberté indépendants d'un nœud i , tel que, pour un vecteur de dimension n donné $\mathbf{u}(\mathbf{q})$, nous ayons $\dot{\mathbf{u}} = \sum_i \nabla_i \mathbf{u} \dot{\mathbf{q}}_i$ où $\nabla_i \mathbf{u}$ est une matrice $n \times d$.

La Jacobienne est obtenue par différentiation de l'équation (5.2), décomposée suivant :

$$\mathbf{J}_i = \nabla_i \mathbf{p} = \frac{\partial \mathbf{p}}{\partial \mathbf{b}'} \frac{\partial \mathbf{b}'}{\partial \mathbf{b}} \frac{\partial \mathbf{b}}{\partial \mathbf{q}_i} \nabla_i \mathbf{q}_i = w_i \mathbf{Q} \mathbf{N} \bar{\mathbf{T}}_i \mathbf{L}_i \quad (5.8)$$

Les matrices $\bar{\mathbf{T}}_i(8 \times 8)$ et \mathbf{L}_i sont introduites dans la section 5.2. La matrice $\mathbf{N}(8 \times 8)$ est une projection qui filtre le changement de norme du quaternion dual. La matrice $\mathbf{Q}(3 \times 8)$ encode

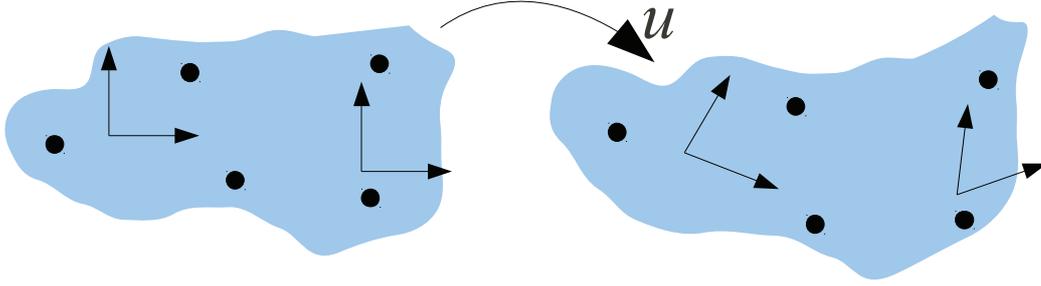


FIGURE 5.4: Le déplacement u d'un objet déformable (bleu) est discrétisé suivant les nœuds (cercles noirs). La déformation est mesurée d'après la déformation des repères locaux (flèches noires), calculés sur les points d'intégration du matériau.

le déplacement d'un point donné associé au changement du quaternion dual normalisé \mathbf{b}' . Les expressions et les détails de ces matrices sont consultables en annexe 3.

5.4.1 Pré-calcul

Par simplicité, les termes croisés sont négligés (c.f. mass lumping) : $\mathbf{M}_{ij} = 0, \forall i \neq j$. La matrice résultante est une matrice diagonale par blocs et chaque bloc \mathbf{M}_{ii} est une matrice carrée, simplifiant l'étape d'intégration temporelle sans artefacts visibles.

La matrice de masse est pré-calculée à l'initialisation, comme expliqué au chapitre 1.8. Lors de l'exécution, un changement de base est effectué pour chaque bloc en fonction des rotations des primitives correspondantes au bloc traité. Dans le cas linéaire, les masses sont constantes pour les primitives affines et quadratiques et ne nécessitent pas de changement de base lors de l'exécution, accélérant encore le temps de calcul.

5.5 Déformations & Forces internes

5.5.1 Déformation

Les forces élastiques peuvent être dérivées suivant la méthodologie classique hyperélastique. Considérons un matériau dont les positions à l'état de repos $\bar{\mathbf{p}}(\theta)$ sont paramétrées par des coordonnées locales curvilignes θ (comme les coordonnées de texture). Quand le matériau subit une déformation, les points sont déplacés aux nouvelles positions $\mathbf{p}(\theta) = \bar{\mathbf{p}}(\theta) + \mathbf{u}(\theta)$, comme illustré sur la figure 5.4.

En tous points matériels, les dérivées premières de la fonction position \mathbf{p} par rapport aux coordonnées θ sont les vecteurs formant une base locale, représentée à l'aide de repères dans la figure 5.4, et ils sont les colonnes d'une matrice 3×3 \mathbf{F} , appelée le gradient de déformation.

$$\mathbf{F} = \frac{d\mathbf{p}}{d\bar{\mathbf{p}}} \quad (5.9)$$

Sur les figures 5.1 et 5.4, les colonnes du gradient de déformation au repos $\bar{\mathbf{F}}$ forment un repère orthonormal, tandis qu'en état déformé, les termes de compression correspondent au changement de longueur de vecteurs de cette base, et ceux de cisaillement aux angles

entre ces vecteurs. Dans le cas général, la déformation locale du matériau est la partie non rigide de la transformation $\bar{F}^{-1}F$ entre l'état initial et l'état courant.

Le tenseur de déformation, ϵ , est une mesure de la déformation. Différentes fonctions de déformation ont été utilisées, mais l'ensemble de ces fonctions sont compatibles avec cette dérivation. Pour le tenseur de Green-Lagrange : $[\mathbf{E}] = (\mathbf{F}^T \mathbf{F} - \mathbf{I})/2$ représenté sous forme d'un vecteur colonne suivant $\{\mathbf{E}\} = [\epsilon_{xx} \ \epsilon_{yy} \ \epsilon_{zz} \ \epsilon_{xy} \ \epsilon_{yz} \ \epsilon_{zx}]^T$.

Cas des quaternions duaux : La dérivation plus complexe nécessaire au calcul de F est décomposée comme suit :

$$\mathbf{F} = \frac{d\mathbf{p}(\bar{\mathbf{p}}, \mathbf{b}'(\bar{\mathbf{p}}))}{d\bar{\mathbf{p}}} = \frac{\partial \mathbf{p}}{\partial \bar{\mathbf{p}}} + \frac{\partial \mathbf{p}}{\partial \mathbf{b}'} \frac{\partial \mathbf{b}'}{\partial \mathbf{b}} \frac{\partial \mathbf{b}}{\partial \bar{\mathbf{p}}} = \mathbf{R} + \mathbf{QNW} \quad (5.10)$$

La matrice $\mathbf{W}_{8 \times 8}$ encode la dépendance de la somme pondérée \mathbf{b} au points et dépend des variations locales $\mathbf{W}_i = \frac{\partial w_i}{\partial \bar{\mathbf{p}}}$ des fonctions de poids dans le matériau (voir l'annexe 3).

5.5.2 Énergie de déformation et effort

La réponse élastique $\sigma(\epsilon)$ génère les forces élastiques internes, et est une caractéristique physique du matériau. Son produit avec le tenseur de déformation est homogène à la densité d'une énergie. L'énergie élastique d'un objet déformé est le travail accompli par les forces élastiques de l'état déformé à l'état courant, intégré sur l'ensemble de l'objet : $\mathcal{W} = \int_{\mathcal{V}} \int_0^\epsilon \sigma d\epsilon$.

Exemple : Pour la formulation de l'élasticité de Hooke d'un matériau de Saint-Venant-Kirchhoff, l'effort \mathbf{S} sous forme vectorielle est $\mathbf{S} = \mathbf{HE}$, où \mathbf{H} est la matrice de raideur standard de Hooke. La densité de l'énergie de déformation : $\mathcal{W} = \mathbf{E}^T \mathbf{S}/2$, soit :

$$\mathcal{W} = \frac{1}{2} \mathbf{E}^T \mathbf{H} \mathbf{E} \quad (5.11)$$

Des modèles non-linéaires peuvent facilement être obtenus par mise à jour de \mathbf{H} et de ses dérivées en fonction de \mathbf{E} à chaque pas de temps. Pour les modèles non-uniformes, \mathbf{H} dépend simplement de $\bar{\mathbf{p}}$ et pour des matériaux anisotropes, \mathbf{H} prend la forme généralisée discutée dans [Bat96].

Des modèles plus généraux peuvent être dérivés d'après cette méthodologie.

5.5.3 Forces internes et matrice de raideur

Soit \mathbf{q} le vecteur de degrés de liberté indépendants de l'objet, responsables des déplacements \mathbf{u} . Les forces élastiques associées \mathbf{f} sont calculées par dérivation de l'énergie par rapport aux degrés de liberté :

$$\mathbf{f}^T = -\frac{\partial \mathcal{W}}{\partial \mathbf{q}} = -\int_{\mathcal{V}} \frac{\partial \epsilon}{\partial \mathbf{q}} \sigma \quad (5.12)$$

Une dérivation supplémentaire fonction de la raideur $\frac{\partial \mathbf{f}}{\partial q}$, est utilisée pour la résolution de solveurs itératifs implicites et statiques. La plupart des solveurs itératifs, comme c'est le cas pour le gradient conjugué utilisé dans nos tests, traite seulement la matrice par le biais de son produit avec un vecteur, en calculant le changement de force $\delta(\mathbf{f})$ correspondant à un changement infinitésimal de position $\delta(\mathbf{q})$. Cela évite d'avoir à calculer la matrice de raideur de manière explicite, et permet de calculer seulement les changements des termes dans l'expression de la force et d'accumuler leur contributions :

$$\delta(\mathbf{f}^T) = - \int_{\mathcal{V}} \frac{\partial \epsilon}{\partial \mathbf{q}} \frac{\partial \sigma}{\partial \epsilon} \frac{\partial \epsilon^T}{\partial \mathbf{q}} \delta(\mathbf{q}) - \int_{\mathcal{V}} \delta\left(\frac{\partial \epsilon}{\partial \mathbf{q}}\right) \sigma \quad (5.13)$$

Le premier terme correspond au changement de l'intensité des contraintes. Le second correspond à un changement de direction, dû à la non linéarité, et peut être nul ou négligeable, en fonction de l'interpolation et des fonctions de déformation. Les forces d'amortissements, basées sur la vitesse, peuvent être directement dérivés et ajoutées aux forces élastiques.

Cas des quaternions duaux : Le calcul des forces est basé sur la dérivée de la déformation par rapport aux degrés de liberté :

$$\nabla_i \mathbf{E} = \frac{1}{2} (\mathbf{F}^T \nabla_i \mathbf{F} + \nabla_i \mathbf{F}^T \mathbf{F}) \quad (5.14)$$

Ce tenseur \mathbf{B}_i du troisième ordre, de dimension $3 \times 3 \times 6$, est calculé en dérivant l'équation (5.9). Nous l'écrivons sous la forme d'un bloc 6×6 en utilisant la forme vectorielle du tenseur de déformation :

$$\mathbf{B}_i = \nabla_i \mathbf{E} \quad (5.15)$$

La force généralisée \mathbf{f}_i agissant sur le repère i est l'opposé du gradient de l'énergie par rapport aux degrés de liberté :

$$\mathbf{f}_i = - \int_{\mathcal{V}} \nabla_i \mathcal{W}^T d\mathcal{V} = - \int_{\mathcal{V}} \mathbf{B}_i^T \mathbf{H} \mathbf{E} d\mathcal{V} \quad (5.16)$$

La matrice de raideur \mathbf{K} utilisée dans les méthodes d'intégration implicite est la Jacobienne de la force. Chaque bloc 6×6 \mathbf{K}_{ij} encode la variation de la force sur i dû au déplacement de j :

$$\mathbf{K}_{ij} = \nabla_j \mathbf{f}_i = - \int_{\mathcal{V}} [\mathbf{B}_i^T \mathbf{H} \mathbf{B}_j + (\nabla_j \mathbf{B}_i^T) \mathbf{H} \mathbf{E}] d\mathcal{V} \quad (5.17)$$

Le premier terme encode le changement d'intensité des forces internes, tandis que le second induit un tenseur du troisième ordre qui encode le changement de direction. Dans nos expérimentations, ce dernier peut être ignoré, excepté pour de grandes vitesses angulaires.

5.5.4 Optimisation

Lors du calcul des forces internes, le gradient de déformation F est utilisé explicitement lors du calcul de la force :

$$\mathbf{f}^T = - \frac{\partial \mathcal{W}}{\partial \mathbf{q}} = - \int_{\mathcal{V}} \frac{\partial \epsilon}{\partial \mathbf{q}} \sigma = - \int_{\mathcal{V}} \frac{\partial \epsilon}{\partial F} \frac{\partial F}{\partial \mathbf{q}} \sigma \quad (5.18)$$

Cela permet une grande modularité : le calcul du tenseur des contraintes nécessite $\frac{\partial \epsilon}{\partial \mathbf{F}}$, et $\frac{\partial F}{\partial \mathbf{q}}$ est calculé lors de l'interpolation. Ces deux valeurs sont directement réutilisées pour le calcul de la force.

Une déduction similaire intervient pour le calcul du produit de la matrice de raideur avec un vecteur :

$$\delta(\mathbf{f}^T) = - \int_{\mathcal{V}} \frac{\partial \epsilon}{\partial \mathbf{F}} \frac{\partial F}{\partial \mathbf{q}} \frac{\partial \sigma}{\partial \epsilon} \frac{\partial F^T}{\partial \mathbf{q}} \frac{\partial \epsilon}{\partial \mathbf{F}}^T \delta(\mathbf{q}) - \int_{\mathcal{V}} \left(\delta \left(\frac{\partial \epsilon}{\partial \mathbf{F}} \right) \frac{\partial F}{\partial \mathbf{q}} + \frac{\partial \epsilon}{\partial \mathbf{F}} \delta \left(\frac{\partial F}{\partial \mathbf{q}} \right) \right) \sigma, \quad (5.19)$$

Cette modularité permet d'implémenter le mélange pour les primitives rigides, affines et quadratiques et de les combiner aisément avec une variété de mesure de déformations. Il est à noter que les autres méthodes d'interpolation, tels que les éléments finis et celles des méthodes particulières, sont compatibles avec cette architecture. Nous avons implémenté les déformations par corotationnel et de Green-Lagrange, et les lois de comportements Hookéenne. L'incompressibilité est prise en compte par la mesure du changement de volume, $\|F\| - 1$, et en appliquant une réponse scalaire à l'aide du module d'incompressibilité (bulk modulus). D'autres modèles tels que Mooney-Rivlin et Arruda-Boyce pourraient facilement être ajoutés.

Préservation de volume : Pour les matériaux à volume invariant, il est nécessaire de calculer le changement unitaire de volume par rapport aux degrés de liberté. Le volume unitaire est donné par le déterminant du gradient de déformation : $J = \det(\mathbf{F})$. Sa dérivée est calculée à partir des matrices existantes :

$$\nabla_i J_k = \frac{\partial J}{\partial q_{ik}} = \det(\mathbf{F}) \sum_{lj} \mathbf{F}^{-T}{}_{lj} \nabla_i \mathbf{F}_{ljk}$$

Les éléments inversés ont une déformation et une énergie de déformation nulles, résultant en une configuration stable mais non souhaitable. Pour éviter l'inversion, un terme d'énergie k (bulk modulus) est généralement ajouté pour pénaliser le changement de volume à partir de la configuration initiale. Les expressions possibles pour la densité d'énergie, la force et la raideur sont :

$$\mathcal{W} = \frac{k}{2}(J - 1)^2, \quad \mathbf{f}_i = -k(J - 1)\nabla_i J^T, \quad \mathbf{K}_{ij} \approx -k\nabla_i J^T \nabla_j J$$

Matériaux hyperélastiques isotropes : une approche couramment répandue en ingénierie mécanique est d'exprimer la densité d'énergie \mathcal{W} d'après les trois invariants du tenseur de déformation de Cauchy $\mathbf{C} = \mathbf{F}^T \mathbf{F}$; définis en fonction de la déformation et du volume :

$$\begin{aligned} I1(\mathbf{C}) &= Tr(\mathbf{C}) = 2\varepsilon_{xx} + 2\varepsilon_{yy} + 2\varepsilon_{zz} + 3 \\ I2(\mathbf{C}) &= (Tr(\mathbf{C})^2 - Tr(\mathbf{C}^2))/2 \\ &= 4(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz} + \varepsilon_{xx}\varepsilon_{yy} + \varepsilon_{xx}\varepsilon_{zz} \\ &\quad + \varepsilon_{yy}\varepsilon_{zz} - \varepsilon_{xy}^2 + \varepsilon_{xz}^2 + \varepsilon_{yz}^2) + 3 \\ I3(\mathbf{C}) &= \det(\mathbf{C}) = J^2 \end{aligned}$$

Leurs dérivées sont obtenues à l'aide des lignes de \mathbf{B} :

$$\begin{aligned}\nabla_i I1 &= 2(\mathbf{B}_{i1} + \mathbf{B}_{i2} + \mathbf{B}_{i3}) \\ \nabla_i I2 &= 2\nabla_i I1 - 8(\varepsilon_{xy}\mathbf{B}_{i4} + \varepsilon_{xz}\mathbf{B}_{i5} + \varepsilon_{yz}\mathbf{B}_{i6}) + 4(\varepsilon_{xx} \\ &\quad (\mathbf{B}_{i2} + \mathbf{B}_{i3}) + \varepsilon_{yy}(\mathbf{B}_{i1} + \mathbf{B}_{i3}) + \varepsilon_{zz}(\mathbf{B}_{i1} + \mathbf{B}_{i2})) \\ \nabla_i I3 &= 2J\nabla_i J\end{aligned}$$

La relation $\frac{\partial \mathcal{W}}{\partial I_i}$ est donnée par le modèle du matériau (e.g., Mooney-Rivlin). En insérant cette relation dans les équations suivantes, les forces et matrice de raideur sont définies suivant :

$$\begin{aligned}\mathbf{f}_i &= -\left[\frac{\partial \mathcal{W}}{\partial I1}\nabla_i I1 + \frac{\partial \mathcal{W}}{\partial I2}\nabla_i I2 + \frac{\partial \mathcal{W}}{\partial I3}\nabla_i I3\right]^T \\ \mathbf{K}_{ij} &\approx -\left[\nabla_i I1^T \nabla_j \frac{\partial \mathcal{W}}{\partial I1} + \nabla_i I2^T \nabla_j \frac{\partial \mathcal{W}}{\partial I2} + \nabla_i I3^T \nabla_j \frac{\partial \mathcal{W}}{\partial I3}\right]\end{aligned}$$

Pour les matériaux incompressibles, il est possible d'utiliser les invariants. Ils sont déduits des variables précédemment introduites :

$$\begin{aligned}\bar{I}1 &= J^{-2/3} I1 \\ \bar{I}2 &= J^{-4/3} I2 \\ \bar{I}3 &= J\end{aligned}\tag{5.20}$$

5.5.5 Pré-calculs

Le temps de calcul des intégrales volumiques est proportionnelle au nombre d'échantillons volumiques. Un compromis entre vitesse et précision peut être choisi en modifiant la résolution de la grille d'échantillonnage. Pré-calculer un ensemble représentatif de voxels à l'initialisation dans l'esprit de [AKJ08], comme nous l'avons fait avec les points d'intégration, accélère sensiblement la méthode. Afin de l'accélérer encore plus, nous avons tenté de pré-calculer les raideurs, à la manière de la matrice de masse, pour éviter toute intégration spatiale pendant la boucle de simulation. Cette technique a été développée pour des repères rigides et seul le principe est expliqué ici, les calculs étant détaillés dans [GBFP11]. La matrice de raideur initiale est transformée en un réseau de ressorts généralisés $6d$ (prenant en compte le déplacement en translation et rotation), dont le principe est illustré sur la figure 5.5.

A l'initialisation, pour chaque paire de repères (i, j) , les coordonnées \bar{l}_i et \bar{l}_j d'un repère virtuel (en vert) placé entre les repères i et j sont calculées. Le bloc de raideur $\bar{\mathbf{K}}_{ij}$ est utilisé pour initialiser la raideur correspondante au ressort $\bar{\mathbf{K}}_m$ en fonction des transformations $-\bar{l}_i$ et $-\bar{l}_j$.

En position déformée, l'écart des repères $m_i = q_i \circ \bar{q}_i^{-1} \circ \bar{m}$ et $m_j = q_j \circ \bar{q}_j^{-1} \circ \bar{m}$, respectivement attachés aux repères i et j , est la mesure de déformation engendrée par le déplacement relatif des repères. Le bloc \mathbf{K}_m est alors mis à jour suivant la rotation du repère virtuel m : $\mathbf{K}_m = \mathbf{R}_m \bar{\mathbf{K}}_m \mathbf{R}_m^T$. Et les forces engendrées par le ressort de raideur \mathbf{K}_m sont ensuite distribuées sur les repères en fonction des transformations l_i et l_j au temps t .

Cette optimisation permet d'obtenir des exécutions extrêmement rapides, dont les temps sont fournis dans le tableau de la section 6.1.2, mais elle s'avère exacte et stable en petits

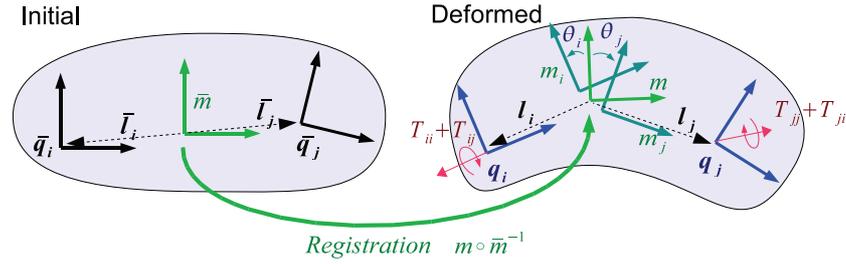


FIGURE 5.5: Principe des ressorts pré-calculés 6d. Les coordonnées d'un repère virtuel (en vert) sont calculées entre chaque couple de repères (en noir). Après déplacements des repères (en bleu), les forces sont calculées suivant la différence en translation et rotation des repères associés (en turquoise).

déplacements uniquement. Face à l'instabilité de la méthode sur les repères rigides, nous n'avons pas étendu le modèle aux repères déformables.

5.6 Surfaces de contacts et de rendu

Par simplicité d'implémentation et pour plus de flexibilité, les modèles physiques, de contacts et visuels sont indépendants. Les surfaces de rendu et de contact peuvent être attachées à l'objet déformable à l'aide de la méthode de skinning présentée dans la section 4.1.2. Ces surfaces peuvent être utilisées pour générer les données volumiques, comme illustré dans l'exemple de la section 6.2.2.

Chaque sommet des maillages est mis en relation avec les repères par le champ de déplacement. Les distances de compliance des sommets sont interpolées dans la carte volumique des matériaux, et les fonctions de forme correspondantes sont calculées. Cette méthode n'entraîne aucune restriction quant aux méthodes de détection et de réponse aux collisions employées.

5.6.1 Report des forces externes

Différentes forces externes sont appliquées au modèle physique. Il peut s'agir de forces résultantes de collisions entre modèles, de champs de forces (modélisant le vent, la flottaison, etc) ou encore la gravité. Dans certains cas où la force est identique en tout point de l'espace comme pour la gravité, il peut être plus rapide d'appliquer directement la force sur le modèle physique. Dans le cas où la force est ponctuelle et appliquée à la surface de l'objet, chaque force externe 3d \mathbf{f}_{ext} appliquée au point \mathbf{p} doit être transformée en force généralisée d'une dimension dépendant du type de nœud utilisé, étant donné que l'équation dynamique (5.4) est appliquée aux nœuds. En appliquant la loi de conservation des puissances $\mathbf{f}_{\text{ext}}^T \dot{\mathbf{p}} = \sum_i \mathbf{f}_i^T \dot{\mathbf{q}}_i$ pour chaque $\dot{\mathbf{q}}_i$, toute force \mathbf{f}_p appliquée sur un point \mathbf{p} de la surface de contact peut être accumulée sur les nœuds de contrôle suivant la relation :

$$\mathbf{f}_+ = \frac{\partial \mathbf{u}^T}{\partial \mathbf{q}} \mathbf{f}_p \quad (5.21)$$

$\frac{\partial \mathbf{u}}{\partial \mathbf{q}}$ étant la Jacobiennne \mathbf{J} , utilisée pour la construction de la matrice de masse et dans les calculs des quaternions duaux de l'annexe 3.

5.6.2 Mise à jour du modèle visuel

Le mapping de mise en relation des repères avec le maillage de rendu conserve uniquement les poids associés aux sommets. A la fin de la boucle de simulation, les sommets du maillage de rendu sont mis à jour par skinning, d'après la nouvelle position des nœuds.

Chapitre 6

Résultats et autres travaux

Contents

6.1	Modèle physique par repères	86
6.1.1	Validation	86
6.1.2	Performances	86
6.1.3	Points d'intégration	88
6.2	Exemples de simulations	88
6.2.1	Application pour l'animation	89
6.2.2	Application médicale	89
6.2.3	Enrichissement de reconstruction de mouvements	89
6.3	Simulation de découpe	90
6.3.1	Performances	90
6.3.2	Intégration du modèle physique par repères	91
6.3.3	My Corporis Fabrica	94

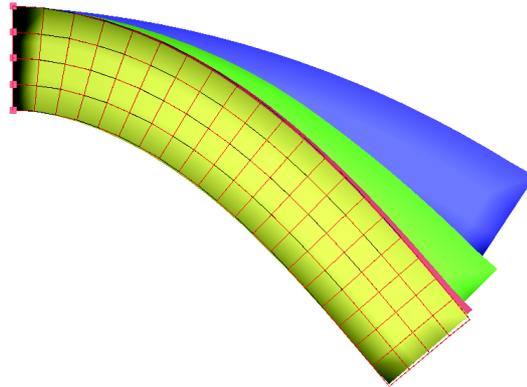


FIGURE 6.1: Comparaison de nos modèles (couleurs pleines) avec les éléments finis (en fil de fer rouge). Bleu : avec deux repères affines. Vert : trois repères affines. Rouge : cinq repères affines. Jaune : neuf repères affines.

6.1 Modèle physique par repères

6.1.1 Validation

La validation de notre modèle a été réalisée par comparaison avec la méthode des éléments finis, dont les résultats sont similaires. Nous présentons ici une simulation de poutres de propriétés homogènes soumises à la gravité. La figure 6.1 montre les résultats obtenus à l'aide des éléments finis de St-Venant-Kirchhoff (en fil de fer rouge), et de notre modèle (en couleurs pleines) comprenant un nombre croissant de repères affines : 2 (en bleu), 3 (en vert), 5 (en rouge) et 9 (en jaune). Les repères sont uniformément répartis sur l'axe médian de la barre, des fonctions de forme linéaires sont calculées analytiquement. Le comportement converge lorsque le nombre de repères est augmenté. Le maillage des éléments finis comporte $20 \times 4 \times 4$ éléments construits sur $21 \times 5 \times 5$ points, soit 1575 degrés de liberté. Contre 108 pour le modèle simulé par 9 repères affines. Sur cet exemple, 9 repères affines approximent déjà correctement le comportement obtenu par un maillage de bonne qualité d'éléments finis.

Concernant l'encodage de la raideur dans les fonctions de forme, nous proposons la mesure du déplacement de la ligne centrale d'une plaque fine de 10×4 , comme illustré sur la figure 6.2. La partie gauche est fixée, tandis qu'une traction est uniformément appliquée sur la partie droite. Comme attendu, les résultats par éléments finis et avec notre méthode sont similaires, en utilisant des paramètres matériaux identiques. Une légère sur-extension apparaît lors d'une distribution dense des repères, probablement due à des problèmes numériques lors de l'intégration de l'énergie de déformation sur les voxels.

6.1.2 Performances

Les temps de calcul sont difficiles à comparer rigoureusement car nous utilisons un solveur itératif reposant sur l'algorithme du gradient conjugué. Dans le test du matériau homogène de la figure 6.2, nous mesurons le nombre d'itérations du GC appliquées pour

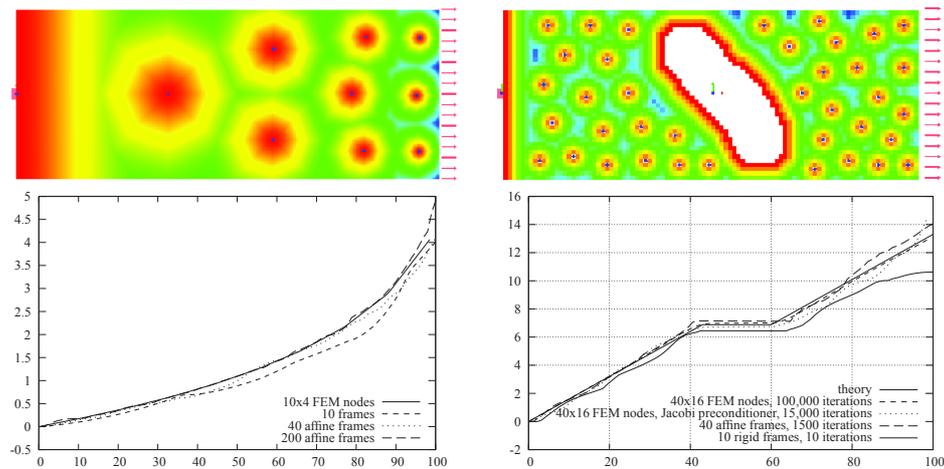


FIGURE 6.2: Comparaison de l'extension d'une plaque avec les éléments finis. Gauche : avec le gradient de raideur. Droite : Raideur uniforme et partie rigide. Haut : Champ de distance de compliance de chaque cellule de Voronoï, gauche : grille 500×200 , droite : grille 100×40

atteindre l'état final avec moins d'1% de précision. Le modèle par repères converge une à trois fois plus vite, grâce au nombre réduit de degrés de liberté. Pour les éléments finis, nous utilisons un maillage régulier. Un maillage prenant en compte les raideurs permettrait certainement de meilleures performances. Malheureusement, nous ne disposons pas d'une telle implémentation. Des maillages bien conditionnés peuvent grandement accélérer la convergence des éléments finis. Cependant, résoudre les détails géométriques requiert des maillages fins avec un nombre important de nœuds, et dans le cas de grandes variations de la raideur, des problèmes numériques réduisent considérablement la vitesse de convergence. La capacité de notre méthode d'encoder les raideurs dans les fonctions de forme réduit non seulement le nombre de nœuds nécessaires, mais semble aussi réduire les problèmes de conditionnement. Les temps d'initialisation varient de moins d'une seconde pour 10 repères sur une grille de voxels 100×40 à 10 minutes pour 200 repères sur une grille de 500×200 . De nombreuses optimisations de notre implémentation pourraient être réalisées, notamment par parallélisation.

La table 6.1 présente les taux de rafraîchissement obtenus sur un pc standard (quadcore $2.67Hz$, $8GB$, Nvidia $295GTx$). Les scènes du dragon et du ruban ne sont pas cités dans cette thèse mais sont inclus dans la vidéo accompagnant la publication [FGBP11], consultable à l'adresse <http://hal.inria.fr/docs/00/61/55/07/VIDEO/video.avi>. Ils comprennent tous les calculs, incluant les étapes de détection de collisions et le rendu. Les temps de calcul dépendent fortement du nombre de points d'intégration, ce qui suggère qu'une implémentation GPU du calcul des forces pourrait accroître de manière drastique les performances. Une relaxation des nœuds plus rapide [LWL⁺09] pourrait aussi accélérer les pré-calculs des grilles fines.

La mesure de déformation corotationnelle est approximativement 1.5 fois plus rapide que celle de Green-Lagrange, dû au degré d'intégration plus faible. Cependant, dans notre implémentation, elle n'est pas aussi robuste du fait que la partie rotation de F n'est pas

Modèle	# repères	# échantillons	# sommets	# voxels	t_{ini}	IPS
Steak	3	10	5k	67k	3s	500
Steak	10	53	5k	67k	8s	100
Steak	20	140	5k	67k	12s	40
Dragon	3	6	20k	7M	100s	300
Dragon	10	41	20k	7M	220s	150
Dragon	20	158	20k	7M	360s	27
Ruban	5	9	4k	60k	4s	200
Genou	10	200	35k	500k	11s	10
Rat	30	230	600k	1.5M	90s	8

TABLE 6.1: Temps de simulation du modèle par repères.

dérivée pour calculer les forces (dans l'équation 5.18). Leurs précisions sont comparables dans nos tests.

Les primitives rigides et affines exhibent des temps de calculs similaires. Les primitives quadratiques sont à peu près 15% plus lentes pour le même nombre de points d'intégration de même degrés. Nous pensons que le meilleur compromis entre vitesse et précision est fourni par les primitive affines : elles ont plus de degrés de liberté que les repères rigides, et capturent ainsi de plus hautes fréquences de déformations, mais nécessitent bien moins de points d'intégration que les primitives quadratiques, si le développement du gradient de déformation est au premier ordre et le degré d'intégration de 4 (= 30 termes polynômiaux). En théorie, les primitives quadratiques devraient bénéficier d'un développement de F d'ordre deux et d'une intégration d'ordre 8 (= 165 termes polynômiaux). Nous croyons que cela serait trop coûteux. Les repères rigides et affines requièrent seulement un point d'intégration par région avec des fonctions de forme linéaires, permettant tous les modes de déformation d'une corde par deux repères et un point d'intégration (voir figure 4.2).

6.1.3 Points d'intégration

Le choix de placer les points d'intégration dans les régions de recouvrement d'influences est une stratégie intéressante, dans la mesure où cela permet une bonne approximation linéaire de la fonction de forme définie sur les voxels : dans nos tests, la différence moyenne est de 0.05 (la fonction de forme étant définie entre 0 et 1, résultant en une erreur d'approximation de moins de 5%). Ce résultat montre aussi que la normalisation de la fonction noyau ne change pas significativement la linéarité. Le choix de points d'échantillonnage uniformément distribués accroît la raideur de manière non réaliste, car moyenniser des régions d'échantillonnage assigne de fortes raideurs aux parties souples.

6.2 Exemples de simulations

Trois exemples de simulations sont présentées ici afin d'illustrer les possibilités d'utilisations de notre méthode.



FIGURE 6.3: La chair et le gras, bien qu'interpolés par deux repères identiques, montrent des déformations différentes, dues aux différentes raideurs.

6.2.1 Application pour l'animation

La figure 6.3 montre une vue rapprochée de la simulation du steak (fig 4.1 page 58), qui s'exécute à une vitesse compatible avec des instruments haptiques. Le gras subit plus de déformation que la chair, car celui-ci est plus souple, même si ces deux parties sont interpolées par les mêmes repères. La méthode de [NKJF09a] simule aussi les matériaux hétérogènes de manière réaliste, mais un os rigide (grande raideur) au sein des différents éléments entraînerait des instabilités numériques, même en ayant recours à un maillage adapté. A l'inverse, notre méthode gère parfaitement les parties rigides, indépendamment de leur géométrie. La carte de raideur utilisée pour initialiser les fonctions de forme peut provenir d'images médicales ou être facilement créée à la main.

6.2.2 Application médicale

Notre méthode permet des simulations interactives de systèmes biologiques complexes tel que l'articulation du genou de la figure 6.4. Dans cette scène, nous ne disposons pas de carte de raideur pour ce modèle mais uniquement des maillages 3D de chaque partie anatomique. Quatre tissus différents sont simulés : des os, des muscles, du gras et des ligaments, auxquels sont associés des raideurs théoriques. A l'initialisation, les modèles sont utilisés pour générer les données volumiques. Chaque partie possède ainsi une raideur uniforme. L'influence des nœuds englobe plusieurs modèles et leurs noyaux encodent la variation de raideur.

Avec seulement 10 nœuds, nous sommes capable de simuler de façon réaliste la flexion et des mouvements fins tels que le déplacement de la rotule à 10 images par seconde, sans aucune connaissance a priori de la structure articulaire. Les forces sont transmises du quadriceps au tibia, suggérant que des modèles dynamique précis de l'anatomie, prenant en compte la réaction musculaire, pourraient être construits. Quelques modifications dans la voxélisation et les propriétés matérielles permettraient des discontinuités de mouvement entre les tissus en contact et une simulation plus précise des fibres biologiques (hautement anisotropiques et non linéaires).

6.2.3 Enrichissement de reconstruction de mouvements

Les deux scènes précédentes reflètent l'utilisation classique à laquelle notre modèle est destiné. A titre d'élargissement, nous présentons une démonstration dans laquelle notre

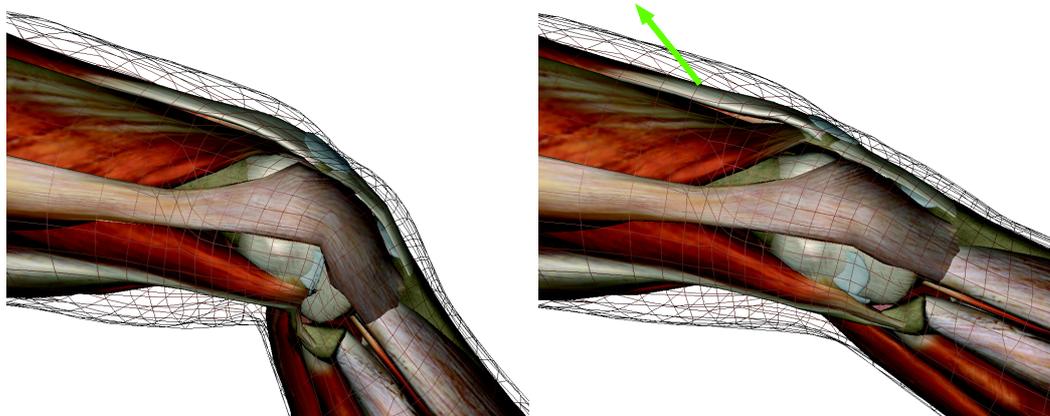


FIGURE 6.4: Simulation interactive du genou par 10 repères. Tirer sur le quadriceps déplace le tibia.

méthode est utilisée pour ajouter des mouvements secondaires pseudos physiques sur une animation de squelette issue de captures de mouvements (figure 6.5). La peau et le squelette complet d'un rat ont été acquis par micro-CT. Une capture de mouvement optique a été utilisée pour déterminer les mouvements des membres, de la tête et de trois os à l'arrière. Des nœuds sont placés sur les degrés de liberté de la capture de mouvement, dont l'animation est contrainte. Notre méthode est appliquée pour échantillonner automatiquement les tissus intermédiaires et placer des nœuds additionnels, permettant une déformation plus flexible du ventre et de la queue notamment. Une animation en temps réel, ainsi que l'interaction avec le modèle est consultable dans la vidéo jointe à [FGBP11] (<http://hal.inria.fr/docs/00/61/55/07/VIDEO/video.avi>).

6.3 Simulation de découpe

Ce chapitre présente les résultats de la méthode de découpe actuelle, les travaux prévus à court et moyen terme afin d'améliorer la technique de découpe et introduit un projet ambitieux à plus long terme : *My Corporis Fabrica*.

6.3.1 Performances

La scène de la figure 3.1 p. 48 comprend 96k voxels sur lesquels sont construits le modèle physique (519 éléments finis) et les maillages triangulaires (93k chacun). Lors des découpes, où toutes les étapes sont exécutées, le temps d'exécution est de 16 ips. La scène de la figure 3.7 p.56 contient 104 éléments finis, 13k triangles et s'exécute à 40 ips.

En moyenne, 60% du temps de calcul est passé dans la résolution des EDP, le calcul des forces des éléments finis et l'intégration des positions. La détection de collisions effectuée sur le GPU prend 15%, et la propagation dans les voxels 2%, la mise à jour des éléments finis 10% et la mise à jour du maillage triangulaire 2%.



FIGURE 6.5: *Mélange d'une animation et d'une simulation*

L'algorithme de détection de collisions utilisé dans ces tests et lors des tests de la méthode de simulation par repères est identique. Le temps d'exécution de la partie physique est bien plus important que pour les opérations de découpe. Il est dû au nombre important d'éléments finis générés, entraîné par l'échantillonnage régulier.

La vidéo associée à la publication [JBB⁺10] est consultable à l'adresse http://hal.inria.fr/docs/00/55/02/67/ANNEX/Breast_Cut_Omni_1.avi.

6.3.2 Intégration du modèle physique par repères

Notre modèle de simulation par repères permet l'utilisation d'un nombre extrêmement réduit de nœuds. Son intégration dans le pipeline de découpe permettrait sans doute d'obtenir de meilleurs résultats, en plus des autres avantages inhérents au modèle. Ce travail est en cours de réalisation. Certains aspects ont été validés par une publication, d'autres sont présentés ici en tant que travaux futurs pour la technique de découpe.

Nous prévoyons d'utiliser les modèles triangulaires visuels et de collision générés à partir des voxels. Lors d'une découpe, la suppression de voxels entraîne des changements topologiques sur les maillages triangulaires. Étant donné que les modèles physique, de collision et visuel sont indépendants et basés sur les voxels, la mise en relation des maillages triangulaires avec le modèle physique pour le transfert de forces et de positions demande peu de changements ; la majeure partie des modifications portent sur le modèle physique.

Adaptativité des nœuds

Travaux existants : L'insertion de nœuds au cours de la simulation est déjà implémentée et a été validée dans un autre contexte. Dans [GBFP11], nous présentons l'insertion d'un repère dans l'objet déformé afin d'obtenir des déformations locales, comme illustré sur la figure 6.6, ou d'améliorer la précision des calculs, en ajoutant des degrés de liberté (figure 6.7). Lors de la publication de cet article, nous employions des fonctions de forme basées sur les coordonnées harmoniques, pour matériaux homogènes.

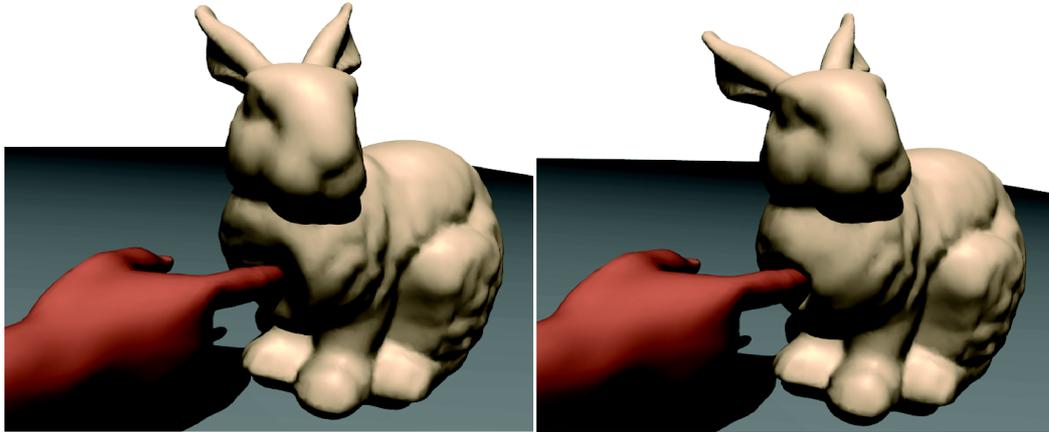


FIGURE 6.6: Déformations plus ou moins locales dues à l'utilisation de fonctions de forme différentes pour le nouveau repère inséré.

Un nouveau repère est inséré lors d'une collision (ou d'une interaction utilisateur). Afin d'accélérer les calculs, une nouvelle fonction de forme à influence décroissante suivant la distance géodésique est attribuée au nouveau repère. Elle est calculée dans la position non déformée, par propagation en une passe sur la grille de voxels jusqu'à atteindre une distance seuil. Dans cet article, la partition de l'unité des fonctions de formes est assurée par une renormalisation intrinsèque au champ de déplacement. Ainsi, les fonctions de formes associées aux autres repères restent inchangées. Le choix du seuil de distance associé au nouveau repère détermine la localité des déformations.

Cet ajout nécessite un nouveau bloc dans la matrice de masse ainsi que de nouveaux blocs dans la matrice de raideur. De plus, tous les blocs des matrices de masse et de raideur dont le domaine intersecte celui du nouveau repère inséré doivent être recalculés.

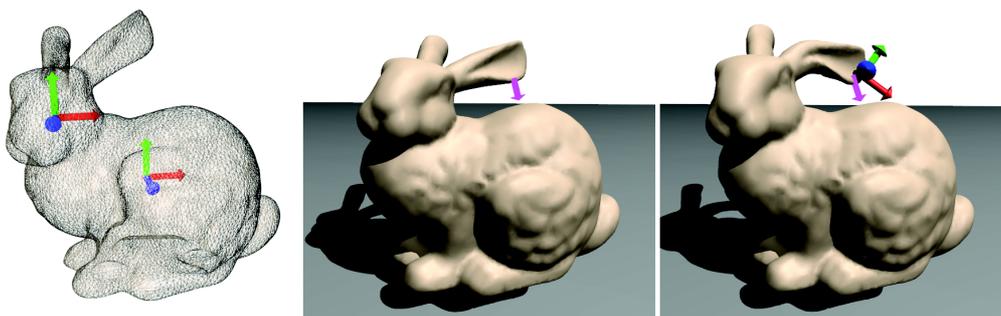


FIGURE 6.7: Adaptativité des repères. Réalisme physique obtenu avec deux repères (à Gauche). Contrôle intuitif de l'animation : la tête et l'oreille du lapin sont pratiquement rigides (Milieu), mais en insérant un repère additionnel (Droite), l'oreille devient flexible.

La suppression de nœuds est simple : il suffit de supprimer les données associées, dont la fonction de forme et mettre à jour les blocs des matrices de masses et de raideur.

Travaux futurs : Concernant la découpe, la suppression de matière implique l'ajout et la suppression de nœuds dans la simulation. Il doit être possible d'insérer des repères en cours de simulation. Par exemple dans le cas où certaines parties du maillage se trouvent déconnectées de tous repères, ou lorsqu'une partie rigide (ne possédant qu'un repère) est scindée en deux parties. De la même manière, lorsque l'intégralité de la matière associée à un repère est supprimée par exemple, il doit être supprimé de la simulation.

Aussi, l'échantillonnage peut être très faible avec ce modèle et dépend des propriétés de la matière. Afin d'obtenir le meilleur comportement possible avec un nombre limité de repère donné, il pourrait être préférable de le réévaluer en déplaçant certains repères lorsque la géométrie du modèle est changée. Néanmoins, cette opération demanderait de recalculer beaucoup de données associées au modèle et s'avèrerait sûrement trop coûteuse.

Le raffinement du comportement lorsque surviennent des collisions présenté précédemment est intéressant. Lors de nos tests, le seuil de distance d'un nouveau repère était arbitraire et il serait préférable d'automatiser son calcul en fonction de la géométrie et des matériaux des modèles en collision.

Ainsi, différentes stratégies de modification des nœuds s'offre à nous pour la découpe et nous évaluons actuellement les meilleurs critères d'insertion et de suppression dynamique.

Relation avec les modèles visuel et de collision

Les changements des modèles triangulaires suite à la suppression de voxels demandent de mettre à jour les mappings physique et visuel, employés pour transférer les forces du modèle de collision aux nœuds et les position des nœuds aux modèles triangulaires.

Les références des points supprimés sont enlevés des mappings. Celles des points nouvellement insérés sont calculées dans l'espace non déformé, de la même manière que lors de l'utilisation des éléments finis hiérarchiques.

L'implémentation des mappings sur GPU est prévu, ce qui devrait notablement accélérer la méthode.

Recalcul des fonctions de forme

Concernant les fonctions de forme, l'insertion ou le déplacement d'un nœud dans l'espace non déformé nécessite de (re)calculer une fonction de forme ainsi que de renormaliser celles des nœuds voisins ayant une influence commune avec celle du nœud pour conserver la partition de l'unité. Actuellement, le calcul des fonctions de forme est réalisé lors de l'initialisation et leur calcul relativement long est ainsi peu problématique. Leurs mises à jour pendant l'exécution nous demandera d'accélérer leur temps de calcul. La première amélioration à implémenter repose sur la parallélisation du calcul des gradients de poids sur GPU. Ce calcul équivaut à la résolution de l'équation d'Eikonal et plusieurs articles proposent actuellement des algorithmes efficaces, notamment le Fast-Marching [Set96, MAC04]. Les calculs pourraient être restreint à la zone d'influence de la fonction de forme plutôt qu'effectués sur l'ensemble des voxels.

De la même manière, la suppression de voxels entraîne un changement des poids. Mais la modification est plus locale dans ce cas. Il serait peut être plus avantageux d'appliquer

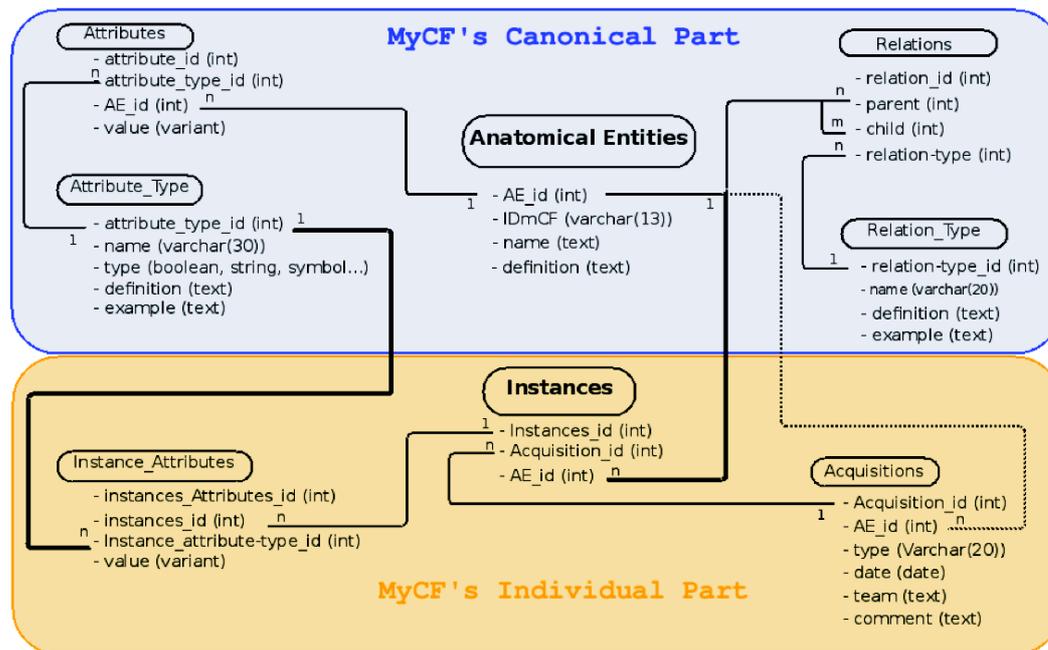


FIGURE 6.8: Organisation des données de My Corporis Fabrica

un algorithme de mise à jour des gradients par propagation locale à partir des voxels supprimés.

6.3.3 My Corporis Fabrica

L'automatisation des simulations spécifiques au patient est souhaitée dans le domaine médical. Nous avons présenté dans l'exemple 6.2.2 que la simulation d'un genou de manière automatisée est possible, alors que nous n'avons aucune connaissance a priori. Les seules données utilisées sont les cartes de densités issues d'acquisitions d'IRM. Nous avons assimilé celles-ci aux raideurs des matériaux, ce qui fournit un comportement correct mais inexact. Concernant le rendu, les modèles de l'exemple présenté ne sont pas spécifiques au patient. Pour ce faire, les données IRM sont généralement segmentées afin de créer des maillages correspondant au patient. Or, certaines structures n'apparaissent pas sur les IRM. Ainsi, il serait bon de disposer d'informations complémentaires a priori pour enrichir les simulations. Ce que propose, entre autres, de réaliser My Corporis Fabrica.

Ce projet collaboratif se concentre sur la création d'une ontologie anatomique, la description des relations entre chaque partie anatomique et leurs propriétés génériques. Son objectif est de constituer une base de connaissances pour les anatomistes et le développement de nouvelles technologies.

Au modèle générique, contenu dans une base de données s'ajoute la possibilité d'inscrire ou consulter des acquisitions anatomiques réelles, dont les données fournies (par exemple des propriétés physiques de raideur, densité, masse, etc) peuvent être complétées par le modèle théorique.

L'élaboration du modèle informatique permettant d'organiser les données et les re-

lations de manière générique a donné lieu à une publication [PBJ⁺09] dans laquelle le modèle est expliqué plus en détails. Pour en faire une rapide présentation, la base de données est scindée en deux parties (voir figure 6.8) : la partie canonique (en bleu) et celle spécifique aux acquisitions réelles (en jaune).

Le design de la base est simple mais permet une grande généralité. La partie canonique décrit le modèle anatomique théorique. Les entités anatomiques (table "Anatomical Entities") existantes y sont listées possédant des propriétés par défaut (tables "Attributes" et "Attributes Type") et mises en relation (tables "Relations" et "Relation Type").

Plus concrètement, "Anatomical Entities" contient les entités : "*os*", "*fémur*" et "*jambe*". Chaque entité possède des attributs par défaut donnant la valeur du type associé (position, masse, maillage, couleur...). Les types de relation permettent de définir des graphes dont les nœuds sont les entités. Par exemple, la relation de type "*is a*" indique que *fémur* est un *os*. La relation de type "part of" indique que *fémur* fait partie de la *jambe*.

La partie individuelle (en jaune) décrit des acquisitions réelles. Elle contient différentes acquisitions (table "Acquisition"), chacune comprenant des instances d'entités anatomiques (table "Instances") ainsi que leurs propriétés associées.

Concrètement, "Acquisitions" contient une entrée *acquisition IRM du genou*. Celle-ci inclut plusieurs instances *fémur*, *tibia*, etc, associées à l'entité anatomique du modèle théorique qu'ellesinstancient. Chaque instance définit ses attributs : *maillage*, *couleur*, *position*, etc ; surchargeant ainsi les propriétés par défaut du modèle théorique.

Néanmoins, si l'attribut *couleur* n'a pas été donné pour le *tibia*, et qu'il est nécessaire à l'exécution du programme exploitant cette acquisition, il peut être récupéré à partir du modèle théorique.

Conclusion

1 Récapitulatif des contributions

Dans cette thèse, nous avons présenté deux modèles interactifs de simulation, permettant la déformation et la découpe de solides à géométrie complexe.

Ainsi, dans le troisième chapitre, le modèle présenté permet la simulation interactive de scènes volumineuses par l'utilisation d'éléments finis hiérarchiques et des nœuds virtuels. Le corps simulé réagit aux contacts avec un objet tranchant avant qu'une découpe ne survienne dans le cas d'une pression trop importante, facilitant l'immersion et la manipulation de la scène virtuelle. Définir chacun des modèles d'après les voxels facilite leur découplage et la mise à jour des modèles triangulaires par *marching cubes* en espace non déformé. Le rendu se base sur les données de segmentation au lieu d'une texture en trois dimensions, dont les cliniciens ne disposent pas lors de simulations chirurgicales, facilitant les modélisations spécifiques aux patients. Néanmoins, d'autres modèles physiques se prêtent mieux aux changements topologiques et le modèle utilisé présente certains inconvénients et instabilités numériques.

De ce fait, un nouveau modèle physique meshless est présenté dans les deux chapitres suivants. La simulation par repères déformables présente un grand potentiel et un avantage par rapport aux éléments finis.

Concernant l'échantillonnage, les primitives peuvent être réparties de manière arbitraires et inégales dans le matériau, assurant un bon compromis entre vitesse et précision, sans compromettre la stabilité de cette technique. La possibilité de fixer un nombre arbitrairement faible de nœuds, combiné à la stratégie de distribution reposant sur les propriétés matérielles, permet la simulation rapide de modèles présentant des modes de déformation pertinents. L'extension des nœuds à des repères déformables permet la définition de fonctions de forme plus complexes, interpolant la déformation en ces points plutôt que le déplacement uniquement.

Ces nouvelles fonctions de forme, anisotropes, permettent d'encoder des cartes de raideurs détaillées grâce à la définition d'une distance de compliance reposant sur la souplesse du matériau. Elles peuvent être combinées avec la méthode de déformation classique du skinning pour simuler pseudo physiquement de grandes déformations sur des objets hétérogènes complexes, à l'aide d'un faible nombre de nœuds de contrôles et bénéficiant d'un temps d'exécution très rapide.

Contrairement aux éléments finis classiques et aux méthodes utilisant des fonctions de forme géométriques, notre approche découple la résolution matérielle de celle des fonctions de déplacement par l'encodage des propriétés matérielles dans les fonctions de forme et l'introduction des points d'intégration. Les pré-calculs de la masse et d'un réseau de ressorts généralisés contribuent aussi à l'accélération du temps de simulation.

L'adaptativité des primitives, dont les travaux initiaux sont présentés au chapitre six, permet de raffiner les résultats pendant l'animation de manière simple ; et à terme d'unifier les deux modèles.

2 Limitations

Les méthodes présentées comportent néanmoins quelques limitations :

- **Découpe** : la découpe nécessite la création d'un maillage interne de découpe, qui peut être généré automatiquement mais dont il serait préférable de s'abstraire. La suppression de matière génère des variations importantes des forces de contact entre deux pas de temps, qui restent faibles si la résolution de la grille de voxels est suffisamment élevée.
- **Ressorts** : l'approximation de la matrice de raideur en un réseau de ressorts 6D n'est stable qu'en petits déplacements.
- **Fonctions de forme** : leur définition par un scalaire rend le mélange linéaire rapide mais ne peut pas représenter correctement l'anisotropie du matériau. De plus, leur calcul est coûteux ; ce qui ne pose pas de problème car cette étape est actuellement réalisée pendant l'initialisation, mais une approche plus rapide devra être envisagée pour l'adaptativité des nœuds en cours de simulation.

3 Perspectives

A fin d'unifier notre modèle physique avec la méthode de découpe, le travail sur l'adaptativité expliqué en 6.3.2 p.91 est envisagé en premier. Les points importants sont les critères d'insertion et de suppression de repères, afin de raffiner localement les déformations et de prendre en compte les changements topologiques, et le calcul des poids en temps réel.

Concernant la méthode de découpe, il serait bien d'éviter la création d'un second maillage pour modéliser la pression lors de la découpe. La pression isostatique est l'opposé du tiers de la trace du tenseur de contrainte (sous forme matricielle) et il est possible de l'évaluer au centre d'un voxel dans notre modèle physique.

De nombreuses étapes actuelles (la calcul des poids, les calculs physiques et le déplacement des sommets des modèles visuels et de collision) sont matriciels et hautement parallélisables. Leur calcul sur GPU apporterait un gain de temps substantiel à notre modèle physique déjà rapide.

Bibliographie

Bibliographie

- [ACK99] SN Atluri, JY Cho, and H.G. Kim. Analysis of thin beams, using the meshless local petrov–galerkin method, with generalized moving least squares interpolations. *Computational Mechanics*, 24(5) :334–347, 1999.
- [AD08] Barbara André and Hervé Delingette. Versatile design of changing mesh topologies for surgery simulation. In *ISBMS*, pages 147–156, 2008.
- [AKJ08] Steven S. An, Theodore Kim, and Doug L. James. Optimizing cubature for efficient integration of subspace deformations. *ACM Trans. Graph.*, 27(5) :1–10, 2008.
- [Ale02] Marc Alexa. Linear combination of transformations. *ACM Trans. Graph.*, 21 :380–387, July 2002.
- [AOW⁺08] B. Adams, M. Ovsjanikov, M. Wand, H-P. Seidel, and L.J. Guibas. Meshless modeling of deformable shapes and their motion. In *Symposium on Computer Animation*, pages 77–86, 2008.
- [Bar84] Alan H. Barr. Global and local deformations of solid primitives. *SIGGRAPH Comput. Graph.*, 18 :21–30, January 1984.
- [Bat96] K. Bathe. *Finite Element Procedures*. Prentice Hall, 1996.
- [BB99] Ted Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5) :601–620, 1999.
- [Bey00] J. Bey. Simplicial grid refinement : on freudenthal’s algorithm and the optimal number of congruence classes. *Numerische Mathematik*, 85(1) :1–29, 2000.
- [BKO⁺96] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods : an overview and recent developments. *Computer methods in applied mechanics and engineering*, 139(1) :3–47, 1996.
- [BMG99] Daniel Bielser, Volker A. Maiwald, and Markus H. Gross. Interactive Cuts through 3-Dimensional Soft Tissue. In *Proceedings of the European Association for Computer Graphics 20th Annual Conference. Eurographics’99.*, volume 18/3, pages 31–38, September 1999.
- [BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. *SIGGRAPH Comput. Graph.*, 32 :106–117, 1998.

- [CDA00] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. A hybrid elastic model for real-time cutting, deformations and force feedback for surgery training and simulation. *The Visual Computer*, 16(7) :437–452, 2000.
- [CDC⁺96] S. Cotin, H. Delingette, J. M. Clement, J.M. Clément V. Tasseti, J. Marescaux, and N. Ayache. Volumetric deformable models for simulation of laparoscopic surgery, 1996.
- [DCA99] H. Delingette, S. Cotin, and N. Ayache. Efficient linear elastic models of soft tissues for real time surgery simulation. In *Medecine Meets Virtual Reality VII*, Interactive Technology and the New Paradigm for Healthcare, pages 139–151. IOS Press, January 1999.
- [DDCB01] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic real-time deformations using space and time adaptive sampling. In *28th Annual Conference on Computer Graphics, SIGGRAPH 2001, August, 2001*, pages 31–36, Los Angeles, California, Etats-Unis, 2001. ACM.
- [DMG05] J. Dequidt, D. Marchal, and L. Grisoni. Time-critical animation of deformable solids : Collision detection and deformable objects. *Comput. Animat. Virtual Worlds*, 16 :177–187, July 2005.
- [DO00] John Dingliana and Carol O’Sullivan. Graceful degradation of collision handling in physically based animation. *Computer Graphics Forum*, 19 :239–247, 2000.
- [FBAF08] F. Faure, S. Barbier, J. Allard, and F. Falipou. Image-based collision detection and response between arbitrary volumetric objects. In *ACM Siggraph/Eurographics Symposium on Computer Animation*, 2008.
- [FDA02] C. Forest, H. Delingette, and N. Ayache. Removing tetrahedra from a manifold mesh. In *Computer Animation, 2002. Proceedings of*, pages 225–229. IEEE, 2002.
- [FGBP11] François Faure, Benjamin Gilles, Guillaume Bousquet, and Dinesh Pai. Sparse meshless models of complex deformable solids. *ACM Trans. Graph.*, 30(4), July 2011. Special Issue : Proceedings of ACM SIGGRAPH 2011.
- [FM03] T-P. Fries and H.G. Matthies. Classification and overview of meshfree methods. Technical report, TU Brunswick, Germany, 2003.
- [GBFP11] Benjamin Gilles, Guillaume Bousquet, François Faure, and Dinesh Pai. Frame-based elastic models. *ACM Trans. Graph.*, 30(2), 2011.
- [GCMS00] F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno. A multiresolution model for soft objects supporting interactive cuts and lacerations. In *Computer Graphics Forum*, volume 19, pages 271–281. Wiley Online Library, 2000.
- [Gdo05] E.E. Gdoutos. *Fracture mechanics : an introduction*, volume 123. Kluwer Academic Pub, 2005.
- [GKS02] E. Grinspun, P. Krysl, and P. Schröder. Charms : A simple framework for adaptive simulation. *SIGGRAPH Comput. Graph.*, pages 281–290, 2002.
- [GP07] M. Gross and H. Pfister. *Point-Based Graphics*. Morgan Kaufmann, 2007.

- [Hej04] J. et al. Hejl. Hardware skinning with quaternions. In *Symposium on Interactive 3D Graphics and Games*, 2004.
- [HH00] T.J.R. Hughes and Hughes. *The finite element method : linear static and dynamic finite element analysis*. Dover Publications, 2000.
- [HJCW06] Min Hong, Sunhwa Jung, Min-Hyung Choi, and Samuel W. J. Welch. Fast volume preservation for a mass-spring system. *IEEE Comput. Graph. Appl.*, 26 :83–91, September 2006.
- [HWL08] D-P. Han, Q. Wei, and Z-X. Li. Kinematic control of free rigid bodies using dual quaternions. *International Journal of Automation and Computing*, 5(3) :319–324, 2008.
- [ITF06] G. Irving, J. Teran, and R. Fedkiw. Tetrahedral and hexahedral invertible finite elements. *Graphical models*, 68(2) :66–89, 2006.
- [JBB⁺10] Lenka Jeřábková, Guillaume Bousquet, Sébastien Barbier, François Faure, and Jérémie Allard. Volumetric modeling and interactive cutting of deformable bodies. *Prog. Biophys. Mol. Biol.*, 103(2-3) :217–224, December 2010. Special Issue on Biomechanical Modelling of Soft Tissue Motion.
- [JK09] Lenka Jeřábková and Torsten Kuhlen. Stable cutting of deformable objects in virtual environments using the XFEM. *IEEE Computer Graphics and Applications*, 29(2) :61–71, 2009.
- [JMD⁺07] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 26(3) :71, 2007.
- [KCZO07] Kavan, Collins, Zara, and O’Sullivan. Skinning with dual quaternions. In *Symposium on Interactive 3D graphics and games*, pages 39–46, 2007.
- [KCZO08] Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O’Sullivan. Geometric skinning with approximate dual quaternion blending. volume 27. ACM Press, 2008.
- [Kea05] Kavan and Zara et al. Spherical bend skinning : a real time deformation of articulated model. In *Symposium on Interactive 3D Graphics and Games*, 2005.
- [KMBG08] Peter Kaufmann, Sebastian Martin, Mario Botsch, and Markus Gross. Flexible simulation of deformable models using discontinuous galerkin fem. In *Symposium on Computer Animation*, 2008.
- [KZ03] Jan Klein and Gabriel Zachmann. Adb-trees : Controlling the error of time-critical collision detection. In *IN 8TH INTERNATIONAL FALL WORKSHOP VISION, MODELING, AND VISUALIZATION (VMV)*, pages 37–46. Springer, 2003.
- [Lev98] D. Levin. The approximation power of moving least-squares. *Mathematics of computation*, 67(224) :1517–1532, 1998.
- [LL96] S. Li and W.K. Liu. Moving least-square reproducing kernel method part ii : Fourier analysis. *Computer Methods in Applied Mechanics and Engineering*, 139(1) :159–193, 1996.

- [LLH⁺04] S. Li, H. Lu, W. Han, W.K. Liu, and D.C. Simkins. Reproducing kernel element method part ii : Globally conforming^m/_n hierarchies. *Computer Methods in Applied Mechanics and Engineering*, 193(12) :953–987, 2004.
- [LS81] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155) :141–158, 1981.
- [LWL⁺09] Yang Liu, Wenping Wang, Bruno Lévy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. On Centroidal Voronoi Tessellation Energy Smoothness and Fast Computation. *ACM Trans. Graph.*, 28, 08 2009.
- [MAC04] D. Marchal, F. Aubert, and C. Chaillou. Collision between deformable objects using fast-marching on tetrahedral models. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 121–129. Eurographics Association, 2004.
- [MBF04] Neil Molino, Zhaosheng Bao, and Ronald Fedkiw. A virtual node algorithm for changing mesh topology during simulation. *ACM Transactions on Graphics*, 23(3) :385–392, August 2004. Special Issue : Proceedings of the 2004 SIGGRAPH Conference.
- [MC97] P. Meseure and C. Chaillou. Deformable body simulation with adaptive subdivision and cuttings. In *Proceedings of the WSCG*, volume 97, pages 361–370. Citeseer, 1997.
- [MHTG05] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Transactions on Graphics*, 24(3) :471–478, 2005.
- [MK00] Andrew B. Mor and Takeo Kanade. Modifying Soft Tissue Models : Progressive Cutting with Minimal New Element Creation. In *Proceedings of Medical Image Computing & Computer Assisted Intervention*, pages 598–607, 2000.
- [MKB⁺08] Sebastian Martin, Peter Kaufmann, Mario Botsch, Martin Wicke, and Markus Gross. Polyhedral finite elements using harmonic basis functions. *Comput. Graph. Forum*, 27(5), 2008.
- [MKB⁺10] S. Martin, P. Kaufmann, M. Botsch, E. Grinspun, and M. Gross. Unified simulation of elastic rods, shells, and solids. *SIGGRAPH Comput. Graph.*, 29(3), 2010.
- [MKN⁺04] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *Symposium on Computer Animation*, pages 141–151, 2004.
- [MMDJ01] M. Müller, L. McMillan, J. Dorsey, and R. Jagnow. Real-time simulation of deformation and fracture of stiff materials. *Computer Animation and Simulation 2001*, pages 113–124, 2001.
- [MO05] C. MENDOZA and C.A.N.N. O’SULLIVAN. Towards time-critical collision detection for deformable objects based on reduced models. 2005.
- [MTLT88] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint dependent local deformations for hand animation and object grasping. In *Graphics interface*, pages 26–33, 1988.

- [Nea04] A. Nealen. An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. URL : <http://www.nealen.com/projects>, 2004.
- [Nie03] H.W. Nienhuys. Cutting in deformable objects. *Institute for Information and Computing Sciences, Utrecht University*, 2003.
- [NKJF09a] M. Nesme, P. Kry, L. Jerabkova, and F. Faure. Preserving Topology and Elasticity for Embedded Deformable Models. *SIGGRAPH Comput. Graph.*, 2009.
- [NKJF09b] Matthieu Nesme, Paul Kry, Lenka Jerabkova, and François Faure. Preserving Topology and Elasticity for Embedded Deformable Models. *ACM Transaction on Graphics (proc. of SIGGRAPH 2009)*, 2009.
- [NMK⁺05] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. In *Comput. Graph. Forum*, volume 25 (4), pages 809–836, 2005.
- [NPF06] Matthieu Nesme, Yohan Payan, and François Faure. Animating shapes at arbitrary resolution with non-uniform stiffness. In *3rd Workshop in Virtual Reality Interaction and Physical Simulation, VRIPHYS '06*, Madrid, Spain, November 2006.
- [NvdS01] Han-Wen Nienhuys and A. Frank van der Stappen. Supporting cuts and finite element deformation in interactive surgery simulation. Technical report, University of Utrecht, 2001.
- [OH99] J. O'Brien and J. Hodgins. Graphical models and animation of brittle fracture. *SIGGRAPH Comput. Graph.*, pages 137–146, 1999.
- [PBJ⁺09] Olivier Palombi, Guillaume Bousquet, David Jospin, Sahar Hassan, Lionel Revéret, and François Faure. My corporis fabrica : a unified ontological, geometrical and mechanical view of human anatomy. In *2nd workshop on 3D Physiological Human, 3DPH2009, December, 2009*, volume 5903 of *Lecture Notes in Computer Science*, pages 209–219, Zermatt, Suisse, December 2009. Springer.
- [PKA⁺05] Mark Pauly, Richard Keiser, Bart Adams, Philip Dutre, Markus Gross, and Leonidas J. Guibas. Meshless animation of fracturing solids. In *Proceedings of Siggraph*, volume 24, pages 957–964, New York, NY, USA, 2005. ACM Press.
- [Pow90] M. J. D. Powell. The theory of radial basis function approximation. *University Numerical Analysis Report*, 1990.
- [RI97] M.C. RIVARA and P. Inostroza. Using longest-side bisection techniques for the automatic refinement of delaunay triangulations. *International journal for numerical methods in Engineering*, 40(4) :581–597, 1997.
- [SDF07] Eftychios Sifakis, Kevin G. Der, and Ronald Fedkiw. Arbitrary cutting of deformable tetrahedralized objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '07*, pages 73–80, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

- [Set96] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4) :1591, 1996.
- [Sho85] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '85, pages 245–254, New York, NY, USA, 1985. ACM.
- [SMMB00] N. Sukumar, N. Moës, B. Moran, and T. Belytschko. Extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering*, 48(11) :1549–1570, 2000.
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In Maureen C. Stone, editor, *SIGGRAPH Comput. Graph.*, volume 21, pages 205–214, July 1987.
- [TWAD09] Jane Tournois, Camille Wormser, Pierre Alliez, and Mathieu Desbrun. Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. *ACM Trans. Graph.*, 28 :75 :1–75 :9, July 2009.
- [WBG07] Martin Wicke, Mario Botsch, and Markus Gross. A Finite Element Method on Convex Polyhedra. In *Proceedings of Eurographics '07*, 2007.
- [WDGT01] Xunlei Wu, Michael S. Downes, Tolga Goktekin, and Frank Tendick. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Computer Graphics Forum*, pages 349–358, 2001.
- [WSG05] M. Wicke, D. Steinemann, and M. Gross. Efficient animation of point-sampled thin shells. In *Computer Graphics Forum*, volume 24, pages 667–676. Wiley Online Library, 2005.

Annexes

Formules utiles au mélange par quaternions duaux

$$\mathbf{Q}(\bar{\mathbf{p}}, \mathbf{b}') = \frac{\partial \mathbf{P}}{\partial \mathbf{b}'} = 2 [\mathbf{Q}_1 \ \mathbf{Q}_2] \quad , \quad \mathbf{Q}_2 = \begin{bmatrix} w_0 & -c_0 & b_0 & -a_0 \\ c_0 & w_0 & -a_0 & -b_0 \\ -b_0 & a_0 & w_0 & -c_0 \end{bmatrix} \quad (1)$$

$$\mathbf{Q}_1 = \begin{bmatrix} -w_\varepsilon + a_0x + b_0y + c_0z & c_\varepsilon - b_0x + a_0y + w_0z & -b_\varepsilon - c_0x - w_0y + a_0z & a_\varepsilon + w_0x - c_0y + b_0z \\ -c_\varepsilon + b_0x - a_0y - w_0z & -w_\varepsilon + a_0x + b_0y + c_0z & a_\varepsilon + w_0x - c_0y + b_0z & b_\varepsilon + c_0x + w_0y - a_0z \\ b_\varepsilon + c_0x + w_0y - a_0z & -a_\varepsilon - w_0x + c_0y - b_0z & -w_\varepsilon + a_0x + b_0y + c_0z & c_\varepsilon - b_0x + a_0y + w_0z \end{bmatrix}$$

$$\Delta \mathbf{R}(\mathbf{V}) = 2 \begin{bmatrix} -2(b_0v_1 + c_0v_2) & -c_0v_3 + b_0v_0 + a_0v_1 - w_0v_2 & b_0v_3 + c_0v_0 + w_0v_1 + a_0v_2 \\ c_0v_3 + b_0v_0 + a_0v_1 + w_0v_2 & -2(a_0v_0 + c_0v_2) & -a_0v_3 - w_0v_0 + c_0v_1 + b_0v_2 \\ -b_0v_3 + c_0v_0 - w_0v_1 + a_0v_2 & a_0v_3 + w_0v_0 + c_0v_1 + b_0v_2 & -2(a_0v_0 + b_0v_1) \end{bmatrix} \quad (2)$$

$$\Delta \mathbf{Q}(\mathbf{V}) = 2 [\Delta \mathbf{Q}_1 \ \Delta \mathbf{Q}_2] \quad , \quad \Delta \mathbf{Q}_2 = \begin{bmatrix} v_3 & -v_2 & v_1 & -v_0 \\ v_2 & v_3 & -v_0 & -v_1 \\ -v_1 & v_0 & v_3 & -v_2 \end{bmatrix} \quad (3)$$

$$\Delta \mathbf{Q}_1 = \begin{bmatrix} xv_0 + yv_1 + zv_2 - v_7 & zv_3 + yv_0 - xv_1 + v_6 & -yv_3 + zv_0 - xv_2 - v_5 & xv_3 + zv_1 - yv_2 + v_4 \\ -zv_3 - yv_0 + xv_1 - v_6 & xv_0 + yv_1 + zv_2 - v_7 & xv_3 + zv_1 - yv_2 + v_4 & yv_3 - zv_0 + xv_2 + v_5 \\ yv_3 - zv_0 + xv_2 + v_5 & -xv_3 - zv_1 + yv_2 - v_4 & xv_0 + yv_1 + zv_2 - v_7 & zv_3 + yv_0 - xv_1 + v_6 \end{bmatrix}$$

Quaternion dual correspondant à la position actuelle d'un repère rigide :

$$\mathbf{q}_i = \begin{bmatrix} \mathbf{q}_0^i \\ \mathbf{q}_\varepsilon^i \end{bmatrix} = [a_0^i \ b_0^i \ c_0^i \ w_0^i \ a_\varepsilon^i \ b_\varepsilon^i \ c_\varepsilon^i \ w_\varepsilon^i]^T$$

où \mathbf{q}_0^i est la partie réelle représentant la rotation et $\mathbf{q}_\varepsilon^{iT} = \frac{1}{2}(\mathbf{t}_{i\wedge} \circ \mathbf{q}_0^i)$ la partie duale encodant la translation à l'origine du repère $\mathbf{t}_i = [t_x^i \ t_y^i \ t_z^i]^T$ et $\mathbf{t}_{i\wedge} = [t_x^i \ t_y^i \ t_z^i \ 0]^T$.

Transformations relatives à la configuration au repos :

$$\mathbf{q}_i \circ \bar{\mathbf{q}}_i^{-1} = \bar{\mathbf{T}}_i \mathbf{q}_i \quad , \quad \bar{\mathbf{T}}_i = \begin{bmatrix} H^-[(\bar{\mathbf{q}}_0^i)^{-1}] & \mathbf{0} \\ H^-[(\bar{\mathbf{q}}_\varepsilon^i)^{-1}] & H^-[(\bar{\mathbf{q}}_0^i)^{-1}] \end{bmatrix}$$

$$H^-(\mathbf{q}) = \begin{bmatrix} w & c & -b & a \\ -c & w & a & b \\ b & -a & w & c \\ -a & -b & -c & w \end{bmatrix} \quad , \quad \mathbf{q} = \begin{bmatrix} a \\ b \\ c \\ w \end{bmatrix}$$

où H^- est l'opérateur de Hamilton qui encode la multiplication à droite d'un quaternion.

Le mélange linéaire de quaternions duaux est :

$$\mathbf{b} = \sum w_i \bar{\mathbf{T}}_i \mathbf{q}_i = \begin{bmatrix} \mathbf{Q}_0 \\ \mathbf{Q}_\varepsilon \end{bmatrix}$$

Normalisation de quaternion dual :

$$\begin{aligned} \mathbf{b}' &= \frac{\mathbf{b}}{\|\mathbf{b}\|} = \frac{1}{\|\mathbf{Q}_0\|} \begin{bmatrix} \mathbf{Q}_0 \\ \mathbf{Q}_\varepsilon \end{bmatrix} - \frac{\mathbf{Q}_\varepsilon \cdot \mathbf{Q}_0}{\|\mathbf{Q}_0\|^3} \begin{bmatrix} \mathbf{0} \\ \mathbf{Q}_0 \end{bmatrix} \\ &= [\mathbf{q}_0^T \mathbf{q}_\varepsilon^T]^T = [a_0 \ b_0 \ c_0 \ w_0 \ a_\varepsilon \ b_\varepsilon \ c_\varepsilon \ w_\varepsilon]^T \end{aligned}$$

L'extraction de la translation et de la rotation du quaternion dual \mathbf{b}' suite au mélange se fait par [KCZO08] :

$$\begin{aligned} \mathbf{R}(\mathbf{b}') &= \begin{bmatrix} 1 - 2b_0^2 - 2c_0^2 & 2a_0b_0 - 2w_0c_0 & 2a_0c_0 + 2w_0b_0 \\ 2a_0b_0 + 2w_0c_0 & 1 - 2a_0^2 - 2c_0^2 & 2b_0c_0 - 2w_0a_0 \\ 2a_0c_0 - 2w_0b_0 & 2b_0c_0 + 2w_0a_0 & 1 - 2a_0^2 - 2b_0^2 \end{bmatrix} \\ \text{et } \mathbf{t}(\mathbf{b}') &= 2 \begin{bmatrix} -w_\varepsilon a_0 + a_\varepsilon w_0 - b_\varepsilon c_0 + c_\varepsilon b_0 \\ -w_\varepsilon b_0 + a_\varepsilon c_0 + b_\varepsilon w_0 - c_\varepsilon a_0 \\ -w_\varepsilon c_0 - a_\varepsilon b_0 + b_\varepsilon a_0 + c_\varepsilon w_0 \end{bmatrix} \end{aligned}$$

Il y a une relation linéaire entre le vecteur vitesse et la dérivée temporelle du repère : $\dot{\mathbf{q}}_i = \frac{1}{2}(\hat{\omega}_i \circ \mathbf{q}_i)$ [HWL08]. Elle peut être représentée par une matrice \mathbf{L}_i de dimension 8×6 tel que $\dot{\mathbf{q}}_i = \mathbf{L}_i \dot{q}_i$. Ce qui donne :

$$\begin{aligned} \mathbf{L}_i(\mathbf{q}_i) &= \nabla_i \mathbf{q}_i = \frac{1}{2} \begin{bmatrix} \mathbf{L}_{i0} & \mathbf{0} \\ \mathbf{L}_{i\varepsilon} & \mathbf{L}_{i0} \end{bmatrix}, \quad \mathbf{L}_{i0} = \begin{bmatrix} w_0^i & c_0^i & -b_0^i \\ -c_0^i & w_0^i & a_0^i \\ b_0^i & -a_0^i & w_0^i \\ -a_0^i & -b_0^i & -c_0^i \end{bmatrix} \\ \mathbf{L}_{i\varepsilon} &= \begin{bmatrix} w_\varepsilon^i + c_0^i t_z^i + b_0^i t_y^i & c_\varepsilon^i - w_0^i t_z^i - b_0^i t_x^i & -b_\varepsilon^i + w_0^i t_y^i - c_0^i t_x^i \\ -c_\varepsilon^i + w_0^i t_z^i - a_0^i t_y^i & w_\varepsilon^i + c_0^i t_z^i + a_0^i t_x^i & a_\varepsilon^i - w_0^i t_x^i - c_0^i t_y^i \\ b_\varepsilon^i - w_0^i t_y^i - a_0^i t_z^i & -a_\varepsilon^i + w_0^i t_x^i - b_0^i t_z^i & w_\varepsilon^i + b_0^i t_y^i + a_0^i t_z^i \\ -a_\varepsilon^i - b_0^i t_z^i + c_0^i t_y^i & -b_\varepsilon^i + a_0^i t_z^i - c_0^i t_x^i & -c_\varepsilon^i - a_0^i t_y^i + b_0^i t_x^i \end{bmatrix} \end{aligned}$$

Pour calculer la Jacobienne (équation 5.8) et le gradient de déformation (équation 5.9), il est nécessaire de calculer les dérivés premières \mathbf{Q} , \mathbf{N} et \mathbf{W} qui sont données par (voir aussi l'équation longue 1) :

$$\begin{aligned} \mathbf{N}(\mathbf{b}, \mathbf{b}') &= \frac{\partial \mathbf{b}'}{\partial \mathbf{b}} = \begin{bmatrix} \mathbf{N}_0 & \mathbf{0} \\ \mathbf{N}_\varepsilon & \mathbf{N}_0 \end{bmatrix} \\ \text{avec } \mathbf{N}_0 &= -\frac{1}{\|\mathbf{Q}_0\|} (\mathbf{q}_0 \mathbf{q}_0^T - \mathbf{I}) \\ \text{et } \mathbf{N}_\varepsilon &= -\frac{1}{\|\mathbf{Q}_0\|} (\mathbf{q}_0 \mathbf{q}_\varepsilon^T + \mathbf{q}_\varepsilon \mathbf{q}_0^T) - \frac{\mathbf{Q}_\varepsilon \cdot \mathbf{Q}_0}{\|\mathbf{Q}_0\|^3} (\mathbf{I} - \mathbf{q}_0 \mathbf{q}_0^T) \\ \mathbf{W}(\bar{\mathbf{p}}) &= \frac{\partial \mathbf{b}}{\partial \bar{\mathbf{p}}} = \sum \bar{\mathbf{T}}_i \mathbf{q}_i \mathbf{W}_i \quad \text{avec} \quad \mathbf{W}_i = \frac{\partial w_i}{\partial \bar{\mathbf{p}}} \end{aligned}$$

\mathbf{W}_i représente les gradients de poids et sont numériquement calculés sur la grille de voxel par différences centrales.

La dérivée du gradient de déformation est nécessaire au calcul de la dérivée de l'effort (équation 5.14). En appliquant les lois de dérivations et la convention d'ajout d'Einstein, le tenseur au troisième ordre s'écrit :

$$\begin{aligned} \nabla_i \mathbf{F}_{abc} &= w_i \left[\frac{\partial \mathbf{R}}{\partial \mathbf{b}'} \right]_{abd} \mathbf{N}_{de} \bar{\mathbf{T}}_i \mathbf{e}_f \mathbf{L}_i \mathbf{f}_c + \mathbf{Q}_{ad} \mathbf{N}_{de} \left[\frac{\partial \mathbf{W}}{\partial \mathbf{q}_i} \right]_{ebf} \mathbf{L}_i \mathbf{f}_c \\ &+ w_i \left[\frac{\partial \mathbf{Q}}{\partial \mathbf{b}'} \right]_{ade} \mathbf{N}_{ef} \bar{\mathbf{T}}_i \mathbf{f}_g \mathbf{L}_i \mathbf{g}_c \mathbf{N}_{dh} \mathbf{W}_{hb} + w_i \mathbf{Q}_{ad} \left[\frac{\partial \mathbf{N}}{\partial \mathbf{b}} \right]_{def} \bar{\mathbf{T}}_i \mathbf{f}_g \mathbf{L}_i \mathbf{g}_c \mathbf{W}_{eb} \end{aligned}$$

Les notations peuvent être simplifiées en exprimant chaque sous matrice $3 \times 3 \nabla_i \mathbf{F}_k = \partial \mathbf{F} / \partial \dot{q}_{ik}$ en utilisant les opérateurs $\Delta \mathbf{R}$, $\Delta \mathbf{W}$, $\Delta \mathbf{Q}$ et $\Delta \mathbf{N}$:

$$\begin{aligned} \nabla_i \mathbf{F}_k &= w_i \Delta \mathbf{R}([\mathbf{N} \bar{\mathbf{T}}_i \mathbf{L}_i]^k) + \mathbf{Q} \mathbf{N} \bar{\mathbf{T}}_i \Delta \mathbf{W}([\mathbf{L}_i]^k) \\ &+ w_i \Delta \mathbf{Q}([\mathbf{N} \bar{\mathbf{T}}_i \mathbf{L}_i]^k) \mathbf{N} \mathbf{W} + w_i \mathbf{Q} \Delta \mathbf{N}([\bar{\mathbf{T}}_i \mathbf{L}_i]^k) \mathbf{W} \end{aligned}$$

Chaque opérateur produit une matrice à partir d'un vecteur colonne 8×1

$$\mathbf{V} = [\mathbf{V}_0^T \ \mathbf{V}_\varepsilon^T]^T = [v_0 \ v_1 \ v_2 \ v_3 \ v_4 \ v_5 \ v_6 \ v_7]^T$$

(par exemple la k^{ime} colonne de $\mathbf{N} \bar{\mathbf{T}}_i \mathbf{L}_i$). Ils sont obtenus par différentiation de \mathbf{R} , \mathbf{W} , \mathbf{Q} et \mathbf{N} (voir aussi les équations longues 2 et 3) :

$$\Delta \mathbf{W}(\mathbf{V}) = \mathbf{V} \mathbf{W}_i$$

$$\Delta \mathbf{N}(\mathbf{V}) = \begin{bmatrix} \Delta \mathbf{N}_0 & \mathbf{0} \\ \Delta \mathbf{N}_\varepsilon + \Delta \mathbf{N}_c & \Delta \mathbf{N}_0 \end{bmatrix} \quad \text{with :}$$

$$\Delta \mathbf{N}_0 = -\frac{1}{\|\mathbf{Q}_0\|^2} [\mathbf{q}_0 \mathbf{V}_0^T + \mathbf{V}_0 \mathbf{q}_0^T + \mathbf{V}_0 \cdot \mathbf{q}_0 (\mathbf{I} - 3\mathbf{q}_0 \mathbf{q}_0^T)]$$

$$\Delta \mathbf{N}_\varepsilon = -\frac{1}{\|\mathbf{Q}_0\|^2} [\mathbf{q}_0 \mathbf{V}_\varepsilon^T + \mathbf{V}_\varepsilon \mathbf{q}_0^T + \mathbf{V}_\varepsilon \cdot \mathbf{q}_0 (\mathbf{I} - 3\mathbf{q}_0 \mathbf{q}_0^T)]$$

$$\begin{aligned} \Delta \mathbf{N}_c &= -\frac{1}{\|\mathbf{Q}_0\|^2} [\mathbf{q}_\varepsilon \mathbf{V}_0^T + \mathbf{V}_0 \mathbf{q}_\varepsilon^T + \mathbf{V}_0 \cdot \mathbf{q}_\varepsilon (\mathbf{I} - 3\mathbf{q}_0 \mathbf{q}_0^T) \\ &\quad - 3\mathbf{V}_0 \cdot \mathbf{q}_0 (\mathbf{q}_\varepsilon \mathbf{q}_0^T + \mathbf{q}_0 \mathbf{q}_\varepsilon^T)] - 2 \frac{\mathbf{Q}_\varepsilon \cdot \mathbf{Q}_0}{\|\mathbf{Q}_0\|^2} \Delta \mathbf{N}_0 \end{aligned}$$

Table des figures

1.1	Champ de déplacement	9
1.2	Déformations linéaires	10
1.3	Les tenseurs des déformations de Cauchy et Green-Lagrange	11
1.4	Contrainte en réponse à une force	12
1.5	Loi de comportement plastique	13
1.6	Le module d'Young et le coefficient de Poisson	15
1.7	Géométrie et propriétés du matériau	15
2.1	Inversion de tétraèdre	21
2.2	Modélisation par éléments finis	23
2.3	Courbes d'interpolation linéaire et quadratique	23
2.4	Simulation par cages de déformation	24
2.5	Raffinement des fonctions de bases	25
2.6	Technique meshless basée sur les déplacements	27
2.7	Positionnement d'une surface par skinning.	29
2.8	Espaces des rotations	30
2.9	Comparaison des mélanges de transformations	31
2.10	Fonctions de forme harmoniques	33
2.11	Les moindres carrés	36
2.12	Découpe de [MKN ⁺ 04]	40
2.13	Modification du noyau suivant la visibilité [PKA ⁺ 05]	41
3.1	Aperçu de notre méthode de découpe	48
3.2	Éléments finis étendus et hiérarchiques	50
3.3	Layer Depth Image (LDI)	52
3.4	Propagation sur les voxels	53
3.5	Mise à jour du maillage triangulaire	54
3.6	Modélisation de la pression	55
3.7	Simulation de foie	56
4.1	Exemple du steak	58
4.2	Modes de déformations (dont skinning et GMLS)	61
4.3	Comparaison des distances	63

4.4	Modélisation des poids par l'utilisateur	64
4.5	Distance de compliance	66
4.6	Comparaison de fonctions de forme	67
4.7	Fonctions de forme et subdivision de Voronoï	69
4.8	Distribution des nœuds	70
5.1	Un objet déformé par skinning à l'aide de deux repères.	73
5.2	Échantillons et intégration spatiale	76
5.3	Erreur d'approximation des points d'intégrations	77
5.4	Déplacement et déformation	78
5.5	Ressorts 6D	83
6.1	Comparaison avec les éléments finis	86
6.2	Comparaison des fonctions de formes relatives au matériau	87
6.3	Déformations d'un matériau à propriétés hétérogènes	89
6.4	Simulation de l'articulation du genou	90
6.5	Mélange d'une animation et d'une simulation	91
6.6	Adaptativité : déformations locales	92
6.7	Adaptativité : ajout de degré de liberté	92
6.8	Organisation des données de My Corporis Fabrica	94

Table des matières

Introduction	1
<hr/>	
Contexte scientifique	7
<hr/>	
1 Introduction à la théorie des milieux continus	7
1.1 Discrétisation et degrés de liberté	8
1.2 Vue d'ensemble	8
1.3 Déplacements	9
1.4 Tenseur des déformations	10
1.4.1 Le tenseur de Cauchy	11
1.4.2 Le tenseur de Green-Lagrange	11
1.5 Tenseur des contraintes	12
1.6 Loi de comportement	12
1.6.1 La loi élastique	13
1.6.2 La loi de Hooke	13
1.7 Matrice de rigidité	15
1.7.1 Formulation linéaire	15
1.7.2 Formulation non linéaire	16
1.8 Matrice de masse	16
1.9 Intégration dynamique	16
1.10 Conclusion	18
2 Modèles déformables discrets	19
2.1 Systèmes masses-ressorts	20
2.1.1 Force de rappel	20
2.1.2 Force d'amortissement	20
2.1.3 Force extérieure	21
2.1.4 Discussion	21
2.2 Éléments finis	22
2.2.1 Principe	22
2.2.2 Maillage	22
2.2.3 Fonctions d'interpolation	22

2.2.4	Cages de déformation	23
2.2.5	Multirésolution	24
2.2.6	Adaptativité	25
2.2.7	Conclusion	26
2.3	Méthodes particulières (alias Meshless)	26
2.3.1	Principe	26
2.3.2	Techniques basées sur les déplacements	26
2.3.3	Techniques basées sur l'énergie de déformation	27
2.3.4	Conclusion	28
2.4	Enrobage (alias Skinning)	28
2.4.1	Principe	29
2.4.2	La pondération	30
2.4.3	Interpolation de transformations rigides	30
2.4.4	Conclusion	32
2.5	Les fonctions de forme	32
2.5.1	Interpolation dans un élément	32
2.5.2	Interpolation sans élément	34
2.6	Simulation de découpe	38
2.6.1	Modèles physiques maillés	38
2.6.2	Modèles meshless	40
2.7	Conclusion	41
Déformation par repères déformables		45
3	Simulation de découpe	47
3.1	Présentation de la méthode	48
3.2	Modèles physique, visuel et de collision	49
3.2.1	Topologies des modèles	49
3.2.2	Modèle physique	50
3.2.3	Modèles triangulaires	51
3.3	Processus de découpe	53
3.3.1	Rastérisation de la scène	53
3.3.2	Propagation dans les hexaèdres	53
3.3.3	Mise à jour du maillage triangulaire	54
3.4	Réponse au contact	54
3.5	Rendu	55
4	Fonctions de forme relatives aux matériaux et Echantillonnage	57
4.1	Présentation du modèle	58
4.1.1	Rappel lexical	59
4.1.2	Choix d'échantillonnage et du modèle interpolant	59
4.2	Fonctions de forme relatives aux matériaux	62

4.2.1	Première approche : les matériaux homogènes	62
4.2.2	Fonctions de forme relatives au matériau	64
4.3	Distribution des nœuds	69
5	Dynamique par repères déformables	71
5.1	Algorithme de simulation	72
5.2	Champ de déplacement	73
5.3	Intégration temporelle & spatiale	74
5.3.1	Équations différentielles	74
5.3.2	Intégration spatiale	75
5.4	Masse	77
5.4.1	Pré-calcul	78
5.5	Déformations & Forces internes	78
5.5.1	Déformation	78
5.5.2	Énergie de déformation et effort	79
5.5.3	Forces internes et matrice de raideur	79
5.5.4	Optimisation	80
5.5.5	Pré-calculs	82
5.6	Surfaces de contacts et de rendu	83
5.6.1	Report des forces externes	83
5.6.2	Mise à jour du modèle visuel	84
6	Résultats et autres travaux	85
6.1	Modèle physique par repères	86
6.1.1	Validation	86
6.1.2	Performances	86
6.1.3	Points d'intégration	88
6.2	Exemples de simulations	88
6.2.1	Application pour l'animation	89
6.2.2	Application médicale	89
6.2.3	Enrichissement de reconstruction de mouvements	89
6.3	Simulation de découpe	90
6.3.1	Performances	90
6.3.2	Intégration du modèle physique par repères	91
6.3.3	My Corporis Fabrica	94
	Conclusion	99
1	Récapitulatif des contributions	99
2	Limitations	100
3	Perspectives	100

Bibliographie	103
Annexes	111
Formules utiles au mélange par quaternions duaux	111
Tables détaillées	116
Table des figures	116
Table des matières	117