

Titre: Analyse des méthodes par éléments finis et méthodes sans
maillage pour la déformation de corps mous en simulation
chirurgicale
Title:

Auteur: Jean-Nicolas Brunet
Author:

Date: 2017

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Brunet, J.-N. (2017). Analyse des méthodes par éléments finis et méthodes sans
maillage pour la déformation de corps mous en simulation chirurgicale [Master's
thesis, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/2529/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2529/>
PolyPublie URL:

**Directeurs de
recherche:** Benoît Ozell
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

ANALYSE DES MÉTHODES PAR ÉLÉMENTS FINIS ET MÉTHODES SANS
MAILLAGE POUR LA DÉFORMATION DE CORPS MOUS EN SIMULATION
CHIRURGICALE

JEAN-NICOLAS BRUNET
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLOME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
MAI 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ANALYSE DES MÉTHODES PAR ÉLÉMENTS FINIS ET MÉTHODES SANS
MAILLAGE POUR LA DÉFORMATION DE CORPS MOUS EN SIMULATION
CHIRURGICALE

présenté par : BRUNET Jean-Nicolas

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. KADOURY Samuel, Ph. D., président

M. OZELL Benoît, Ph. D., membre et directeur de recherche

M. REGGIO Marcelo, Ph. D., membre

DÉDICACE

À mes parents...

REMERCIEMENTS

J'aimerais d'abord remercier mon père pour l'aide exceptionnelle qu'il m'a donnée lors de la rédaction de ce long mémoire. Merci également à ma mère pour son apport constant en chocolat. Merci à mon directeur de recherche pour ses sages conseils et sa confiance. Merci à toute l'équipe Mimesis à Strasbourg de m'avoir accueilli pendant 6 merveilleux mois. Merci à Aurélie pour son appui et sa patience lors des dernières interminables semaines de rédaction. Merci aux professeurs Samuel Kadoury et Marcelo Reggio d'avoir accepté d'être membres de mon jury. Merci à OSSimTech, Mitacs-Globalink et l'INRIA pour leur confiance et soutien financier. Finalement, merci à tous mes amis et proches qui ont su m'encourager et me soutenir dans les moments les plus difficiles de cette recherche et sans qui je n'aurais pas pu me rendre jusqu'ici.

RÉSUMÉ

Ce projet de recherche s'intéresse au problème de *la simulation chirurgicale* qui constitue un des grands défis du domaine de l'animation en temps réel de corps virtuellement déformables. Le processus de simulation doit représenter le comportement d'un organe déformable qui réagit aux manipulations de l'utilisateur et de son outil haptique, et ce en temps réel. La modélisation repose sur la résolution d'un système d'équations aux dérivées partielles complexe pour lequel la méthode des éléments finis est généralement favorisée. Cependant, cette méthode nécessite une discrétisation du modèle déformable en une suite d'éléments géométriques connectés entre eux, un processus fastidieux et mal adapté pour des simulations qui nécessitent des découpes ou autres manipulations entraînant un changement de la topologie initiale de l'objet simulé. Ce travail cherche à confirmer l'hypothèse que des méthodes de discrétisation basées sur des particules, et donc sans maillage d'éléments, sont suffisamment rapides, précises et stables pour pallier le problème de l'approche traditionnelle des éléments finis. Pour y arriver, deux méthodes de discrétisation sans maillage sont construites. Les deux utilisent une formulation faible des équations d'équilibre de la théorie de l'élasticité linéaire en mécanique des milieux continus. Le premier modèle, la méthode basée sur les points (MBP), utilise une approximation du déplacement d'une particule basée sur un voisinage compact bâti autour de celle-ci à l'aide d'une fonction noyau. Pour intégrer l'équation d'équilibre sur le domaine non déformé, le modèle utilise le volume d'une particule décrit comme le rapport entre sa masse et la densité de son voisinage. Deux fonctions d'approximation du déplacement sont formulées : la fonction de forme "Smooth Particle Hydrodynamics" (SPH) et la fonction de forme "Moving least square" (MLS). Le deuxième modèle est basé sur la méthode Meshless Total Lagrangian Dynamics (MTLED) qui utilise également une approximation du déplacement d'une particule basée sur un voisinage compact, mais cette fois en utilisant un maillage d'éléments pour l'intégration de l'équation d'équilibre sur le domaine non déformé. Un algorithme intégrant ces deux modèles est proposé. Pour valider l'approche sans maillage, un modèle de référence basé sur la méthode des éléments finis avec des éléments tétraédriques et hexaédriques linéaires est décrit et également implémenté. La performance des méthodes sans maillage et de la méthode des éléments finis est finalement comparée. La comparaison est effectuée sur différents scénarios d'étirement et de fléchissement. Une analyse de la résolution statique du système d'équilibre et de la matrice de rigidité est également abordée. Les techniques d'intégration dans le temps et de gestion des forces de contact sont aussi testées. Les résultats obtenus amènent à l'invalidation de l'hypothèse de départ. Pour arriver à atteindre du temps réel, seules les méthodes d'intégration explicites dans le temps peuvent

être utilisées sur des objets de taille considérable, c'est-à-dire comportant quelques milliers de particules. D'ailleurs, même avec une méthode d'intégration explicite dans le temps, un nombre très restreint de voisins est nécessaire, entraînant ainsi une mauvaise propagation des forces. De plus, si le nombre de particules du volume est diminué en augmentant la taille du voisinage pour ainsi accélérer le temps de calcul, la faible consistance des fonctions d'approximation apporte de graves conséquences sur la précision et la stabilité de l'animation. Pour ces raisons, l'utilisation des deux approches sans maillage étudiées lors d'une modélisation complète du comportement d'un organe dans une simulation en temps réel n'apporte aucun gain par rapport à la méthode des éléments finis.

ABSTRACT

Virtual reality applications and computer simulations are currently undergoing exponential growth. This project focus on the field of *surgery simulation* which relies on computer animations concepts and complex physical models in order to realistically animate the behaviour of deformable soft tissues and organs under the manipulations of a haptic device driven by the user. The modelization involves a system of partial differential equations for which the finite element methods (FEM) are well adapted and have been used over the past three decades. Yet, those methods rely on a huge discretization process that breaks the simulated object into a group of geometrical elements connected together within a topology. Avoiding the discretization process would result in a major breakthrough for real time surgical simulation. This research suggests that particle-based discretization methods, also referred to as meshless methods, currently used in other simulation applications are adequately fast, precise and stable to meet the surgery simulation constraints. To sustain this hypothesis, the study and implementation of two meshless discretization models built around the weak form of elasticity equilibrium of the theory of continuum mechanics are performed. The first model, the point-based animation method (MBP), uses an approximation function of a displacement based on a compact neighborhood of particles built from a kernel function. To integrate the weak form of the equilibrium equations formulated within a lagrangian description, the model uses the volume of a particle as the ratio between its mass and the density of its neighborhood. Two approximation functions of the displacement are studied: the Smooth Particle Hydrodynamics (SPH) and the Moving Least Squares (MLS). The second meshless model is based on the Meshless Total Lagrangian Explicit Dynamics (MTLED) method and also uses an MLS approximation function, but takes advantage of a background integration grid to integrate the weak form of the equilibrium equations. To validate the meshless approaches, the implementation of a reference solution based on a finite element method using linear tetrahedral and hexahedral elements is carried out. The finite element and the two meshless models are tested on different stretching and bending behaviours. An analysis of the static case of equilibrium system and thereby the stiffness matrix is performed. Likewise, time integration schemes and contact penalty forces are tested. The results of the trials invalidate the research's hypothesis. In real time simulations, only the explicit time integration scheme was able to produce fast enough results on simulations involving large objects, that is, with more than a few thousands particles. Those particles were restrained to a low number of neighbours, causing a weak propagation of forces through the objects. Furthermore, because of the low consistency order of the approximations, decreasing the number of particles by

extending the neighborhood in order to reduce computation delays will result in severe instabilities inducing in most cases the end of the simulation process. For these reasons, the conclusion is that the two particle based methods presented in this research bring no valuable gains over the finite elements method in the field of real-time simulation of soft tissues deformation required by surgical simulators.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xii
LISTE DES FIGURES	xiii
LISTE DES SIGLES ET ABRÉVIATIONS	xv
LISTE DES ANNEXES	xvi
CHAPITRE 1 INTRODUCTION	1
1.1 Contexte de la simulation chirurgicale	1
1.2 Éléments de la problématique	3
1.2.1 Précision	4
1.2.2 Rapidité	5
1.2.3 Stabilité	5
1.3 Solution envisagée, question et hypothèse de recherche	6
1.4 Objectifs	7
1.5 Plan du mémoire	8
CHAPITRE 2 REVUE DES CONCEPTS DE BASE EN MÉCANIQUE DES MILIEUX CONTINUS	9
2.1 Description lagrangienne d'une déformation en mécanique des milieux continus	9
2.2 Description de la force engendrée par une déformation	12
2.3 Relation de comportement entre la déformation et les contraintes	14
2.4 Élastodynamique	15
2.5 Résumé	16

CHAPITRE 3	REVUE DE LITTÉRATURE	18
3.1	Introduction des premiers modèles déformables	18
3.2	Simulation des corps déformables dans le domaine médical	19
3.3	Modèles sans maillage	28
3.3.1	Dynamique des particules internes	29
CHAPITRE 4	MÉTHODE DES ÉLÉMENTS FINIS	36
4.1	Formulation faible	36
4.2	Discrétisation du domaine et intégration numérique	38
4.3	Interpolation et fonction de forme	41
4.4	Simulation semi-statique	42
4.5	Simulation dynamique et intégration temporelle	45
4.5.1	Intégration explicite	47
4.5.2	Intégration semi-implicite	49
4.6	Approche corotationnelle	49
4.7	Forces de contact et retour de force pour outils haptiques	51
4.8	Algorithme final du modèle FEM	53
CHAPITRE 5	MÉTHODES SANS MAILLAGE	54
5.1	Noyaux et fonctions de forme	54
5.2	Choix du noyau et analyse du domaine d'influence d'une particule	56
5.3	Élasticité : Animation basée sur les points (Méthode MBP)	58
5.3.1	Fonction de forme : approximation "Moving least square" (MLS) . . .	63
5.3.2	Fonction de forme : approximation "Smooth Particle Hydrodynamics" (SPH)	65
5.4	Élasticité : méthode Galerkin-free	66
5.4.1	Méthode approximative du modèle MTLED : approximation "Moving least square" (MLS)	68
5.5	Approche corotationnelle	71
5.6	Simulation dynamique et intégration temporelle	73
5.7	Synthèse	76
5.8	Algorithme final des méthodes MBP et MTLED	77
CHAPITRE 6	RÉSULTATS ET DISCUSSION	78
6.1	Analyse des modèles en mode statique	78
6.2	Simulation dynamique	82
6.2.1	Étirement	82

6.2.2	Fléchissement	86
6.2.3	Collision et intégration temporelle	91
6.3	Bilan	93
CHAPITRE 7 CONCLUSION		95
7.1	Synthèse des travaux	95
7.2	Limitations de la solution proposée	96
7.3	Améliorations futures	97
RÉFÉRENCES		98
ANNEXES		104

LISTE DES TABLEAUX

Tableau 6.1	Différence des déplacements des DDLs au bas d'un cube avec un modèle FEM $10 \times 10 \times 10$ de référence. La première colonne indique la moyenne des différences entre les déplacements des 25 particules du bas et celles du modèle de référence. Les deux colonnes suivantes indiquent le temps de construction et d'inversion de \mathbf{K} . La dernière colonne indique le conditionnement de \mathbf{K}	83
Tableau 6.2	Temps d'exécution - Étirement linéaire	87
Tableau 6.3	Temps d'exécution - Étirement linéaire avec corotation	89
Tableau 6.4	Temps d'exécution - Étirement non linéaire	90
Tableau A.1	Poids et coordonnées barycentriques pour n points d'intégration à l'intérieur d'un tétraèdre. (Yu (1984); Keast (1986))	104
Tableau A.2	Poids et coordonnées élémentaires pour n points d'intégration à l'intérieur d'un hexaèdre.	105

LISTE DES FIGURES

Figure 1.1	Simulateur orthopédique, OSSimTech	2
Figure 2.1	Déplacement d'un point	10
Figure 2.2	Vecteur de traction \mathbf{T} d'un point \mathbf{P} avec ses deux composants : le vecteur de traction normal σ_n et le vecteur de cisaillement τ_n (auteur Sanpaz, Wikipedia)	13
Figure 3.1	Interaction du modèle hybride masse-tenseur et semi-statique, Cotin et al. (1998)	25
Figure 3.2	(A-D) Utilisation du petit tenseur de déformation linéaire, (E) Utilisation du tenseur de déformation non linéaire (F-G) Tenseur de déformation linéaire vs non linéaire, Picinbono et al. (2003)	27
Figure 3.3	Potentiel de Lennard-Jones avec $n = 12$, $m = 6$, $\epsilon = \frac{B^2}{4A}$ et $\sigma = \sqrt[6]{\frac{A}{B}}$	29
Figure 4.1	Reconstruction 3D d'un foie à partir d'images 2D d'un scanneur	39
Figure 4.2	Décomposition SVD d'une transformation. Auteur Georg-Johann, Wikipédia	50
Figure 5.1	Supports d'influence monotones et décroissants d'un point I	55
Figure 5.2	Noyaux utilisés dans nos expériences (1D)	57
Figure 5.3	Noyaux utilisés dans nos expériences (2D)	57
Figure 5.4	Différences d'approximation et d'intégration entre les méthodes implémentées. Les points rouges sont les degrés de liberté (particules et sommets FEM), les croix représentent les points d'intégration, les lignes vertes représentent les liens d'approximation entre un point d'intégration et un degré de liberté et les lignes noires représentent un maillage d'éléments.	68
Figure 6.1	$4 \times 4 \times 4$ hexaèdres avec FEM : a) position non déformée, b) position déformée.	79
Figure 6.2	$4 \times 4 \times 4$ hexaèdres avec : a) 1 point d'intégration au milieu de chaque hexaèdre, b) 2 tétraèdres de 3 points d'intégration dans chaque hexaèdre, c) 64 points d'intégration (degré 4) d) FEM 1 point d'intégration	80
Figure 6.3	Matrices de rigidité \mathbf{K} : a) 1 point d'intégration au milieu de chaque hexa, b) 2 tétras de 3 pts dans chaque hexa, c) 64 points d'intégration (degré 4)	81

Figure 6.4	Étirement d'un parallélépipède de $10 \times 40 \times 10$ particules : de gauche à droite, l'approximation SPH avec respectivement 10, 20 et 30 voisins pour chaque particule. À droite complètement, la méthode FEM de référence.	84
Figure 6.5	Étirement d'un parallélépipède de $10 \times 40 \times 10$ particules : de gauche à droite, l'approximation MLS avec respectivement 10, 20 et 30 voisins pour chaque particule. À droite complètement, la méthode FEM de référence.	85
Figure 6.6	Fléchissement d'un parallélépipède de $10 \times 40 \times 10$ particules : comparaison entre tenseurs de déformation et corotation. a-c) SPH 10 voisins, d-e) SPH 20 voisins, g-i) SPH 30 voisins, j-l) FEM	87
Figure 6.7	Fléchissement d'un parallélépipède de $10 \times 40 \times 10$ particules : comparaison entre tenseurs de déformation et corotation. a-c) MLS 10 voisins, d-e) MLS 20 voisins, g-i) MLS 30 voisins, j-l) FEM	88
Figure 6.8	Fléchissement d'un parallélépipède de $10 \times 40 \times 10$ particules avec 3159 pts d'intégration : comparaison entre tenseurs de degré de consistance a-c) Galerkin-Free 10 voisins, d-f) Galerkin-Free 15 voisins, g-i) Galerkin-Free 20 voisins	90
Figure 6.9	Collision par multiplicateurs de Lagrange : ligne du haut méthode d'intégration temporelle explicite, ligne du bas méthode d'intégration temporelle semi-implicite. Pas de temps de 0.001 seconde. Méthode FEM corotationnelle avec des cubes de $10 \times 10 \times 10$ DDLs, $\nu = 0.3$, $k = 2000$	92
Figure 6.10	Collision par pénalités : ligne du haut pénalité par multiplicateurs de Lagrange, ligne du bas pénalité par ressort. Pas de temps de 0.001 seconde avec intégration semi-implicite. Méthode corotationnelle avec des cubes de $10 \times 10 \times 10$ DDLs, $\nu = 0.3$, $k = 2000$	94

LISTE DES SIGLES ET ABRÉVIATIONS

DDL	Degré(s) de liberté
FEM	Méthode des éléments finis (Finite element method)
MBP	Méthode basée sur les points de Müller et al. (2004)
MTLED	“Meshless Total Lagrangian Explicit Dynamic” de Horton et al. (2010)
MLS	Moving least squares
SPH	Smooth Particle Hydrodynamics
CG	Conjugate gradient (solveur itératif du gradient conjugué)

LISTE DES ANNEXES

Annexe A	Intégration numérique et quadratures	104
Annexe B	Notation et termes utilisés	106

CHAPITRE 1 INTRODUCTION

Le domaine de la simulation par ordinateur et de la réalité virtuelle est maintenant très largement ancré dans notre société. On peut penser aux jeux vidéos, aux simulateurs de vols et d'exploration de l'espace qui prennent de plus en plus d'espace dans notre quotidien. Certains professionnels font également appel aux simulateurs dans le cadre de leur travail. L'aspect virtuel de la simulation leur permet entre autres de s'entraîner et de se former sans danger.

Dans ce projet de recherche, nous nous intéresserons à la simulation dans un cadre professionnel bien précis : la *simulation chirurgicale*. Dans la prochaine section, nous expliquerons l'importance de ce type de simulation et pourquoi ce domaine de recherche a pris autant d'envergure de nos jours. Nous présenterons ensuite les problèmes qui se rattachent à la simulation chirurgicale ainsi que les contraintes dont elle est confrontée. Nous discuterons par la suite du problème qui est au coeur de ce projet et de l'hypothèse que nous désirons vérifier. Nous terminerons le chapitre par une présentation claire de l'ensemble de nos objectifs et du plan détaillé de ce mémoire.

1.1 Contexte de la simulation chirurgicale

Il va de soi que les tâches d'un chirurgien sont très complexes. Elles peuvent entre autres engendrer de graves conséquences pour le patient. En contrepartie, une chirurgie bien réalisée peut grandement améliorer la condition d'un patient, voir lui sauver la vie. Le chirurgien doit donc avoir recours à un entraînement rigoureux.

Malheureusement, l'entraînement d'un chirurgien comporte beaucoup d'obstacles. Dans un premier temps, s'exercer sur des animaux, morts ou vivants, représente un coût logistique et monétaire important, sans parler des problèmes d'éthique qui en découle. L'entraînement sur un vrai patient sous la supervision d'un chirurgien expérimenté comporte également son lot d'inconvénients. On peut penser à la difficulté d'obtenir un patient ayant la condition médicale recherchée, ou encore les limites d'un processus d'apprentissage par observation (Cover et al. (1993)). Les méthodes traditionnelles d'entraînement comportent donc des difficultés non négligeables : les dépenses liées au matériel requis et à la logistique, le temps restreint d'un chirurgien, les considérations d'éthique, la difficulté d'obtenir et de répéter les conditions d'une opération chirurgicale complexe et la singularité du corps d'un patient.

C'est ici qu'intervient la simulation virtuelle d'une chirurgie. Plutôt que de s'entraîner sur des

animaux ou des cadavres, les chirurgiens peuvent s'exercer dans un environnement virtuel qui vient calquer l'environnement de chirurgie et les propriétés des organes et tissus humains (ce type d'environnement virtuel est illustré à la figure 1.1). Dans Feudner et al. (2009), 31 étudiants en médecine et 32 résidents en ophtalmologie participent à une étude de trois semaines où chacun doit effectuer trois capsulorhexis (type chirurgie pour soigner la cataracte) sur des porcs au début et à la fin de l'étude. Parmi ces candidats, un échantillon de participants sélectionnés au hasard s'entraînera à l'aide du simulateur virtuel ophtalmologique EYESI. Les résultats sont convaincants : les utilisateurs du simulateur démontrent d'importantes améliorations sur leurs performances globales et une meilleure consistance dans leurs résultats. De la même façon, 16 résidents en chirurgie prennent part à une étude réalisée dans Seymour et al. (2002). Les résidents effectuent des cholécystectomies (chirurgie laparoscopique consistant à l'ablation de la vésicule biliaire). Les auteurs démontrent que la dissection est 29% plus rapide chez les résidents entraînés par un simulateur virtuel. Ces mêmes résidents étaient également six fois moins portés à faire des erreurs. Des études similaires sont réalisées dans Haluck et al. (2001); Haller et al. (2009); Selvander et Åsman (2012).



Figure 1.1 Simulateur orthopédique, OSSimTech

Les avantages des simulateurs chirurgicaux sont multiples. Toutefois, trois d'entre eux sont particulièrement bien documentés (Gibson et al. (1997)). Un premier avantage, l'entraînement par simulation permet de réduire les coûts et la logistique reliés à l'obtention de cadavres ou d'animaux vivants. Avec le simulateur, l'utilisateur du simulateur peut répéter autant de fois qu'il le souhaite un exercice de chirurgie, et ce sans l'utilisation d'êtres vivants et toutes les

considérations qui en découlent.

Un second avantage est la planification. Dans certains cas, l'anatomie d'un patient ou la singularité d'une opération peut devenir un problème important. La reconstruction virtuelle de certaines parties du corps du patient peut alors être intégrée dans un simulateur pour aider les chirurgiens à planifier une opération difficile et à faciliter la communication entre eux et entre les médecins et leurs patients.

Finalement, la simulation virtuelle ouvre la porte vers la chirurgie guidée par l'image et la réalité augmentée. Durant une opération chirurgicale, le chirurgien peut obtenir une aide visuelle présentant une vue plus globale de l'opération en temps réel. De plus, la simulation peut guider le chirurgien et l'empêcher d'effectuer une fausse manoeuvre dans une région sensible du patient.

Avec l'arrivée d'ordinateurs de plus en plus puissants, nous pouvons facilement prévoir que les gains associés à la simulation chirurgicale iront en s'accroissant au cours des prochaines années et que les avantages seront encore plus nombreux.

1.2 Éléments de la problématique

Comme nous pouvons l'imaginer, la simulation chirurgicale fait appel à des notions de biomécaniques. Elle fait également appel à des modèles physiques pour représenter toutes sortes de phénomènes incluant des déformations de corps viscoélastiques.

Évidemment, le problème de modélisation de déformations de corps viscoélastiques n'est pas réservé uniquement à la simulation chirurgicale. D'autres types de simulation font face aux mêmes défis. On a qu'à penser aux simulations utilisées pour créer des effets spéciaux au cinéma ou encore pour animer des objets mous dans des jeux vidéos. Toutefois, contrairement à ces applications, la simulation chirurgicale doit produire un résultat à l'intérieur d'un temps très court, en temps réel. En effet, le simulateur doit donner à l'utilisateur l'impression d'être en présence d'un organe réel. La simulation doit également être doublée d'un excellent rendu visuel des déformations des corps afin de bien refléter l'impact des gestes et actions du chirurgien pendant l'intervention. Finalement, notre processus de simulation doit intégrer l'interaction d'un dispositif haptique pour permettre au chirurgien de manipuler et ressentir l'objet virtuel affiché à son écran.

L'équilibre entre le réalisme et la rapidité d'exécution du simulateur représente évidemment l'objectif ultime dans ce type de simulation. Imaginons un instant la simulation d'un organe virtuel présenté à l'écran d'un ordinateur où l'utilisateur peut interagir physiquement avec l'organe à l'aide d'un outil haptique. Dans cette simulation, l'outil haptique fait bouger un

instrument virtuel quelconque à l'écran, par exemple un scalpel. Lorsque l'instrument virtuel touche l'organe, ce dernier doit se déformer en fonction de la collision avec l'instrument. L'outil haptique, quant à lui, doit offrir une résistance (ou un retour de force) à l'utilisateur pour lui donner une sensation d'immersion avec la simulation. Imaginons également qu'en appuyant sur un bouton, l'utilisateur peut se servir de l'instrument virtuel pour découper l'organe. On se retrouve alors avec des modifications importantes de la structure topologique initiale de l'objet simulé.

La simulation chirurgicale relève donc de l'animation en temps réel d'un corps virtuellement déformable, un sous domaine de l'infographie. Avec ce type d'animation, trois contraintes importantes doivent être prises en considération : la précision des résultats, la rapidité de calcul et stabilité des solutions. Regardons d'un peu plus près l'implication de ces trois contraintes dans le contexte chirurgical qui nous intéresse.

1.2.1 Précision

Dans le cadre d'une simulation chirurgicale, la précision sera évaluée en fonction du réalisme du comportement élastique du corps mou simulé et de la réaction avec l'instrument virtuel dirigé par l'utilisateur. Que ce soit un foie, un cerveau, des reins ou bien toutes autres parties du corps humain qui présentent des caractéristiques élastiques, nous souhaitons que nos modèles représentent l'objet d'une manière plausible et très proche de la réalité.

Comme nous pourrions le voir dans notre revue de la littérature au chapitre 3, la théorie de l'élasticité en mécanique des milieux continus est souvent l'approche la plus recommandée pour produire des simulations très réalistes. Avec cette approche, des tenseurs physiques sont utilisés pour décrire l'état d'un matériau ayant subi un effort externe. En réalité, ces tenseurs décrivent l'équilibre entre les forces internes et externes qui sont appliquées sur un objet élastique. Les déformations de l'objet simulé sont obtenues une fois que l'équilibre du système a été atteint. Pour y arriver, un système d'équations aux dérivées partielles doit être résolu. Cette résolution peut se faire en deux approches distinctes. Dans un premier temps, la déformation peut être obtenue graduellement dans le temps par un passage des états intermédiaires de l'objet entre sa forme initiale et sa forme finale lorsque l'état d'équilibre est atteint. Nous parlons alors d'une simulation dynamique. Sinon, la déformation est obtenue en calculant l'équilibre optimal de notre système d'un seul coup à chaque fois qu'une force externe est appliquée ou change d'intensité et de direction. On parle alors d'une simulation statique.

Dans les deux types de simulation, le calcul des déplacements passe par le calcul d'une intégrale. Dans le cas d'intégration dans l'espace, la simulation fait appel à des méthodes de

discrétisations. Dans le cas de simulation dynamique, les états de l'objet jusqu'à sa forme d'équilibre doivent être calculés. Dans ce cas, on fait appel à des méthodes numériques d'intégration dans le temps. Dans les deux cas, des méthodes d'approximations seront requises. Ces méthodes de discrétisations et d'approximations deviennent potentiellement de grandes sources d'erreurs visuelles portant atteinte à la précision de la simulation.

La précision de la simulation intervient également au niveau des forces de contact avec l'objet haptique. Lorsqu'une collision est détectée entre l'instrument et l'organe virtuels, la force de contact doit être calculée et transmise à l'outil haptique de l'utilisateur. De plus, cette force doit également être insérée dans le calcul de la déformation de l'organe. Différentes approches peuvent être utilisées pour déterminer cette force. Ce choix doit se faire en fonction du compromis entre le niveau de précision et la rapidité du processus de simulation recherché.

1.2.2 Rapidité

La contrainte de rapidité est directement reliée avec la sensation d'immersion de l'utilisateur. Comme nous l'avons vu plus haut, nous devons respecter des temps très courts autant pour le rendu visuel de la simulation que pour le rendu haptique des forces de contact entre l'instrument et l'organe virtuel.

Pour que la simulation soit réaliste à l'oeil de l'utilisateur, une image doit être rendue à l'écran à une fréquence d'environ 30 à 60 images par seconde. Si nous choisissons 30 images par seconde, dans le cas d'une simulation statique, le système d'équations aux dérivées partielles devra être résolu 30 fois par seconde. Parallèlement, si la simulation est dynamique, le système d'équations aux dérivées partielles sera intégré dans le temps en discrétisant la simulation en pas de temps généralement constants. La longueur d'un pas de temps devra donc être inférieur à $1/30$ d'une seconde.

En ce qui a trait au rendu haptique, la simulation statique ou dynamique devra respecter une fréquence de 1000 rendus de force à chaque seconde (Lin et Otaduy (2008)). Nous verrons qu'il n'est généralement pas possible à cette fréquence d'intégrer cette force à la dynamique de l'organe déformable. On doit donc séparer le rendu visuel et haptique.

1.2.3 Stabilité

La stabilité est très souvent dépendante des méthodes numériques utilisées (par exemple lors des processus d'intégration dans le temps et dans l'espace). Un manque de stabilité intervient directement avec le réalisme de la simulation (et donc la contrainte de précision) puisqu'une simulation instable produit des artéfacts visuels et haptiques. Il n'est pas rare

même qu’une simulation instable fasse diverger complètement le système menant donc à la fin de la simulation. De plus, puisque la simulation est en temps réel, la contrainte de rapidité nous restreint à l’utilisation de méthodes numériques peu coûteuses en temps, mais facilement perturbables, posant alors de graves problèmes de stabilité.

Deux types de stabilité doivent être considérés : la stabilité en temps et la stabilité en interaction. Pour la stabilité en temps, la simulation doit pouvoir récupérer les passages de temps irréguliers. En d’autres termes, si la résolution du système au temps $T = t - 1$ prend un peu plus de temps (une application externe pourrait, par exemple, utiliser beaucoup de ressources en même temps que la simulation), la solution au temps $T = t$ doit demeurer cohérente. Dans le cas de stabilité en interaction, le corps virtuel étant déformé par l’interaction manuelle d’un utilisateur, on ne peut prévoir les manipulations effectuées. La simulation doit rester stable pour tout type d’interactions externes.

1.3 Solution envisagée, question et hypothèse de recherche

Une des particularités avec la simulation chirurgicale est que les organes doivent généralement se déformer en fonction d’un comportement élastique. Comme nous l’avons mentionné plus haut (nous le verrons plus en détail au chapitre 2), la modélisation de ce comportement est basée sur un système d’équations différentielles et sur d’autres éléments empruntés à la théorie de l’élasticité en mécanique des milieux continus. Le calcul des intégrales de ce système requiert évidemment l’utilisation de techniques de discrétisation du domaine volumétrique de notre objet (l’organe dans le cas de la simulation chirurgicale) et l’application de méthodes d’approximation numériques.

Les techniques de discrétisation utilisés en animation par ordinateur sont généralement regroupés en deux grandes catégories : *les méthodes avec maillage* dont fait partie l’approche des éléments finis et *la méthode sans maillage*. Comme l’indique son nom, avec l’approche des éléments finis, la discrétisation du domaine se fait à partir d’éléments géométriques d’interpolation connectés entre eux (triangles, tétraèdres et hexaèdres). Le processus d’intégration se fait alors directement sur ces éléments géométriques (une description plus détaillée de cette approche sera fournie au chapitre 4).

Dans le cas des méthodes sans maillage, les particules ne sont pas regroupées en objets géométriques dans le processus de discrétisation du domaine. L’intégration se fait plutôt sur des espaces compacts d’approximation entourant l’ensemble des points du domaine (une description des méthodes sans maillage sera fournie au chapitre 5). Ces méthodes ont donc un avantage important par rapport l’approche des éléments finis puisqu’elles ne requièrent

aucun maillage. La création d'un maillage est en effet un processus long et complexe qui devient un défi important dans un processus de simulation en temps réel. Ceci est d'autant plus important lorsque la déformation implique une découpe de l'objet, comme c'est souvent le cas en simulation chirurgicale. Dans ce cas, un nouveau processus de maillage des objets produit par la découpe doit être répété ce qui entraîne des temps de calcul importants, sans parler des sources d'erreurs liées à une mauvaise topologie autour de la découpe.

Évidemment, l'absence d'une structure géométrique (ou éléments) au niveau de la discrétisation du domaine volumétrique vient grandement affecter notre processus d'intégration. L'impact se fera sentir tant au niveau de la précision que du temps de calcul de notre simulation. La question est donc de savoir si malgré ces impacts, *l'approche sans maillage est suffisamment rapide, précise et stable pour rencontrer les exigences du domaine chirurgical et ainsi offrir une option de choix pour les simulations impliquant des coupes*.

Pour cette recherche, nous supposons que les méthodes sans maillage peuvent supporter des simulations chirurgicales en temps réel. À partir de cette hypothèse, nous chercherons donc à démontrer que les modèles sans maillage rencontrent les contraintes de rapidité, précisions et stabilité énumérées plus tôt dans ce chapitre et qu'elles sont appropriées pour les simulations chirurgicales.

1.4 Objectifs

Dans ce projet de recherche, nous aborderons le problème de déformations des corps déformables dans un contexte de simulation chirurgicale en temps réel et avec l'interaction d'un outil haptique.

L'objectif principal de notre projet sera de valider si l'utilisation des méthodes sans maillage couramment utilisées dans le domaine de la théorie de l'élasticité linéaire en mécanique des milieux continus est adaptée pour ce type de simulation. Nous chercherons entre autres à comparer ces méthodes à la méthode traditionnelle des éléments finis qui est bien documentée dans la littérature. Cet objectif est divisé en quatre parties :

Objectif 1 Uniformiser les parties de la théorie de l'élasticité linéaire en mécanique des milieux continus qui nous sont nécessaires pour notre cadre de simulation

Objectif 1.1 Identifier les tenseurs physiques et leurs simplifications

Objectif 1.2 Identifier les lois de comportement

Objectif 1.3 Décrire les outils mathématiques qui se rattachent à la théorie

Objectif 2 Méthode avec maillage

Objectif 2.1 Implémenter la méthode des éléments finis avec des éléments tétraédriques et hexaédriques linéaires

Objectif 2.2 Intégrer les forces de contact avec un simple facteur de rigidité

Objectif 2.3 Intégrer les forces de contact avec les multiplicateurs de Legendre

Objectif 3 Méthodes sans maillage

Objectif 3.1 Analyser les méthodes d'approximation basées sur des particules

Objectif 3.2 Analyser les méthodes d'intégration basées sur des particules

Objectif 3.3 Choisir et implémenter des méthodes d'approximation et d'intégration

Objectif 3.4 Identifier les limites et sources d'erreurs de ces méthodes

Objectif 4 Évaluer les méthodes sans maillage en fonction des critères de précision, rapidité et stabilité

1.5 Plan du mémoire

Le plan du mémoire découle directement des objectifs de recherche proposés à la section précédente. Nous débuterons par un chapitre introduisant la *théorie de l'élasticité linéaire*. Ce chapitre posera de façon explicite le système d'équations aux dérivées partielles qui devra être résolu de manière semi-statique ou dynamique dans nos simulations. Le chapitre suivant présentera une *revue de la littérature*. Cette revue permettra d'établir rapidement les différentes techniques d'animation de corps déformables, puis de manière plus approfondie, l'utilisation de la méthode des éléments finis et des méthodes sans maillage dans un contexte de simulation temps réel. Le chapitre 4 présentera ensuite notre analyse de la *méthode des éléments finis*. L'analyse *des méthodes sans maillage* sera ensuite présentée au chapitre 5. Finalement, le chapitre 6 proposera *un bilan* général des expériences que nous avons effectuées avec les deux types de méthodes et exposera les limites et les forces des méthodes sans maillage. Nous terminerons au chapitre 7 avec une *conclusion* qui viendra résumer l'ensemble des observations de ce mémoire et nos recommandations pour des recherches futures.

CHAPITRE 2 REVUE DES CONCEPTS DE BASE EN MÉCANIQUE DES MILIEUX CONTINUS

Les modèles numériques présentés dans les prochains chapitres feront appel à la théorie de l'élasticité en mécanique des milieux continus pour la modélisation ou l'animation de corps déformables. Cette approche permet de reproduire une déformation réaliste d'un objet à partir d'équations différentielles impliquant une mesure de cette déformation ainsi que la réaction qu'elle impose à un matériau.

Dans ce chapitre, nous exposerons quelques concepts de base sur l'analyse de la déformation et des forces de contraintes (réaction à la déformation) d'un matériau en réponse à un effort externe, et la relation qui les relie. Nous nous référerons à ces concepts tout au long de ce mémoire.

2.1 Description lagrangienne d'une déformation en mécanique des milieux continus

Dans le cas de simulation d'un corps purement élastique, on s'intéresse précisément à la force qui vise à minimiser l'énergie potentielle d'une déformation. En d'autres mots, on cherche la force interne (\mathbf{F}^{int}) qui pousse un corps à reprendre sa forme initiale, ou de repos, après l'application d'une force externe (\mathbf{F}^{ext}) sur le domaine. Ce principe est illustré à la figure 2.1 ci-dessous.

La figure présente un objet ovale dans sa forme initiale au temps $t = t_0$ à gauche de la ligne pointillée ainsi qu'à sa forme déformée au temps $t = t_1$ à droite de la ligne pointillée. On représente le domaine volumétrique de référence (non déformé) d'un objet par $\Omega \subset \mathbb{R}^3$. La **description lagrangienne** du système définit la position $\mathbf{X} = [x, y, z]^T \in \Omega$ d'une particule de l'objet à son état initial. Nous disons que \mathbf{X} est la **coordonnée matérielle** de la particule. La fonction $\phi(\mathbf{X}, t) = \mathbf{x}$ du domaine au temps t décrit la transformation de la coordonnée lagrangienne à sa **description eulérienne**, également appelée la **coordonnée spatiale** d'une particule. Finalement, le déplacement $\mathbf{u} = \phi(\mathbf{X}, t) - \mathbf{X} = [u, v, w]^T$ correspond au vecteur qui relie la coordonnée matérielle \mathbf{X} de la particule et sa coordonnée spatiale \mathbf{x} dans la configuration actuelle après déformation au temps t_1 .

On dit qu'une transformation est rigide si elle préserve la distance entre chaque point du domaine. On peut alors la représenter par

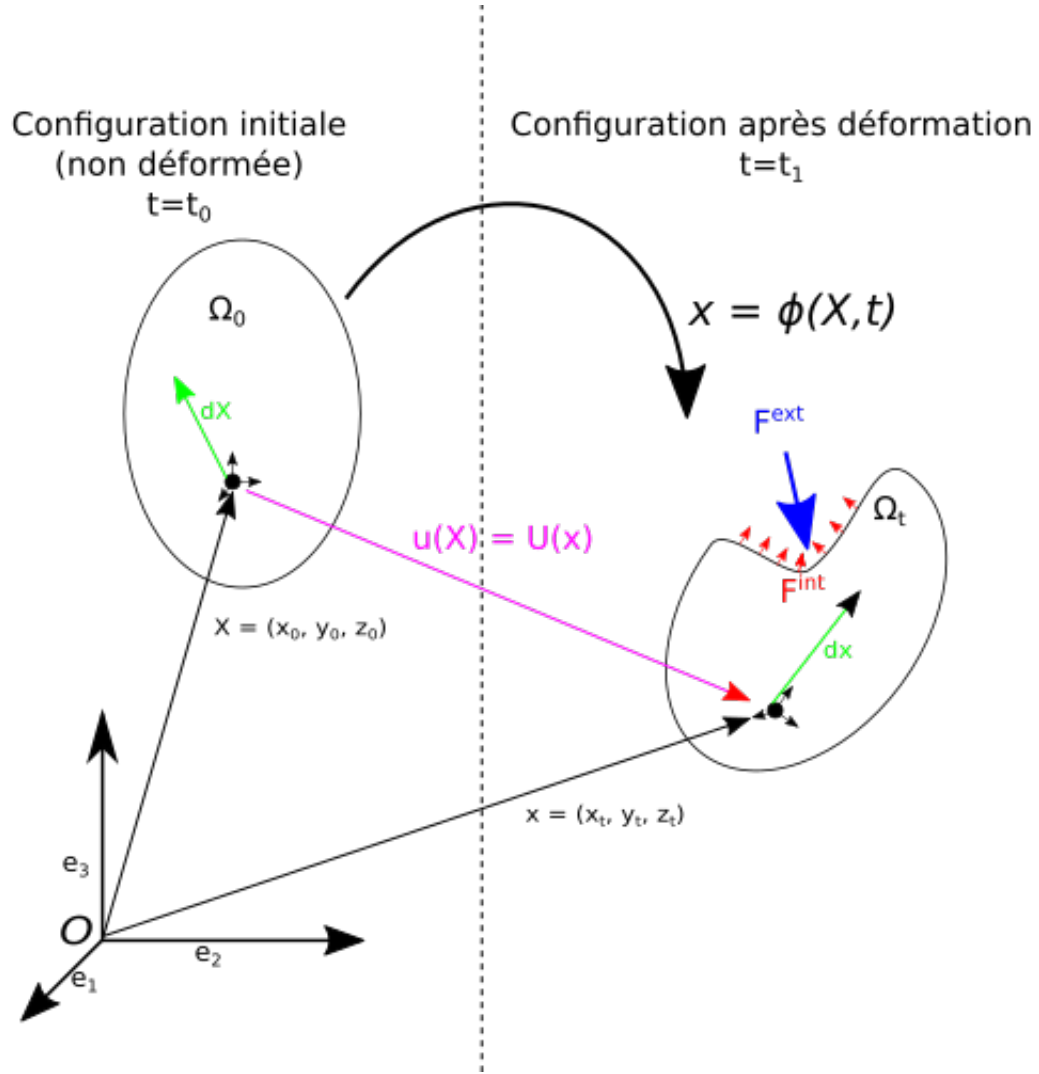


Figure 2.1 Déplacement d'un point

$$\phi(\mathbf{X}, t) = \mathbf{R}\mathbf{X} + \mathbf{T} \quad (2.1)$$

où \mathbf{R} est une matrice de rotation (orthogonale) avec son déterminant $|\mathbf{R}| = 1$ et \mathbf{T} est un vecteur de translation. Nous verrons plus tard qu'une transformation rigide ne produit pas de force interne. Si nous regardons de nouveau la figure 2.1, on peut voir que la force interne (les petites flèches rouges) provient uniquement de la région déformée de l'objet, et non de la translation et de la rotation.

Malheureusement, dans le cas d'une simulation en temps réel, on ne connaît pas explicitement ϕ . Nous pouvons cependant décrire une déformation localement en introduisant le tenseur de transformation \mathbf{F} qui met en relation un vecteur tangent $d\mathbf{X}$ (on peut penser à un très

petit segment entre deux points) d'une particule i du domaine initial à son vecteur tangent $d\mathbf{x}$ dans le domaine déformé (voir les vecteurs verts de la figure 2.1).

$$d\mathbf{x} = \mathbf{F}d\mathbf{X} \quad (2.2)$$

En utilisant le gradient du déplacement $\nabla \mathbf{u}$, on définit alors le tenseur de transformation comme suit :

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \nabla \phi(\mathbf{X}, t) = \nabla(\mathbf{X} + \mathbf{u}) = \mathbf{I} + \nabla \mathbf{u} = \begin{bmatrix} 1 + u_{,x} & u_{,y} & u_{,z} \\ v_{,x} & 1 + v_{,y} & v_{,z} \\ w_{,x} & w_{,y} & 1 + w_{,z} \end{bmatrix} \quad (2.3)$$

On note que le tenseur de transformation \mathbf{F} correspond à la matrice jacobienne de la déformation et l'on représente son déterminant par $J = \det(\mathbf{F})$. Lors d'une transformation, si le déterminant est plus grand que 1, le volume de l'objet augmente, et si ce dernier est inférieur à 1, le volume diminue. Dans le cas d'un matériau incompressible, le déterminant de \mathbf{F} est donc toujours 1.

Ayant défini la relation entre $d\mathbf{X}$ et $d\mathbf{x}$, on peut alors introduire une métrique permettant de quantifier la déformation :

$$\begin{aligned} \|d\mathbf{x}\|^2 - \|d\mathbf{X}\|^2 &= \|\mathbf{F}d\mathbf{X}\|^2 - \|d\mathbf{X}\|^2 \\ &= ((\mathbf{F}d\mathbf{X})^T \cdot \mathbf{F}d\mathbf{X}) - ((d\mathbf{X})^T \cdot d\mathbf{X}) \\ &= d\mathbf{X}^T (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) d\mathbf{X} \end{aligned} \quad (2.4)$$

En utilisant la partie symétrique de la relation précédente, on pose

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u}^T \cdot \nabla \mathbf{u}) \quad (2.5)$$

qui est également appelé le tenseur de déformation de Green-Lagrange. Dans le contexte d'une simulation chirurgicale, l'hypothèse de petites déformations est souvent utilisée : la résistance offerte à l'utilisateur de l'outil haptique augmente proportionnellement à l'importance de la déformation (Cotin et al. (1998)). Ainsi, il n'est pas rare que le terme non linéaire du tenseur de déformation soit ignoré, obtenant ainsi le petit tenseur de déformation.

$$\boldsymbol{\epsilon} = \frac{1}{2}(\mathbf{F} + \mathbf{F}^T) - \mathbf{I} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \quad (2.6)$$

2.2 Description de la force engendrée par une déformation

À partir de la mesure de la déformation décrite à la section précédente, on peut alors décrire la force par unité de surface que subit notre objet en faisant appel à la théorie de l'analyse de contraintes, ou encore l'analyse du stress d'un matériau. On distingue en fait deux catégories de contraintes selon le type de l'unité de surface. Dans le cas d'une unité de surface non déformée $d\mathbf{A} = \mathbf{N}dA$ où \mathbf{N} correspond au vecteur normal de cette surface, nous dirons qu'il s'agit de contraintes lagrangiennes. Sinon, pour une unité de surface déformée $d\mathbf{a} = \mathbf{n}da$, avec \mathbf{n} correspondant au vecteur normal de cette surface, nous dirons qu'il s'agit de contraintes eulériennes, dites "vrai stress". Puisque la géométrie de la surface déformée est difficile à obtenir, nous nous limiterons aux contraintes lagrangiennes pour nos simulations.

On peut définir l'intensité de la force (ou encore la force par unité de contact) par un vecteur \mathbf{T} dit vecteur de traction, ou vecteur de stress

$$\mathbf{T}(\mathbf{x}, t) = \frac{d\mathbf{P}}{da} \quad (2.7)$$

$$\mathbf{T}(\mathbf{X}, t) = \frac{d\mathbf{P}}{dA} \quad (2.8)$$

où $d\mathbf{P}$ est la force agissant sur la surface da (respectivement dA). Dans le cas de petites déformations ($da \approx dA$), nous avons $\mathbf{T}(\mathbf{x}, t) \approx \mathbf{T}(\mathbf{X}, t)$.

Le vecteur de traction est alors décomposé en deux composantes (voir figure 2.2), l'une est dépendante à la normale \mathbf{n} de la surface et est appelée le vecteur de traction normal :

$$\boldsymbol{\sigma}_n = \frac{d\mathbf{P}_n}{da} \quad (2.9)$$

et l'autre, parallèle à la surface, est appelée le vecteur de cisaillement :

$$\boldsymbol{\tau}_n = \frac{d\mathbf{P}_s}{da} \quad (2.10)$$

Selon le théorème de contraintes de Cauchy, il existe un tenseur de deuxième ordre $\boldsymbol{\sigma}$, appelé le tenseur des contraintes de Cauchy, qui est indépendant du vecteur normal à la surface \mathbf{n} et qui permet d'obtenir le vecteur de traction \mathbf{T} à partir de l'équation linéaire suivante :

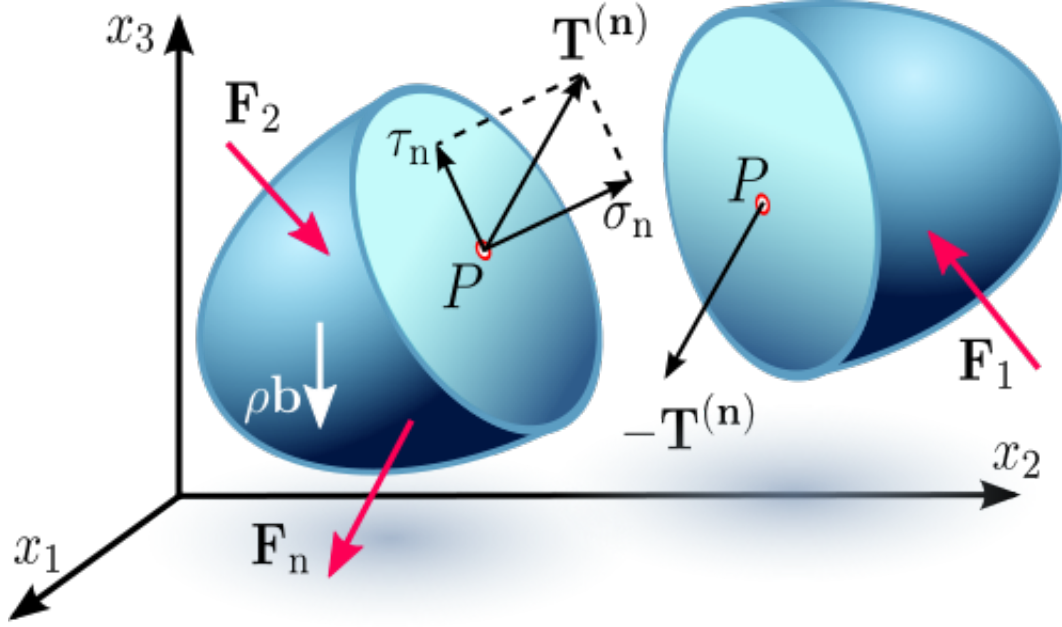


Figure 2.2 Vecteur de traction \mathbf{T} d'un point \mathbf{P} avec ses deux composantes : le vecteur de traction normal σ_n et le vecteur de cisaillement τ_n (auteur Sanpaz, Wikipedia)

$$\mathbf{T}(\mathbf{n}) = \mathbf{n} \cdot \boldsymbol{\sigma} \quad (2.11)$$

où $\boldsymbol{\sigma}$ est construit comme suit :

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{00} & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11} & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \quad (2.12)$$

La première composante (x, y ou z) en indice indique que la contrainte est sur le plan parallèle à cette composante. La deuxième indique la direction de la contrainte (dans le cas du cisaillement). Nous pouvons démontrer que ce tenseur de contraintes est symétrique (Taber (2004)). Ainsi, il n'est pas rare de le décrire par la notation de Voigt :

$$\boldsymbol{\sigma} = [\sigma_x \ \sigma_y \ \sigma_z \ \tau_{yz} \ \tau_{xz} \ \tau_{xy}]^T \quad (2.13)$$

Le tenseur de contraintes de Cauchy est formulé en termes de coordonnées spatiales. Pour une description lagrangienne (en coordonnées matérielles), on introduit le second tenseur des contraintes de Piola-Kirchhoff (Wriggers (2008))

$$\mathbf{S} = J\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \quad (2.14)$$

avec \mathbf{F} le tenseur de transformation introduit à l'équation 2.3 et J son déterminant.

2.3 Relation de comportement entre la déformation et les contraintes

Il reste donc à définir explicitement la relation entre les valeurs du tenseur des contraintes $\boldsymbol{\sigma}$ et le tenseur de déformation \mathbf{E} (ou $\boldsymbol{\epsilon}$ dans le cas du tenseur linéaire pour les petites déformations). Cette relation entre les contraintes et la déformation se nomme une loi de comportement et dépend des caractéristiques du matériau utilisé dans la simulation. Il existe donc plusieurs lois de comportement, certaines sont linéaires (où le tenseur de contraintes $\boldsymbol{\sigma}$ dépend linéairement du tenseur de déformation \mathbf{E}), d'autres sont non linéaires. Pour ce mémoire, nous nous concentrerons uniquement sur des matériaux linéaires et isotropes (c'est-à-dire que les caractéristiques du matériau sont les mêmes dans toutes les directions d'un point à l'intérieur de l'objet).

Soit $W(\mathbf{E})$ une densité d'énergie de la déformation, fonction du tenseur de déformation. On dit alors qu'un matériau est hyperélastique (ou élastique de Green) s'il est possible d'en dériver le tenseur de contraintes ainsi :

$$\mathbf{S} = \frac{\partial W}{\partial \mathbf{E}} \quad (2.15)$$

Il s'agit donc d'une représentation idéaliste de l'élasticité puisqu'elle ne dépend de rien d'autre que de la densité d'énergie de la déformation. Dans le cas d'un matériau isotrope, la densité d'énergie de la déformation est fonction uniquement des trois invariants de la déformation I_1 , I_2 et I_3 définis comme

$$I_1 = \text{tr} \mathbf{C} \quad (2.16)$$

$$I_2 = \frac{1}{2} \left[(\text{tr} \mathbf{C})^2 - \text{tr} \mathbf{C}^2 \right] \quad (2.17)$$

$$I_3 = \det \mathbf{C} \quad (2.18)$$

où $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ est le tenseur de déformation droit de Cauchy-Green. Tel que mentionné au début du chapitre, le déterminant du tenseur de déformation \mathbf{F} est 1 pour un matériau incompressible et il s'en suit donc que $I_3 = \det \mathbf{C} = \det(\mathbf{F}^T \mathbf{F}) = 1$ dans ce cas particulier.

En posant $J_1 = \frac{1}{2}(I_1 - 3)$ et $J_2 = \frac{1}{4}(I_2 - 2I_1 + 3)$ (Taber (2004)), nous pouvons définir explicitement la densité W pour un matériau linéaire isotrope incompressible comme

$$W = C_1 J_1^2 + C_2 J_2 \quad (2.19)$$

avec $C_1 = \frac{1}{2}(\lambda + 2\mu)$, $C_2 = -2\mu$. Ici λ et μ sont les coefficients de Lamé définis par

$$\begin{aligned} \lambda &= \frac{k\nu}{(1+\nu)(1-2\nu)} \\ \mu &= \frac{k}{2(1+\nu)} \end{aligned} \quad (2.20)$$

où k est le module de Young (la constante qui relie la contrainte de traction - ou de compression - et le début de la déformation du matériau élastique isotrope) et ν est le coefficient de Poisson (la constante caractérisant la contraction du matériau élastique isotrope perpendiculairement à la direction de l'effort appliqué).

Puisque nous idéalisons les propriétés élastiques de notre matériau avec la description hyperélastique, nous dérivons tout simplement W par rapport à la déformation \mathbf{E} pour finalement obtenir une description explicite du second tenseur des contraintes de Piola-Kirchhoff qui est exprimé comme suit :

$$\mathbf{S} = \lambda \text{tr}(\mathbf{E}) \mathbf{I} + 2\mu \mathbf{E} \quad (2.21)$$

avec \mathbf{I} la matrice identité. L'équation précédente est une généralisation de la loi de Hooke pour un matériau linéaire, isotrope et incompressible, ce qui est exactement ce que nous utiliserons durant nos simulations.

2.4 Élastodynamique

Maintenant que les mesures de déformation et des contraintes d'un objet en coordonnées matérielles et spatiales sont définies, la prochaine étape consiste à présenter les équations d'équilibres du système. L'équation d'équilibre du mouvement se pose comme suit

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{r} = \rho \ddot{\mathbf{u}} \quad (2.22)$$

où $\boldsymbol{\sigma}$ est le tenseur des contraintes de Cauchy, ρ est la masse volumique du matériau et \mathbf{r} est un vecteur de force volumique (par exemple, la force gravitationnelle).

L'équation d'équilibre du moment angulaire se réduit simplement à

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad (2.23)$$

et vient renforcer la symétrie du tenseur des contraintes de Cauchy et de Piola-Kirchhoff.

Nous verrons dans les chapitres 4 et 5 les méthodes numériques pour intégrer l'équation 2.22 dans le temps.

Nous pouvons également ignorer la variable de temps et ainsi obtenir le système d'équations différentielles suivant :

$$-\nabla \cdot \boldsymbol{\sigma} = \mathbf{r} \quad (2.24)$$

$$\Updownarrow$$

$$-\left(\frac{\partial \sigma_{00}}{\partial x} + \frac{\partial \sigma_{01}}{\partial y} + \frac{\partial \sigma_{02}}{\partial z}\right) = r_1 \quad (2.25)$$

$$-\left(\frac{\partial \sigma_{10}}{\partial x} + \frac{\partial \sigma_{11}}{\partial y} + \frac{\partial \sigma_{12}}{\partial z}\right) = r_2 \quad (2.26)$$

$$-\left(\frac{\partial \sigma_{20}}{\partial x} + \frac{\partial \sigma_{21}}{\partial y} + \frac{\partial \sigma_{22}}{\partial z}\right) = r_3 \quad (2.27)$$

La résolution de ce système élastostatique nous permet d'obtenir les déplacements \mathbf{u} de l'objet établissant l'équilibre entre les forces élastiques et les contraintes externes du système. Si la variable de temps est requise, la résolution dans le temps du système élastodynamique nous donnera les états de l'objet simulé entre le début de la simulation et l'état d'équilibre du système.

Il est également possible de résoudre \mathbf{u} à partir d'une formulation faible de l'équation 2.24 et un processus d'intégration dans l'espace est alors requis. Les méthodes numériques associées à ce processus seront discutées au chapitre 4 et 5.

2.5 Résumé

Nous avons défini dans ce chapitre les notions de déplacement \mathbf{u} et de déformation d'un corps mou à l'aide des tenseurs de déformations linéaire $\boldsymbol{\epsilon}$ et non linéaire \mathbf{E} qui permettent de

quantifier la déformation d'un objet. Ensuite, en supposant que nos matériaux sont linéaires et isotropes, nous avons posé une relation de comportement linéaire permettant d'obtenir le tenseur de contraintes exposant la force de traction d'un point de l'espace en coordonnées matérielles (non déformées) grâce à la description de la déformation donnée par le tenseur de déformation. Finalement, le principe d'équilibre du mouvement linéaire nous offre une formulation différentielle du déplacement \mathbf{u} dans le temps et dans l'espace.

Le prochain chapitre repassera en revue les méthodes numériques fréquemment utilisées dans la littérature pour inclure ces concepts dans un contexte de simulation. Les chapitres 4 et 5 présenteront quant à eux les méthodes de résolutions dans l'espace et dans le temps que nous avons implémentées à partir des concepts théoriques discutés dans les sections précédentes.

CHAPITRE 3 REVUE DE LITTÉRATURE

Dans ce chapitre, nous passerons en revue les méthodes numériques utilisées dans le contexte de simulation de corps mous. Nous débuterons par une revue des premiers modèles déformables qui ont été proposés dans la littérature. Ensuite, nous examinerons les différents modèles de simulation des corps déformables qui ont été proposés à ce jour dans le domaine chirurgical. Nous terminerons ce chapitre par une revue des modèles sans maillage actuellement disponibles.

3.1 Introduction des premiers modèles déformables

Lasseter (1987) s'intéresse d'abord à l'extension des animations 2D par ordinateur à une scène en 3D. Les objets de cette scène ont une particularité intéressante : ils peuvent se compresser et s'étirer. Dans cette discussion, Lasseter pose les principes de base d'une animation 3D. Il impose une condition stricte à ces objets : peu importe le niveau de compression ou d'étiement, le volume d'un objet doit impérativement rester le même. Dans le cas contraire, un objet qui se déforme donnerait l'impression de rétrécir ou de grossir suivant sa transformation. Il y décrit par la suite les principes de temps d'animation, d'actions exécutées par cette dernière, des transitions entre les actions et de plusieurs autres principes de bases hérités des animations 2D.

Parallèlement, Terzopoulos introduit le terme de modèles déformables dans Terzopoulos et al. (1987). Il préconise l'idée de modèle "actif" d'animation basé sur les principes de physique faisant ainsi contraste aux modèles "passifs" basés sur la cinématique simple des objets où les objets sont statiques ou bien se déplacent suivant une interpolation paramétrique (modèles polynomial, spline et superquadrique, Barr (1984); Sederberg et Parry (1986)). En particulier, Terzopoulos cherche à représenter l'énergie potentielle des objets déformables en les soumettant à des lois simplifiées de la théorie de l'élasticité en mécanique des milieux continus. Pour ce faire, il met au point un tenseur métrique $\mathbf{G}_{ij}(\mathbf{r}(\mathbf{a})) = \frac{\partial \mathbf{r}}{\partial a_i} \cdot \frac{\partial \mathbf{r}}{\partial a_j}$ qu'il utilise pour comparer l'objet déformé à sa forme initiale initiale \mathbf{G}_{ij}^0 . Puisque l'énergie potentielle d'une configuration doit être nulle si cette dernière est identique à la configuration initiale, Terzopoulos obtient l'énergie d'une déformation en intégrant la norme des deux configurations sur le domaine :

$$\xi(\mathbf{r}) = \int_{\Omega} \|\mathbf{G} - \mathbf{G}^0\|_{\alpha}^2 d\mathbf{a} \quad (3.1)$$

Pour simplifier l'équation, l'auteur utilise une approximation de la norme avec une fonction

de poids quadrique représentant la résistance du matériau aux déformations. L'équation de la dynamique de mouvement (équation 2.22) est ainsi abordée sous une forme discrète par la méthode des différences finies sur un maillage en 2D. Ce système est finalement intégré dans le temps avec une méthode semi-implicite, dite “alternating-direction implicit” (ADI), éventuellement améliorée par Baraff et Witkin (1998).

Dans Terzopoulos et Fleischer (1988), les auteurs modifient la fonction de densité d'énergie décrite à l'équation 3.1 pour y introduire un terme visqueux (représenté par une spline). Le matériau n'est donc plus parfaitement élastique, mais plutôt viscoélastique : l'énergie potentielle de la déformation dépend non seulement du tenseur métrique entre la configuration courante et initiale, mais également de l'historique des déformations.

En suivant la seconde loi de Newton, cette approche permet de résoudre la dynamique de ces objets dans le temps.

Depuis que Terzopoulos a introduit les modèles déformables basés sur la théorie de l'élasticité en animation par ordinateur, plusieurs chercheurs se sont intéressés à l'intégration de l'équation 2.22 en développant des techniques de discrétisation du domaine déformable. On peut regrouper la majorité de ces techniques dans deux grandes catégories de discrétisation : les méthodes avec maillage où l'intégration se fait sur des éléments géométriques d'interpolation connectés entre eux (triangles, tétraèdres et hexaèdres) et ; les méthodes sans maillage où l'intégration se fait sur des espaces compacts d'approximation entourant des points (ou particules) non connectés du domaine. Ce sont ces deux types de modèles que nous analyserons plus en détail dans ce mémoire.

3.2 Simulation des corps déformables dans le domaine médical

Tel que mentionné en introduction de ce mémoire, la simulation chirurgicale présente un défi de taille : les organes doivent se déformer en suivant un comportement élastique et en présentant des propriétés de résistance à la déformation et d'incompressibilité. Les chercheurs de Cover et al. (1993) proposent d'animer un organe déformable à l'aide d'une méthode qu'ils introduisent comme un modèle de “contour actif”, appelé par plusieurs le modèle de “surface déformable”. Un contour actif est une minimisation d'énergie le long d'une spline. Soit $\mathbf{v}(s) = \{x(s), y(s)\}$ une représentation paramétrique de la surface, les auteurs décrivent l'énergie interne par la distance allouée entre les segments et la flexibilité du contour : $E_{int} = \frac{\alpha s \|v_s(s)\|^2 + \beta(s) \|v_{ss}(s)\|^2}{2}$. Les termes α et β sont alors utilisés pour caractériser le contour. Les auteurs de Cover et al. (1993) utilisent cette méthode pour implémenter un simulateur de cholécystectomie laparoscopique (ablation de la vésicule biliaire à l'aide d'une laparotomie).

Ce simulateur ne comporte cependant pas d'outil haptique avec retour de force.

Une équipe de chercheurs au Georgia Institute of Technology fut parmi la première à implémenter un simulateur chirurgical avec retour de force haptique. Dans Sinclair et al. (1995), ces chercheurs décrivent un simulateur de la chirurgie des yeux où une image créée par ordinateur d'un oeil est présenté à l'utilisateur et où ce dernier peut interagir avec un stylet permettant d'obtenir la position de l'outil dans l'espace. Le stylet est connecté à trois moteurs générant un retour de force. Pour déterminer la force de retour du dispositif haptique, les auteurs utilisent une jauge de déformation accrochée sur un scalpel et effectuent des coupes dans des yeux réels. L'oeil virtuel est représenté par le modèle de surface déformable présenté par Cover et al. (1993).

La simulation basée sur les contours actifs comporte cependant un problème important : l'intérieur de l'organe n'est pas représenté. Or, ce volume est nécessaire lors de découpes ou déchirures du modèle représentant un corps en trois dimensions.

Cotin et al. (1996a) élimine ce problème en utilisant une approche similaire à Terzopoulos et al. (1987) pour mettre en place un simulateur de chirurgie laparoscopique du foie. Ces auteurs utilisent une approche classique de la méthode des éléments finis (approche avec maillage). Le volume est alors discrétisé en un ensemble de tétraèdres où une fonction d'interpolation linéaire lagrangienne est utilisée pour interpoler les déformations du volume à l'intérieur d'un élément (seules les valeurs aux sommets sont utilisées et les équations sont formulées en fonction du domaine initial). La déformation d'un point à l'intérieur d'un tétraèdre e est alors donnée par :

$$\mathbf{u}(\mathbf{x}) = \sum_{j=0}^3 b_j^e(\mathbf{x}) \mathbf{u}_j^e \quad (3.2)$$

Ici e est l'élément contenant le point \mathbf{x} , b_j^e est une fonction de base linéaire et \mathbf{u}_j^e est le déplacement du sommet j de l'élément e . La solution du problème de déformation est atteinte lorsque l'énergie potentielle du système est minimale : $\delta \xi(\mathbf{u}) = 0$. Bien que les propriétés biomécaniques d'un organe humain soient peu connues au temps de la rédaction, les auteurs estiment qu'un matériau ayant des propriétés d'élasticité linéaire est suffisant pour approximer les caractéristiques du modèle déformable. Un premier problème survient alors : la force externe appliquée par l'outil haptique est inconnue (seule la position de l'outil est disponible). Cotin et al. (1996a) propose alors, dans un premier temps, de résoudre ces forces en fonction du déplacement des sommets du modèle déformable de leurs positions P_0 dans la configuration initiale (sans déformation) vers leurs positions P^* dans un modèle déformé par l'outil haptique, puis dans un deuxième temps, d'introduire ces forces comme contraintes au

système d'équations.

Puisque le petit tenseur de déformation linéaire ϵ (équation 2.6) est utilisé, couplé du tenseur d'élasticité de Lamé (équation 2.20), l'équation d'équilibre peut alors être réécrite en isolant le terme de déformation $\mathbf{u}(\mathbf{x})$:

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{f}^e \quad (3.3)$$

où \mathbf{K} est la matrice de rigidité de l'élément et est constante dans le temps s'il n'y a pas de changement topologique ou de déformations plastiques. L'assemblage des matrices élémentaires produit une matrice globale creuse de rigidité :

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (3.4)$$

Ce système d'équations est cependant trop gros pour être utilisé dans une simulation en temps réel.

Bro-Nielsen et Cotin (1996) propose alors de réduire le temps d'exécution en effectuant une étape de pré-simulation. Les auteurs supposent que lors d'une simulation chirurgicale, le comportement des noeuds de surface est généralement suffisant puisque ce sont seulement ces noeuds qui sont visibles. Le système d'équations 3.4 est alors réduit en utilisant une technique de condensation : en assumant que la numérotation des noeuds du modèle d'éléments finis est faite de telle sorte que les noeuds de surface sont numérotés en premier, on obtient la matrice par bloc suivante :

$$\left[\begin{array}{c|c} \mathbf{K}_{ss} & \mathbf{K}_{si} \\ \hline \mathbf{K}_{is} & \mathbf{K}_{ii} \end{array} \right] \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_s \\ \mathbf{f}_i \end{bmatrix} \quad (3.5)$$

où l'indice s est associé à un noeud de surface et l'indice i à un noeud interne. Un nouveau système d'équations contenant uniquement les noeuds de surface est alors extrait :

$$\mathbf{K}_{ss}^* \mathbf{u}_s = \mathbf{f}_s^* \quad (3.6)$$

avec

$$\begin{aligned} \mathbf{K}_{ss}^* &= \mathbf{K}_{ss} - \mathbf{K}_{si} \mathbf{K}_{ii}^{-1} \mathbf{K}_{is} \\ \mathbf{f}_s^* &= \mathbf{f}_s - \mathbf{K}_{si} \mathbf{K}_{ii}^{-1} \mathbf{f}_i \end{aligned} \quad (3.7)$$

La matrice résultante sera pleine, mais puisque les auteurs prévoient l'inverser au stade de pré-simulation, cela ne présente pas d'inconvénient.

Cotin et al. (1996b) propose une autre phase de pré-calcul à la simulation, toujours dans le but d'accélérer la rapidité d'exécution. Pour chaque noeud sur la surface du maillage, un déplacement unitaire est appliqué sur ce celui-ci et le champ de déformation \mathbf{u} est résolu et enregistré dans un fichier sous la forme d'un tenseur de déformation 3x3 $[\mathbf{T}_{nk}^u]$ (pour n un noeud libre, k un noeud libre de surface et \mathbf{u} le déplacement fait sur le noeud k) pour tous les noeuds "libres" (sans contrainte) du domaine. De plus, un second tenseur 3x3 $[\mathbf{T}_k^f]$, cette fois de force, est également résolu et enregistré dans le fichier. Ainsi, à partir d'un déplacement unitaire sur un noeud de surface, deux tenseurs (un de déplacement, l'autre de force) sont obtenus et enregistrés. Il devient ensuite possible d'obtenir la force et/ou le déplacement d'une déformation à l'aide d'une simple combinaison linéaire de ces deux tenseurs. Durant la simulation, la contrainte \mathbf{u}_k^* au sommet k découlant d'une collision avec l'outil haptique virtuel est utilisée pour calculer le déplacement \mathbf{u}_n pour tout noeud n et la force \mathbf{f}_k au sommet de surface k comme suit :

$$\begin{aligned}\mathbf{u}_n &= [\mathbf{T}_{nk}^u] \frac{\mathbf{u}_k^*}{\|\mathbf{u}_k^*\|} \\ \mathbf{f}_k &= [\mathbf{T}_k^f][\mathbf{u}_k^*]\end{aligned}\tag{3.8}$$

Cotin, Stéphane and Delingette, Hervé and Ayache, Nicholas (1999) met en oeuvre cette technique en l'implémentant pour une simulation de déformation d'un foie. Dans cette simulation, l'opérateur (le chirurgien) déplace un outil haptique fournissant au logiciel la position et l'orientation de l'outil chirurgical virtuel. Lorsqu'une collision est détectée entre l'outil virtuel et le foie simulé, la position des sommets du maillage en collision est récupérée et utilisée comme contrainte de déplacements \mathbf{u}^* . Les contraintes sont imposées en utilisant la méthode des multiplicateurs de Lagrange. Le système à résoudre devient donc :

$$\left[\begin{array}{c|c} \mathbf{K} & \overline{\mathbf{K}} \\ \hline \overline{\mathbf{K}}^T & 0 \end{array} \right] \begin{bmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{u}^* \end{bmatrix}\tag{3.9}$$

où $\overline{\mathbf{K}}$ contient 1 à la i ième colonne, 0 ailleurs. Une fois le système résolu, λ_i contient l'inverse de la force nécessaire pour bouger le sommet libre \mathbf{u}_i vers \mathbf{u}_i^* .

Cotin, Stéphane and Delingette, Hervé and Ayache, Nicholas (1999) propose ensuite d'améliorer ce système en simulant des réponses non linéaires (dites "quasi non-linéaires") obtenues

empiriquement sur des morceaux cylindriques d'un cerveau de porc (Chinzei et Miller (1997)). Ainsi, les auteurs bâtissent deux fonctions polynomiales $P(d_r^L(x))$ et $Q(f^L(x))$ approximant les courbes $d_r(d_r^L(x))$ et $f(f^L(x))$ avec $d_r^L(x) = \nu \frac{rx}{h}$ la déformation linéaire, $f^L(x) = \frac{skx}{h}$ la force linéaire de la déformation linéaire, k est le module de Young, ν le coefficient de Poisson et x est le déplacement axial du cylindre. Les fonctions polynomiales P et Q approximent donc les fonctions de la force de chargement (sur le cylindre) et de la déformation radiale du cylindre, toutes deux fonctions du déplacement axial. De cette façon en posant un déplacement linéaire d'un sommet n du maillage du foie par $u_n = u_n^t + u_n^r$ (où u_n^t et u_n^r sont respectivement colinéaire et orthogonal à la direction du déplacement obtenu en 3.8) et la force résultante au sommet par $f_n = f_n^t + f_n^r$, alors le déplacement non linéaire peut être approximé par :

$$\underline{u}_n = u_n^t + P(\|u_n^r\|) \frac{u_n^r}{\|u_n^r\|} \quad (3.10)$$

et la force quasi non-linéaire peut être approximée par :

$$\underline{f}_n = Q(\|f_n^t\|) \frac{f_n^t}{\|f_n^t\|} + f_n^r \quad (3.11)$$

Grâce à ces différentes techniques de prétraitement, la simulation en temps réel du comportement élastique linéaire via la méthode des éléments finis est possible. Cependant, ces techniques ne permettent pas les changements topologiques se produisant lors de découpes ou de déchirures chirurgicales, car de tels changements invalident l'inversion de la matrice de rigidité ainsi que les tenseurs de déplacements et de forces précalculées.

Cotin et al. (1998) utilise également la méthode décrite par Bro-Nielsen et Cotin (1996) et, Cotin et al. (1996b) mais l'intègre dans un système de résolution dynamique implicite dans le temps en ajoutant un terme de masse et d'amortissement dans l'équation, contrairement aux méthodes précédentes qui utilisaient un système semi-statique (les positions au temps $t + 1$ sont les solutions du système au temps t). Cette nouvelle méthode, dite méthode de **masse-tenseur**, permet les changements topologiques. Son nom découle de la similitude avec le modèle de masse-ressort, largement utilisé dans les premiers simulateurs chirurgicaux.

Les auteurs partent de la méthode des éléments finis avec des éléments linéaires de Lagrange (P1) tétraédrique pour obtenir une formulation de la force locale sur un sommet du maillage. Ainsi, le déplacement d'un point \mathbf{x} à l'intérieur du tétraèdre T_i est interpolé à l'aide des coordonnées barycentriques $b_j(\mathbf{x})$ linéaires, fonction des quatre sommets $T_i(j)$, $j = 0, \dots, 3$ du tétraèdre i :

$$\mathbf{u}_{T_i}(\mathbf{x}) = \sum_{j=0}^3 b_j^i(\mathbf{x}) \mathbf{u}_j \quad (3.12)$$

À l'aide de cette fonction de forme et de la dérivation de la fonction de densité d'énergie découlant de la loi de Hooke généralisée avec les coefficients de Lamé (2.21), la force sur un sommet i devient :

$$\mathbf{F}_i = \left[\sum_{j \in \text{TetraVoisins}(P_i)} [\mathbf{K}_{ii}^{T_j}] \right] \mathbf{u}_i + \sum_{j \in \text{SommetsVoisins}(P_i)} \left(\left[\sum_{j \in \text{TetraVoisins}(P_i)} [\mathbf{K}_{ii}^{T_j}] \right] \mathbf{u}_j \right) \quad (3.13)$$

avec

$$[\mathbf{K}_{jk}^{T_i}] = \frac{1}{36V(T_i)} (\lambda_i \mathbf{M}_k \mathbf{M}_j^T + \mu_i \mathbf{M}_j \mathbf{M}_k^T + \mu_i (\mathbf{M}_j \mathbf{M}_k) [\mathbf{I}]) \quad (3.14)$$

$$\mathbf{M}_j = \mathbf{P}_{T_i(j+1)}^0 \wedge \mathbf{P}_{T_i(j+2)}^0 + \mathbf{P}_{T_i(j+2)}^0 \wedge \mathbf{P}_{T_i(j+3)}^0 + \mathbf{P}_{T_i(j+3)}^0 \wedge \mathbf{P}_{T_i(j+1)}^0$$

et où $V(T_i)$ est le volume du tétraèdre T_i .

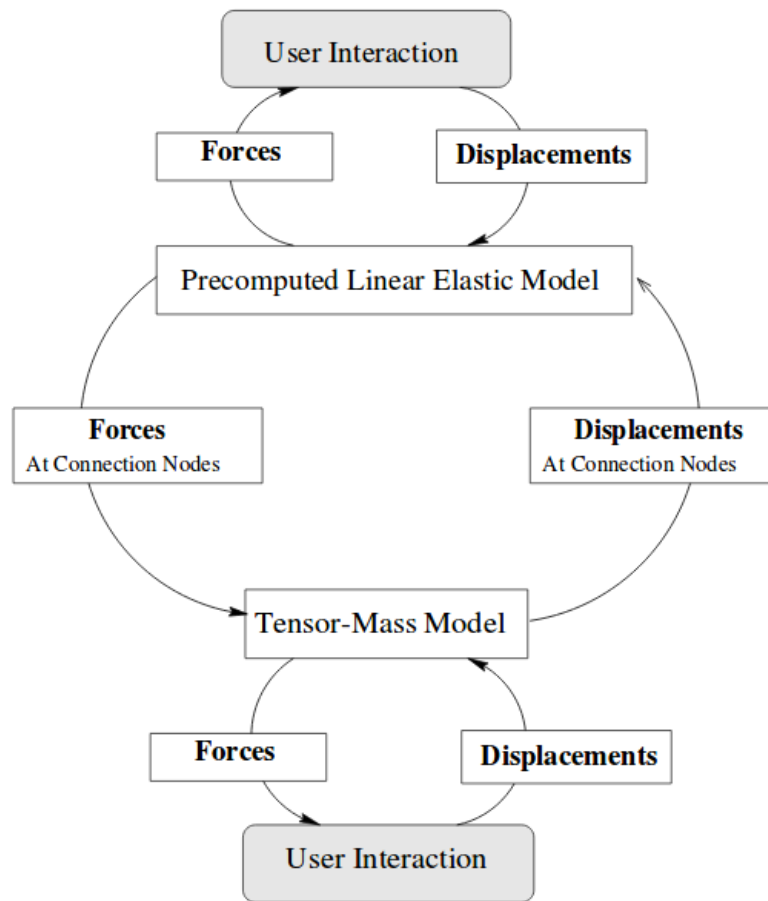
Le système est intégré dans le temps à l'aide d'une représentation explicite de l'équation différentielle (la méthode de Runge-Kutta). Lors d'un changement topologique, seules les matrices élémentaires \mathbf{K}^{T_i} autour de la région affectée doivent être modifiées, ajoutées ou supprimées.

Les auteurs de Cotin et al. (1998) proposent finalement un modèle hybride. Puisque la méthode quasi statique est beaucoup plus efficiente que la méthode de masse-tenseur, mais ne permet pas de changements topologiques, elle sera utilisée sur l'ensemble du domaine où aucune découpe ne peut avoir lieu, et la méthode dynamique de masse-tenseur sera utilisée sur le domaine critique qui recevra les découpes.

L'interaction entre les deux modèles est montrée à la figure 3.1.

Jusqu'à présent, seule l'approximation du tenseur de déformation non linéaire de Green vers le petit tenseur de déformation linéaire ϵ est utilisée. Cette approximation apporte deux limitations importantes aux simulations : le tenseur représente mal les déformations qui dépassent 10 % de la taille du maillage ; et les rotations du maillage peuvent générer de l'énergie de déformation superflue (force fantômes, Cotin, Stéphane and Delingette, Hervé and Ayache, Nicholas (1999)). Dans certains cas, la première limitation peut être ignorée en appliquant l'hypothèse de petites déformations : la résistance offerte à l'utilisateur de l'outil haptique augmente proportionnellement à l'importance de la déformation (Cotin et al.

Figure 3.1 Interaction du modèle hybride masse-tenseur et semi-statique, Cotin et al. (1998)



(1998)).

Felippa (2000) propose d'ailleurs une modification aux systèmes précédents en ajoutant une étape de corotation des éléments lors du calcul des forces. Puisque le petit tenseur de déformation ϵ produit des forces fantômes lors de rotations, une décomposition polaire du gradient de déformation $\mathbf{F} = \mathbf{R}\mathbf{U}$ est effectuée pour extraire la rotation \mathbf{R} .

Dans Székely et al. (2000), les auteurs développent un cadre d'application pour la simulation laparoscopique. Dans ce papier, une méthode pour paralléliser le calcul des forces sur chaque noeud est décrite. Cette méthode tire avantage de la discrétisation de l'énergie de déformation de chaque élément en suivant le système dynamique de masse-tenseur décrit précédemment par Cotin et al. (1998).

Picinbono et al. (2003) reprend également le modèle de masse-tenseur du simulateur de chirurgie laparoscopique du foie (Cotin et al. (1998)), mais utilise cette fois le tenseur de déformation non linéaire de Green-St-Venant (équation 2.5) : $\mathbf{E} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u}^T \cdot \nabla \mathbf{u})$. L'énergie élastique devient alors le polynôme de degré quatre, fonction de $\nabla \mathbf{u}$:

$$\begin{aligned} W(\mathbf{E}) &= \frac{\lambda}{2}(\text{tr} \mathbf{E})^2 + \mu \text{tr}(\mathbf{E}^2) \\ W(\nabla \mathbf{u}) &= \frac{\lambda}{2}(\nabla \cdot \mathbf{u})^2 + \mu \|\nabla \mathbf{u}\|^2 - \frac{\mu}{2} \|\text{rot} \mathbf{u}\|^2 \\ &+ \frac{\lambda}{2}(\nabla \cdot \mathbf{u}) \|\nabla \mathbf{u}\|^2 + \frac{\lambda}{8} \|\nabla \mathbf{u}\|^4 + \mu(\nabla \mathbf{u} : \nabla \mathbf{u}^T \nabla \mathbf{u}) + \frac{\mu}{4} \|\nabla \mathbf{u}^T \cdot \nabla \mathbf{u}\|^2 \end{aligned} \quad (3.15)$$

La différence entre le petit tenseur de déformation linéaire et le tenseur de déformation non linéaire de Green-St-Venant peut être visualisée à la figure 3.2

La dernière équation 3.15 est formulée pour un matériau suivant un comportement isotrope. Picinbono et son équipe formule un terme d'énergie d'élasticité qui pénalise l'étirement du matériau dans une direction \mathbf{a}_0 donnée, rendant ainsi le matériau anisotropique, ou plutôt "transversalement isotrope" :

$$W_{\text{trans_iso}}(\mathbf{E}) = W + \left(\frac{\lambda^L - \lambda}{2} + \mu^L - \mu \right) ((\mathbf{a}_0)^T \mathbf{E} (\mathbf{a}_0)) \quad (3.16)$$

Finalement, les auteurs de Picinbono et al. (2003) introduisent une force d'incompressibilité pour éviter qu'un tétraèdre s'inverse lors de fortes contraintes externes (lorsque que le coefficient de Poisson ν tend vers 4.5 et donc que λ tend vers l'infini). Ainsi, les auteurs pénalisent les variations du volume d'un tétraèdre en ajoutant une force sur chaque sommet p en direction de la norme sa \mathbf{N}_p :

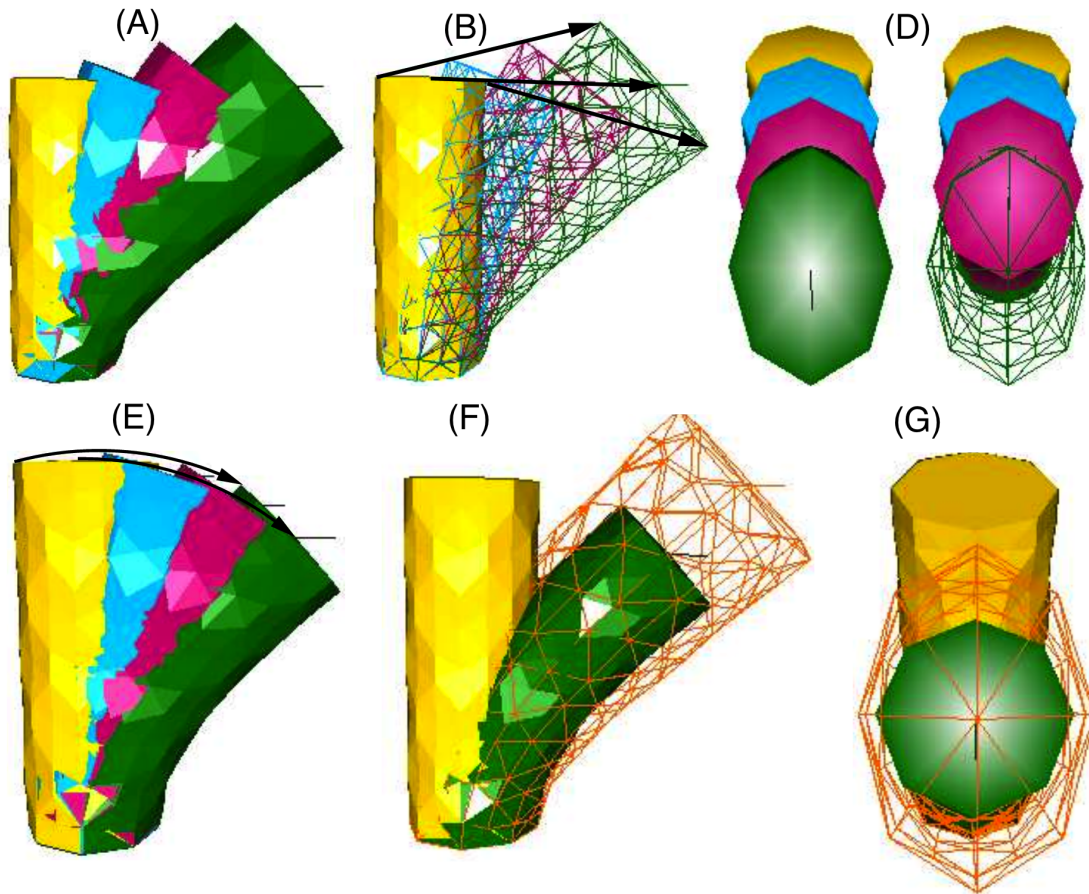


Figure 3.2 (A-D) Utilisation du petit tenseur de déformation linéaire, (E) Utilisation du tenseur de déformation non linéaire (F-G) Tenseur de déformation linéaire vs non linéaire, Picinbono et al. (2003)

$$\mathbf{F}_{\text{incomp}}^p = \left(\frac{V - V_0}{V_0}\right)^2 \mathbf{N}_p \quad (3.17)$$

Berkley et al. (2004) reprend l'idée de Bro-Nielsen et Cotin (1996) et inclut la formulation condensée de la matrice de rigidité dans un simulateur de suture en utilisant, de manière semblable à Cotin, Stéphane and Delingette, Hervé and Ayache, Nicholas (1999), des contraintes pour régir les forces de contact entre un outil haptique et le maillage de la simulation.

3.3 Modèles sans maillage

Les méthodes numériques sans maillage ont été introduites en animation par ordinateur d'une part pour pallier les problèmes causés par un mauvais maillage et, par le fait même, pour alléger la lourde tâche de vérification du maillage. Également, les méthodes sans maillage sont bien adaptées pour permettre de coupler différents modèles physiques (comportement élastique et fluide). Elles permettent aussi de facilement découper le volume initial puisqu'il n'y a pas de changements topologiques à effectuer et donc, pas de remaillage à faire. Finalement, ces méthodes permettent de remplir le domaine (ou un sous-domaine particulier) de points, ou particules, non connectés qui partagent un ensemble de propriétés (par exemple, une vitesse, une masse, un potentiel, une couleur, etc.). À partir de ces points, il devient alors possible de faire une approximation des propriétés de l'espace local compact entourant ceux-ci. Cet espace est aussi appelé le voisinage ou l'ensemble de particules voisines d'un point.

Avant même de s'intéresser au corps déformable, Reeves (1983) introduit une particule comme étant un point généré durant la simulation, pouvant se déplacer dans l'espace et qui mourra après un certain temps. Le chercheur remplit un volume de particules pour représenter visuellement diverses animations telles que du feu, des nuages ou de l'eau. La dynamique est établie selon un modèle stochastique prédéfini et les particules n'interagissent pas entre elles.

Par la suite, plusieurs équipes de chercheurs tenteront d'étendre le travail de Reeves (1983) à des comportements élastiques et fluides. Voici un survol d'une partie de ces méthodes, en particulier les méthodes permettant de modéliser un comportement élastique en temps réel ou presque réel avec une formulation lagrangienne, donc en fonction des coordonnées matérielles.

3.3.1 Dynamique des particules internes

Contrairement à Reeves (1983) qui modélise des particules sans interaction entre elles, l'animation de solides requiert une interaction entre les particules. Dans Luciani et al. (1991), les particules représentent des balles (ou des bulles) de différentes tailles et qui, selon la distance entre elles, s'attireront ou bien se repousseront entre elles. Un objet est alors approximé à l'aide d'agglomération de balles. Pour définir l'interaction entre les particules, les auteurs prennent exemple sur le calcul des forces intermoléculaires. Dans ce domaine, il est fréquent d'utiliser le potentiel de Lennard-Jones $\phi_{LJ}(r) = \frac{B}{r^n} - \frac{A}{r^m}$ pour régir le niveau d'attraction et de répulsion entre deux molécules ayant une distance r entre elles. Lorsque la distance est courte, les molécules se repoussent ; lorsque la distance est grande, les molécules s'attirent (voir figure 3.3). Les auteurs s'appuieront sur ce potentiel pour représenter les forces entre les particules de l'animation.

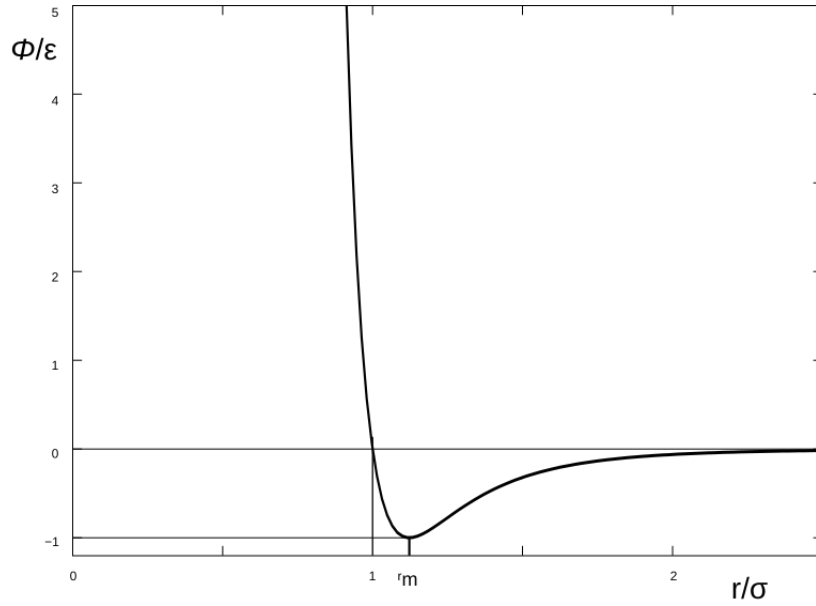


Figure 3.3 Potentiel de Lennard-Jones avec $n = 12$, $m = 6$, $\epsilon = \frac{B^2}{4A}$ et $\sigma = \sqrt[6]{\frac{A}{B}}$.

Dans Szeliski et Tonnesen (1992), les auteurs souhaitent influencer les particules à former naturellement des surfaces. Or, les particules d'un système basé sur le potentiel de Lennard-Jones ont tendance à se regrouper en amas compact pour ainsi minimiser l'énergie de leur potentiel. Les auteurs introduisent alors le concept de particules orientées où ces dernières possèdent, en plus d'une masse et d'une vitesse, un plan local (x_i, y_i, z_i) où l'axe local z_i représente le vecteur normal \mathbf{n}_i de la particule i . L'énergie potentielle de Lennard-Jones ϕ_{LJ} sera augmentée en introduisant des potentiels de coplanarité ϕ_P (équation 3.18), de co-

normalité ϕ_N (équation 3.19) et de co-circularité ϕ_C (équation 3.20) poussant les particules à former des surfaces plutôt que des volumes. Les auteurs parleront alors de particules de surface : un surfel “(surface element)”. En définissant une fonction de poids $\psi(r)$ décroissante monotone suivant la distance entre deux particules $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, ces auteurs ajoutent les potentiels suivants :

$$\phi_P(\mathbf{n}_i, \mathbf{r}_{ij}) = (\mathbf{n}_i \cdot \mathbf{r}_{ij})^2 \psi(\|\mathbf{r}_{ij}\|) \quad (3.18)$$

$$\phi_N(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}) = \|\mathbf{n}_i - \mathbf{n}_j\|^2 \psi(\|\mathbf{r}_{ij}\|) \quad (3.19)$$

$$\phi_C(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}) = ((\mathbf{n}_i + \mathbf{n}_j) \cdot \mathbf{r}_{ij})^2 \psi(\|\mathbf{r}_{ij}\|) \quad (3.20)$$

L'énergie entre deux particules est alors pondérée ainsi :

$$E_{ij} = \alpha_{LJ} \phi_{LJ}(\mathbf{r}_{ij}) + \alpha_P \phi_P(\mathbf{n}_i, \mathbf{r}_{ij}) + \alpha_N \phi_N(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}) + \alpha_C \phi_C(\mathbf{n}_i, \mathbf{n}_j, \mathbf{r}_{ij}) \quad (3.21)$$

Pour chaque particule i du système, Szeliski et Tonnesen (1992) accumuleront les forces \mathbf{f}_i et les moments τ_i , puis ils intégreront les équations de mouvement de Newton par rapport au temps en utilisant la méthode explicite d'Euler.

Bien que les forces basées sur le potentiel de Lennard-Jones ont l'avantage d'être faciles à implémenter, la paramétrisation de ce potentiel en vue d'obtenir un comportement élastique est difficile à réaliser. De plus, les particules ont tendance à osciller lorsqu'elles ont une distance très proche de l'équilibre d'attraction-répulsion. Pour cette raison, Desbrun et Gascuel (1996) se penchera sur l'interaction entre particules pour simuler des objets déformables, mais utilisera plutôt une technique basée sur la méthode “Smoothed Particle Hydrodynamics” (SPH) utilisée initialement par des astrophysiciens puis par des physiciens pour l'étude de la dynamique des fluides. Dans leur recherche, Desbrun et Gascuel (1996) présente une particule comme étant un petit volume de matériau non élastique qui se déplace dans le temps. Autour d'une particule, une fonction de lissage W_h dite “noyau” donne la distribution de sa masse sur une distance h autour d'elle. La valeur et la dérivée d'une propriété φ contenues dans les particules sont alors approximées pour un point donné $\mathbf{x} = (x, y, z)$ par :

$$\tilde{\varphi}(\mathbf{x}) = \sum_i m_i \frac{\varphi(\mathbf{x}_i)}{\rho_i} W_h(\mathbf{x} - \mathbf{x}_i) \quad (3.22)$$

$$\nabla \tilde{\varphi}(\mathbf{x}) = \sum_i m_i \frac{\varphi(\mathbf{x}_i)}{\rho_i} \nabla W_h(\mathbf{x} - \mathbf{x}_i) \quad (3.23)$$

où la somme se fait sur l'ensemble des particules du système et où m_i et ρ_i sont respectivement la masse et la densité de la particule i . En pratique, on sommerait sur un ensemble restreint de particules proche du point \mathbf{x} .

Dans le cas de la dynamique des fluides, le lissage de la densité en un point \mathbf{x} est donné par :

$$\rho(\mathbf{x}) = \sum_i m_i W_h(\mathbf{x} - \mathbf{x}_i) \quad (3.24)$$

Ici, la force \mathbf{F}_i exercée sur une particule i est décrite par :

$$\mathbf{F}_i = -m_i \sum_{j \neq i} m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_h(\mathbf{x}_i - \mathbf{x}_j) \quad (3.25)$$

où la pression initiale est obtenue pour un gaz idéal par la relation $PV = k$ avec k une constante et V le volume du fluide. Cette force produit un effet de répulsion uniquement et devra être modifiée pour l'utilisation dans une simulation d'un corps élastique. Pour produire un effet d'attraction-répulsion comme le fait le potentiel de Lennard-Jones, les auteurs remplacent la pression précédente P par la relation $P = k(\rho - \rho_0)$ avec $V = \frac{1}{\rho}$ le volume par unité de masse et $P_0 = k\rho_0$. La force exercée sur une particule devient :

$$\mathbf{F}_i = -km_i \left(\frac{\rho_i - \rho_0}{\rho_i^2} \sum_{j \neq i} m_j \nabla_i W_h(\mathbf{x}_i - \mathbf{x}_j) + \sum_{j \neq i} m_j \frac{\rho_j - \rho_0}{\rho_j^2} \nabla_i W_h(\mathbf{x}_i - \mathbf{x}_j) \right) \quad (3.26)$$

De cette manière, une particule aura tendance à vouloir reprendre son volume initial. La paramétrisation de la rigidité du volume est faite avec la constante k . Les auteurs ajouteront une propriété de viscosité aux particules à l'aide d'une force d'amortissement.

Les auteurs définissent ensuite la fonction de noyau se rapprochant d'une spline gaussienne. Contrairement au noyau de la spline gaussienne, celle-ci est mieux adaptée pour représenter la force d'attraction-répulsion :

$$W_h(r) = \frac{15}{\pi(4h)^3} \begin{cases} (2 - \frac{r}{h})^3 & \text{si } 0 \leq r \leq 2h \\ 0 & \text{si } r > 2h \end{cases} \quad (3.27)$$

L'introduction d'une méthode SPH par Desbrun et Gascuel (1996) constitue un pas important dans le domaine des déformations élastiques. Müller et al. (2004) poussera un peu plus loin

cette idée en y introduisant cette fois-ci la théorie de l'élasticité linéaire utilisée alors sur des éléments finis connectés entre eux. Pour évaluer le gradient de déplacement $\nabla \mathbf{u}_i$ de la particule i , Müller et ses coéquipiers utilisent une approximation MLS ("Moving Least Squares") de Lancaster et Salkauskas (1981) qui consiste à minimiser l'erreur d'une estimation de premier degré d'un développement de Taylor du déplacement \mathbf{u}_j des particules voisines j : $\tilde{\mathbf{u}}_j = \mathbf{u}_i + \nabla \mathbf{u}_i \cdot (\mathbf{x}_j - \mathbf{x}_i) + O(\|(\mathbf{x}_j - \mathbf{x}_i)\|)$. Ils obtiennent l'approximation suivante :

$$\nabla \mathbf{u}_i = \mathbf{A}^{-1} \left(\sum_j (\mathbf{u}_j - \mathbf{u}_i)(\mathbf{x}_j - \mathbf{x}_i) W_h(\|\mathbf{x}_j - \mathbf{x}_i\|) \right) \quad (3.28)$$

où $\mathbf{A} = \sum_j ((\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T W_h(\|\mathbf{x}_j - \mathbf{x}_i\|))$ est la matrice dite de moment et peut être précalculée et préinversée avant la simulation. Une attention particulière doit être portée au placement initial des particules puisque si une particule contient moins de quatre voisines dans son support h , ou bien si la configuration est coplanaire ou colinéaire, la matrice \mathbf{A} sera singulière. Pour éviter tout problème relié à une matrice de moment mal conditionnée, les auteurs utilisent une décomposition SVD ("Singular Value Decomposition") de celle-ci pour l'inversion.

Müller et al. (2004) utilise enfin un noyau polynomial proposé dans Müller et al. (2003) qui est décrit comme suit :

$$W_h(r) = \frac{15}{\pi(4h)^3} \begin{cases} \frac{315}{64\pi h^9} (h^2 - r^2)^3 & \text{si } 0 \leq r \leq h \\ 0 & \text{si } r > h \end{cases} \quad (3.29)$$

Les particules avoisinantes sont trouvées par une grille de hachage introduite dans Teschner et al. (2003). À partir du moment où on a une estimation de $\nabla \mathbf{u}_i$, on peut trouver les forces aux particules de manière semblable aux sommets des éléments d'un système FEM. Pour ce faire, le tenseur quadratique de Green-Saint-Venant mesure la déformation $\boldsymbol{\epsilon}_i = \mathbf{J}_i^T \mathbf{J}_i - \mathbf{I}$ où $\mathbf{J}_i = \nabla \mathbf{x}_i^0 + \nabla \mathbf{u}_i^T = \mathbf{I} + \nabla \mathbf{u}_i^T$ est le Jacobien du déplacement $\mathbf{x}_i = \mathbf{x}_i^0 + \mathbf{u}_i$. On obtient $\boldsymbol{\epsilon}_i = \nabla \mathbf{u}_i + \nabla \mathbf{u}_i^T + \nabla \mathbf{u}_i \nabla \mathbf{u}_i^T$ et la relation de comportement linéaire Hookéenne est utilisée $\boldsymbol{\sigma}_i = \mathbf{C} : \boldsymbol{\epsilon}_i$ avec \mathbf{C} le tenseur d'élasticité linéaire de Lamé (voir 2.20). Les auteurs posent alors pour une particule i de volume v_i : $\mathbf{F}_e = -2v_i \mathbf{J}_i \boldsymbol{\sigma}_i$ la force élastique et $\mathbf{F}_v = -v_i k_v (\|\mathbf{J}\| - 1) \left[(\mathbf{J}_v \mathbf{X} \mathbf{J}_w)^T (\mathbf{J}_w \mathbf{X} \mathbf{J}_u)^T (\mathbf{J}_u \mathbf{X} \mathbf{J}_v)^T \right]^T$ la force de conservation du volume (pour rétablir les déformations inverses au Jacobien négatif). Ils obtiennent alors les forces internes pour une particule i et ses voisines j :

$$\begin{aligned}\mathbf{f}_i &= (\mathbf{F}_e + \mathbf{F}_v) \mathbf{A}^{-1} \left(- \sum_j (\mathbf{x}_j - \mathbf{x}_i) W_h(\|\mathbf{x}_j - \mathbf{x}_i\|) \right) \\ \mathbf{f}_j &= (\mathbf{F}_e + \mathbf{F}_v) \mathbf{A}^{-1} ((\mathbf{x}_j - \mathbf{x}_i) W_h(\|\mathbf{x}_j - \mathbf{x}_i\|))\end{aligned}\tag{3.30}$$

En terminant, Müller et al. (2004) intègre explicitement la dynamique newtonienne.

$$\frac{\partial^2 \mathbf{u}_i}{\partial t^2} = \frac{\mathbf{f}_i}{\rho_i}\tag{3.31}$$

dans le temps à l'aide de la méthode “Leap Frog” pour un pas de temps Δt suffisamment petit. L'approche de Müller et al. (2004) sera d'ailleurs utilisée dans le chapitre sur les méthodes sans maillage (chapitre 5) pour développer une des méthodes que nous avons implémentées dans ce projet.

Keiser et al. (2005) se basera sur le travail de Müller et al. (2004) pour augmenter le modèle d'élasticité linéaire en incorporant les équations de Navier-Stokes utilisées en dynamique des fluides à l'aide de la méthode SPH. Ceci leur permettra de fusionner ou de solidifier des objets de la simulation en temps réel. Les équations de conservation de moment et de masse pour un fluide newtonien incompressible sont posées par :

$$\begin{aligned}\rho \frac{Dv}{Dt} &= \nabla \sigma^f(v) + f \\ \frac{\partial \rho}{\partial t} + \nabla(\rho v) &= 0\end{aligned}\tag{3.32}$$

Puisqu'une approche Lagrangienne est utilisée et que chaque particule contient une masse constante m_i et se déplace avec le fluide, l'équation de conservation de masse est ignorée et l'équation 3.32 devient

$$\rho \frac{\partial^2 u}{\partial t^2} = \nabla \sigma(v) + f\tag{3.33}$$

La contrainte d'élasticité de Müller et al. (2004) est ajoutée à l'équation 3.33 pour obtenir

$$\rho \frac{\partial^2 u}{\partial t^2} = \nabla \sigma(u, v) + f\tag{3.34}$$

où $\sigma(u, v)$ cumule les contraintes élastiques, visqueuses et de pression. Les auteurs utilisent la méthode SPH et les noyaux définis dans Müller et al. (2003) pour calculer les forces

visqueuses, de pression et de tension, respectivement pour une particule i aux voisins j :

$$\begin{aligned}\mathbf{f}_i^v &= \sum_j m_j \frac{\mu_i + \mu_j}{2} \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W_h^v(\|\mathbf{x}_j - \mathbf{x}_i\|) \\ \mathbf{f}_i^p &= - \sum_j m_j \frac{\Phi_i + \Phi_j}{2\rho_j} \nabla^2 W_h^p(\|\mathbf{x}_j - \mathbf{x}_i\|) \\ \mathbf{f}_i^t &= -k^t \nabla^2 c(\mathbf{x}_i) \frac{\nabla c(\mathbf{x}_i)}{\|\nabla c(\mathbf{x}_i)\|}, \quad c(\mathbf{x}) = \sum_j m_j \frac{1}{\rho_j} W_h^t(\|\mathbf{x} - \mathbf{x}_j\|)\end{aligned}$$

où μ_i est le coefficient de viscosité, $\Phi_i = k^g(\rho_i - \rho_{i0})$ la pression scalaire, k^g une constante pour un gaz idéal, k^t une constante du matériau et ρ_{i0} la densité de masse initiale. En reprenant la force élastique calculée par la méthode MLS de Müller et al. (2004) (équation 3.30), les auteurs de Keiser et al. (2005) intégreront la dynamique newtonienne dans le temps (l'équation 3.31) avec $\mathbf{f}_i = \mathbf{f}_i^v + \mathbf{f}_i^p + \mathbf{f}_i^t + \mathbf{f}_i^e$ la somme des forces internes de la particule i . Tout comme Müller et al. (2004), la méthode explicite de Leap Frog est utilisée. En ajoutant l'équation de transfert de chaleur, les auteurs interpolent les coefficients de rigidité, de viscosité et de tension pour une valeur de température minimum à une température maximum pour ainsi créer des transitions entre la fusion et la solidification d'un matériau. Idem à Müller et al. (2004), les particules avoisinantes sont trouvées à l'aide de la grille de hachage de Teschner et al. (2003).

Solenthaler et al. (2007) reprend le concept précédent de Müller et al. (2004) et Keiser et al. (2005), mais utilise la méthode SPH plutôt que MLS pour estimer $\nabla \mathbf{u}_i$ éliminant ainsi les contraintes de positionnement des particules (configuration étendue, colinéaire ou coplanaire). Puisque la méthode SPH à une consistance de niveau zéro (la notion de consistance sera développée à la section 5.1 du chapitre sur les méthodes sans maillage), leur méthode introduit des forces fantômes lors des déformations rotationnelles.

Peu importe la méthode d'approximation du tenseur gradient du déplacement $\nabla \mathbf{u}_i$ d'une particule i , les auteurs de Müller et al. (2004), Keiser et al. (2005) et Solenthaler et al. (2007) ont toujours utilisé le tenseur de déformation non linéaire de Green-Saint-Venant pour mesurer la déformation d'une configuration. Becker et al. (2009) propose plutôt d'utiliser le petit tenseur de déformation linéaire (voir l'équation 2.6) en adaptant la technique de corotation des modèles FEM proposée par Felippa (2000) pour ainsi éliminer les forces fantômes résultant d'une déformation rotationnelle. Cette amélioration sera également détaillée dans le chapitre 5 puisqu'il s'agit d'une des méthodes que nous avons implémentées.

Ces travaux, appuyés sur Müller et al. (2004), forment une branche des méthodes sans

maillage nommée animation basée sur les points. Cette branche d’animation de comportement élastique fut récemment réintroduite dans la littérature pour l’utilisation dans des simulations chirurgicales. Nous pouvons citer notamment les travaux suivants : Zhu et al. (2010); Hao et Huang (2011); Shrivastava et Das (2014); Luo et al. (2015); Peng et al. (2017).

Finalemt, Horton et al. (2010) propose une méthode d’approximation MLS semblable à Müller et al. (2004), mais utilise un maillage pour l’intégration numérique du système d’équations à résoudre. Ainsi, sa méthode nommée “Meshless Total Lagragian Explicit Dynamics” (MTLED) se trouve entre la méthode complètement sans élément de Müller et al. (2004) et la méthode des éléments finis. Le système est par la suite intégré dans le temps par la méthode des différences centrées (méthode d’intégration temporelle basée sur la méthode d’Euler explicite). La méthode est ensuite appliquée à une simulation de la déformation d’un cerveau, cependant la simulation est trop lente pour du temps réel dû au trop grand nombre de DDLs et de points d’intégration utilisé.

Joldes et al. (2010) utilisera la méthode de Horton et al. (2010) mais accélèrera la rapidité des calculs en les parallélisant sur processeurs graphiques. La simulation n’est cependant toujours pas assez rapide pour du temps réel.

Comme nous le verrons au chapitre 5, le modèle MTLED de Horton et al. (2010) est une des méthodes sans maillage que nous avons choisi d’analyser et d’implémenter.

CHAPITRE 4 MÉTHODE DES ÉLÉMENTS FINIS

Au chapitre 2 nous avons présenté un résumé de la théorie de l'élasticité linéaire en mécanique des milieux continus. Nous avons entre autres présenté une formulation du principe d'équilibre du moment linéaire qui décrit le déplacement $\mathbf{u}(\mathbf{X})$ d'une configuration initiale d'un objet à sa configuration déformée. Pour calculer ce déplacement, un système d'équations impliquant des dérivées partielles doit être résolu.

$$-\nabla \cdot \boldsymbol{\sigma} = \mathbf{r} \quad (4.1)$$

Ce système d'équations différentielles correspond à la *formulation forte* du problème du système élastostatique qui nous intéresse.

Pour faciliter la résolution de ce système d'équations, une formulation faible est souhaitable. C'est cette approche qui est à la base de la méthode des éléments finis et du modèle FEM que nous avons implémenté et que nous décrivons dans ce chapitre.

4.1 Formulation faible

La formulation faible de notre problème contient un degré différentiel de moins que la version forte et fait ressortir des termes aux frontières du domaine qui nous permettent d'appliquer des conditions limites en déplacement \mathbf{u} . Nous pouvons démontrer que la solution au problème faible peut être également solution du problème fort (Fortin et Garon (2011)).

Pour obtenir la *formulation faible*, nous multiplions la formulation forte (équation 4.1) par une fonction vectorielle \mathbf{w} quelconque comme suit :

$$-\mathbf{w} \nabla \cdot \boldsymbol{\sigma} = \mathbf{w} \mathbf{r} \quad (4.2)$$

En intégrant cette équation par parties et en tirant avantage de la symétrie du tenseur des contraintes $\boldsymbol{\sigma}$ (voir également le théorème A.7 en annexe A de Fortin et Garon (2011)), on trouve la formulation faible du problème

$$\int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon}(\mathbf{w}) dv - \int_{\Gamma} (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{w} da = \int_{\Omega} \mathbf{r} \cdot \mathbf{w} dv \quad (4.3)$$

où Γ est la surface aux frontières de notre domaine Ω , da est un élément d'aire de la surface,

dv est un élément de volume de l'objet et \mathbf{n} est la normale.

Si le terme de temps doit être considéré, le même procédé peut être appliqué à l'équation de mouvement de Cauchy (équation 2.22) pour obtenir la formulation faible suivante :

$$\int_{\Omega} \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} dv + \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon}(\mathbf{w}) dv = \int_{\Gamma} (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{w} da + \int_{\Omega} \mathbf{r} \cdot \mathbf{w} dv \quad (4.4)$$

Pour simplifier notre analyse, les trois termes de l'équation précédente peuvent être décomposés comme suit :

$$\dot{\Pi}_{cin} = \int_{\Omega} \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} dv \quad (4.5)$$

$$\dot{\Pi}_{int} = \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon}(\mathbf{w}) dv \quad (4.6)$$

$$\dot{\Pi}_{ext} = \int_{\Gamma} (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{w} da + \int_{\Omega} \mathbf{r} \cdot \mathbf{w} dv \quad (4.7)$$

Le premier terme correspond au changement d'énergie cinétique du système. Le second terme est lié au taux de changement de l'énergie interne élastique de déformation. Finalement, le troisième correspond à l'effort effectué par la charge externe volumique et de surface. L'équilibre de l'énergie totale du système peut alors être formulé plus simplement par

$$\dot{\Pi}_{cin} + \dot{\Pi}_{int} = \dot{\Pi}_{ext} \quad (4.8)$$

Pour la formulation statique, nous avons simplement

$$\dot{\Pi}_{int} = \dot{\Pi}_{ext} \quad (4.9)$$

Les dernières équations 4.8 et 4.9 sont formulées sur la configuration courante (eulérienne), c'est-à-dire qu'elles font intervenir les coordonnées spatiales sur le volume déformé. Comme nous l'avons mentionné au chapitre 2, la définition explicite de cette géométrie n'est malheureusement pas connue dans nos simulations. Nous devons alors faire intervenir le second tenseur des contraintes de Piola-Kirchoff $\mathbf{S} = J \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}$ où \mathbf{F} est le tenseur de déformation introduit à l'équation 2.3 et J son déterminant. Nous définissons également les quantités suivantes à partir de la formule de Nanson (Fortin et Garon (2011)) :

$$d\mathbf{a} = J\mathbf{F}^{-T} \cdot d\mathbf{A} \quad (4.10)$$

$$\mathbf{n}da = J\mathbf{F}^{-T} \cdot \mathbf{N}dA \quad (4.11)$$

$$dv = JdV \quad (4.12)$$

$$\rho = \frac{\rho_0}{J} \quad (4.13)$$

$$\boldsymbol{\sigma} = \frac{1}{J}\mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T \quad (4.14)$$

Finalement, dans le cas de matériaux hyperélastiques et isotropes, il est possible de démontrer que les termes de l'équation 4.8 sont formulés en coordonnées matérielles comme ceci :

$$\dot{\Pi}_{cin} = \int_{\Omega^0} \rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} dV \quad (4.15)$$

$$\dot{\Pi}_{int} = \int_{\Omega^0} \mathbf{S} : \boldsymbol{\epsilon}(\mathbf{w}) dV \quad (4.16)$$

$$\dot{\Pi}_{ext} = \int_{\Gamma^0} (J\boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathbf{N}) \cdot \mathbf{w} dA + \int_{\Omega^0} \mathbf{r} \cdot \mathbf{w} J dV \quad (4.17)$$

4.2 Discrétisation du domaine et intégration numérique

Cette formulation faible du problème d'élasticité nous permet de résoudre le déplacement de notre objet lorsqu'il est soumis à des forces externes en respectant des contraintes aux bords du domaine, tout cela en coordonnées matérielles. Malheureusement, les équations 4.15, 4.16 et 4.17 font intervenir une intégration matérielle, c'est-à-dire sur le domaine non déformé. Or, nous n'avons toujours pas une représentation explicite de la géométrie de l'organe simulé (surface et volume). On s'intéresse alors aux méthodes d'intégration numérique et à la discrétisation matérielle de notre organe.

À l'aide des outils modernes de la médecine, il est maintenant possible d'obtenir des images en deux ou trois dimensions des organes du corps humain. Il y a actuellement tout un domaine de recherche sur la reconstruction d'un organe en un nuage de points 3D (voir figure 4.1b). Puis, une reconstruction de la surface en un maillage de triangle est effectuée (voir figure 4.1c). Finalement, le volume de l'organe est discrétisé en une suite d'éléments volumiques où chaque élément partage ses sommets avec les éléments voisins (voir figure 4.1d).

Pour notre étude, nous partirons plutôt d'un maillage d'éléments tétraédriques et hexaédriques déjà calculé. Il est important de noter que ces étapes de reconstruction et de maillage

volumique constituent un problème très complexe et entraînent une source d'erreurs numériques très importante visant directement nos contraintes de précision et de stabilité énoncées à la section 1.2. Ainsi, lorsque des images 2D ou 3D d'un patient sont obtenues, la tâche de reconstruction demande l'analyse d'un maillage 3D souvent faite par des ingénieurs spécialisés dans ce domaine. Cette complexité est l'une des grandes motivations des méthodes sans maillage décrites au chapitre 5.

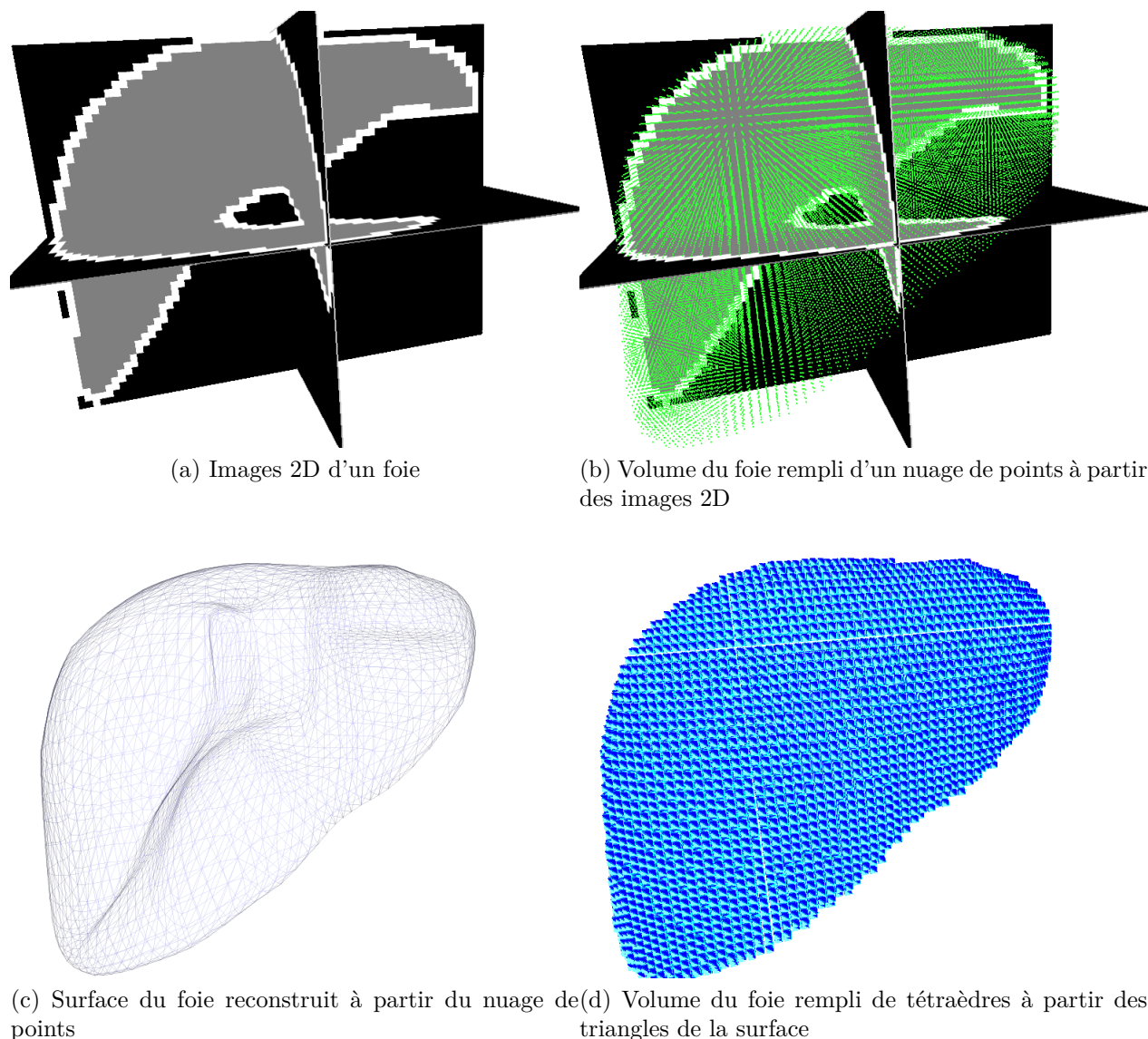


Figure 4.1 Reconstruction 3D d'un foie à partir d'images 2D d'un scanner

Avec l'approche des éléments finis, nous supposons donc que notre ensemble d'éléments volumiques est connu et remplit notre organe virtuel. À partir de cet ensemble, nous intégrons le volume initial en additionnant la valeur de l'intégration du volume de l'élément, et ce, pour

chaque élément de notre ensemble. Par exemple, pour un matériau hyperélastique linéaire, l'intégrale du terme d'énergie interne d'élasticité de l'équation 4.16 devient :

$$\dot{\Pi}_{int}(\mathbf{X}) = \int_{\Omega^0} \mathbf{S} : \boldsymbol{\epsilon} dV \quad (4.18)$$

$$= \sum_{e=1}^{n_e} \int_{\Omega_e^0} \mathbf{S}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{w}) dV_e \quad (4.19)$$

$$= \sum_{e=1}^{n_e} \int_{\Omega_e^0} (\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u})) : \boldsymbol{\epsilon}(\mathbf{w}) dV_e \quad (4.20)$$

L'addition doit être faite sur chacun des n_e éléments volumiques contenus dans notre domaine discrétisé.

Connaissant la géométrie des tétraèdres et des hexaèdres, nous pouvons ensuite intégrer numériquement les termes de notre équation à l'aide d'une règle de quadrature. Ces règles sont détaillées à l'annexe A. Pour nos simulations, nous utiliserons un seul degré de précision pour la quadrature, c'est-à-dire que nous placerons un seul point d'intégration au centre de l'élément. La valeur de l'intégrande sera évaluée à la position de ce point d'intégration en pondérant en fonction du volume de l'élément

$$\int_{\Omega_e^0} f(X, Y, Z) dV_e = V_e \sum_{i=1}^n W_i f(\xi_1^i, \xi_2^i, \xi_3^i, \xi_4^i) \quad (4.21)$$

Ici V_e est le volume de l'élément, n est le nombre de points d'intégration contenu dans l'élément, W_i est le poids du point i et $(\xi_1^i, \xi_2^i, \xi_3^i, \xi_4^i)$ sont les coordonnées élémentaires du point d'intégration i dans l'élément. Pour un tétraèdre ayant pour sommet \mathbf{P}_1 , \mathbf{P}_2 , \mathbf{P}_3 et \mathbf{P}_4 , la relation entre la position du point d'intégration (X_i, Y_i, Z_i) dans l'espace matériel Ω^0 et $(\xi_1^i, \xi_2^i, \xi_3^i, \xi_4^i)$ dans l'espace élémentaire Ω_e est la suivante :

$$(X_i, Y_i, Z_i)^T = \xi_1^i \mathbf{P}_1 + \xi_2^i \mathbf{P}_2 + \xi_3^i \mathbf{P}_3 + \xi_4^i \mathbf{P}_4 \quad (4.22)$$

Dans notre cas, puisque nous utilisons qu'un seul point d'intégration au centre de l'élément, nous prenons $n = 1$, $\xi_1 = \xi_2 = \xi_3 = \xi_4 = \frac{1}{4}$ et $W = 1$ pour ainsi obtenir :

$$\dot{\Pi}_{int}(\mathbf{X}) = \sum_{e=1}^{n_e} V_e (\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u}(\mathbf{X}_I^e))) : \boldsymbol{\epsilon}(\mathbf{w}) \quad (4.23)$$

pour $\mathbf{X}_I^e = \frac{1}{4}(\mathbf{P}_1^e + \mathbf{P}_2^e + \mathbf{P}_3^e + \mathbf{P}_4^e)$ le vecteur position du point d'intégration de l'élément e . Nous utilisons la même procédure pour intégrer dans l'espace les termes $\dot{\Pi}_{cin}$ et $\dot{\Pi}_{ext}$. Pour les hexaèdres, en choisissant un sommet de l'élément, le point d'intégration se trouve à l'intersection du milieu de ses trois arêtes.

4.3 Interpolation et fonction de forme

La méthode de la section précédente nous donne une méthode pour intégrer numériquement le volume initial de notre organe. Il reste maintenant à trouver une définition explicite du déplacement $\mathbf{u}(\mathbf{X}_I^e)$. Pour ce faire, nous définissons les coordonnées barycentriques $[\phi_0 \ \phi_1 \ \phi_2 \ \phi_3]^T$ d'un point \mathbf{x} dans un tétraèdre e en fonction de la position de ses 4 sommets \mathbf{P}_i^e ainsi :

$$\phi_j = -\frac{\mathbf{M}_j \cdot (\mathbf{x} - \mathbf{P}_{(j+1)}^e)}{6V_e} \quad (4.24)$$

$$\mathbf{M}_j = \mathbf{P}_{(j+1)}^e \wedge \mathbf{P}_{(j+2)}^e + \mathbf{P}_{(j+2)}^e \wedge \mathbf{P}_{(j+3)}^e + \mathbf{P}_{(j+3)}^e \wedge \mathbf{P}_{(j+1)}^e \quad (4.25)$$

Ces coordonnées barycentriques sont utilisées pour interpoler linéairement le déplacement d'un point \mathbf{x} dans le tétraèdre. Ses coordonnées sont appelées les **fonctions de forme** de l'élément e . Ainsi la valeur $\mathbf{u}(\mathbf{x})$ devient :

$$\mathbf{u}(\mathbf{x}) = \sum_{j=0}^3 \phi_j \mathbf{u}_j^e \quad (4.26)$$

où \mathbf{u}_j^e est le déplacement du sommet j de l'élément e contenant le point \mathbf{x} .

Le tenseur gradient du déplacement $\nabla \mathbf{u}(\mathbf{x})$ quant à lui est obtenu simplement en dérivant la fonction de forme par rapport à x , y et z :

$$\nabla \mathbf{u}(\mathbf{x}) = \sum_{j=0}^3 \nabla \phi_j \mathbf{u}_j^e \quad (4.27)$$

$$= \sum_{j=0}^3 \begin{bmatrix} \phi_{j,x} u_j^e & \phi_{j,y} u_j^e & \phi_{j,z} u_j^e \\ \phi_{j,x} v_j^e & \phi_{j,y} v_j^e & \phi_{j,z} v_j^e \\ \phi_{j,x} w_j^e & \phi_{j,y} w_j^e & \phi_{j,z} w_j^e \end{bmatrix} \quad (4.28)$$

$$= \sum_{j=0}^3 \begin{pmatrix} u_j^e \\ v_j^e \\ w_j^e \end{pmatrix} [\phi_{j,x} \ \phi_{j,y} \ \phi_{j,z}] \quad (4.29)$$

$$(4.30)$$

Puisque la fonction d'interpolation (fonction de forme) ϕ est linéaire, il s'en suit que ses dérivées sont constantes et donc $\nabla \phi$ peut être précalculée pour chaque élément au début de la simulation.

4.4 Simulation semi-statique

La formulation faible de l'équation aux dérivées partielles d'équilibre du mouvement décrite dans les dernières sections propose trois termes d'énergie, soit $\dot{\Pi}_{kin}$, $\dot{\Pi}_{int}$ et $\dot{\Pi}_{ext}$ (équations 4.15, 4.16 et 4.17) en fonction du déplacement

$\mathbf{u}(\mathbf{X}) = [u(\mathbf{X}) \ v(\mathbf{X}) \ w(\mathbf{X})]$ d'un point matériel $\mathbf{X} \in \Omega_0$. Cette formulation est donc faite en fonction du domaine initial non déformé de notre organe ce qui est un avantage important. En effet, avec cette approche il devient possible de discrétiser l'organe en un ensemble d'éléments tétraédriques ou hexaédriques dont nous connaissons bien la géométrie avant la simulation. À l'aide de cette discrétisation, nous arrivons à intégrer le volume en utilisant des points d'intégration situés au milieu de chaque élément.

Le déplacement de ces points d'intégration peut maintenant être interpolé à l'aide des déplacements des sommets entourant le point. La procédure que nous proposons ici fait ressortir les inconnus de notre système : les déplacements \mathbf{u}_s aux sommets s de nos éléments. Ces inconnus sont appelés les **degrés de liberté** de la simulation et forment le coeur d'une simulation statique, ou plutôt semi-statique, puisque le système devra être résolu à chaque fois que les forces externes changent durant la simulation.

Nous reprenons d'abord les termes des équations 4.16 et 4.17 en laissant tomber le terme de l'énergie cinétique (puisque la simulation statique ne dépend pas du temps). Nous obtenons

donc l'équilibre suivant à résoudre :

$$\int_{\Omega^0} \mathbf{S} : \boldsymbol{\epsilon}(\mathbf{w}) dV = \int_{\Gamma^0} (J \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathbf{N}) \cdot \mathbf{w} dA + \int_{\Omega^0} \mathbf{r} \cdot \mathbf{w} J dV \quad (4.31)$$

En utilisant la discrétisation vue précédemment, nous réécrivons le terme de gauche de l'équation (les termes de droite sont remplacés de manière similaire) :

$$\int_{\Omega^0} \mathbf{S} : \boldsymbol{\epsilon}(\mathbf{w}) dV = \sum_{e=0}^{n_e} \int_{\Omega_e^0} (\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u})) : \boldsymbol{\epsilon}(\phi) dV_e \quad (4.32)$$

Nous reformulons également le petit tenseur de déformation en utilisant les fonctions de formes :

$$\boldsymbol{\epsilon}^e = \frac{1}{2} (\nabla \mathbf{u}_e + \nabla \mathbf{u}_e^T) \quad (4.33)$$

$$= \frac{1}{2} \left(\sum_{j=0}^3 \begin{bmatrix} \phi_{j,x} u_j^e & \phi_{j,y} u_j^e & \phi_{j,z} u_j^e \\ \phi_{j,x} v_j^e & \phi_{j,y} v_j^e & \phi_{j,z} v_j^e \\ \phi_{j,x} w_j^e & \phi_{j,y} w_j^e & \phi_{j,z} w_j^e \end{bmatrix} + \begin{bmatrix} \phi_{j,x} u_j^e & \phi_{j,y} u_j^e & \phi_{j,z} u_j^e \\ \phi_{j,x} v_j^e & \phi_{j,y} v_j^e & \phi_{j,z} v_j^e \\ \phi_{j,x} w_j^e & \phi_{j,y} w_j^e & \phi_{j,z} w_j^e \end{bmatrix}^T \right) \quad (4.34)$$

$$= \frac{1}{2} \left(\sum_{j=0}^3 \begin{bmatrix} 2\phi_{j,x} & \phi_{j,y} + \phi_{j,x} & \phi_{j,z} + \phi_{j,x} \\ \phi_{j,x} + \phi_{j,y} & 2\phi_{j,y} & \phi_{j,z} + \phi_{j,y} \\ \phi_{j,x} + \phi_{j,z} & \phi_{j,y} + \phi_{j,z} & 2\phi_{j,z} \end{bmatrix} \begin{bmatrix} u_j^e \\ v_j^e \\ w_j^e \end{bmatrix} \right)^T \quad (4.35)$$

$$= \left(\sum_{j=0}^3 \begin{bmatrix} \phi_{j,x} & \frac{1}{2}(\phi_{j,y} + \phi_{j,x}) & \frac{1}{2}(\phi_{j,z} + \phi_{j,x}) \\ \frac{1}{2}(\phi_{j,y} + \phi_{j,x}) & \phi_{j,y} & \frac{1}{2}(\phi_{j,z} + \phi_{j,y}) \\ \frac{1}{2}(\phi_{j,z} + \phi_{j,x}) & \frac{1}{2}(\phi_{j,z} + \phi_{j,y}) & \phi_{j,z} \end{bmatrix} \begin{bmatrix} u_j^e \\ v_j^e \\ w_j^e \end{bmatrix} \right)^T \quad (4.36)$$

En exploitant la symétrie de $\boldsymbol{\epsilon}^e$ et en utilisant la formulation de Voigt, on obtient :

$$\begin{bmatrix} \epsilon_{00}^e \\ \epsilon_{11}^e \\ \epsilon_{22}^e \\ 2\epsilon_{01}^e \\ 2\epsilon_{02}^e \\ 2\epsilon_{12}^e \end{bmatrix} = \begin{bmatrix} \phi_{j,x} & 0 & 0 \\ 0 & \phi_{j,y} & 0 \\ 0 & 0 & \phi_{j,y} \\ \phi_{j,y} & \phi_{j,x} & 0 \\ 0 & \phi_{j,z} & \phi_{j,y} \\ \phi_{j,z} & 0 & \phi_{j,x} \end{bmatrix} \begin{bmatrix} u_j^e \\ v_j^e \\ w_j^e \end{bmatrix} \quad (4.37)$$

$$= \mathbf{B}_j^e \mathbf{u}_j^e \quad (4.38)$$

Ici ϕ_j est la fonction de forme du sommet j de l'élément e évaluée à la coordonnée matérielle \mathbf{X}_I^e du point d'intégration de l'élément e et \mathbf{B}_j^e est la matrice de déformation-déplacement du sommet j .

En exploitant cette formulation et en isolant les \mathbf{u}_j^e , nous obtenons :

$$\dot{\Pi}_{int} = \int_{\Omega^0} \mathbf{S} : \boldsymbol{\epsilon}(\mathbf{w}) dV \quad (4.39)$$

$$= \sum_{e=0}^{n_e} \int_{\Omega_e^0} (\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u})) : \boldsymbol{\epsilon}(\phi) dV_e \quad (4.40)$$

$$= \sum_{e=0}^{n_e} V_e \sum_{j=0}^3 (\mathbf{B}_j^e)^T \mathbf{C} \sum_{k=0}^3 \mathbf{B}_k^e \mathbf{u}_k^e \quad (4.41)$$

$$= \sum_{e=0}^{n_e} \sum_{j=0}^3 \sum_{k=0}^3 V_e (\mathbf{B}_j^e)^T \mathbf{C} \mathbf{B}_k^e \mathbf{u}_k^e \quad (4.42)$$

$$= \sum_{e=0}^{n_e} \sum_{j=0}^3 \sum_{k=0}^3 \mathbf{K}_{jk}^e \mathbf{u}_k^e \quad (4.43)$$

Enfin, le système élémentaire précédent est assemblé dans un système de $3n$ équations à $3n$ inconnus pour n degrés de liberté de trois composants (u , v et w) :

$$\mathbf{K} \mathbf{u} = \mathbf{F} \quad (4.44)$$

où \mathbf{K} est la matrice de rigidité $3n \times 3n$. Ici \mathbf{K}_{ij} contient la somme des contributions des matrices élémentaires \mathbf{K}_e de sorte que les sommets i et j font partie de l'élément e , \mathbf{u} est le vecteur inconnu de taille $3n$ contenant les déplacements des sommets et \mathbf{F} est le vecteur des forces externes du système.

La matrice \mathbf{K} peut être assemblée de façon à ce qu'elle soit sous la forme d'une matrice à bande (où les termes non nuls sont près de la diagonale). Également, puisque les dérivées des fonctions de forme sont constantes, la matrice peut être précalculée et donc préinversée une seule fois. La simulation revient donc simplement à accumuler les forces externes dans \mathbf{F} et à le multiplier avec \mathbf{K}^{-1} :

$$\mathbf{u} = \mathbf{K}^{-1} \mathbf{F} \quad (4.45)$$

ce qui heureusement est très rapide. Malheureusement, un changement topologique (par exemple, la suppression ou la création d'un élément durant la simulation) vient invalider la matrice \mathbf{K} et donc \mathbf{K}^{-1} . Dans la prochaine section, nous examinerons justement la simulation dynamique avec des méthodes d'intégration utilisant les éléments de façon presque indépendante entre eux. Ceci permet de simplifier les changements topologiques. Par contre, elles alourdissent du même coup le temps de calcul.

4.5 Simulation dynamique et intégration temporelle

Nous avons vu dans les sections précédentes une méthode d'approximation du déplacement \mathbf{u} pour un point $\mathbf{x} \in \Omega^0$ grâce à une discrétisation par éléments géométriques. Nous avons ensuite décrit l'énergie potentielle de déformation du volume en fonction d'une loi de comportement linéaire entre les mesures de déformation et des contraintes. Cette énergie devient nulle lorsque l'objet simulé reprend sa forme au repos (sans contraintes externes).

Dans cette section, on s'intéresse aux états intermédiaires de l'objet simulé dans le temps. Pour ce faire, en plus de la discrétisation de l'espace volumique utilisée pour obtenir l'énergie de la déformation, nous désirons ajouter une discrétisation isochrone de l'espace temps de notre simulation. Nous choisirons un temps initial de la simulation, t_0 , ainsi qu'un pas de temps Δt de telle sorte que la discrétisation dans le temps devient $t_{n+1} = t_n + \Delta t$. Puisque notre simulation est en temps réel, le temps de calcul d'un pas de temps devra être inférieur à ce dernier. Le choix du pas de temps devra également assurer la stabilité de la simulation puisque certaines méthodes d'intégration dans le temps deviennent instables lorsque le pas de temps est trop grand ou si ce dernier n'est pas régulier. Nous en verrons quelques exemples dans nos expériences du chapitre 6.

Pour obtenir une formulation dans le temps, nous reprenons l'équation de mouvement de Cauchy (équation 4.4), mais cette fois en y gardant le terme de l'énergie cinétique (équation

4.15) :

$$\underbrace{\int_{\Omega^0} \rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} dV}_{\dot{\Pi}_{cin}} + \underbrace{\int_{\Omega^0} \mathbf{S} : \boldsymbol{\epsilon}(\mathbf{w}) dV}_{\dot{\Pi}_{int}} = + \underbrace{\int_{\Gamma^0} (J \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \cdot \mathbf{N}) \cdot \mathbf{w} dA + \int_{\Omega^0} \mathbf{r} \cdot \mathbf{w} J dV}_{\dot{\Pi}_{ext}} \quad (4.46)$$

où ρ_0 est la masse volumique initiale du matériau, $\boldsymbol{\sigma}$ est le tenseur des contraintes de Cauchy, \mathbf{S} est le second tenseur des contraintes de Piola-Kirchoff, \mathbf{w} est une fonction test quelconque et \mathbf{r} est la force volumique externe. Nous pouvons reformuler l'équation précédente avec la description lagrangienne du déplacement sous la forme matricielle suivante :

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{F}^{\text{int}}(\mathbf{u}) = \mathbf{F}^{\text{ext}}(\mathbf{u}) \quad (4.47)$$

où \mathbf{M} et \mathbf{D} sont respectivement les matrices de masse et d'amortissement de taille $3n \times 3n$ pour n degrés de liberté, \mathbf{u} , $\dot{\mathbf{u}}$ et $\ddot{\mathbf{u}}$ sont respectivement le vecteur de déplacement, la dérivée et la dérivée seconde par rapport au temps des déplacements, \mathbf{F}^{int} est le vecteur de force interne de déformation obtenu dans les sections précédentes et \mathbf{F}^{ext} est le vecteur de forces externes (force gravitationnelle, forces de contact et autres).

Pour les méthodes d'intégration, nous aurons besoin d'une formulation pour obtenir la force interne $\mathbf{F}^{\text{int}}(\mathbf{u}_i) = \mathbf{f}_i$ d'un sommet i du maillage. Pour ce faire, nous dérivons le terme d'énergie interne (équation 4.23) des éléments adjacents au sommet i par rapport à son déplacement :

$$\mathbf{f}_i = \nabla_{\mathbf{u}_i} \sum_{e_i} \dot{\Pi}_{int}^e \quad (4.48)$$

$$= \nabla_{\mathbf{u}_i} \sum_{e_i} V_{e_i} (\mathbf{C} : \mathbf{E}(\mathbf{u}(\mathbf{X}_I^{e_i}))) : \boldsymbol{\epsilon}(\phi) \quad (4.49)$$

où e_i est un élément de volume V_{e_i} avec un point d'intégration $\mathbf{X}_I^{e_i}$ au milieu de son volume et dont i est l'un de ses sommets. Dans le cas de l'utilisation du petit tenseur de déformation linéaire $\boldsymbol{\epsilon}$, nous obtenons :

$$\mathbf{f}_i = \nabla_{\mathbf{u}_i} \sum_{e_i} V_{e_i} (\mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u}(\mathbf{X}_I^{e_i}))) : \boldsymbol{\epsilon}(\phi) \quad (4.50)$$

$$= \nabla_{\mathbf{u}_i} \sum_{e_i} V_{e_i} \sum_{j=0}^3 (\mathbf{B}_j^{e_i})^T \mathbf{C} \sum_{k=0}^3 \mathbf{B}_k^{e_i} \mathbf{u}_k \quad (4.51)$$

$$= \sum_{e_i} V_{e_i} \sum_{j=0}^3 (\mathbf{B}_j^{e_i})^T \mathbf{C} \mathbf{B}_i^{e_i} \mathbf{u}_i \quad (4.52)$$

Cette dernière équation est celle utilisée dans nos expériences. Nous ajoutons également la formulation pour le tenseur de déformation non linéaire de Green-Lagrange \mathbf{E} qui n'a cependant pas été implémentée dans notre code :

$$\mathbf{f}_i = \nabla_{\mathbf{u}_i} \sum_{e_i} V_{e_i} (\mathbf{C} : \mathbf{E}(\mathbf{u}(\mathbf{X}_I^{e_i}))) : \boldsymbol{\epsilon}(\phi) \quad (4.53)$$

$$= \sum_{e_i} V_{e_i} \mathbf{J}^{e_i} (\mathbf{B}_i^{e_i})^T \mathbf{S}^{e_i} \quad (4.54)$$

Les deux prochaines sections décrivent les méthodes utilisées pour intégrer l'équation 4.47 dans le temps.

4.5.1 Intégration explicite

Les méthodes d'intégration explicite sont utilisées pour intégrer l'équation 4.47 à la fin d'un pas de temps pour résoudre le déplacement des sommets (DDLs) au temps suivant. Il s'agit des méthodes d'intégration dans le temps les plus rapides, en plus d'être très parallélisables. Toutefois, elles ont l'inconvénient d'être dépendantes de la taille du pas de temps choisi, c'est-à-dire que passé un certain seuil, la simulation devient instable.

Dans ce projet, nous avons utilisé deux méthodes d'intégration explicite. La première, très connue, est la méthode d'Euler explicite où nous découplons l'équation 4.47 en trois parties :

$$\ddot{\mathbf{u}}^{t+\Delta t} = \mathbf{M}^{-1} \mathbf{F}(\mathbf{u}^t, \dot{\mathbf{u}}^t) \quad (4.55)$$

$$\dot{\mathbf{u}}^{t+\Delta t} = \dot{\mathbf{u}}^t + \Delta t \ddot{\mathbf{u}}^t \quad (4.56)$$

$$\mathbf{u}^{t+\Delta t} = \mathbf{u}^t + \Delta t \dot{\mathbf{u}}^t \quad (4.57)$$

où $\mathbf{F}(\mathbf{u}^t, \dot{\mathbf{u}}^t) = \mathbf{F}^{\text{int}}(\mathbf{u}^t) + \mathbf{D}\dot{\mathbf{u}}^t + \mathbf{F}^{\text{ext}}(\mathbf{u}^t)$. Dans nos expériences, nous utiliserons une matrice de masse agrégée $\tilde{\mathbf{M}}$, c'est-à-dire que $\tilde{\mathbf{M}}$ est diagonale avec $\tilde{M}_{ii} = \frac{m}{n}$ où m est la somme des masses de la ligne i de \mathbf{M} , et n est le nombre de degrés de liberté sur la ligne. La partie de droite de l'équation 4.55 peut donc être obtenue en divisant l'élément i du vecteur \mathbf{F} par \tilde{M}_{ii} .

La seconde méthode d'intégration explicite que nous avons utilisée est la méthode des différences centrées (voir LS-DYNA). L'équation 4.47 est découplée comme suit :

$$\ddot{\mathbf{u}}^t = \frac{1}{(\Delta t)^2}(\mathbf{u}^{t+\Delta t} - 2\mathbf{u}^t + \mathbf{u}^{t-\Delta t}) \quad (4.58)$$

$$\dot{\mathbf{u}}^t = \frac{1}{2\Delta t}(\mathbf{u}^{t+\Delta t} - \mathbf{u}^{t-\Delta t}) \quad (4.59)$$

$$\mathbf{u}^{t+\Delta t} = (\mathbf{M} + \frac{1}{2}\Delta t \mathbf{D})^{-1} \left[\Delta t^2 (\mathbf{F}^{\text{ext}}(\mathbf{u}^t) - \mathbf{F}^{\text{int}}(\mathbf{u}^t)) + 2\mathbf{M}\mathbf{u}^t - \left(\mathbf{M} - \frac{\Delta t}{2} \mathbf{D} \right) \mathbf{u}^{t-\Delta t} \right] \quad (4.60)$$

L'équation 4.60 est obtenue en substituant les équations 4.58 et 4.59 dans l'équation 4.47 et nous montre que la position au prochain pas de temps est fonction des positions au pas de temps courant et du précédent uniquement.

Nous remarquons que les deux méthodes explicites sont facilement parallélisables. En effet, nous pouvons calculer les déplacements \mathbf{u}_i et leurs dérivées par rapport au temps t de façon indépendante entre les DDLs.

Pour assurer la stabilité de la simulation, nous utilisons le critère de Horton et al. (2010) pour déterminer la largeur maximale du pas de temps. Ainsi, soit la vitesse d'onde de dilatation du matériau utilisé :

$$c = \sqrt{\frac{\lambda + 2\mu}{\rho}} \quad (4.61)$$

Le pas de temps maximal pour un point d'intégration I est donné par :

$$\Delta t^{\text{max}} = \frac{2}{\sqrt{g}c} \quad (4.62)$$

où λ et μ sont les coefficients de Lamé, ρ est la masse volumique du matériau et g est la somme des carrés des dérivées des fonctions de forme au point I .

4.5.2 Intégration semi-implicite

Toujours dans le but d'intégrer l'équation 4.47 dans le temps, les méthodes d'intégration implicites sont formulées de telle sorte que l'intégration ne dépend plus du pas de temps choisi. Pour ce faire, nous utilisons la méthode d'Euler implicite où l'équation 4.47 est découplée comme suit :

$$\ddot{\mathbf{u}}^{t+\Delta t} = \mathbf{M}^{-1} \mathbf{F}(\mathbf{u}^{t+\Delta t}, \dot{\mathbf{u}}^{t+\Delta t}) \quad (4.63)$$

$$\dot{\mathbf{u}}^{t+\Delta t} = \dot{\mathbf{u}}^t + \Delta t \ddot{\mathbf{u}}^{t+\Delta t} \quad (4.64)$$

$$\mathbf{u}^{t+\Delta t} = \mathbf{u}^t + \Delta t \dot{\mathbf{u}}^{t+\Delta t} \quad (4.65)$$

Avec les forces $\mathbf{F}(\mathbf{u}^{t+\Delta t}, \dot{\mathbf{u}}^{t+\Delta t}) = \mathbf{F}^{\text{int}}(\mathbf{u}^{t+\Delta t}) + \mathbf{D}\dot{\mathbf{u}}^{t+\Delta t} + \mathbf{F}^{\text{ext}}(\mathbf{u}^{t+\Delta t})$. Puisque ces forces sont inconnues au temps t , nous devons trouver une approximation avec un développement de Taylor du premier ordre :

$$\mathbf{F}(\mathbf{u}^{t+\Delta t}, \dot{\mathbf{u}}^{t+\Delta t}) \approx \mathbf{F}(\mathbf{u}^t, \dot{\mathbf{u}}^t) + \frac{\partial}{\partial \mathbf{u}} \mathbf{F}(\mathbf{u}^t, \dot{\mathbf{u}}^t) + \frac{\partial}{\partial \dot{\mathbf{u}}} \mathbf{F}(\mathbf{u}^t, \dot{\mathbf{u}}^t) \quad (4.66)$$

$$= \mathbf{F}^{\text{int}}(\mathbf{u}^t) + \mathbf{F}^{\text{ext}}(\mathbf{u}^t) + \dot{\mathbf{K}} d\mathbf{u} + \mathbf{D} d\dot{\mathbf{u}} \quad (4.67)$$

où $\dot{\mathbf{K}}$ et \mathbf{D} sont respectivement les matrices tangentes de rigidité et d'amortissement, $d\mathbf{u} = \mathbf{u}^{t+\Delta t} - \mathbf{u}^t$ et $d\dot{\mathbf{u}} = \dot{\mathbf{u}}^{t+\Delta t} - \dot{\mathbf{u}}^t$.

En insérant 4.67 dans 4.63, nous obtenons le système d'équations suivant à résoudre :

$$(\mathbf{M} - \Delta t \mathbf{D} - \Delta t^2 \dot{\mathbf{K}}) \ddot{\mathbf{u}}^{t+\Delta t} = \Delta t (\mathbf{F}^{\text{int}}(\mathbf{u}^t) + \mathbf{F}^{\text{ext}}(\mathbf{u}^t)) + \Delta t^2 \dot{\mathbf{K}} \dot{\mathbf{u}}^t \quad (4.68)$$

Pour résoudre ce système, nous utilisons le solveur itératif du gradient conjugué (Baraff et Witkin (1998)) avec un seuil d'erreur de 0.00005. Durant nos expériences, le nombre d'itérations ne dépassait jamais 25.

4.6 Approche corotationnelle

Nous avons vu dans la section 2.1 que la déformation d'un corps mou de sa configuration initiale (au repos) à sa configuration actuelle (au temps t) est rigide s'il peut être décrit par

une transformation rigide, c'est-à-dire par une translation et/ou une rotation. Une transformation rigide contient donc une énergie potentielle de déformation nulle, en d'autres termes, le volume transformé n'a pas subi de déformation, simplement un déplacement.

Si la consistance de la fonction de forme est faible, si le petit tenseur de déformation linéaire est utilisé ou bien si une relation de comportement entre le tenseur des contraintes et le tenseur de déformation est linéaire, les rotations seront mal représentées et l'énergie potentielle de déformation sera non nulle pour une transformation rigide. En conséquence, des forces superflues, dites force fantômes, apparaissent comme une augmentation du volume de l'objet simulé proportionnellement au degré de rotation de la transformation.

Pour tenter de minimiser l'erreur due à la rotation, nous profitons des méthodes de factorisation matricielle telles que la décomposition en valeurs singulières (SVD) et la décomposition polaire d'une matrice de transformation. En observant la figure 4.2, nous remarquons que la décomposition SVD permet de séparer une matrice de transformation en deux composantes de rotation (\mathbf{U} et \mathbf{V}) et un composant d'étirement ($\mathbf{\Sigma}$).

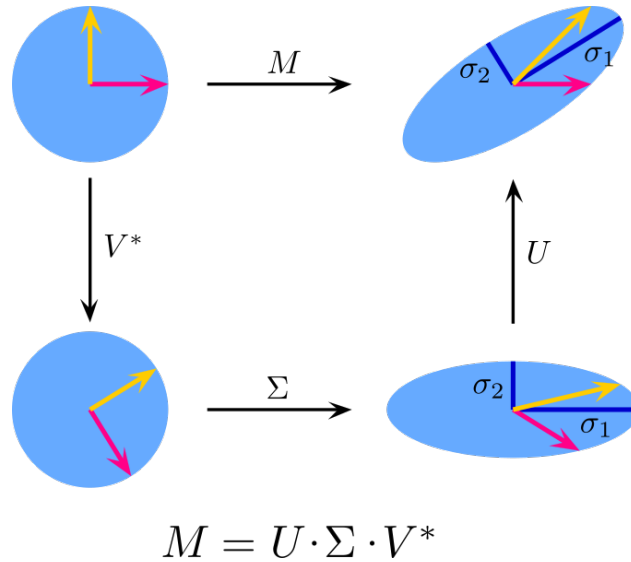


Figure 4.2 Décomposition SVD d'une transformation. Auteur Georg-Johann, Wikipédia

Voici une brève description de l'approche corotationnelle de Nesme et al. (2005). Ici la une matrice de transformation est obtenue à l'aide des trois segments connectés à un sommet d'un tétraèdre ou d'un hexaèdre :

$$\mathbf{J} = [\mathbf{e}_1^0 \ \mathbf{e}_2^0 \ \mathbf{e}_3^0]^{-1} [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3] \quad (4.69)$$

où \mathbf{e}_i est un des trois segments de l'élément et \mathbf{e}_i^0 le même segment dans le volume non déformé. Nesme et al. (2005) utilise une décomposition polaire pour extraire la matrice de rotation. Nous avons plutôt implémenté la méthode de décomposition SVD stable décrite dans Press et al. (1992) pour obtenir $\mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. La matrice de rotation \mathbf{R} de l'élément est ainsi extraite : $\mathbf{R} = \mathbf{U}\mathbf{V}^T$.

Nous transposons ensuite la matrice de rotation \mathbf{R}_e d'un élément e pour obtenir son inverse de sorte qu'on puisse obtenir le tenseur gradient du déplacement (4.27) sans rotation :

$$\nabla \mathbf{u}_e^T = \sum_{j=0}^3 \nabla \phi_j (\mathbf{R}_e^T \mathbf{x}_i - \mathbf{x}_i^0)^T \quad (4.70)$$

Finalement, la rotation \mathbf{R}_e est réappliquée sur les forces d'un sommet i (équation 4.48). Pour le petit tenseur de déformation \mathbf{e} :

$$\mathbf{f}_i = \sum_{e_i} \mathbf{R}_{e_i} V_{e_i} \sum_{j=0}^3 (\mathbf{B}_j^{e_i})^T \mathbf{C} \mathbf{B}_i^{e_i} (\mathbf{R}_{e_i}^T \mathbf{x}_i - \mathbf{x}_i^0) \quad (4.71)$$

Puis, si le tenseur de déformation non linéaire de Green-Lagrange \mathbf{E} est utilisé, nous avons :

$$\mathbf{f}_i = \sum_{e_i} \mathbf{R}_{e_i} V_{e_i} \mathbf{J}^{e_i} (\mathbf{B}_i^{e_i})^T \mathbf{S}^{e_i} \quad (4.72)$$

où les tenseurs \mathbf{J}^{e_i} et \mathbf{S}^{e_i} sont obtenus avec $\nabla \mathbf{u}_e^T$ sans rotation (équation 4.70).

4.7 Forces de contact et retour de force pour outils haptiques

Nous avons pour l'instant un modèle déformable sous l'effet de forces externes volumiques. Or, une simulation chirurgicale comprend également des forces de contact lorsqu'une collision entre deux objets est détectée. Dans le cas où l'un des deux objets est l'instrument virtuel dirigé par un outil haptique, la force de contact devra être retournée à l'outil haptique pour offrir une résistance physique à l'utilisateur.

Pour résoudre ces forces, nous avons utilisé les outils fournis par le logiciel **Sofa Framework** (Allard et al. (2007)). Ainsi, nous n'avons pas eu à implémenter ces méthodes. Voici tout de même une description de ces dernières.

La première approche, la méthode *des pénalités par ressort*, permettant résoudre la force de contact entre deux objets consiste simplement à ajouter une force de ressort entre les points surfaciques en collision et la distance de pénétration des deux objets. Dans un premier temps, les collisions entre les éléments 2D en surface des objets (points, lignes, triangles et quadrilatères) sont détectées. Soit d la distance de pénétration d'une collision, la force de pénalité est

$$\mathbf{f}_{\text{pen}} = -\mathbf{n}kd \quad (4.73)$$

où \mathbf{n} est le vecteur normal de la surface et k est un facteur de rigidité. Les forces sont ensuite réintroduites dans le vecteur de forces externe de la simulation.

La seconde approche, la méthode *des pénalités par multiplicateurs de Lagrange* (Courtecuisse et al. (2010)), ajoute des contraintes \mathbf{H}_i au système d'équations à résoudre pour modéliser des interactions complexes telles que la friction et les contacts. Ainsi, le système à résoudre pour les deux objets devient :

$$\mathbf{A}_1 \mathbf{u}_1 = \mathbf{b}_1 + \mathbf{H}_1 \boldsymbol{\lambda} \quad (4.74)$$

$$\mathbf{A}_2 \mathbf{u}_2 = \mathbf{b}_2 + \mathbf{H}_2 \boldsymbol{\lambda} \quad (4.75)$$

où \mathbf{H} est la loi d'interaction entre les objets (par exemple, les déplacements des DDLs suite à une collision), et $\boldsymbol{\lambda}$ est le vecteur de force inconnue répondant à cette loi. Les détails pour résoudre ce système peuvent être retrouvés dans Courtecuisse et al. (2010).

Les forces de contact entre un organe et un instrument virtuel sont obtenues en choisissant l'une de ces deux méthodes. Elles sont ensuite retournées à l'outil haptique à partir d'un fil d'exécution différent de celui de la simulation permettant de retourner les forces à une très grande fréquence sans que la simulation en soit impactée.

4.8 Algorithme final du modèle FEM

Définitions :

$\mathbf{X}^t = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n]^T$ l'ensemble des positions des sommets (DDLs) du système au temps t

$\mathbf{U}^t = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n]^T$ l'ensemble des déplacements \mathbf{u}_i d'un DDL i au temps t

$\mathbf{V}_e = [\mathbf{x}_1^e \ \mathbf{x}_2^e \ \dots \ \mathbf{x}_n^e]^T$ l'ensemble des sommets de l'élément e

$\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \dots \ \omega_3]^T$ l'ensemble des poids ω_e du point d'intégration au centre de l'élément e

$\mathbf{B}_e = [\mathbf{B}_1^e \ \mathbf{B}_2^e \ \dots \ \mathbf{B}_n^e]$ l'ensemble des matrices de déformation-déplacement \mathbf{B}_i^e du sommet i de l'élément e (équation 4.37)

$\mathbf{F}_{\text{int}} = [\mathbf{f}_1 \ \mathbf{f}_2 \ \dots \ \mathbf{f}_n]^T$ l'ensemble des forces internes des DDLs

Initialisation :

1. Pour chaque élément e ,
 - (a) $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} \cup \boldsymbol{\omega}_e$ (les valeurs de $\boldsymbol{\omega}_e$ sont définies à l'annexe A)
 - (b) Pour chaque sommet i de l'élément e
 - i. $\nabla \phi_i^e \leftarrow$ équation 4.27
 - ii. $\mathbf{B}_e \leftarrow \mathbf{B}_e \cup \mathbf{B}_i(\nabla \phi_i^e)$ où $\mathbf{B}_i(\nabla \phi_i^e)$ est décrite à l'équation 4.37

Pour chaque pas de temps Δt :

1. Pour chaque élément e ,
 - (a) $\mathbf{R}_e \leftarrow \mathbf{U}\mathbf{V}^T$ avec $\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ définie à l'équation 4.69
 - (b) Pour chaque sommet i de l'élément e
 - i. Pour chaque sommet j de l'élément e
 - A. $\mathbf{f}_i \leftarrow \mathbf{f}_i + \omega_e \mathbf{R}_e \mathbf{B}_j^T \mathbf{C} \mathbf{B}_i (\mathbf{R}_e^T \mathbf{x}_i^t - \mathbf{x}_i^0)$
2. $\mathbf{U}^{t+1} \leftarrow$ équation 4.57, 4.60 ou 4.68
3. $\mathbf{X}^t \leftarrow \mathbf{X}^0 + \mathbf{U}^{t+1}$

CHAPITRE 5 MÉTHODES SANS MAILLAGE

Comme nous l'avons mentionné au chapitre précédent, l'approche basée sur les éléments finis (avec maillage) comporte certaines difficultés dues à la topologie de l'objet qui doit être considérée ainsi qu'aux changements que doit subir cette topologie. Les méthodes sans maillage permettent d'aborder cette problématique puisque, comme l'indique le nom, ces méthodes n'ont pas d'élément topologique.

Ce chapitre débute par une présentation des deux éléments de base des méthodes sans maillage, soit le noyau et la fonction de forme. Nous présenterons ensuite les cinq noyaux que nous avons implémentés. Nous décrirons par la suite les deux méthodes que nous avons retenues pour construire et résoudre la dynamique d'un objet déformable dans un environnement sans maillage, soit la méthode d'animation basée sur les points (modèle MBP) et la méthode Garlerkin-free (modèle MTLED). L'algorithme intégrant ces deux modèles est décrit à la fin de ce chapitre.

5.1 Noyaux et fonctions de forme

Tout comme la méthode des éléments finis, les méthodes sans maillage permettent d'estimer un déplacement $u(\mathbf{x})$ à une position \mathbf{x} du domaine Ω . Cependant, plutôt que d'utiliser les propriétés géométriques d'un élément pour interpoler ce déplacement, les méthodes sans maillage utilisent un nuage de points où chaque point I (appelé fréquemment "*particule*") comporte un support compact d'influence, communément appelé le *domaine d'influence* Ω_I de la particule I . L'approximation d'une valeur $u^h(\mathbf{x}) \approx u(\mathbf{x})$ à une position \mathbf{x} du domaine est alors obtenue à l'aide d'une fonction d'approximation, dite *fonction de forme*, qui est évaluée pour chaque particule voisine comme suit :

$$u^h(\mathbf{x}) = \sum_I \phi_I(\mathbf{x}) u_I \quad (5.1)$$

Ici ϕ_I et u_I sont respectivement la fonction de forme et la valeur d'une particule I voisine à \mathbf{x} , c'est-à-dire $\forall \Omega_I$ t.q. $\mathbf{x} \in \Omega_I$.

Le support de la fonction de forme d'une particule est généralement bâti à partir d'une fonction de poids $w(\mathbf{x} - \mathbf{x}_I, h_I)$ (souvent appelée la fonction "noyau", ou "fenêtre") . Cette fonction est monotone et décroît de façon proportionnelle en fonction de la distance d'influence h_I entourant le point I à l'intérieur du domaine Ω_I . Les supports les plus fréquents

sont circulaires ou rectangulaires (voir figure 5.1).

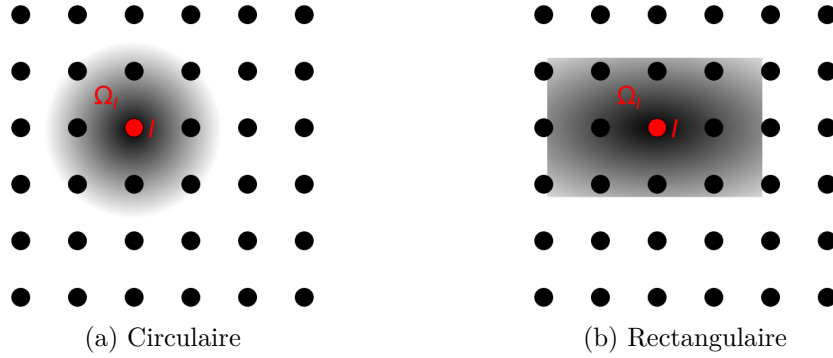


Figure 5.1 Supports d'influence monotones et décroissants d'un point I

Pour notre projet, nous avons choisi une formulation lagrangienne pour formuler la fonction de poids lagrangienne. Ainsi, le poids d'une position dans le support compact d'une particule restera constant tout au long de la simulation et sera précalculé une seule fois au départ de la simulation. Évidemment cette approche permet des gains de vitesse importants par rapport à une formulation eulérienne. Mais ces gains ont un coût non négligeable : une particule sera limitée à de petits déplacements uniquement (Fries et Matthies (2003)). Il est important de noter cependant que nous n'avons trouvé aucune mention de ce désavantage dans le cadre de la simulation chirurgicale, le coût d'utilisation d'un noyau eulérien étant trop grand de toute manière pour du temps réel.

Le choix de la fonction de poids varie normalement d'une application à l'autre. Par contre, la fonction doit répondre à certaines conditions pour assurer une bonne représentation locale et une bonne consistance de l'approximation vers sa solution (Belytschko et al. (1996)) . Ces conditions sont :

1. $w(\mathbf{x} - \mathbf{x}_I, h_I) > 0$ pour $x \in \Omega_I$
2. $w(\mathbf{x} - \mathbf{x}_I, h_I) = 0$ pour $x \notin \Omega_I$
3. $\int_{\Omega} w(\mathbf{x} - \mathbf{x}_I, h_I) d\Omega = 1$
4. $w(\mathbf{x}_1 - \mathbf{x}_I, h_I) > w(\mathbf{x}_2 - \mathbf{x}_I, h_I)$ lorsque $\|\mathbf{x}_1 - \mathbf{x}_I\| < \|\mathbf{x}_2 - \mathbf{x}_I\|$
5. $w(\mathbf{x} - \mathbf{x}_I, h_I) \rightarrow \delta(\|\mathbf{x} - \mathbf{x}_I\|)$ lorsque $h_I \rightarrow 0$, $\delta(\|\mathbf{x} - \mathbf{x}_I\|)$ est la fonction de Dirac.

Le critère de consistance quant à lui dépendra directement de la qualité de l'approximation et du degré de l'équation aux dérivées partielles à résoudre. Donc, pour qu'une approximation ait une consistance de degré k , elle devra représenter exactement une polynomiale de degré k . Dit autrement, pour une équation aux dérivées partielles de degré $2k$, l'approximation devra

représenter exactement une solution de degré k . Si l'équation aux dérivées partielles est du second degré, l'approximation sera consistante au premier degré (linéairement consistante), et produira exactement le champ linéaire de la solution.

5.2 Choix du noyau et analyse du domaine d'influence d'une particule

Plusieurs noyaux ont été documentés dans la littérature. Les noyaux que nous avons implémentés sont les suivants :

$$\textbf{Spiky} \text{ (Desbrun et Gascuel (1996))} \quad w(q) \in C^2 = \begin{cases} \frac{15}{\pi h^3} (1-q)^3 & q \leq 1 \\ 0 & q > 1 \end{cases} \quad (5.2)$$

$$\textbf{Poly6} \text{ (Müller et al. (2003))} \quad w(q) \in C^2 = \begin{cases} \frac{315}{64\pi h^3} (1-q^2)^3 & q \leq 1 \\ 0 & q > 1 \end{cases} \quad (5.3)$$

$$\textbf{SPH} \text{ (Solenthaler et al. (2007))} \quad w(q) \in C^\infty = \begin{cases} \frac{\cos(\frac{\pi}{2}(1+q))+1}{4h^3(\frac{\pi}{3}-\frac{8}{\pi}+\frac{16}{\pi^2})} & q \leq 1 \\ 0 & q > 1 \end{cases} \quad (5.4)$$

$$\textbf{Spline cubique} \quad w(q) \in C^2 = \begin{cases} \frac{8}{\pi h^3} (1-6q^2+6q^3) & q \leq \frac{1}{2} \\ \frac{16}{\pi h^3} (1-q)^3 & \frac{1}{2} \leq q \leq 1 \\ 0 & q > 1 \end{cases} \quad (5.5)$$

$$\textbf{Spline quartique} \text{ (Horton et al. (2010))} \quad w(q) \in C^2 = \begin{cases} 1-6q^2+8q^3-3q^4 & q \leq 1 \\ 0 & q > 1 \end{cases} \quad (5.6)$$

Avec $q = \frac{\|\mathbf{x}-\mathbf{x}_I\|}{h}$.

Une représentation graphique de ces noyaux est présentée à la figure 5.2 et à la figure 5.3 à la fin de cette section.

Nous avons effectué plusieurs tests sur ces différents noyaux. Nos expériences ont permis de démontrer que le choix du noyau n'a pas un grand impact puisque les différences entre les noyaux d'une approximation restent petites en général.

La distance d'influence d'une particule, ou dans notre cas, le rayon h du noyau, aura quant à elle une relation directe avec la qualité de l'approximation puisqu'elle définit la taille et le support de la fonction de forme utilisée. Bien que la valeur optimale de h reste un problème ouvert (Fries et Matthies (2003)), nous avons testé trois méthodes d'affectations du rayon durant nos expériences.

La première, la plus simple, consiste simplement à assigner la même valeur h à toutes les particules du système (équation 5.7). Cette méthode fonctionne bien si les particules ont toutes une densité semblable (donc à peu près la même quantité de voisins dans un même rayon).

$$h_I = h \quad (5.7)$$

La seconde méthode (équation 5.8) récupère le j plus proche voisin de la particule I et multiplie la distance entre ces deux points par un facteur de dilatation α (généralement entre 1 et 3).

$$h_I = \alpha r_I \quad (5.8)$$

$$r_I = \|\mathbf{x}_I - \mathbf{x}_j\|$$

Finalement, la dernière méthode calcule la distance moyenne entre les n plus proches voisins et multiplie cette valeur par le facteur de dilatation (équation 5.9).

$$h_I = \alpha r_I \quad (5.9)$$

$$r_I = \frac{1}{n} \sum_j \|\mathbf{x}_I - \mathbf{x}_j\|$$

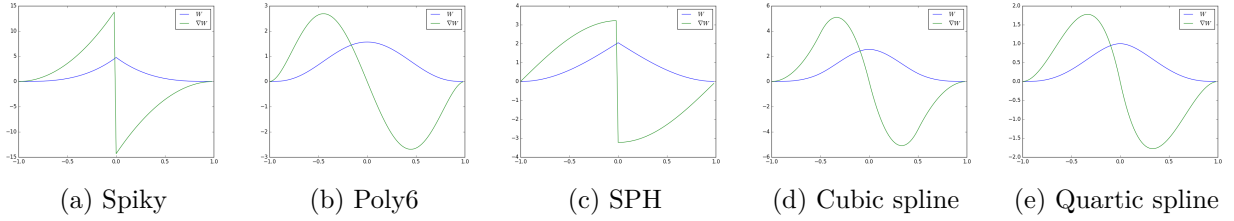


Figure 5.2 Noyaux utilisés dans nos expériences (1D)

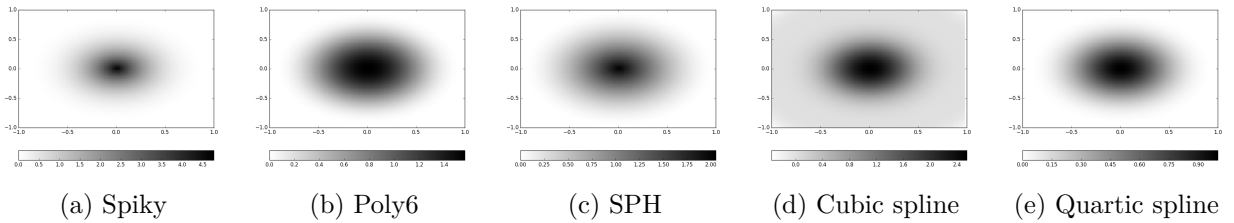


Figure 5.3 Noyaux utilisés dans nos expériences (2D)

5.3 Élasticité : Animation basée sur les points (Méthode MBP)

Il existe plusieurs approches pour résoudre la dynamique d'élasticité d'un objet déformable. La première que nous avons considérée est la méthode d'animation élastique basée sur les points introduite dans Müller et al. (2004). Avec cette approche, les équations d'élasticité en théorie de la mécanique continue sont appliquées aux particules du domaine. Voici comment nous avons construit notre modèle à partir de cette approche.

Au début de la simulation, nous déterminons pour chaque particule le support de celle-ci en choisissant l'une des trois méthodes de calcul d'une distance d'influence (équations 5.7 à 5.9) ainsi qu'une des cinq méthodes de calcul du noyau (équations 5.2, 5.3, 5.4, 5.5 et 5.6). Une fois le support d'influence obtenu, nous gardons pour chaque particule un vecteur de particules voisines ainsi que le poids (l'influence) de chacune.

Nous posons ensuite la densité d'énergie de déformation d'une particule en utilisant le volume de celle-ci comme approximation de l'intégrale de l'équation 4.23. L'équation suivante :

$$U_i = \int_{\Omega_i^0} \frac{1}{2} (\boldsymbol{\epsilon}_i \cdot \boldsymbol{\sigma}_i) d\mathbf{X} \quad (5.10)$$

devient alors

$$U_i = v_i \frac{1}{2} (\boldsymbol{\epsilon}_i \cdot \boldsymbol{\sigma}_i) \quad (5.11)$$

où v_i est le volume de la particule i , $\boldsymbol{\epsilon}_i$ est le tenseur de déformation (linéaire ou non linéaire) et $\boldsymbol{\sigma}_i$ est le tenseur de stress.

Pour trouver le volume v_i , nous avons utilisé la méthode proposée par Müller et al. (2003) où la densité d'une particule (appelée "phyxel" dans Müller et al. (2004)) est déterminée à l'aide de la méthode d'approximation "Smooth Particle Hydrodynamics" (SPH). Avec cette méthode, une valeur est approchée en un point $\mathbf{x} \in \Omega$ en sommant l'influence des particules voisines au point comme suit :

$$u^h(\mathbf{x}) = \sum_{j \text{ t.q. } \mathbf{x}_j \in V(\mathbf{x})} m_j \frac{u_j}{\rho_j} w(\|\mathbf{x} - \mathbf{x}_j\|, h) \quad (5.12)$$

Ici $V(\mathbf{x})$ est l'ensemble des particules voisines de \mathbf{x} (donc lorsque le noyau $w(\|\mathbf{x} - \mathbf{x}_j\|, h) > 0$), u_j est la valeur u à la position \mathbf{x}_j et finalement ρ_j et m_j sont respectivement la densité et la masse de la particule j .

La densité d'une particule i est donc déterminée par

$$\rho^h(\mathbf{x}_i) = \sum_{j \text{ t.q. } \mathbf{x}_j \in V(\mathbf{x}_i)} m_j \frac{\rho_j}{\rho_j} w(\|\mathbf{x}_i - \mathbf{x}_j\|, h) \quad (5.13)$$

$$= \sum_{j \text{ t.q. } \mathbf{x}_j \in V(\mathbf{x}_i)} m_j w(\|\mathbf{x}_i - \mathbf{x}_j\|, h) \quad (5.14)$$

Étant donné que le volume est défini comme

$$v_i = \frac{m_i}{\rho_i} \quad (5.15)$$

la dernière étape consiste à déterminer la masse m_i .

Pour ce faire, nous trouvons la distance moyenne r_i entre la particule i et ses k plus proches voisins. Soit la densité ρ du matériau simulé, on pose alors $m_i = sr_i^3 \rho$ avec s un facteur permettant de rapprocher ρ_i à ρ . On trouve s en minimisant l'erreur $e = \sum_i (\rho_i - \rho)^2$:

$$\frac{de}{ds} = \frac{d}{ds} \sum_i (\rho_i - \rho)^2 \quad (5.16)$$

$$= \frac{d}{ds} \sum_i \left(\left(\sum_{j \in V(i)} sr_j^3 \rho w_{ij} \right) - \rho \right)^2 \quad (5.17)$$

où $w_{ij} = W(\|\mathbf{x}_i - \mathbf{x}_j\|, h)$. En posant $\frac{de}{ds} = 0$, on trouve

$$s = \frac{\sum_i a_i}{\sum_i a_i^2} \quad (5.18)$$

Avec $a_i = \sum_{j \in V(i)} r_j^3 w_{ij}$.

Nous avons donc une formulation pour le volume v_i de la particule i . Il nous reste maintenant à déterminer dans l'équation 5.11 les tenseurs $\boldsymbol{\epsilon}_i$ et $\boldsymbol{\sigma}_i$ qui sont dépendants du gradient du déplacement $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x}))$.

Nous avons implémenté deux tenseurs : les tenseurs de déformation linéaire et non linéaire de Green-Lagrange décrit aux équations 2.6 et 2.5 que nous reprenons ici pour des raisons de clarté :

$$\mathbf{E}_i = \frac{1}{2}(\nabla \mathbf{u}_i + \nabla \mathbf{u}_i^T + \nabla \mathbf{u}_i^T \cdot \nabla \mathbf{u}_i) \quad (5.19)$$

et

$$\boldsymbol{\epsilon}_i = \frac{1}{2}(\nabla \mathbf{u}_i + \nabla \mathbf{u}_i^T) \quad (5.20)$$

Puisque ces deux tenseurs sont symétriques, nous pouvons les écrire sous leur forme matricielle à partir de la formulation de Voigt :

$$\begin{bmatrix} \boldsymbol{\epsilon}_{00}^i \\ \boldsymbol{\epsilon}_{11}^i \\ \boldsymbol{\epsilon}_{22}^i \\ \boldsymbol{\epsilon}_{01}^i \\ \boldsymbol{\epsilon}_{02}^i \\ \boldsymbol{\epsilon}_{12}^i \end{bmatrix} = \begin{bmatrix} u_{i,x} \\ v_{i,y} \\ w_{i,z} \\ \frac{1}{2}(v_{i,x} + u_{i,y}) \\ \frac{1}{2}(w_{i,x} + u_{i,z}) \\ \frac{1}{2}(w_{i,y} + v_{i,x}) \end{bmatrix} \quad (5.21)$$

et

$$\begin{bmatrix} \mathbf{E}_{00}^i \\ \mathbf{E}_{11}^i \\ \mathbf{E}_{22}^i \\ \mathbf{E}_{01}^i \\ \mathbf{E}_{02}^i \\ \mathbf{E}_{12}^i \end{bmatrix} = \begin{bmatrix} u_{i,x} \\ v_{i,y} \\ w_{i,z} \\ \frac{1}{2}(v_{i,x} + u_{i,y}) \\ \frac{1}{2}(w_{i,x} + u_{i,z}) \\ \frac{1}{2}(w_{i,y} + v_{i,x}) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} u_{i,x}^2 + v_{i,x}^2 + w_{i,x}^2 \\ u_{i,y}^2 + v_{i,y}^2 + w_{i,y}^2 \\ u_{i,z}^2 + v_{i,z}^2 + w_{i,z}^2 \\ u_{i,x}u_{i,y} + v_{i,x}v_{i,y} + w_{i,x}w_{i,y} \\ u_{i,x}u_{i,z} + v_{i,x}v_{i,z} + w_{i,x}w_{i,z} \\ u_{i,y}u_{i,z} + v_{i,y}v_{i,z} + w_{i,y}w_{i,z} \end{bmatrix} \quad (5.22)$$

Dans la méthode d'animation basée sur les points, l'approximation de la déformation \mathbf{u}_i (et son tenseur gradient $\nabla \mathbf{u}_i$) d'une particule i est calculée localement dans son voisinage à partir du déplacement local entre i et ses particules voisines j :

$$\mathbf{u}_i = \sum_{j \in V(i)} \phi_j(\mathbf{x}_i)(\mathbf{u}_j - \mathbf{u}_i) \quad (5.23)$$

$$\nabla \mathbf{u}_i^T = \sum_{j \in V(i)} \nabla \phi_j(\mathbf{x}_i)(\mathbf{u}_j - \mathbf{u}_i)^T \quad (5.24)$$

Ici $\phi_j(\mathbf{x}_i)$ est appelée la fonction de forme de la particule j évaluée à la position de la particule

i et $\nabla \phi_j(\mathbf{x}_i) = \begin{pmatrix} \phi_{j,x} \\ \phi_{j,y} \\ \phi_{j,z} \end{pmatrix}$ est son gradient. Nous avons implémenté deux fonctions de forme ϕ_j qui seront décrites dans les deux prochaines sections.

Puisque nous utilisons un matériau linéaire et isotrope, la relation entre le tenseur de contrainte $\boldsymbol{\sigma}_i$ et le tenseur de déformation $\boldsymbol{\epsilon}_i$ (\mathbf{E}_i pour le tenseur non linéaire) découle directement de la loi de Hooke. Si le tenseur de déformation non linéaire \mathbf{E}_i est utilisé, nous disons que le matériau est un matériau de St-Venant-Kirchhof. Cette relation, appelée loi de comportement, peut alors être écrite ainsi :

$$\boldsymbol{\sigma}_i = \mathbf{C} : \boldsymbol{\epsilon}_i \quad (5.25)$$

où \mathbf{C} est un tenseur d'élasticité du quatrième ordre doublement symétrique, c'est-à-dire qu'il comporte au plus 21 composantes différentes parmi ses 81 valeurs possibles. En utilisant la convention de sommation d'Einstein sur les indices répétés, on exprime ce tenseur ainsi :

$$\mathbf{C}_{ijkl} = \lambda \mathbf{I}_{ij} \mathbf{I}_{kl} + \mu (\mathbf{I}_{ik} \mathbf{I}_{jl} + \mathbf{I}_{jk}) \quad (5.26)$$

Ici, les coefficients de Lamé λ et μ représentent les caractéristiques élastiques du matériau.

Grâce à la symétrie des tenseurs d'élasticité et de déformation, nous pouvons simplifier l'équation 5.25 à l'aide de la notation de Voigt sous la forme matricielle suivante :

$$\begin{bmatrix} \boldsymbol{\sigma}_{00} \\ \boldsymbol{\sigma}_{11} \\ \boldsymbol{\sigma}_{22} \\ \boldsymbol{\sigma}_{01} \\ \boldsymbol{\sigma}_{02} \\ \boldsymbol{\sigma}_{12} \end{bmatrix} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\epsilon}_{00} \\ \boldsymbol{\epsilon}_{11} \\ \boldsymbol{\epsilon}_{22} \\ 2\boldsymbol{\epsilon}_{01} \\ 2\boldsymbol{\epsilon}_{02} \\ 2\boldsymbol{\epsilon}_{12} \end{bmatrix} \quad (5.27)$$

Pour trouver la force qu'exerce une particule sur son voisinage, nous dérivons l'équation de la densité d'énergie de déformation (équation 5.11) par rapport au déplacement d'une particule voisine j :

$$\mathbf{f}_{i \rightarrow j} = -\nabla_{\mathbf{u}_j} U_i = -v_i \boldsymbol{\sigma}_i \nabla_{\mathbf{u}_j} \epsilon_i \quad (5.28)$$

où $\mathbf{f}_{i \rightarrow j}$ est la force que la particule i applique sur sa particule voisine j . La force appliquée sur i devient alors le négatif des forces appliquées sur son voisinage :

$$\mathbf{f}_i = \sum_j -\mathbf{f}_{i \rightarrow j} \quad (5.29)$$

Pour le petit tenseur de déformation linéaire, la force que la particule i applique sur sa particule voisine j devient :

$$\mathbf{f}_{i \rightarrow j} = -v_i \mathbf{B}_j^T \mathbf{C} \sum_{k \in V(i)} \mathbf{B}_k^T (\mathbf{u}_k - \mathbf{u}_i) \quad (5.30)$$

Avec

$$\mathbf{B}_j = \begin{bmatrix} \phi_{j,x} & 0 & 0 \\ 0 & \phi_{j,y} & 0 \\ 0 & 0 & \phi_{j,z} \\ \phi_{j,y} & \phi_{j,x} & 0 \\ 0 & \phi_{j,z} & \phi_{j,y} \\ \phi_{j,z} & 0 & \phi_{j,x} \end{bmatrix} \quad (5.31)$$

La fonction de forme ϕ_j de la particule j est évaluée à la position initiale \mathbf{x}_i de la particule i .

Pour le tenseur de déformation non linéaire, la force que la particule i applique sur sa particule voisine j devient :

$$\mathbf{f}_{i \rightarrow j} = -v_i \mathbf{J}_i \boldsymbol{\sigma}_i \begin{pmatrix} \phi_{j,x} \\ \phi_{j,y} \\ \phi_{j,z} \end{pmatrix} \quad (5.32)$$

Avec $\mathbf{J}_i = \nabla \mathbf{u}_i^T + \mathbf{I}$

La dernière étape consiste à définir une fonction de forme ϕ que nous utiliserons pour approximer localement le champ de déplacement $\mathbf{u}_i = \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix}$ d'une particule i ainsi que le tenseur

gradient de sa déformation $\nabla \mathbf{u}_i = \begin{bmatrix} u_{i,x} & u_{i,y} & u_{i,z} \\ v_{i,x} & v_{i,y} & v_{i,z} \\ w_{i,x} & w_{i,y} & w_{i,z} \end{bmatrix}$. Tel que mentionné précédemment, nous avons implémenté deux fonctions de formes couramment utilisées dans la littérature. Celles-ci sont décrites dans les deux prochaines sections.

5.3.1 Fonction de forme : approximation “Moving least square” (MLS)

Avec la méthode MLS, l’approximation des champs scalaires $u(\mathbf{x})$, $v(\mathbf{x})$ et $w(\mathbf{x})$ est effectuée de façon linéaire par en utilisant leur décomposition en série de Taylor dans le voisinage de \mathbf{x} (Müller et al. (2004)). On obtient alors pour u (le calcul est le même pour v et w) :

$$u(\mathbf{x}_i + \Delta \mathbf{x}) = u_i + \nabla u|_{\mathbf{x}_i} \cdot \Delta \mathbf{x} + O(\|\Delta \mathbf{x}\|^2) \quad (5.33)$$

$$\nabla u|_{\mathbf{x}_i} = \begin{pmatrix} u_{i,x} \\ u_{i,y} \\ u_{i,z} \end{pmatrix} \quad (5.34)$$

et on pose pour une particule j voisine de la particule i

$$\tilde{u}_j = u_i + (\mathbf{x}_j - \mathbf{x}_i)^T \nabla u|_{\mathbf{x}_i} \quad (5.35)$$

On minimise finalement l’erreur de l’approximation en effectuant la méthode des moindres carrés pondérés par le noyau W :

$$\nabla e|_{\mathbf{x}_i} = \nabla_{\mathbf{x}_i} \sum_j (\tilde{u}_j - u_j)^2 w_{ij} = 0 \quad (5.36)$$

Pour obtenir

$$\nabla_{\mathbf{x}_i} u = \left(\sum_j (\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T w_{ij} \right)^{-1} \sum_j (u_j - u_i)(\mathbf{x}_j - \mathbf{x}_i) w_{ij} \quad (5.37)$$

$$= \sum_j [\mathbf{A}^{-1} \mathbf{x}_{ij} w_{ij}] u_{ij} \quad (5.38)$$

$$= \sum_j [\nabla_{\mathbf{x}_i} \phi_j] u_{ij} \quad (5.39)$$

Ici \mathbf{A} est appelée la matrice de moment et $\nabla_{\mathbf{x}_i} \phi_j = [\phi_{j,x} \ \phi_{j,y} \ \phi_{j,z}]^T$ est la dérivée de la fonction de forme. Cette dernière peut être pré-inversée au début de la simulation grâce à la formulation lagrangienne utilisée.

Cette méthode impose cependant un voisinage d'au moins trois particules pour chaque point, et que celles-ci aient une configuration qui n'est ni coplanaire ni colinéaire, sans quoi la matrice \mathbf{A} sera singulière et l'inversion de cette dernière sera difficile. Pour éviter des problèmes numériques qui pourraient survenir si la matrice est mal conditionnée, nous l'inversons avec une décomposition SVD stable décrite dans Press et al. (1992). Il est à noter que cette méthode de décomposition n'est pas une solution à la condition de non-coplanarité et de non-co-linéarité du voisinage, elle minimise simplement l'impact d'une matrice mal conditionnée sur la stabilité de la simulation.

Avec cette approche, la valeur du gradient de déplacement $\nabla \mathbf{u}$ d'un point \mathbf{x}_i peut donc être évaluée à l'aide des particules voisines comme suit :

$$\nabla \mathbf{u}(\mathbf{x}_i) = \nabla \mathbf{u}_i = \begin{bmatrix} u_{i,x} & u_{i,y} & u_{i,z} \\ v_{i,x} & v_{i,y} & v_{i,z} \\ w_{i,x} & w_{i,y} & w_{i,z} \end{bmatrix} \quad (5.40)$$

$$\approx \sum_{j \in V(i)} \begin{bmatrix} \phi_{j,x}(u_j - u_i) & \phi_{j,y}(u_j - u_i) & \phi_{j,z}(u_j - u_i) \\ \phi_{j,x}(v_j - v_i) & \phi_{j,y}(v_j - v_i) & \phi_{j,z}(v_j - v_i) \\ \phi_{j,x}(w_j - w_i) & \phi_{j,y}(w_j - w_i) & \phi_{j,z}(w_j - w_i) \end{bmatrix} \quad (5.41)$$

$$= \sum_{j \in V(i)} \begin{pmatrix} u_j - u_i \\ v_j - v_i \\ w_j - w_i \end{pmatrix} [\phi_{j,x} \ \phi_{j,y} \ \phi_{j,z}] \quad (5.42)$$

$$= \sum_{j \in V(i)} \begin{pmatrix} u_j - u_i \\ v_j - v_i \\ w_j - w_i \end{pmatrix} [\mathbf{A}_i^{-1} \mathbf{x}_{ij}^0 w_{ij}^0]^T \quad (5.43)$$

Les dérivés de la fonction de forme ϕ sont donc

$$\begin{pmatrix} \phi_{j,x} \\ \phi_{j,y} \\ \phi_{j,z} \end{pmatrix} = [\mathbf{A}_i^{-1} \mathbf{x}_{ij}^0 w_{ij}^0] \quad (5.44)$$

5.3.2 Fonction de forme : approximation “Smooth Particle Hydrodynamics” (SPH)

Dans Solenthaler et al. (2007), les auteurs remédient à la limite d’un voisinage coplanaire ou colinéaire en utilisant une approximation du gradient de déplacement $\nabla \mathbf{u}(\mathbf{x}_i)$ via la méthode “Smooth Particle Hydrodynamics” (SPH).

Notre deuxième fonction d’approximation implémentée se base donc sur l’approche de Solenthaler et al. (2007) et reprend la formulation SPH introduite à l’équation 5.12, mais cette fois dans le but d’approcher le déplacement $\mathbf{u}_i = \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix}$ d’une particule dans son voisinage comme suit :

$$\mathbf{u}_i = \sum_{j \text{ t.q. } \mathbf{x}_j^0 \in V(\mathbf{x})} \frac{m_j^0}{\rho_j^0} w(\|\mathbf{x}^0 - \mathbf{x}_j\|^0, h) \begin{pmatrix} u_j - u_i \\ v_j - v_i \\ w_j - w_i \end{pmatrix} \quad (5.45)$$

$$= \sum_{j \in V(i)} v_i w_{ij}^0 (\mathbf{u}_j - \mathbf{u}_i) \quad (5.46)$$

$$= \sum_{j \in V(i)} \phi_j(\mathbf{u}_j - \mathbf{u}_i) \quad (5.47)$$

Un avantage important de cette méthode est que le tenseur gradient du déplacement est obtenu en dérivant le noyau utilisé dans la fonction de forme :

$$\nabla \mathbf{u}(\mathbf{x}_i) = \nabla \mathbf{u}_i = \begin{bmatrix} u_{i,x} & u_{i,y} & u_{i,z} \\ v_{i,x} & v_{i,y} & v_{i,z} \\ w_{i,x} & w_{i,y} & w_{i,z} \end{bmatrix} \quad (5.48)$$

$$\approx \sum_{j \in V(i)} \begin{bmatrix} \phi_{j,x}(u_j - u_i) & \phi_{j,y}(u_j - u_i) & \phi_{j,z}(u_j - u_i) \\ \phi_{j,x}(v_j - v_i) & \phi_{j,y}(v_j - v_i) & \phi_{j,z}(v_j - v_i) \\ \phi_{j,x}(w_j - w_i) & \phi_{j,y}(w_j - w_i) & \phi_{j,z}(w_j - w_i) \end{bmatrix} \quad (5.49)$$

$$= \sum_{j \in V(i)} \begin{pmatrix} u_j - u_i \\ v_j - v_i \\ w_j - w_i \end{pmatrix} [\phi_{j,x} \ \phi_{j,y} \ \phi_{j,z}] \quad (5.50)$$

$$= \sum_{j \in V(i)} \begin{pmatrix} u_j - u_i \\ v_j - v_i \\ w_j - w_i \end{pmatrix} \begin{pmatrix} v_i W_{ij,x}^0 \\ v_i W_{ij,y}^0 \\ v_i W_{ij,z}^0 \end{pmatrix}^T \quad (5.51)$$

Les dérivés de la fonction de forme ϕ sont donc

$$\begin{pmatrix} \phi_{j,x} \\ \phi_{j,y} \\ \phi_{j,z} \end{pmatrix} = \begin{pmatrix} v_i W_{ij,x}^0 \\ v_i W_{ij,y}^0 \\ v_i W_{ij,z}^0 \end{pmatrix} \quad (5.52)$$

5.4 Élasticité : méthode Galerkin-free

La deuxième approche sans maillage que nous avons implémentée pour résoudre la dynamique d'élasticité se réfère toujours à la formulation faible de l'équation de mouvement. Mais cette fois, l'objectif est de pallier l'erreur d'intégration et le faible degré de consistance de la méthode basée sur les points décrite à la section 5.3.

Cette deuxième méthode est décrite dans Horton et al. (2010) et est nommée "Meshless Total Lagrangian Explicit Dynamics" (MTLED). Cependant, nous l'avons légèrement modifiée pour y ajouter une intégration implicite dans le temps ainsi qu'une formulation corotationnelle. Mais voici d'abord une description de cette méthode.

Avec la méthode MTLED, la fonction de forme est appliquée sur le déplacement d'une particule voisine directement, plutôt que sur le déplacement du voisinage (la différence entre le déplacement d'un point et de sa voisine). Ainsi, l'approximation de la méthode basée sur les points du déplacement à un point \mathbf{x} du domaine initial

$$\mathbf{u}_i = \sum_{j \in V(i)} \phi_j(\mathbf{x}_i)(\mathbf{u}_j - \mathbf{u}_i) \quad (5.53)$$

$$\nabla \mathbf{u}_i^T = \sum_{j \in V(i)} \nabla \phi_j(\mathbf{x}_i)(\mathbf{u}_j - \mathbf{u}_i)^T \quad (5.54)$$

devient

$$\mathbf{u}(\mathbf{x}) = \sum_{j \in V(i)} \phi_j(\mathbf{x})\mathbf{u}_j \quad (5.55)$$

$$\nabla \mathbf{u}(\mathbf{x})^T = \sum_{j \in V(i)} \nabla \phi_j(\mathbf{x})\mathbf{u}_j^T \quad (5.56)$$

et se rapproche d'autant plus de l'approximation géométrique utilisée dans la méthode des éléments finis.

Ici, l'intégration géométrique de la densité d'énergie de déformation du domaine initial se fait en utilisant une quadrature de Gauss où le volume est discrétisé en éléments d'intégration et où chaque élément contient un ou plusieurs points d'intégration. Bien que cette intégration soit très semblable à la méthode des éléments finis, l'approximation du déplacement d'un point d'intégration se pose en fonction du déplacement des particules voisines qui, elles, sont indépendantes de la topologie d'intégration. Cette méthode est donc un entre-deux entre la méthode basée sur les points (MBP) et la méthode des éléments finis (FEM). En effet, alors que la méthode basée sur les points et la méthode des éléments finis utilisent respectivement une intégration et approximation sans et avec éléments, la méthode MTLED utilise une intégration sur des éléments et une approximation sans élément. Les différences entre les points d'intégration et les liens d'approximations de ces trois méthodes sont présentées dans la figure 5.4.

Dans la figure 5.4c, une grille d'intégration de fond ("background integration grid") est utilisée. Dans notre version de la méthode MTLED, nous utilisons plutôt un maillage d'intégration où les éléments sont les mêmes que ceux utilisés dans la méthode des éléments finis. Les points d'intégration sont donc identiques dans les deux méthodes. Cependant, le voisinage d'approximation n'est pas le même : la méthode des éléments finis utilise les sommets de l'élément entourant le point d'intégration, alors que la méthode MTLED utilise les particules dans un rayon d'influence autour du point d'intégration. Ainsi, un point d'intégration contenu à l'intérieur d'un élément peut être influencé par des particules à l'extérieur de l'élément.

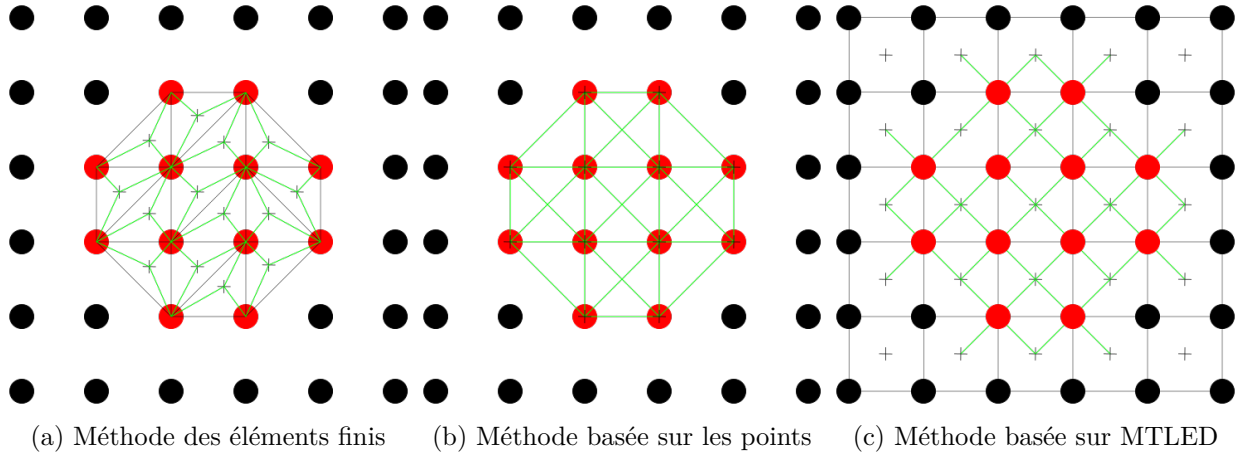


Figure 5.4 Différences d’approximation et d’intégration entre les méthodes implémentées. Les points rouges sont les degrés de liberté (particules et sommets FEM), les croix représentent les points d’intégration, les lignes vertes représentent les liens d’approximation entre un point d’intégration et un degré de liberté et les lignes noires représentent un maillage d’éléments.

Finalement, la masse (volume/densité) d’un point d’intégration est répartie également à tous ses points d’intégration.

La méthode d’intégration du modèle MTLED est basée sur des points de Gauss comme pour le modèle des éléments finis et est décrite en annexe A. La prochaine section présente la version approximative du modèle MTLED que nous avons développée.

5.4.1 Méthode approximative du modèle MTLED : approximation “Moving least square” (MLS)

Pour développer notre approximation, nous cherchons à nous rapprocher des fonctions scalaires des différentes composantes u , v et w du déplacement à un point \mathbf{x} quelconque du domaine, ainsi qu’à leurs dérivées par rapport à x , y et z . Pour ce faire, nous posons pour u (le développement est identique pour v et w) :

$$u^h(\mathbf{x}) = \sum_{i=0}^m p_i(\mathbf{x}) a_i(\mathbf{x}) = \mathbf{P}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) \quad (5.57)$$

où p_i est une fonction de base monomiale et a_i est son coefficient à déterminer. Nous utilisons les fonctions de base suivantes :

$$\mathbf{P}^T(\mathbf{x}) = [1|x \ y \ z|xy \ xz \ yz|x^2 \ y^2 \ z^2] \quad (5.58)$$

en choisissant m parmi 1, 4, 7 et 10 premiers termes en partant de la gauche. Par exemple, si $m = 4$, les fonctions de base sont 1, x , y et z . Théoriquement, des fonctions de bases plus complexes pourraient être utilisées pour mieux représenter certaines parties plus singulières de l'espace solution (semblable en théorie des éléments finis).

Pour trouver les coefficients $a_j(\mathbf{x})$, la méthode des moindres carrés pondérés par le noyau W est utilisée sur le voisinage de \mathbf{x} pour minimiser l'erreur de l'approximation :

$$\frac{\partial}{\partial \mathbf{a}} \left[\sum_{j \in V(\mathbf{x})} W(\|\mathbf{x} - \mathbf{x}_j\|, h) (u^h(\mathbf{x}) - u(\mathbf{x}))^2 \right] = 0 \quad (5.59)$$

où h est déterminée à l'aide d'une des trois méthodes décrites à la section 5.2.

Nous obtenons la formulation suivante pour les coefficients $\mathbf{a}(\mathbf{x})$ des fonctions de base de \mathbf{x} :

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x}) \sum_{j \in V(\mathbf{x})} W(\|\mathbf{x} - \mathbf{x}_j\|, h) \mathbf{P}(\mathbf{x}_j) u_j \quad (5.60)$$

où $\mathbf{A}(\mathbf{x}) = \sum_{j \in V(\mathbf{x})} W(\|\mathbf{x} - \mathbf{x}_j\|, h) \mathbf{P}(\mathbf{x}_j) \mathbf{P}^T(\mathbf{x}_j)$ est la matrice de moment de \mathbf{x} .

Finalement, en substituant 5.60 dans 5.57 nous obtenons :

$$u^h(\mathbf{x}) = \sum_{j \in V(\mathbf{x})} \mathbf{P}(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) W(\|\mathbf{x} - \mathbf{x}_j\|, h) \mathbf{P}(\mathbf{x}_j) u_j \quad (5.61)$$

$$= \sum_{j \in V(\mathbf{x})} \phi_j(\mathbf{x}) u_j \quad (5.62)$$

où $\phi_j(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) W(\|\mathbf{x} - \mathbf{x}_j\|, h) \mathbf{P}(\mathbf{x}_j)$ est la fonction de forme de la particule j évaluée en \mathbf{x} .

Puisque les tenseurs de déformation et de contraintes du point \mathbf{x} dépendent du tenseur gradient de déformation ($\nabla \mathbf{u}$), nous devons dériver la fonction de forme ϕ en x , y , et z . Nous écrivons ici la dérivée partielle de la composante u par rapport à x (la démarche est la même

pour les autres composantes) :

$$\phi_{i,x}(\mathbf{x}) = [\mathbf{P}_{,x} \mathbf{A}^{-1} W_j + \mathbf{P}(\mathbf{A}_{,x}^{-1} W_j + \mathbf{A}^{-1} W_{j,x})] \mathbf{P}_i \quad (5.63)$$

$$= [\mathbf{P}_{,x} \mathbf{A}^{-1} W_j + \mathbf{P}((\mathbf{A}^{-1} \mathbf{A}_{,x} \mathbf{A}^{-1}) W_j + \mathbf{A}^{-1} W_{j,x})] \mathbf{P}_i \quad (5.64)$$

$$= [\mathbf{P}_{,x} \mathbf{A}^{-1} W_j + \mathbf{P}((\mathbf{A}^{-1} \left[\sum_{j \in V(\mathbf{x})} W_{j,x} \mathbf{P}_i \mathbf{P}_i^T \right] \mathbf{A}^{-1}) W_j + \mathbf{A}^{-1} W_{j,x})] \mathbf{P}_i \quad (5.65)$$

Il nous reste donc simplement à calculer les dérivées de $\mathbf{P}(\mathbf{x})$ et $W(\|\mathbf{x} - \mathbf{x}_j\|, h)$.

La consistance de l'approximation dépend du degré des fonctions de base. D'abord, soit m le nombre de fonctions de base dans $\mathbf{P}(\mathbf{x})$ et n le nombre de particules dans le voisinage de \mathbf{x} , n doit être supérieur ou égal à m pour que l'inverse de la matrice de moment soit possible. Dans Horton et al. (2010), les auteurs estiment que n doit être environ le double de la quantité de fonctions de base, soit $n \approx 2m$. Le support d'un point d'intégration doit également contenir plus d'une particule, sans quoi celle-ci recevra une force trop grande et la simulation divergera.

Tout comme à la section 5.3, la force exercée sur une particule s'obtient en dérivant l'énergie de déformation du domaine en fonction du déplacement de cette dernière. Nous avons

$$U = \int_{\Omega^0} \frac{1}{2} (\boldsymbol{\epsilon} \cdot \boldsymbol{\sigma}) d\mathbf{X} \quad (5.66)$$

$$= \sum_{\mathbf{x}_I} \omega_I \frac{1}{2} (\boldsymbol{\epsilon}(\mathbf{x}_I) \cdot \boldsymbol{\sigma}(\mathbf{x}_I)) d\mathbf{X} \quad (5.67)$$

où ω_I est le poids du point d'intégration de Gauss I , pour tous les points d'intégration du domaine.

En dérivant par rapport au déplacement d'une particule i , nous obtenons la force appliquée celle-ci.

$$\mathbf{f}_i = \nabla_{\mathbf{u}_i} U \quad (5.68)$$

$$= \sum_{\mathbf{x}_I} \omega_I \boldsymbol{\sigma}_I \nabla_{\mathbf{u}_i} \boldsymbol{\epsilon}_I \quad (5.69)$$

$$= \sum_{\mathbf{x}_I} \mathbf{f}_{I \rightarrow i} \quad (5.70)$$

En substituant l'approximation de la section 5.4.1, la force $\mathbf{f}_{I \rightarrow i}$ appliquée sur la particule i par le point d'intégration I devient pour le petit tenseur de déformation linéaire

$$\mathbf{f}_{I \rightarrow i} = -\omega_I \mathbf{B}_i^T \mathbf{C} \sum_{j \in V(I)} \mathbf{B}_j \mathbf{u}_j \quad (5.71)$$

et

$$\mathbf{f}_{I \rightarrow i} = -\omega_I \mathbf{J}_I^T \mathbf{B}_i^T \mathbf{S}_I \quad (5.72)$$

pour le tenseur de déformation non linéaire de Green-Lagrange avec $\mathbf{J}_I = \nabla \mathbf{u}_I^T + \mathbf{I}$, $\mathbf{S}_I = \mathbf{C} \mathbf{E}_I$, \mathbf{C} le tenseur d'élasticité du quatrième ordre (voir équation 5.26) et \mathbf{E}_I le tenseur de déformation non linéaire sous sa forme de Voigt (voir équation 5.21) et

$$\mathbf{B}_j = \begin{bmatrix} \phi_{j,x} & 0 & 0 \\ 0 & \phi_{j,y} & 0 \\ 0 & 0 & \phi_{j,y} \\ \phi_{j,y} & \phi_{j,x} & 0 \\ 0 & \phi_{j,z} & \phi_{j,y} \\ \phi_{j,z} & 0 & \phi_{j,x} \end{bmatrix} \quad (5.73)$$

La matrice gradient de la fonction de forme de j évaluée en \mathbf{x}_I où $\phi_{j,x}$, $\phi_{j,y}$ et $\phi_{j,z}$ est obtenue une seule fois au début de la simulation à l'aide de l'équation 5.63.

5.5 Approche corotationnelle

Les méthodes MBP et MTLED comportent malheureusement certaines limites et plusieurs sources d'erreurs numériques liées à la consistance de la fonction de forme, au tenseur de déformation utilisé ainsi qu'à la loi de comportement utilisée.

Nous avons également vu dans la section 4.6 du chapitre des éléments finis une méthode pour séparer une transformation en deux composants, l'un représentant une rotation et l'autre un étirement du domaine. Cette procédure, dite approche corotationnelle, est reprise ici. Cependant une nouvelle stratégie doit être utilisée puisque contrairement à l'approche des éléments finis, nous n'avons plus accès à une représentation géométrique d'un élément entourant un point pour extraire la rotation.

Pour résoudre ce problème, nous utilisons la méthode corotationnelle de Becker et al. (2009) qui est basée sur l'approche corotationnelle qui avait été proposée initialement par Müller et al.

(2002) pour des éléments finis ainsi que la méthode de correspondance de forme proposée par Müller et al. (2005). Puisque les forces fantômes découlent des transformations sur une particule i (respectivement un point d'intégration I dans le cas de la méthode basée sur MTLED) comportant une rotation \mathbf{R}_i (respectivement \mathbf{R}_I), nous tentons d'obtenir le tenseur gradient de déformation sur le voisinage d'une particule dépourvue de cette rotation. Dans la méthode de correspondance de forme, une rotation d'un corps est extraite à partir d'une matrice de transformation représentant la distance entre un nuage de points et son centre de masse, de la configuration initiale à la configuration courante :

$$\mathbf{A} = \left(\sum_i m_i (\mathbf{x}_i - \mathbf{x}_{\text{cm}})(\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0)^T \right) \left(\sum_i m_i (\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0)(\mathbf{x}_i^0 - \mathbf{x}_{\text{cm}}^0)^T \right)^{-1} \quad (5.74)$$

où \mathbf{x}_{cm} est le centre de masse du nuage de points et \mathbf{A} est la matrice de transformation 3x3. La rotation est alors obtenue par une décomposition polaire de \mathbf{A} .

Dans notre cas, on s'intéresse à la rotation d'une particule (respectivement un point d'intégration) avec son voisinage. Le centre de masse de l'équation précédente est donc substitué par la position de la particule, et la somme se fait sur chaque voisin pondéré par le noyau :

$$\mathbf{A}_i = \sum_j m_j W_{ij}(\mathbf{x}_{ij})(\mathbf{x}_{ij}^0)^T \quad (5.75)$$

Pour la méthode basée sur MTLED, nous utilisons plutôt la fonction de forme pour pondérer la matrice de transformation :

$$\mathbf{A}_I = \sum_j \phi_j(\mathbf{x}_j - \mathbf{x}_I)(\mathbf{x}_j^0 - \mathbf{x}_I^0)^T \quad (5.76)$$

De plus, puisque nous n'avons pas accès à la position déformée \mathbf{x}_I du point d'intégration I , nous devons à chaque pas de temps l'approximer en fonction de son voisinage à l'aide de l'équation 5.57.

Dans le cas d'une configuration instable du voisinage, la méthode de décomposition SVD stable est utilisée. Ainsi, pour chaque particule i (respectivement point d'intégration I), une matrice de rotation \mathbf{R}_i (respectivement \mathbf{R}_I) est extraite au début de chaque pas de temps.

Pour la méthode basée sur les points, la rotation \mathbf{R}_i est retirée du calcul du déplacement d'une

particule avec son voisinage $(\mathbf{u}_j - \mathbf{u}_i)$. L'approximation du tenseur gradient de déplacement de i (équation 5.24) devient ainsi :

$$\nabla \mathbf{u}_i^T = \sum_{j \in V(i)} \nabla \phi_j(\mathbf{x}_i) ((\mathbf{R}_i^{-1} \mathbf{x}_j - \mathbf{x}_j^0) - (\mathbf{R}_i^{-1} \mathbf{x}_i - \mathbf{x}_i^0))^T \quad (5.77)$$

$$= \sum_{j \in V(i)} \nabla \phi_j(\mathbf{x}_i) (\tilde{\mathbf{u}}_j - \tilde{\mathbf{u}}_i)^T \quad (5.78)$$

La rotation est alors réappliquée aux forces qu'exercent la particule i sur son voisinage et l'équation 5.28 devient :

$$\mathbf{f}_{i \rightarrow j} = -\mathbf{R}_i \nabla_{\mathbf{u}_j} U_i = -v_i \mathbf{R}_i \boldsymbol{\sigma}_i \nabla_{\mathbf{u}_j} \boldsymbol{\epsilon}_i \quad (5.79)$$

Pour la méthode basée sur MTLED, l'approximation du tenseur gradient de déplacement de I devient :

$$\nabla \mathbf{u}_I^T = \sum_{j \in V(I)} \nabla \phi_j(\mathbf{x}_I) (\mathbf{R}_I^{-1} \mathbf{x}_j - \mathbf{x}_I^0)^T \quad (5.80)$$

$$= \sum_{j \in V(I)} \nabla \phi_j(\mathbf{x}_I) \tilde{\mathbf{u}}_j^T \quad (5.81)$$

Et la rotation est réappliquée à la force que reçoit la particule i par le point d'intégration I :

$$\mathbf{f}_{I \rightarrow i} = -\omega_I \mathbf{R}_I \boldsymbol{\sigma}_I \nabla_{\mathbf{u}_i} \boldsymbol{\epsilon}_I \quad (5.82)$$

5.6 Simulation dynamique et intégration temporelle

Dans le contexte des modèles sans maillage, l'intégration de l'équation de mouvement de Cauchy (équation 4.4) dans le temps que nous avons utilisée pour notre modèle FEM se fait de la même façon que décrite à la section 4.5. Cependant, pour l'intégration implicite, nous devons dériver l'équation de la force appliquée à une particule i (équation 5.28) par rapport au déplacement de ses voisins pour ainsi obtenir la matrice tangente de rigidité $\dot{\mathbf{K}}$ (équation 4.68). Nous obtenons $\dot{\mathbf{K}}_{ij} = \nabla_{\mathbf{u}_j} \mathbf{f}_i$ que nous interprétons comme étant la dépendance de la force appliquée sur la particule i lorsque sa voisine j se déplace.

Pour la méthode MBP, nous avons avec l'utilisation du petit tenseur de déformation ϵ :

$$\dot{K}_{ij} = \nabla_{u_j} f_i \quad (5.83)$$

$$= \nabla_{u_j} \sum_{k \in V(i)} -f_{i \rightarrow k} \quad (5.84)$$

$$= \nabla_{u_j} \sum_{k \in V(i)} - \left[-v_i \mathbf{B}_k^T \mathbf{C} \sum_{l \in V(i)} \mathbf{B}_l^T (\mathbf{u}_l - \mathbf{u}_i) \right] \quad (5.85)$$

$$= \sum_{k \in V(i)} v_i \mathbf{B}_k^T \mathbf{C} \mathbf{B}_j^T \quad (5.86)$$

Et, avec l'utilisation du tenseur de déformation non linéaire \mathbf{E} , nous avons :

$$\dot{K}_{ij} = \nabla_{u_j} f_i \quad (5.87)$$

$$= \nabla_{u_j} \sum_{k \in V(i)} -f_{i \rightarrow k} \quad (5.88)$$

$$= \nabla_{u_j} \sum_{k \in V(i)} v_i \mathbf{B}_k^T \mathbf{J}_i \mathbf{S}_i \quad (5.89)$$

$$= \sum_{k \in V(i)} v_i \mathbf{B}_k^T \left[\mathbf{J}_{i, u_j} \mathbf{S}_i + \mathbf{J}_i \mathbf{S}_{i, u_j} \right] \quad (5.90)$$

$$= \sum_{k \in V(i)} v_i \mathbf{B}_k^T \left[\mathbf{J}_{i, u_j} \mathbf{S}_i + \mathbf{J}_i \mathbf{C} (\mathbf{J}_{i, u_j} \mathbf{J}_i^T + \mathbf{J}_i \mathbf{J}_{i, u_j}^T) \right] \quad (5.91)$$

avec

$$\mathbf{J}_{i, u_j} = \begin{bmatrix} \phi_{j,x} & 0 & 0 \\ 0 & \phi_{j,y} & 0 \\ 0 & 0 & \phi_{j,y} \end{bmatrix} \quad (5.92)$$

Maintenant, pour la méthode MTLED, nous avons exactement le même calcul, seulement nous remplaçons

$$\dot{K}_{ij} = \nabla_{u_j} f_i \quad (5.93)$$

$$= \nabla_{u_j} \sum_{k \in V(i)} -f_{i \rightarrow k} \quad (5.94)$$

par

$$\dot{\mathbf{K}}_{ij} = \nabla_{\mathbf{u}_j} \mathbf{f}_i \quad (5.95)$$

$$= \nabla_{\mathbf{u}_j} \sum_I \mathbf{f}_{I \rightarrow i} \quad (5.96)$$

Pour I un point d'intégration tel quel que la particule i fait partie de son voisinage. La force $\mathbf{f}_{I \rightarrow i}$ est décrite aux équations 5.71 et 5.72.

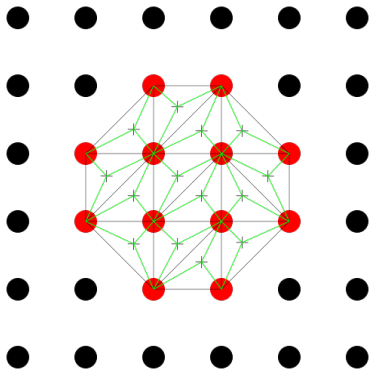
5.7 Synthèse

Dans les sections précédentes, deux méthodes sans maillage ont été abordées pour répondre aux questions suivantes :

- ❶ Comment intégrer la fonction de densité d'énergie de la déformation sur un domaine dont on ne connaît pas une définition géométrique explicite ?
- ❷ Comment obtenir le déplacement à n'importe quelle position du domaine ?

Méthode avec maillage

FEM



$$\text{❶} \quad \int F dx = \sum_e v_e \sum_I \omega_I f_e(\nabla u_I)$$

Quadrature de Gauss

$$\text{❷} \quad \nabla u_I = \sum_{j=1}^{\text{NbSommets}} \phi_{j,x}(\mathbf{x}_I) u_j$$

MLS

⊕ Bonne intégration volumique

⊕ Consistance linéaire et plus de l'approximation

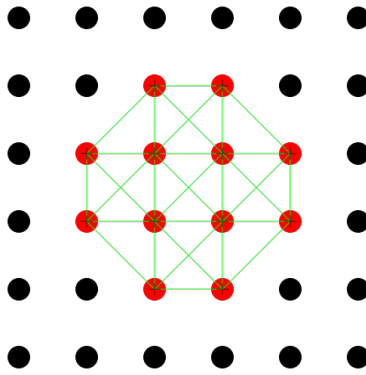
⊖ Maillage requis pour l'intégration

⊖ Maillage requis pour l'approximation

⊕ Nombre d'opérations fixe

Méthode sans maillage

MBP



$$\text{❶} \quad \int F dx = \sum_p v_p F(\nabla u_p)$$

SPH

$$\text{❷} \quad \nabla u_p = \sum_{j=1}^{\text{NbVoisins}} \phi_{j,x}(\mathbf{x}_I) u_j$$

SPH MLS

⊖ Intégration volumique faible

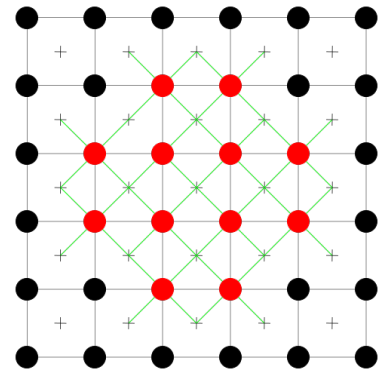
⊕ Consistance linéaire et moins de l'approximation

⊕ Aucun maillage requis pour l'intégration

⊕ Aucun maillage requis pour l'approximation

⊖ Nombre d'opérations proportionnel au nombre de voisins

MTLED



$$\text{❶} \quad \int F dx = \sum_e v_e \sum_I \omega_I f_e(\nabla u_I)$$

Quadrature de Gauss

$$\text{❷} \quad \nabla u_I = \sum_{j=1}^{\text{NbVoisins}} \phi_{j,x}(\mathbf{x}_I) u_j$$

MLS

⊕ Bonne intégration volumique

⊕ Consistance linéaire et plus de l'approximation

⊖ Maillage requis pour l'intégration

⊕ Aucun maillage requis pour l'approximation

⊖ Nombre d'opérations proportionnel au nombre de voisins

5.8 Algorithme final des méthodes MBP et MTLED

Définitions :

- $\mathbf{X}_P^t = [\mathbf{x}_{p_1} \ \mathbf{x}_{p_2} \dots \mathbf{x}_{p_n}]^T$ l'ensemble des positions des particules (DDLs) du système au temps t
 $\mathbf{X}_I = [\mathbf{x}_{I_1} \ \mathbf{x}_{I_2} \dots \mathbf{x}_{I_n}]^T$ l'ensemble des positions des points d'intégration du système
 $\boldsymbol{\omega}_I = [\omega_{I_1} \ \omega_{I_2} \dots \omega_{I_n}]^T$ l'ensemble des poids des points d'intégration du système
 $\mathbf{U}^t = [\mathbf{u}_1 \ \mathbf{u}_2 \dots \mathbf{u}_n]^T$ l'ensemble des déplacements \mathbf{u}_i d'un DDL i au temps t
 $\mathbf{V}_I = [\mathbf{x}_{p_1} \ \mathbf{x}_{p_2} \dots \mathbf{x}_{p_n}]^T$ l'ensemble des particules voisines au point d'intégration I
 $\mathbf{W}_I = [w_{I_1} \ w_{I_2} \dots w_{I_3}]^T$ l'ensemble des valeurs du noyau $w_{I_i} = w(\mathbf{x}_I - \mathbf{x}_{p_i}, h_I)$ pour le point d'intégration I
 $\mathbf{H}_I = [h_1 \ h_2 \dots h_3]^T$ l'ensemble des rayons d'influence du point d'intégration I
 $\boldsymbol{\Phi}_I = [\phi_{1,\mathbf{x}} \ \phi_{2,\mathbf{x}} \dots \phi_{n,\mathbf{x}}]^T$ l'ensemble des dérivées de la fonction de forme ϕ_i évaluée en \mathbf{X}_I
 $\mathbf{F}_{\text{int}} = [\mathbf{f}_1 \ \mathbf{f}_2 \dots \mathbf{f}_n]^T$ l'ensemble des forces internes des particules

Initialisation :

1. MBP : $\mathbf{X}_I \leftarrow \mathbf{X}_P$
2. MTLED : Pour chaque élément e ,
 - (a) $\mathbf{X}_I \leftarrow \mathbf{X}_I \cup \mathbf{X}_I^e$
 - (b) $\boldsymbol{\omega}_I \leftarrow \boldsymbol{\omega}_I \cup \boldsymbol{\omega}_I^e$ (les valeurs de $\boldsymbol{\omega}_I^e$ sont définies à l'annexe A)
3. Pour chaque point d'intégration $\mathbf{x}_I \in \mathbf{X}_I$
 - (a) $\mathbf{H}_I \leftarrow \mathbf{H}_I \cup h_I$ où h_I est trouvé avec l'équation 5.7, 5.8 ou 5.9
 - (b) Trouver n voisins \mathbf{V}_I de \mathbf{x}_I
 - (c) MBP : $\boldsymbol{\omega}_I \leftarrow \boldsymbol{\omega}_I \cup \frac{m_I}{\rho_I}$ où m_I est la masse et la densité ρ_I est trouvée avec l'équation 5.13
 - (d) Pour chaque particule voisine $\mathbf{x}_i \in \mathbf{V}_I$
 - i. $\mathbf{W}_I \leftarrow \mathbf{W}_I \cup w_{I_i}$ où w_{I_i} est trouvé avec l'équation 5.2, 5.3, 5.4, 5.5 ou 5.6
 - ii. $\boldsymbol{\Phi}_I \leftarrow \boldsymbol{\Phi}_I \cup \phi_{i,\mathbf{x}}$ où $\phi_{i,\mathbf{x}}$ est trouvé avec l'équation 5.44, 5.52 ou 5.63

Pour chaque pas de temps Δt :

1. Pour chaque point d'intégration $\mathbf{x}_I \in \mathbf{X}_I$
 - (a) $\nabla \mathbf{u}_I \leftarrow [\mathbf{0}^T \ \mathbf{0}^T \ \mathbf{0}^T]$
 - (b) $\mathbf{R}_I \leftarrow \mathbf{U} \mathbf{V}^T$ avec $\mathbf{A} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$ définie à l'équation 5.74
 - (c) Pour chaque particule voisine $\mathbf{x}_i \in \mathbf{V}_I$
 - i. MBP : $\nabla \mathbf{u}_I \leftarrow \nabla \mathbf{u}_I + \mathbf{R}_I^T (\mathbf{u}_i - \mathbf{u}_I) \phi_{i,\mathbf{x}}^T$
 - ii. MTLED : $\nabla \mathbf{u}_I \leftarrow \nabla \mathbf{u}_I + \mathbf{R}_I^T \mathbf{u}_i \phi_{i,\mathbf{x}}^T$
 - (d) $\mathbf{J}_I \leftarrow \mathbf{I} + \nabla \mathbf{u}_I^T$
 - (e) $\mathbf{S}_I \leftarrow \begin{cases} \mathbf{C}[\frac{1}{2}(\mathbf{J}_I + \mathbf{J}_I^T) - \mathbf{I}] & \text{tenseur de déformation linéaire} \\ \mathbf{C}[\frac{1}{2}(\mathbf{J}_I^T \mathbf{J}_I - \mathbf{I})] & \text{tenseur de déformation non linéaire} \end{cases}$
 - (f) Pour chaque particule voisine $\mathbf{x}_i \in \mathbf{V}_I$
 - i. $\mathbf{f}_i \leftarrow \begin{cases} -\omega_I \mathbf{R}_I \mathbf{S}_I \phi_{i,\mathbf{x}} & \text{tenseur de déformation linéaire} \\ -\omega_I \mathbf{R}_I \mathbf{J}_I \mathbf{S}_I \phi_{i,\mathbf{x}} & \text{tenseur de déformation non linéaire} \end{cases}$
 - ii. MBP : $\mathbf{f}_I \leftarrow (\mathbf{f}_I - \mathbf{f}_i)$
2. $\mathbf{U}^{t+1} \leftarrow$ équation 4.57, 4.60 ou 4.68
3. $\mathbf{X}_P^t \leftarrow \mathbf{X}_P^0 + \mathbf{U}^{t+1}$

CHAPITRE 6 RÉSULTATS ET DISCUSSION

Dans ce chapitre, nous présentons les résultats des expériences que nous avons menées sur les différentes méthodes avec maillage et sans maillage présentées dans les deux derniers chapitres. Nous avons en fait implémenté trois versions de ces méthodes dans notre code. La première est la méthode des éléments finis que nous dénoterons FEM. Pour la méthode sans maillage, nous avons implémenté une version basée sur les points (version MBP), donc complètement sans maillage. Finalement nous avons implémenté la méthode MTLED qui, comme nous l'avons vu au chapitre 5, est à mi-chemin avec l'approche des éléments finis en utilisant une intégration sur maillage, mais une approximation avec particules. À noter ici que l'approche des éléments finis est bien documentée et a fait l'objet de plusieurs analyses dans la littérature. Nous ferons appel à la méthode FEM seulement à titre comparatif pour évaluer l'approche sans maillage.

Notre évaluation portera essentiellement sur les différentes conditions présentées à la section 1.2 de l'introduction de ce mémoire, soit la **précision** des déformations, la **rapidité** des calculs et la **stabilité** de la simulation.

Toutes nos expériences seront lancées à partir du logiciel de simulation chirurgicale **Sofa Framework** (Allard et al. (2007)). Nos implémentations ont été insérées sous la forme de plugiciels. Nous avons utilisé un ordinateur équipé d'un processeur Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz, de 16 Go de mémoire vive à 1333 MHz et du système d'exploitation Linux Mint 17.1. La version de SOFA Framework utilisée est v16.12.0.

6.1 Analyse des modèles en mode statique

Nous débutons l'analyse de nos trois modèles de simulation dans un environnement de simulation statique. Pour notre première expérience, nous avons simulé l'application d'une force 100N sur les particules du bas d'un cube de $5 \times 5 \times 5 = 125$ degrés de liberté (donc divisé en $4 \times 4 \times 4$ hexaèdres). Dans cette expérience, les particules au haut du cube sont fixes. La figure 6.1 présente le cube dans son état initial (à gauche) et son état final après l'application de la force (à droite). Les déplacements \mathbf{u} qui ont été utilisés pour établir la position finale ont été obtenus en inversant la matrice de rigidité \mathbf{K} à l'aide d'une décomposition SVD.

Nous avons répété l'expérience de la figure 6.1 avec la méthode MTLED. Nous avons utilisé une fonction de forme de degré 1 (4 monômes, soit $\mathbf{P} = [1 \ x \ y \ z]^T$) pour cette simulation. La figure 6.2 présente les résultats de nos simulations pour différentes quantités de points

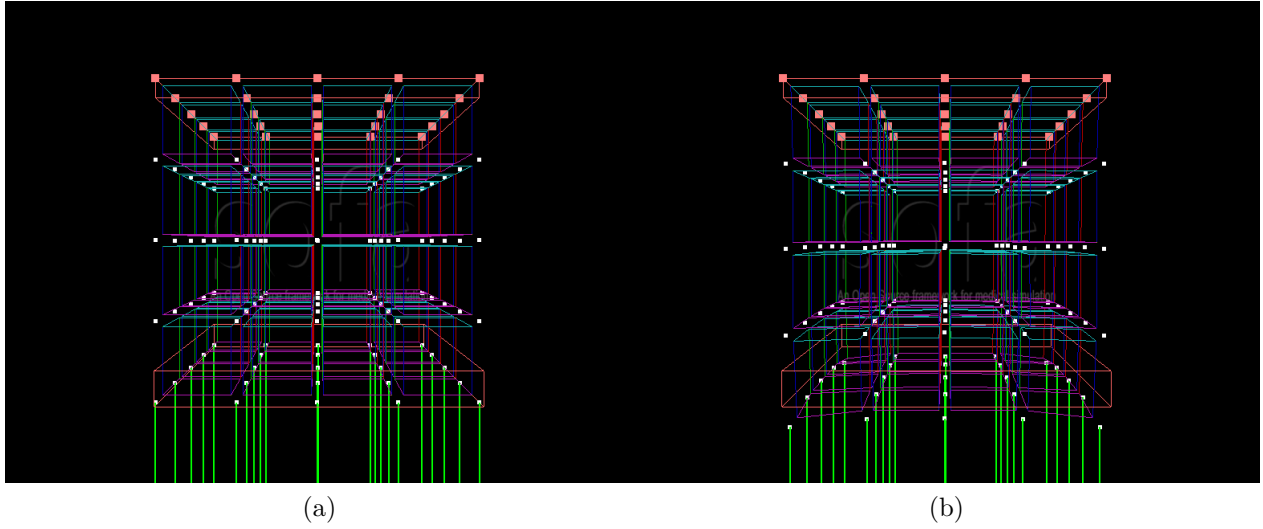


Figure 6.1 $4 \times 4 \times 4$ hexaèdres avec FEM : a) position non déformée, b) position déformée.

d'intégration. Nous avons également inclus en 6.2d la solution de référence obtenue avec la méthode FEM.

Cette expérience permet de mettre en lumière l'importance de la quantité de points d'intégration des méthodes Galerkin Free (section 5.4.1 du chapitre sur les méthodes sans maillage). Dans le cas de notre expérience, chaque point d'intégration possède 10 particules voisines dans sa région d'influence. Les illustrations démontrent clairement que plus on augmente le nombre de points d'intégration à l'intérieur des hexaèdres, plus le modèle MTLED converge vers la solution optimale. L'impact du nombre de points d'intégration sur la précision peut également se déduire en calculant la moyenne des distances entre les déplacements \mathbf{u} générés par nos méthodes et les déplacements de référence obtenus par une approche optimale, et ce sur les 25 points au bas de notre cube. Pour calculer les déplacements de référence, nous utilisons une simulation FEM sur une représentation de $9 \times 9 \times 9$ hexaèdres avec 1 pt intégration/hexaèdre. La distance moyenne obtenue pour chaque simulation est fournie à la deuxième colonne du tableau 6.1. Ces résultats confirment effectivement que plus le nombre de points d'intégration est grand, plus le modèle MTLED se rapproche du déplacement de référence. On constate également qu'il faut au moins 64 points d'intégration par hexaèdre pour obtenir une solution similaire à celle produite par la méthode des éléments finis.

Pour l'analyse de la rapidité et de la stabilité en mode statique, nous devons nous rapporter à la matrice de rigidité \mathbf{K} qui est l'élément clé dans le calcul des déplacements. Comme nous l'avons vu dans les chapitres précédents, le déplacement optimal des particules qui subissent des forces externes se calcule à partir de la résolution du système d'équations $\mathbf{K}\mathbf{u} = \mathbf{F}$ où

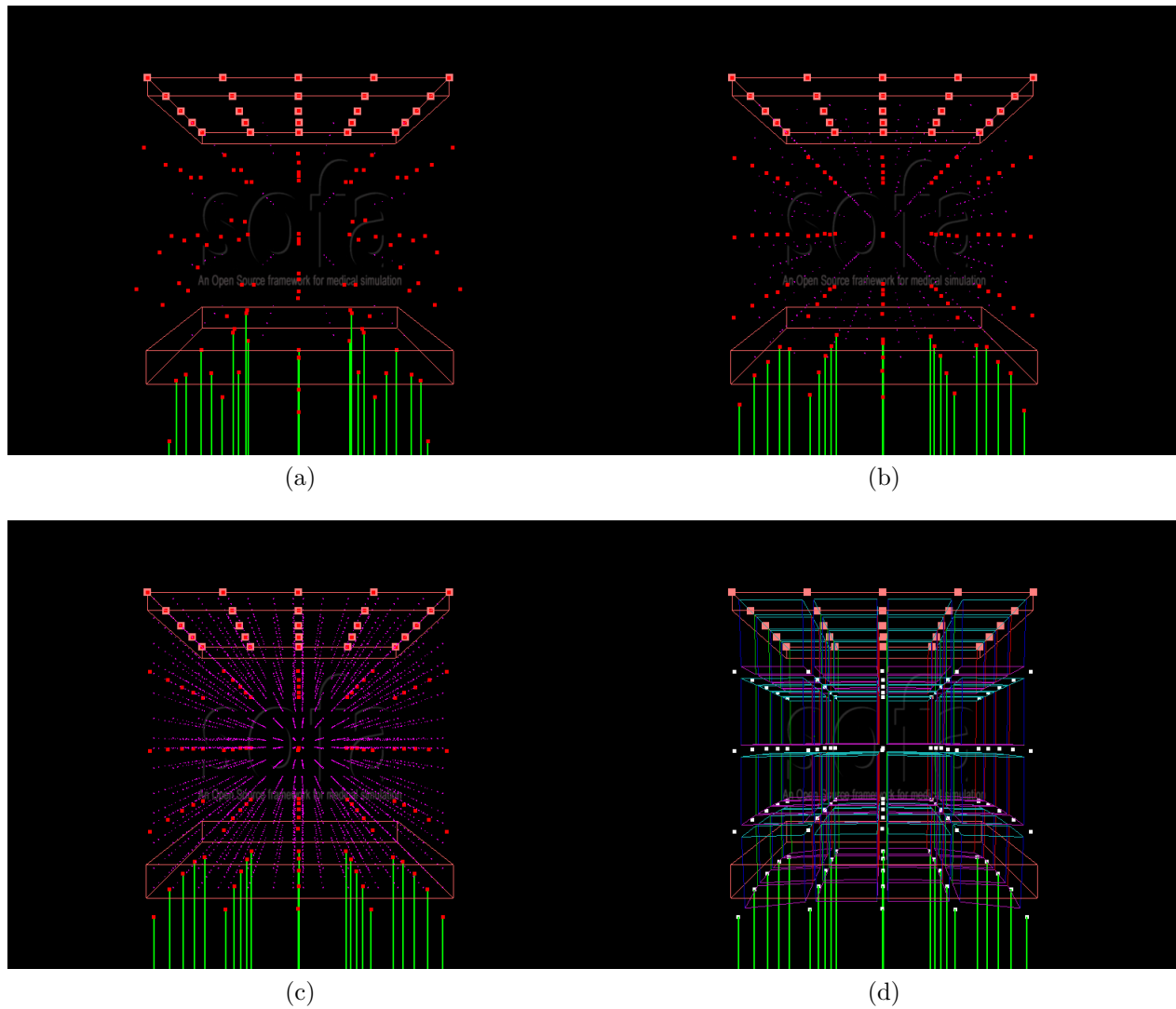


Figure 6.2 $4 \times 4 \times 4$ hexaèdres avec : a) 1 point d'intégration au milieu de chaque hexaèdre, b) 2 tétraèdres de 3 points d'intégration dans chaque hexaèdre, c) 64 points d'intégration (degré 4) d) FEM 1 point d'intégration

\mathbf{u} est le vecteur des déplacements \mathbf{u}_i de chaque degré de liberté (les sommets des éléments pour la méthode des éléments finis, les particules pour les méthodes sans maillage) et où \mathbf{F} est le vecteur des forces externes appliquées à ces derniers.

La structure de matrice de rigidité \mathbf{K} joue donc un rôle très important dans le processus de simulation. En effet, en multipliant la ligne \mathbf{K}_i par \mathbf{u} , on obtient les forces internes de la particule i . Ainsi, chaque sous-matrice \mathbf{K}_{ij} nous donne le lien entre la force interne de la particule i lors d'un déplacement d'une particule voisine j . Si une particule j n'est pas dans le voisinage de la particule i , sa fonction de forme évaluée en i sera nulle et par conséquent \mathbf{K}_{ij} sera égale 0. \mathbf{K} est donc une matrice creuse. Si le processus de numérotation a bien été fait, cette matrice sera bandée, c'est-à-dire que la ligne \mathbf{K}_i contiendra seulement l'influence de ses voisins autour de \mathbf{K}_{ii} , et 0 partout ailleurs. Évidemment, ceci influencera le processus d'inversion de \mathbf{K} et de multiplication au vecteur de force requis pour le calcul du déplacement.

La figure 6.3 plus basse illustre la structure de la matrice de rigidité \mathbf{K} pour la méthode MTLED avec différents points d'intégration. On peut constater que la largeur de la bande de la matrice augmente proportionnellement avec le nombre de points d'intégration. En effet, plus il y a de points d'intégration autour d'une particule i , plus les particules voisines à ces points d'intégration influenceront la force appliquée sur i , et donc leurs déplacements devront être représentés à la ligne i de la matrice \mathbf{K} . La largeur de la bande pour le premier scénario (1 point d'intégration au milieu de chaque hexaèdre) correspond à ce qui serait observé avec les méthodes FEM (la bande de la matrice générée par FEM serait légèrement plus mince puisque chaque point d'intégration à 8 voisins/hexaèdre, contrairement à 10 dans notre méthode MTLED).

Le tableau 6.1 présente trois mesures que nous avons produites lors de nos simulations et qui

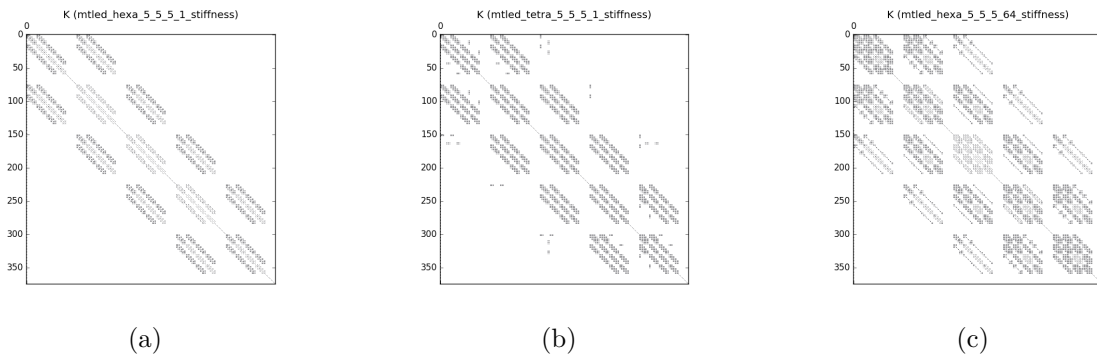


Figure 6.3 Matrices de rigidité \mathbf{K} : a) 1 point d'intégration au milieu de chaque hexa, b) 2 tétras de 3 pts dans chaque hexa, c) 64 points d'intégration (degré 4)

nous permettent de mieux apprécier l'importance de la matrice de rigidité dans la rapidité et stabilité des deux types de méthodes. Les troisième et quatrième colonnes donnent le temps requis pour construire et inverser la matrice. La dernière colonne fournit le conditionnement de cette matrice.

Comme on peut le voir dans ce tableau, la construction de la matrice avec la méthode MTLED utilisant un point d'intégration requiert au moins trois fois plus de temps. Au niveau du temps de calcul pour l'inversion de la matrice, les temps de calcul sont très similaires puisque nous utilisons l'inversion par la décomposition SVD qui ne tire pas avantage de la largeur de la bande de la matrice, le nombre d'opérations étant le même, peu importe l'allure de la matrice à inverser. On constate par contre que le plus grand désavantage de la méthode MTLED se situe au niveau du conditionnement de \mathbf{K} . Le très haut conditionnement obtenu avec un seul point d'intégration laisse présager une grande instabilité qui évidemment affectera la précision de la simulation. En effet, plus le conditionnement est haut, plus une petite perturbation dans le vecteur de forces externes entraînera de grands déplacements dans la solution.

Les résultats au niveau des temps de calcul sont également très préoccupants. Dans un environnement de simulation semi-statique avec un matériau linéaire, la matrice \mathbf{K} serait pré inversé au début de la simulation et nous n'aurions qu'à la multiplier à \mathbf{F} pour obtenir les déplacements. L'impact serait donc négligeable. Mais dans le cas d'un matériau non linéaire où un tenseur de déformation non linéaire \mathbf{E} est utilisé, ou bien lorsque des changements topologiques font partie de la simulation (coupes, déchirements, etc.), l'impact serait beaucoup plus significatif puisque la matrice devait être recalculée et inversée à chaque fois.

6.2 Simulation dynamique

Les prochaines expériences se feront dans un environnement de simulation dynamique où contrairement à la section précédente, la résolution du système d'équations pour calculer les déplacements se fait à partir de méthode d'intégration temporelle (méthode eulérienne). Nous avons choisi de comparer le comportement de nos modèles avec et sans maillage dans trois scénarios bien précis : l'étirement, le fléchissement et la collision.

6.2.1 Étirement

Pour ce premier scénario, notre expérience consiste à observer l'étirement d'un cube sous l'effet de la gravité. Nous avons fixé les DDLs du haut d'un parallélépipède de $10 \times 40 \times 10$ particules avec une densité (masse/volume) de 6.23, un module de Young de 2000 et un coefficient de poisson de 0.3. Nous avons d'abord comparé la performance de la méthode

Tableau 6.1 Différence des déplacements des DDLs au bas d'un cube avec un modèle FEM $10 \times 10 \times 10$ de référence. La première colonne indique la moyenne des différences entre les déplacements des 25 particules du bas et celles du modèle de référence. Les deux colonnes suivantes indiquent le temps de construction et d'inversion de \mathbf{K} . La dernière colonne indique le conditionnement de \mathbf{K} .

Modèle	Moy $\mathbf{u} - \mathbf{u}_{\text{ref}}$	Construction de \mathbf{K} (ms)	inversion de \mathbf{K} (ms)	cond(\mathbf{K})
MTLED $5 \times 5 \times 5$ 1 pt integration/hexa	0.4564	61	981	14.96×10^7
MTLED $5 \times 5 \times 5$ 6 pts integration/hexa	0.1402	101	903	10.10×10^3
MTLED $5 \times 5 \times 5$ 64 pts integration/hexa	0.1203	880	889	11.22×10^3
FEM $5 \times 5 \times 5$ 1 pt integration/hexa	0.1215	0	899	10.68×10^3
FEM $10 \times 10 \times 10$ 1 pt intégration/hexa	0.0000	3	1557628	56.60×10^2

basée sur les points (MBP) à notre méthode de référence (FEM). L'intégration dans le temps a été effectuée avec la méthode des différences centrées (équation 4.60) et un pas de temps de 0.005sec. Pour la fonction de forme du modèle MBP, nous avons utilisé l'approximation SPH. Les résultats des simulations sont illustrés à la figure 6.4.

Les résultats obtenus par cette expérience illustrent une des problématiques que nous avons observées avec la méthode MBP. Pour obtenir une bonne propagation des forces, nous devons augmenter le nombre de voisins. Dans notre exemple, on voit que l'utilisation de 10 voisins par particule se traduit par une très faible propagation des forces. Plus nous augmentons le nombre de voisins, plus la propagation des forces s'améliore. Par contre, ce gain se fait au détriment de la consistance de l'approximation. Nos expériences ont démontré qu'un voisinage d'environ 15 à 20 particules donne les meilleurs résultats.

Comme nous le voyons à la figure 6.5, le problème de consistance peut être corrigé en utilisant l'approximation MLS que nous avons présentée à la section 5.3.1. Avec cette approche, la fonction de forme a un degré de consistance de plus que l'approximation SPH. C'est ce qui explique pourquoi la simulation donne de bien meilleurs résultats. Toutefois, cette simulation n'aurait pas été possible avec des particules ayant un voisinage collinaire, coplanaire ou avec trop peu de voisins. On remarque aussi que même si la consistance est plus élevée, l'approximation peine toujours à approcher les déplacements d'un trop grand voisinage (voir en haut du parallépipède pour un voisinage de 20 et 30 particules de la figure 6.5).

Nous avons également répété notre expérience d'étirement linéaire avec deux versions de

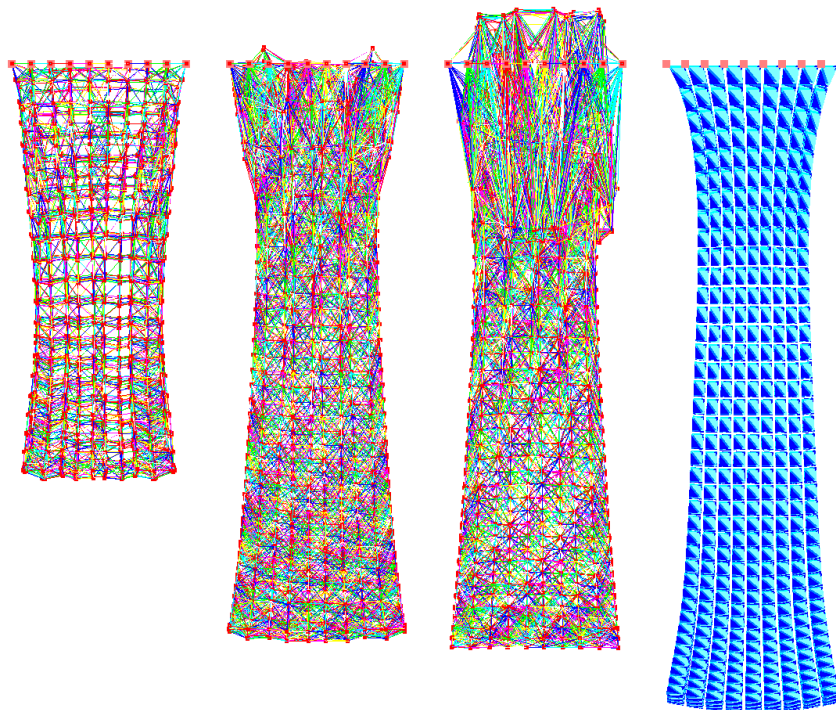


Figure 6.4 Étirement d'un parallélépipède de $10 \times 40 \times 10$ particules : de gauche à droite, l'approximation SPH avec respectivement 10, 20 et 30 voisins pour chaque particule. À droite complètement, la méthode FEM de référence.

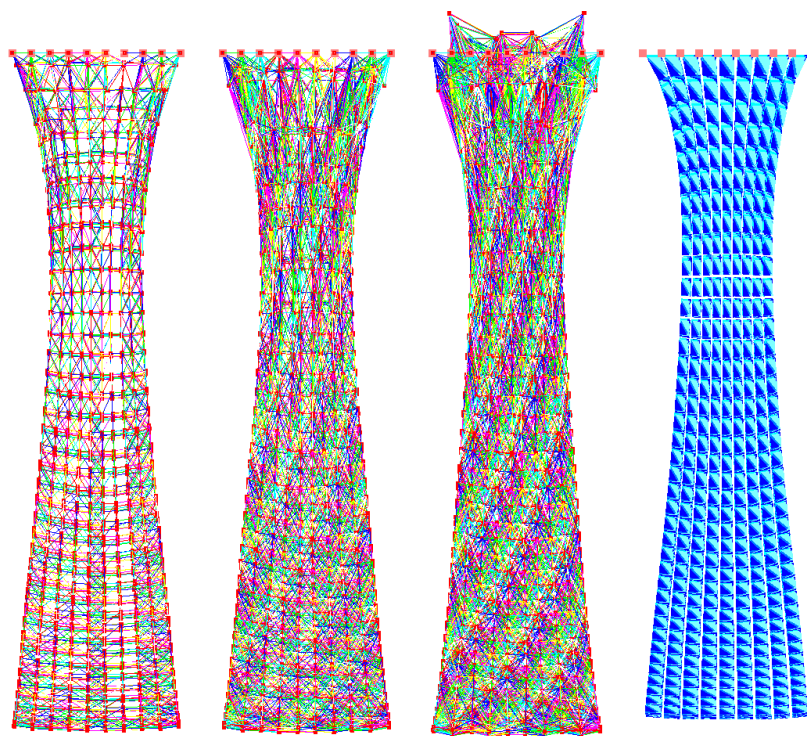


Figure 6.5 Étirement d'un parallélépipède de $10 \times 40 \times 10$ particules : de gauche à droite, l'approximation MLS avec respectivement 10, 20 et 30 voisins pour chaque particule. À droite complètement, la méthode FEM de référence.

la méthode MTLED (1 et 8 points d'intégration par hexaèdre) avec différentes quantités de voisins afin de comparer le temps d'exécution de nos trois méthodes. Les résultats sont présentés au tableau 6.2.

En observant ce tableau, on s'aperçoit rapidement que pour la grande majorité des expériences, la simulation temps réel n'aurait pas été possible. En effet, pour ce scénario, le pas de temps de 0.005 seconde était le plus grand Δt possible pour la méthode d'intégration explicite, sans quoi la simulation divergeait pour la plupart des paramètres utilisés. Donc seule la méthode MTLED avec 1 point d'intégration par hexaèdre peut atteindre cette contrainte de rapidité.

Pour la méthode d'intégration semi-implicite, la stabilité de la simulation est préservée pour un pas de temps plus petit que 0.1 seconde, soit 100 millisecondes. Or, pour atteindre la contrainte de vitesse d'un bon rendu visuel, la simulation doit atteindre 60 rendus par seconde. Ainsi, le maximum de temps d'exécution d'un rendu doit être de 17 millisecondes. On observe qu'aucune des méthodes, y compris FEM, n'arrive à atteindre cette contrainte de rapidité avec la méthode d'intégration semi-implicite.

6.2.2 Fléchissement

Pour ce segment, nous avons voulu analyser la performance de nos méthodes en présence de fléchissement. Pour ce faire, nous avons répété les expériences de la section précédente en changeant la direction de la gravité. Ceci nous a permis de créer un fléchissement de notre parallélépipède.

Lors de ces expériences, les problèmes liés avec la consistance des approximations observés lors de l'étirement linéaire sont également présents dans le cas des fléchissements. D'ailleurs, les erreurs dues à la mauvaise représentation des rotations sont encore plus importantes. Nous avons constaté que l'utilisation d'un tenseur de déformation non linéaire diminue grandement le phénomène des forces fantômes pour ce type de déformations. Toutefois, la faiblesse de la consistance de l'approximation reprend le dessus. Heureusement, l'utilisation de la méthode de corotation qui supprime temporairement les rotations juste avant le calcul des forces permet d'obtenir des résultats beaucoup plus acceptables. Ces résultats sont illustrés dans les deux prochaines figures.

À la figure 6.6, nous présentons les résultats des simulations effectuées avec le modèle MBP utilisant l'approximation SPH avec 3 quantités de voisins par particule différente. Comme dans nos simulations précédentes, les trois scénarios sont comparés à notre modèle de référence (méthode FEM). Les quatre simulations de gauche (a, d, g et i) ont été effectuées avec

Tableau 6.2 Temps d'exécution - Étirement linéaire

Modèle		Nb de voisins	Récup. des forces(ms)	Pas de temps (ms)		Gradient conjugué	
				Explicite	Implicite	nb iterations GC	Résolution GC (ms)
MBP	SPH	10	3.57	5.64	47.54	12	40.70
		20	6.52	8.47	101.40	13	87.97
		30	9.99	10.71	167.29	14	146.69
	MLS	10	3.26	5.81	50.16	13	43.44
		20	6.80	8.30	109.47	14	95.53
		30	10.48	11.98	183.20	15	161.30
MTLED	1 pt/hexa	10	2.19	2.94	37.63	29	33.92
		15	2.96	3.60	63.28	30	58.06
		20	3.53	4.22	56.18	20	49.88
	8 pts/hexa	10	6.35	6.61	41.21	9	32.86
		15	7.91	7.75	56.05	9	45.32
		20	10.06	10.50	84.05	19	69.93
FEM			1.55	2.34	45.44	34	42.39

Temps d'exécution pour l'expérience d'étirement vertical. Comparaison entre la méthode basée sur les points (MBP), la méthode basée sur MTLED et la méthode des éléments finis (FEM)

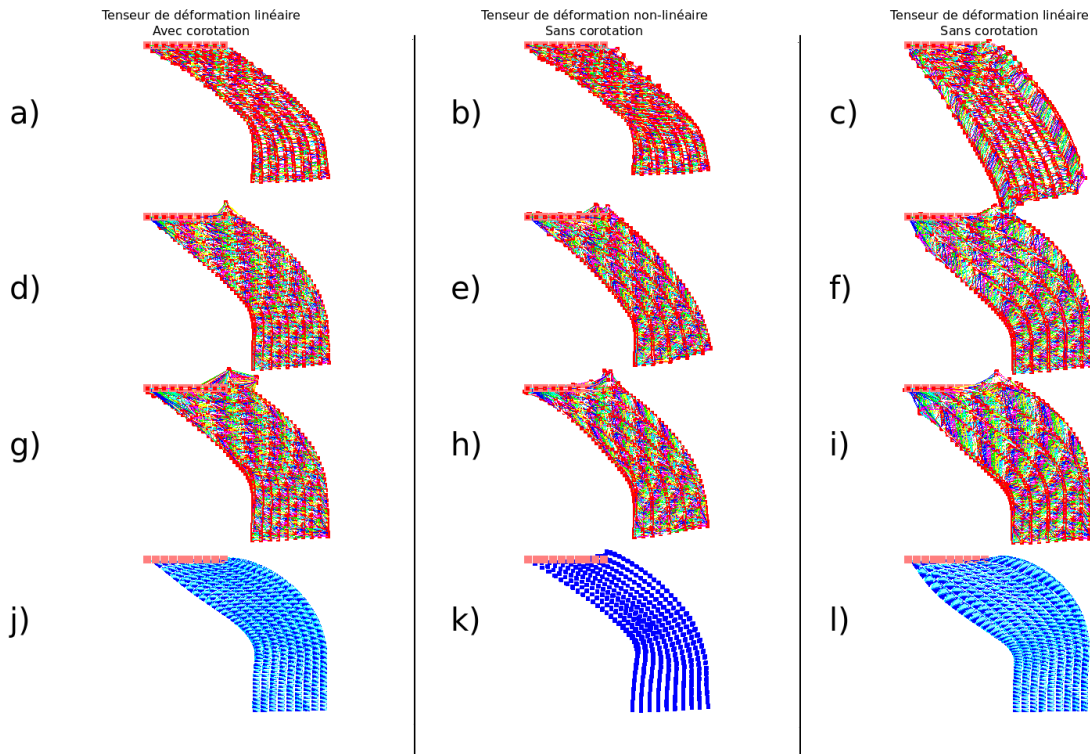


Figure 6.6 Fléchissement d'un parallélépipède de $10 \times 40 \times 10$ particules : comparaison entre tenseurs de déformation et corotation. a-c) SPH 10 voisins, d-e) SPH 20 voisins, g-i) SPH 30 voisins, j-l) FEM

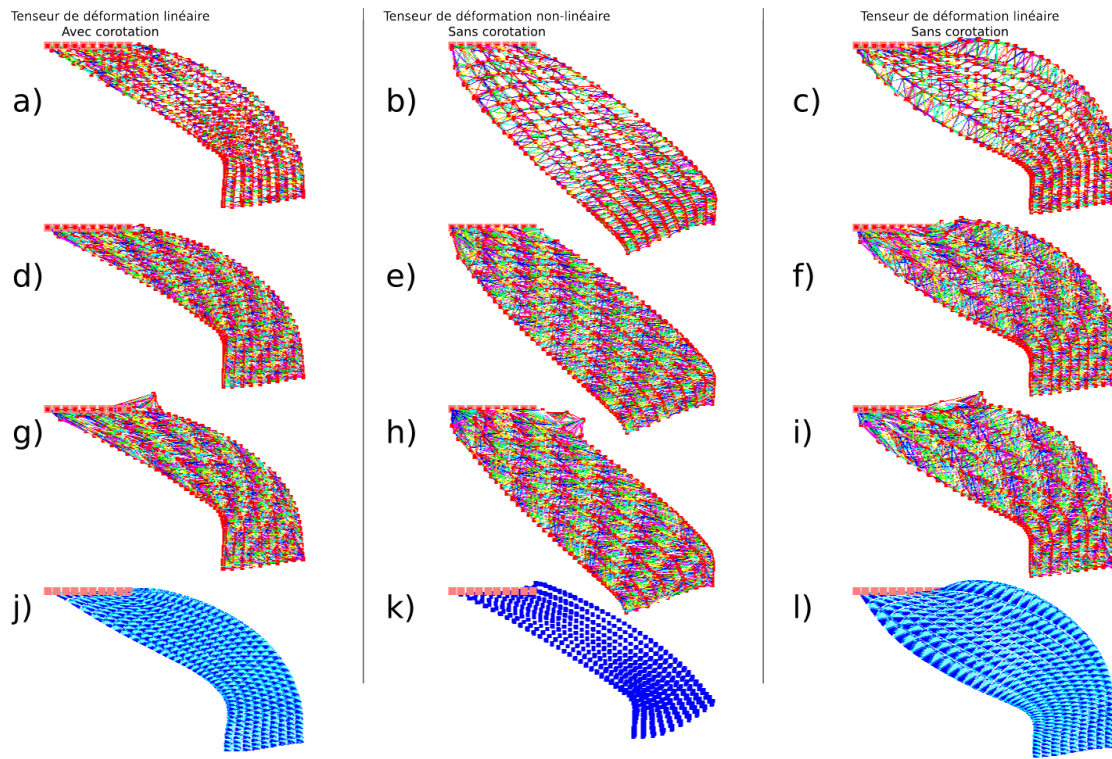


Figure 6.7 Fléchissement d'un parallélépipède de $10 \times 40 \times 10$ particules : comparaison entre tenseurs de déformation et corotation. a-c) MLS 10 voisins, d-e) MLS 20 voisins, g-i) MLS 30 voisins, j-l) FEM

corotation et un tenseur de déformation linéaire. Les quatre simulations de droite (c, f, i, l) utilisent également un tenseur linéaire, mais sans corotation. Finalement, simulations du milieu ont été produites avec un tenseur non linéaire et sans corotation. À la figure 6.7, les mêmes simulations ont été refaites, mais cette fois avec l'approximation MLS. Ces résultats illustrent très bien l'impact positif qui est engendré par l'utilisation de la corotation et des tenseurs non linéaires, et ce pour les deux types d'approximation.

Nous avons observé les mêmes résultats avec la méthode Galerkin-Free (modèle MTLED). Pour cette dernière, nous en avons profité pour mesurer l'effet du degré de consistance de l'approximation MLS avec des fonctions de bases de différent degré (voir la description des monômes à l'équation 5.57). Nous avons donc lancé les mêmes expériences avec un tenseur de déformation non linéaire. Les résultats sont présentés à la figure 6.8.

On remarque rapidement qu'un nombre trop faible de monômes sur une approximation qui dépend de beaucoup de particules voisines donne de mauvaises approximations. Nous avons déterminé qu'en plaçant une plus grande quantité de points d'intégration avec un nombre plus faible de voisins donne des résultats satisfaisants. Malheureusement, la rapidité d'exécution décroît proportionnellement au nombre de points d'intégration utilisé. Si un trop grand nombre de voisins est utilisé, la précision de notre simulation s'en trouvera affectée. L'impact se fait également sentir au regard de la stabilité (voir les illustrations g-i de la figure 6.8).

Ces résultats démontrent que, bien qu'il soit possible d'accélérer la simulation en diminuant le nombre de points d'intégration, nous devons augmenter le nombre de voisins pour pallier le manque de propagation des forces. Cette augmentation peut avoir des conséquences très néfastes sur la qualité de l'animation produite. Il est donc important de trouver le bon équilibre entre ces paramètres afin d'obtenir une simulation optimale. Cette contrainte est bien illustrée aux figures 6.7 et 6.8.

Les tableaux 6.3 et 6.4 présentent une comparaison du temps d'exécution des simulations avec les différents modèles. On peut y voir que la méthode corotationnelle est la plus lente

Tableau 6.3 Temps d'exécution - Étirement linéaire avec corotation

Modèle		Nb de voisins	Récup. des forces(ms)	Pas de temps (ms)		Gradient conjugué	
				Explicite	Implicite	nb iterations GC	Résolution GC (ms)
MBP	SPH	10	7.17	9.11	51.88	12	40.93
		20	12.71	12.83	108.16	13	88.36
		30	18.43	18.57	176.85	14	147.64
FEM			3.06	3.75	58	34	53.06

Temps d'exécution pour l'expérience de fléchissement linéaire avec corotation. Comparaison entre la méthode basée sur les points (MBP) et la méthode des éléments finis (FEM)

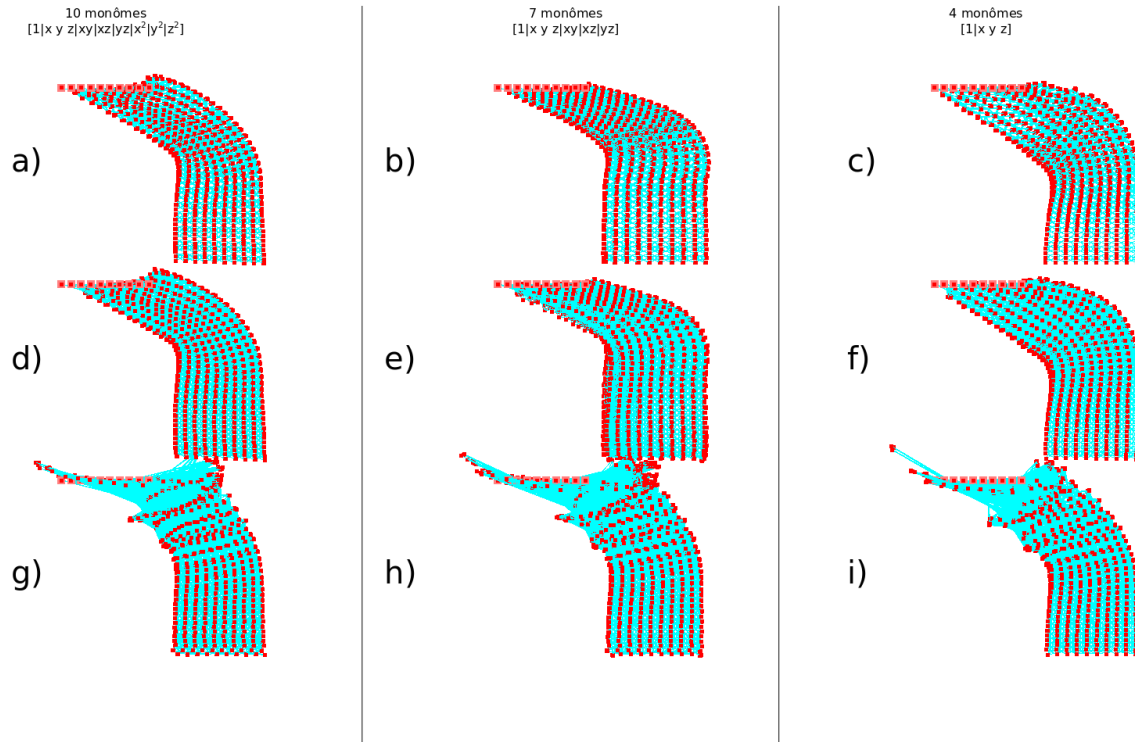


Figure 6.8 Fléchissement d'un parallélépipède de $10 \times 40 \times 10$ particules avec 3159 pts d'intégration : comparaison entre tenseurs de degré de consistance a-c) Galerkin-Free 10 voisins, d-f) Galerkin-Free 15 voisins, g-i) Galerkin-Free 20 voisins

Tableau 6.4 Temps d'exécution - Étirement non linéaire

Modèle		Nb de voisins	Récup. des forces(ms)	Pas de temps (ms)		Gradient conjugué	
				Explicite	Implicite	nb iterations CG	Résolution CG (ms)
MBP	SPH	10	2.55	3.97	50.15	12	44.03
		20	4.63	6.38	101.49	13	89.66
		30	7.68	8.79	167.99	14	149.50
FEM			2.51	2.82	29.70	41	10.03

Temps d'exécution pour l'expérience de fléchissement non linéaire. Comparaison entre la méthode basée sur les points (MBP) et la méthode des éléments finis (FEM)

des méthodes et que, sans surprise, une simulation en temps réel ne serait toujours pas possible avec ces paramètres. Cependant, le temps d'exécution pour l'intégration semi-implicite est sensiblement la même que la méthode linéaire sans corotation, ainsi il serait préférable d'utiliser cette approche si l'intégration semi-implicite est choisie.

6.2.3 Collision et intégration temporelle

Ce dernier segment porte sur la force de collision qui joue un rôle important pour l'intégration de l'objet haptique dans le processus de simulation chirurgicale. Nos expériences visent essentiellement à comparer les deux méthodes de pénalité que nous avons présentées à la section 4.7 de ce mémoire, soit la méthode du ressort et la méthode des multiplicateurs de Lagrange. Nous en profiterons également pour comparer les deux méthodes d'intégration temporelle (semi-implicite et explicite) dans le même contexte (simulation de collision). Pour simplifier notre analyse, toutes les expériences ont été effectuées à partir de l'approche des éléments finis (modèle FEM).

Nos expériences ont été effectuées à partir de deux cubes et d'une surface plane. L'un des deux cubes est placé un au-dessus de l'autre. Cette expérience engendre donc plusieurs collisions. La première survient lorsque l'un des deux cubes tombe sur l'autre. Les autres surviennent lorsque le cube du haut rebondit sur la surface plane. La figure 6.9 présente le résultat des simulations en utilisant la méthode des multiplicateurs de Lagrange (définie par l'équation 4.74). Les deux illustrations démontrent le comportement de cette méthode en fonction des deux techniques d'intégration : la méthode explicite (équation 4.5.1) et la méthode semi-implicite (4.5.2).

Les résultats à la figure 6.9 démontrent bien le manque de précision de la méthode d'intégration temporelle explicite. Bien qu'il soit difficile de visualiser l'animation complète avec quelques figures, on remarque tout de même que le cube qui tombe dans la ligne du haut effectue des déplacements beaucoup plus grands. Ce dernier effectue plusieurs tours sur lui-même, prenant beaucoup de temps pour se stabiliser complètement sur la surface plane. D'ailleurs, dans l'animation complète, nous observons les 8 coins du cube bouger très rapidement dans tous les sens, ce qui donne une impression visuelle de grande instabilité de la simulation. Pour la méthode d'intégration temporelle semi-implicite, les cubes de la ligne du bas sont beaucoup plus stables et les déplacements des DDLs beaucoup moins exagérés qu'à la ligne du haut. Le cube effectue un seul tour sur lui-même, puis se stabilise sur la surface plane rapidement. Les 8 coins du cube sont également beaucoup plus stables que dans la ligne du haut.

Nous avons repris l'expérience avec la technique d'intégration semi-implicite, mais cette fois

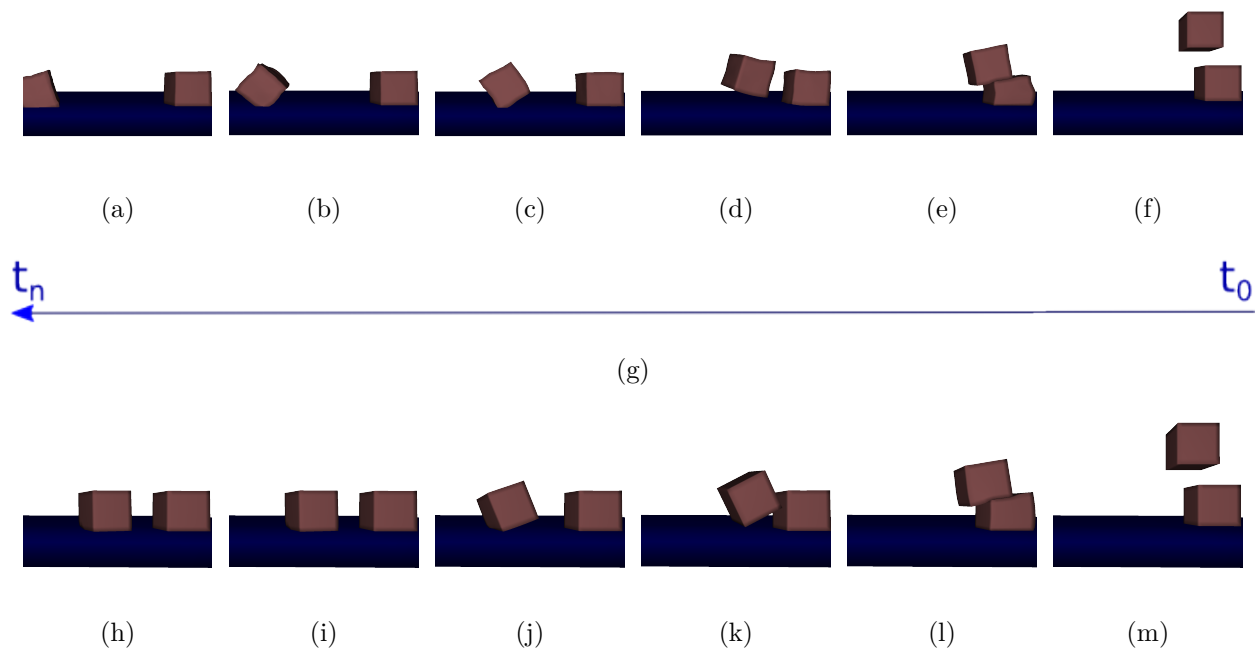


Figure 6.9 Collision par multiplicateurs de Lagrange : ligne du haut méthode d'intégration temporelle explicite, ligne du bas méthode d'intégration temporelle semi-implicite. Pas de temps de 0.001 seconde. Méthode FEM corotationnelle avec des cubes de $10 \times 10 \times 10$ DDLs, $\nu = 0.3$, $k = 2000$.

en changeant les méthodes de pénalités de contact ajoutées au système d'équations. Les résultats sont fournis à la figure 6.10. La ligne du haut montre les pénalités par la méthode des multiplicateurs de Lagrange (équation 4.74) et la ligne du bas par la méthode du ressort (équation 4.73).

La figure 6.10 démontre assez bien que la différence entre les deux techniques est assez négligeable. Cependant, la collision entre les deux cubes est un peu plus raide avec la méthode du ressort, le cube du haut se repliant beaucoup moins sur celui du bas. Cette méthode requiert d'ailleurs une distance de collision un peu plus grande et plusieurs tests pour déterminer le facteur de rigidité à appliquer. Sans ces ajustements, le cube du haut pénétrera largement dans le cube du bas avant que la force de ressort ne puisse le faire repousser vers le haut.

6.3 Bilan

Au cours de notre projet, nous avons réalisé plusieurs expériences sur les modèles de simulation sans maillage, et ce à l'intérieur de différents scénarios. Les résultats principaux observés lors de nos expériences ont été présentés dans les sections précédentes. Dans toutes nos expériences, nous avons remarqué rapidement que, même si les méthodes sans maillage peuvent donner des résultats proches de ceux des éléments finis, elles représentent plusieurs handicaps. D'abord, elles sont beaucoup plus lentes : pour le même nombre de points d'intégration, les méthodes sans maillage devront exécuter l'approximation du déplacement sur tout un voisinage de particules, contrairement à 4 ou 8 sommets pour la méthode des éléments finis. Ensuite, si le voisinage d'un point est trop grand, donc si l'approximation se fait sur trop de particules, la faible consistance des méthodes d'approximation produira des solutions instables et imprécises. De plus, si nous comparons la méthode MTLED avec la méthode MBP-MLS, pour un même degré de consistance de l'approximation, la faiblesse de l'intégration matérielle de la méthode MBP-MLS demandée par la formulation faible du système d'équations produira des résultats également instables et imprécis. Par contre, contrairement à la méthode MTLED, la méthode MBP-MLS reste entièrement sans maillage, alors que la méthode MTLED requiert un maillage d'intégration. Même si l'on trouve le bon nombre de points d'intégration avec une bonne méthode d'intégration et un voisinage adéquat au degré de consistance de l'approximation utilisé, une découpe dans l'objet simulé lors d'une simulation chirurgicale demanderait de réévaluer tous ces paramètres pour assurer la stabilité et la précision de la simulation. Ainsi, même si elles échappent en quelque sorte aux problèmes du remaillage inhérents à la méthode des éléments finis lors des changements topologiques, les méthodes sans maillage offrent très peu d'avantages au niveau d'une simulation chirurgicale en temps réel.

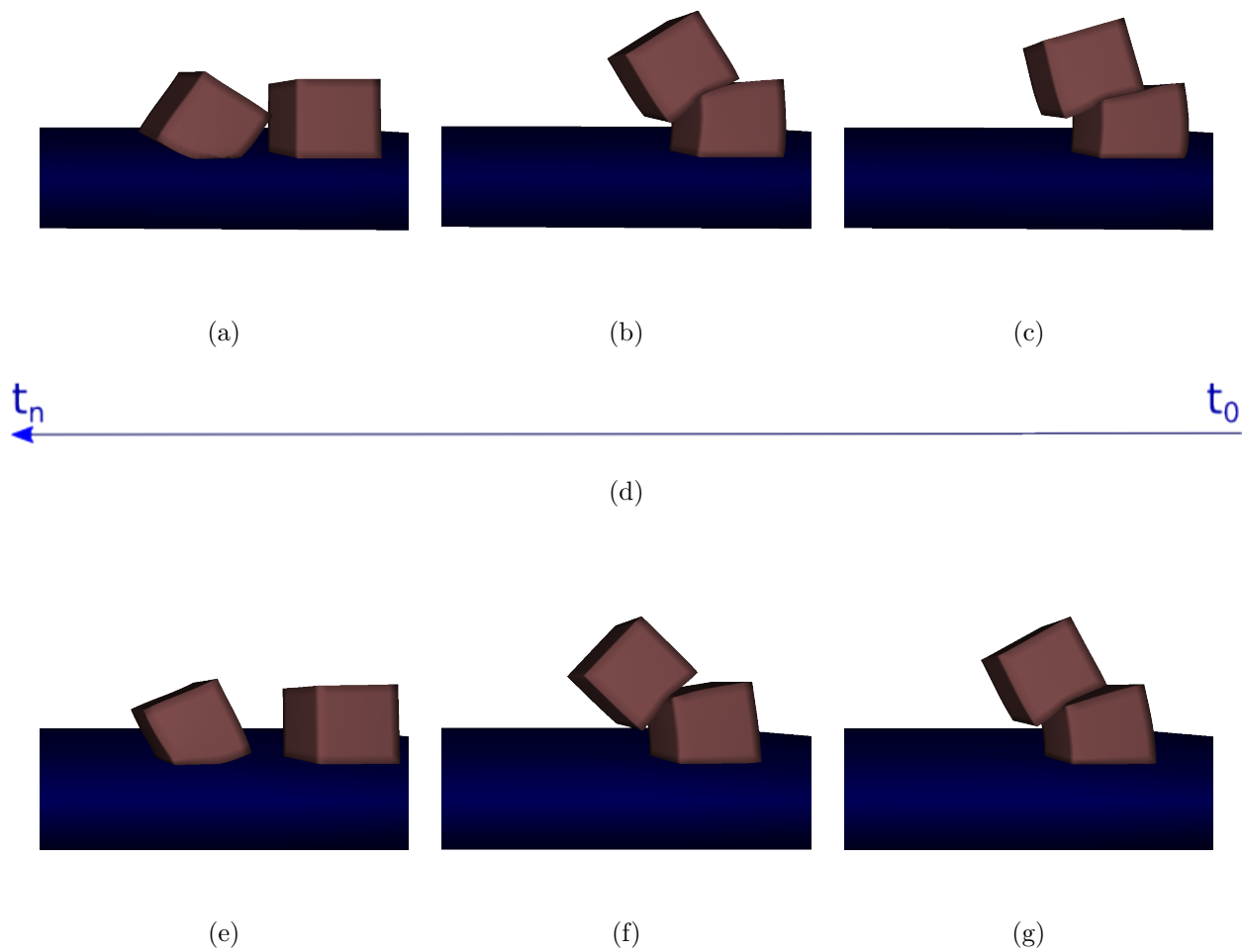


Figure 6.10 Collision par pénalités : ligne du haut pénalité par multiplicateurs de Lagrange, ligne du bas pénalité par ressort. Pas de temps de 0.001 seconde avec intégration semi-implicite. Méthode corotationnelle avec des cubes de $10 \times 10 \times 10$ DDLs, $\nu = 0.3$, $k = 2000$.

CHAPITRE 7 CONCLUSION

Pour aborder le problème de simulation chirurgicale en temps réel où la *rapidité* des calculs, la *précision* et la *stabilité* de la simulation sont des contraintes indispensables à la qualité d’entraînement d’un apprenti chirurgien, nous avons d’abord survolé les différentes techniques d’animation par ordinateur d’un corps déformable documenté dans la littérature. Nous avons vu que la méthode des éléments finis pour résoudre le système d’équations dirigeant l’équilibre entre les forces internes d’une déformation et les forces externes appliquées au modèle était bien adaptée à ce type d’animation et, *ipso facto*, largement utilisée dans des solutions commerciales. Cependant, cette méthode nécessite une discrétisation du modèle en une suite d’éléments géométriques connectés entre eux.

Pour remédier à la complexité souvent problématique de cette étape, nous avons voulu examiner si une autre méthode dépourvue de ce fardeau était envisageable pour le type de simulation en temps réel qui nous intéresse. Nous avons donc posé l’hypothèse que des méthodes de discrétisation sans maillage, et donc sans éléments, étaient suffisamment rapides, précises et stables pour combler les contraintes décrites en introduction.

7.1 Synthèse des travaux

Pour valider notre hypothèse, nous avons entamé une étude approfondie de deux méthodes de discrétisation sans maillage construites à partir d’une formulation faible des équations d’équilibre de la théorie de l’élasticité linéaire en mécanique des milieux continus. Nous avons ainsi abordé cette étude en nous fixant quatre grands objectifs.

En premier, nous avons récolté les différentes notions théoriques du modèle physique de l’élasticité utilisées dans des applications de simulation semblables à la nôtre. Puis, nous avons approfondi cette théorie en y ajoutant des notions utiles à notre application de manière détaillée et homogène, complétant ainsi le premier objectif de notre étude.

Ensuite, avant même d’amorcer les méthodes sans maillage, nous avons décidé d’étudier et d’implémenter une solution de référence fondée sur la méthode des éléments finis avec des éléments tétraédriques et hexaédriques linéaires. À l’aide de différents outils fournis par le cadre d’application “Sofa Framework”, nous avons posé un algorithme complet permettant d’obtenir les forces internes de déformation sur chaque sommet des éléments. Ces forces furent par la suite insérées dans des méthodes d’intégration temporelle du système d’équations d’équilibre du mouvement de notre application. Deux méthodes furent utilisées pour l’introduction

de forces externes qui pénalisent la pénétration de deux objets entrant en collision dans une simulation. Ceci a conclut notre second objectif.

Ayant établi un modèle de référence, nous avons entamé l'étude des méthodes sans maillage. Nous avons implémenté deux approches différentes. La première, la méthode basée sur les points (MBP), utilise une approximation du déplacement d'une particule basée sur un voisinage compact bâti autour de celle-ci à l'aide d'une fonction noyau. Pour intégrer la forme faible de l'équation d'équilibre sur le domaine non déformé, cette approche utilise le volume d'une particule décrit comme le rapport entre sa masse et la densité de son voisinage. Nous avons également formulé deux fonctions d'approximation du déplacement pour cette méthode : la fonction de forme "Smooth Particle Hydrodynamics" (SPH) et la fonction de forme "Moving least square" (MLS). La seconde approche, la méthode basée sur Meshless Total Lagrangian Dynamics (MTLED), utilise également une approximation du déplacement d'une particule basée sur un voisinage compact, mais cette fois en utilisant un maillage d'élément pour l'intégration de l'équation d'équilibre sur le domaine non déformé. Finalement, nous avons construit un algorithme complet pour chacune de ces deux approches, terminant ainsi notre troisième objectif.

Nous avons atteint le quatrième et dernier objectif de notre étude qui consistait à comparer la performance de nos méthodes sans maillage à celle de la méthode des éléments finis en fonction de nos critères de précision, rapidité et stabilité. Les différentes méthodes ont été comparées pour différents scénarios d'étirement et de fléchissement. Une analyse de la résolution statique du système d'équilibre et de la matrice de rigidité fut abordée. Les techniques d'intégration dans le temps et de gestion des forces de contact furent également testées. La rapidité des calculs observés dans ces scénarios nous ont permis de tirer un bilan général des méthodes sans maillage et par conséquent de notre projet de recherche.

7.2 Limitations de la solution proposée

Les résultats obtenus dans notre étude nous amènent à invalider notre hypothèse de départ, avec toutefois une certaine réserve. D'abord, il est évident que les solutions sans maillage implémentées dans nos expériences sont largement plus lentes que la solution de référence utilisant l'approche des éléments finis. Pour arriver à atteindre notre contrainte de rapidité, seules les méthodes d'intégration explicites dans le temps pouvaient être utilisées sur les objets de taille considérable, c'est-à-dire comportant quelques milliers de particules. D'ailleurs, même avec une méthode d'intégration explicite dans le temps, pour que la simulation ne soit pas trop lente, nous devons nous limiter à un nombre très restreint de voisins ce qui, comme nous l'avons observé dans nos expériences, entraîne une mauvaise propagation des forces. De

plus, si nous décidons de diminuer le nombre de particules de notre volume en augmentant la taille du voisinage, la faible consistance de nos fonctions d'approximation apporte de graves conséquences sur la précision et la stabilité de l'animation. Pour terminer, la difficulté que nous avons eue à bien paramétrer tous ces facteurs pour obtenir une simulation qui répond à nos critères de performance nous amène à douter de l'avantage des méthodes sans maillage sur la simulation de découpe. En effet, si nous avons autant de problèmes à stabiliser notre simulation avant même de changer le volume initial, les chances que ces difficultés surviennent après une découpe sont très grandes.

7.3 Améliorations futures

Malgré les résultats décevants que nous avons observés avec les modèles sans maillage, nous croyons que ces derniers offrent plusieurs caractéristiques qui méritent une étude plus poussée. Notamment, l'utilisation d'une intégration temporelle (explicite ou implicite) de l'équation d'équilibre laisse place à une très grande parallélisation des calculs sur processeurs graphiques (GPU). Une telle approche permettrait d'augmenter considérablement la rapidité des calculs. Également, nous avons concentré notre étude sur les méthodes de résolution des équations faibles, ajoutant ainsi le calcul d'une intégration matérielle dans notre solution. Or, il n'est pas rare de voir l'utilisation de l'approximation SPH dans la résolution de la formulation forte des équations de Navier Stokes pour l'animation de fluide. Il serait très intéressant d'observer les résultats d'une résolution de la formulation forte du problème d'élasticité avec une technique semblable, éliminant ainsi le besoin d'intégrer numériquement notre domaine.

Finalement, en sortant légèrement de notre cadre de simulation temps réel, nous pensons que les méthodes sans maillage auraient une place très intéressante dans le contexte de la planification chirurgicale. En effet, ces méthodes permettraient de bâtir un modèle déformable directement avec un nuage de points provenant des images à deux dimensions du corps d'un patient. De plus, les propriétés du matériau pourraient être directement appliquées sur les particules. Il ne serait donc pas nécessaire de passer par tout le processus de construction d'un maillage d'éléments. Puisque ce contexte ne requiert pas nécessairement une simulation en temps réel (des délais assez rapides pour permettre aux chirurgiens de bien planifier l'opération à venir entre la prise des images du patient et la date butoir de l'opération seraient suffisants), la branche des méthodes sans maillage propose une avenue très intéressante qui mérite d'être étudiée.

RÉFÉRENCES

- J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, et L. Grisoni, “Sofa-an open source framework for medical simulation”, dans *MMVR 15-Medicine Meets Virtual Reality*, vol. 125. IOP Press, 2007, pp. 13–18.
- D. Baraff et A. Witkin, “Large steps in cloth simulation”, dans *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, série SIGGRAPH ’98. New York, NY, USA : ACM, 1998, pp. 43–54. DOI : 10.1145/280814.280821
- A. H. Barr, “Global and local deformations of solid primitives”, dans *ACM Siggraph Computer Graphics*, vol. 18, no. 3. ACM, 1984, pp. 21–30.
- M. Becker, M. Ihmsen, et M. Teschner, “Corotated sph for deformable solids.” dans *NPH*. Citeseer, 2009, pp. 27–34.
- T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, et P. Krysl, “Meshless methods : an overview and recent developments”, *Computer methods in applied mechanics and engineering*, vol. 139, no. 1-4, pp. 3–47, 1996.
- J. Berkley, G. Turkiyyah, D. Berg, M. Ganter, et S. Weghorst, “Real-time finite element modeling for surgery simulation : An application to virtual suturing”, *IEEE Trans. Visual. Comput. Graphics*, vol. 10, no. 3, pp. 314–325, may 2004. DOI : 10.1109/tvcg.2004.1272730
- M. Bro-Nielsen et S. Cotin, “Real-time volumetric deformable models for surgery simulation using finite elements and condensation”, dans *Computer graphics forum*, vol. 15, no. 3. Wiley Online Library, 1996, pp. 57–66.
- K. Chinzei et K. Miller, “Compression of swine brain tissue; experiment in vivo”, dans *Proc. XVIth ISB Congress, Tokyo*. Citeseer, 1997.
- S. Cotin, H. Delingette, et N. Ayache, *Real time volumetric deformable models for surgery simulation*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1996, pp. 535–540. DOI : 10.1007/BFb0046995
- S. Cotin, H. Delingette, J.-M. Clément, V. Tasseti, J. Marescaux, et N. Ayache, “Geometric and physical representations for a simulator of hepatic surgery”, dans *Medecine*

Meets Virtual Reality IV, série Studies in Health Technology and Informatics, vol. 29. San Diego, USA, United States : IOS Press, 1996, pp. 139–151.

S. Cotin, H. Delingette, et N. Ayache, “Efficient linear elastic models of soft tissues for real-time surgery simulation”, INRIA, Rapp. tech. RR-3510, Oct. 1998.

Cotin, Stéphane and Delingette, Hervé and Ayache, Nicholas, “Real-time elastic deformations of soft tissues for surgery simulation”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999. DOI : 10.1109/2945.764872

H. Courtecuisse, H. Jung, J. Allard, C. Duriez, D. Y. Lee, et S. Cotin, “Gpu-based real-time soft tissue deformation with cutting and haptic feedback”, *Progress in Biophysics and Molecular Biology*, vol. 103, no. 2-3, pp. 159–168, Déc. 2010, special Issue on Biomechanical Modelling of Soft Tissue Motion. DOI : 10.1016/j.pbiomolbio.2010.09.016

S. A. Cover, N. F. Ezquerra, J. F. O’Brien, R. Rowe, T. Gadacz, et E. Palm, “Interactively deformable models for surgery simulation”, *IEEE Computer Graphics and Applications*, vol. 13, no. 6, pp. 68–75, 1993.

M. Desbrun et M.-P. Gascuel, “Smoothed particles : A new paradigm for animating highly deformable bodies”, dans *Computer Animation and Simulation’96*. Springer, 1996, pp. 61–76.

C. A. Felippa, “A systematic approach to the element-independent corotational dynamics of finite elements”, *Center for Aerospace Structures Document Number CU-CAS-00-03, College of Engineering, University of Colorado*, 2000.

E. M. Feudner, C. Engel, I. M. Neuhaus, K. Petermeier, K.-U. Bartz-Schmidt, et P. Szurman, “Virtual reality training improves wet-lab performance of capsulorhexis : results of a randomized, controlled study”, *Graefe’s Archive for Clinical and Experimental Ophthalmology*, vol. 247, no. 7, p. 955, 2009.

A. Fortin et A. Garon, “Les éléments finis : de la théorie à la pratique”, *Université Laval*, 2011.

T.-P. Fries et H. G. Matthies, “Classification and overview of meshfree methods”, *Department of Mathematics and Computer Science, Technical University of Braunschweig*, 2003.

S. Gibson, J. Samosky, A. Mor, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, S. Nakajima *et al.*, “Simulating arthroscopic knee surgery using

volumetric object representations, real-time volume rendering and haptic feedback”, dans *CVRMed-MRCAS'97*. Springer, 1997, pp. 367–378.

G. Haller, P. S. Myles, P. Taffé, T. V. Perneger, et C. L. Wu, “Rate of undesirable events at beginning of academic year : retrospective cohort study”, *Bmj*, vol. 339, p. b3974, 2009.

R. S. Haluck, R. L. Marshall, T. M. Krummel, et M. G. Melkonian, “Are surgery training programs ready for virtual reality? a survey of program directors in general surgery”, *Journal of the American College of Surgeons*, vol. 193, no. 6, pp. 660–665, 2001.

A. Hao et Z. Huang, “A physical based meshless method for soft tissue deforming”, dans *IT in Medicine and Education (ITME), 2011 International Symposium on*, vol. 2. IEEE, 2011, pp. 293–296.

A. Horton, A. Wittek, G. R. Joldes, et K. Miller, “A meshless total lagrangian explicit dynamics algorithm for surgical simulation”, *International Journal for Numerical Methods in Biomedical Engineering*, vol. 26, no. 8, pp. 977–998, 2010.

G. R. Joldes, A. Wittek, et K. Miller, “Real-time nonlinear finite element computations on gpu—application to neurosurgical simulation”, *Computer methods in applied mechanics and engineering*, vol. 199, no. 49, pp. 3305–3314, 2010.

P. Keast, “Moderate-degree tetrahedral quadrature formulas”, *Computer Methods in Applied Mechanics and Engineering*, vol. 55, no. 3, pp. 339–348, 1986.

R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutré, et M. Gross, “A unified lagrangian approach to solid-fluid animation”, dans *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics, 2005*. IEEE, 2005, pp. 125–148.

P. Lancaster et K. Salkauskas, “Surfaces generated by moving least squares methods”, *Mathematics of computation*, vol. 37, no. 155, pp. 141–158, 1981.

J. Lasseter, “Principles of traditional animation applied to 3d computer animation”, *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 35–44, Août 1987. DOI : 10.1145/37402.37407

M. C. Lin et M. Otaduy, *Haptic rendering : foundations, algorithms, and applications*. CRC Press, 2008.

LS-DYNA, “Lsdyna central differences”, <http://www.dynasupport.com/tutorial/ls-dyna-users-guide/time-integration>.

A. Luciani, S. Jimenez, O. Raoult, C. Cadoz, et J.-L. Florens, “An unified view of multitude behavior, flexibility, plasticity and fractures balls, bubbles and agglomerates”, dans *Modeling in Computer Graphics*. Springer, 1991, pp. 55–74.

J. Luo, S. Xu, et S. Jiang, “A novel hybrid rendering approach to soft tissue cutting in surgical simulation”, dans *Biomedical Engineering and Informatics (BMEI), 2015 8th International Conference on*. IEEE, 2015, pp. 270–274.

M. Müller, J. Dorsey, L. McMillan, R. Jagnow, et B. Cutler, “Stable real-time deformations”, dans *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2002, pp. 49–54.

M. Müller, D. Charypar, et M. Gross, “Particle-based fluid simulation for interactive applications”, dans *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, pp. 154–159.

M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, et M. Alexa, “Point based animation of elastic, plastic and melting objects”, dans *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2004, pp. 141–151.

M. Müller, B. Heidelberger, M. Teschner, et M. Gross, “Meshless deformations based on shape matching”, *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 471–478, 2005.

M. Nesme, Y. Payan, et F. Faure, “Efficient, physically plausible finite elements”, dans *Eurographics*, 2005.

OSSimTech, “Sim-ortho”.

Y. Peng, Y. Ma, Y. Wang, et J. Shan, “The application of interactive dynamic virtual surgical simulation visualization method”, *Multimedia Tools and Applications*, pp. 1–18, 2017.

G. Picinbono, H. Delingette, et N. Ayache, “Non-linear anisotropic elasticity for real-time surgery simulation”, *Graphical models*, vol. 65, no. 5, pp. 305–321, 2003.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, et B. P. Flannery, “Numerical recipes in c : The art of scientific computing (; cambridge”, 1992.

W. T. Reeves, “Particle systems—a technique for modeling a class of fuzzy objects”, *ACM Transactions on Graphics (TOG)*, vol. 2, no. 2, pp. 91–108, 1983.

- T. W. Sederberg et S. R. Parry, “Free-form deformation of solid geometric models”, *ACM SIGGRAPH computer graphics*, vol. 20, no. 4, pp. 151–160, 1986.
- M. Selvander et P. Åsman, “Virtual reality cataract surgery training : learning curves and concurrent validity”, *Acta ophthalmologica*, vol. 90, no. 5, pp. 412–417, 2012.
- N. E. Seymour, A. G. Gallagher, S. A. Roman, M. K. O’Brien, V. K. Bansal, D. K. Andersen, et R. M. Satava, “Virtual reality training improves operating room performance : results of a randomized, double-blinded study”, *Annals of surgery*, vol. 236, no. 4, pp. 458–464, 2002.
- P. Shrivastava et S. Das, “Particle coding for meshfree cutting of deformable assets”, dans *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*. ACM, 2014, p. 6.
- M. J. Sinclair, J. W. Peifer, R. Halebian, M. N. Luxenberg, K. Green, et D. S. Hull, “Computer-simulated eye surgery : a novel teaching method for residents and practitioners”, *Ophthalmology*, vol. 102, no. 3, pp. 517–521, 1995.
- B. Solenthaler, J. Schläfli, et R. Pajarola, “A unified particle model for fluid–solid interactions”, *Computer Animation and Virtual Worlds*, vol. 18, no. 1, pp. 69–82, 2007.
- G. Székely, C. Brechbühler, R. Hutter, A. Rhomberg, N. Ironmonger, et P. Schmid, “Modelling of soft tissue deformation for laparoscopic surgery simulation”, *Medical Image Analysis*, vol. 4, no. 1, pp. 57–66, 2000.
- R. Szeliski et D. Tonnesen, *Surface modeling with oriented particle systems*. ACM, 1992, vol. 26, no. 2.
- L. A. Taber, *Nonlinear theory of elasticity : applications in biomechanics*. World Scientific, 2004.
- D. Terzopoulos et K. Fleischer, “Modeling inelastic deformation : viscoelasticity, plasticity, fracture”, dans *ACM Siggraph Computer Graphics*, vol. 22, no. 4. ACM, 1988, pp. 269–278.
- D. Terzopoulos, J. Platt, A. Barr, et K. Fleischer, “Elastically deformable models”, dans *ACM Siggraph Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 205–214.
- M. Teschner, B. Heidelberger, M. Müller, D. Pomerantes, et M. H. Gross, “Optimized spatial hashing for collision detection of deformable objects.” dans *VMV*, vol. 3, 2003, pp.

47–54.

P. Wriggers, *Nonlinear finite element methods*. Springer Science & Business Media, 2008.

J. Yu, “Symmetric gaussian quadrature formulae for tetrahedronal regions”, *Computer Methods in Applied Mechanics and Engineering*, vol. 43, no. 3, pp. 349–353, 1984.

B. Zhu, L. Gu, X. Peng, et Z. Zhou, “A point-based simulation framework for minimally invasive surgery”, dans *International Symposium on Biomedical Simulation*. Springer, 2010, pp. 130–138.

ANNEXE A Intégration numérique et quadratures

Soit une discrétisation d'un espace 3D Ω en une suite d'espaces géométriques élémentaires $\Omega = \cup \Omega_e$. Nous décrivons dans cette annexe les règles de quadrature pour des éléments tétraédriques et hexaédriques.

Nous avons la règle de quadrature suivante :

$$\int \int \int_{\Omega_e} f(x, y, z) dx dy dz \approx v_e \sum_{i=1}^n \omega_i f(\xi_1^i, \xi_2^i, \xi_3^i, \xi_4^i) \quad (\text{A.1})$$

où v_e est le volume de l'élément, ω_i est le poids du point d'intégration i et ξ_j^i sont les coordonnées barycentriques du point d'intégration i à l'intérieur de l'élément pour n points d'intégration.

Le tableau A.1 présente le poids et les coordonnées barycentriques de n points d'intégration contenus dans un tétraèdre. Le facteur de multiplication M indique le nombre de permutations à effectuer pour les coordonnées barycentriques du point d'intégration présenté à cette ligne du tableau.

Pour l'hexaèdre, nous utilisons plutôt les coordonnées élémentaires ξ , η et ζ pour exprimer la position d'un point d'intégration dans l'élément. Notons qu'on peut passer facilement des coordonnées élémentaires aux coordonnées barycentriques, dans le cas de l'hexaèdre nous utilisons les coordonnées élémentaires pour simplifier le tableau. Nous avons alors

$$\int \int \int_{\Omega_e} f(x, y, z) dx dy dz \approx v_e \sum_{i=1}^n \omega_i f(\xi_i, \eta_i, \zeta_i) \quad (\text{A.2})$$

Les poids ω_i des n points d'intégration aux coordonnées élémentaires ξ_i , η_i et ζ_i d'un hexaèdre

Tableau A.1 Poids et coordonnées barycentriques pour n points d'intégration à l'intérieur d'un tétraèdre. (Yu (1984); Keast (1986))

n	ω_i	ξ_1^i	ξ_2^i	ξ_3^i	ξ_4^i	M
1	1	1/4	1/4	1/4	1/4	1
4	1/4	0.585410196624969	0.138196601125011	0.138196601125011	0.138196601125011	4
5	-4/5	1/4	1/4	1/4	1/4	1
	9/20	1/2	1/6	1/6	1/6	4
11	-0.01315556	1/4	1/4	1/4	1/4	1
	0.00762222	0.785714285714286	0.071428571428571	0.071428571428571	0.071428571428571	4
	0.02488889	0.399403576166799	0.100596423833201	0.399403576166799	0.100596423833201	6

sont présentés au tableau A.2.

Tableau A.2 Poids et coordonnées élémentaires pour n points d'intégration à l'intérieur d'un hexaèdre.

n	ω_i	ξ_i	η_i	ζ_i	M
1	1	$1/2$	$1/2$	$1/2$	1
8	$1/8$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
	$1/8$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	1
	$1/8$	$-1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
	$1/8$	$-1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	1
	$1/8$	$1/\sqrt{3}$	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1
	$1/8$	$1/\sqrt{3}$	$-1/\sqrt{3}$	$1/\sqrt{3}$	1
	$1/8$	$1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	1
	$1/8$	$1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	1

ANNEXE B Notation et termes utilisés

$\mathbf{X} = [X \ Y \ Z]^T$: majuscule pour coordonnée matérielle (volume non déformé)

$\mathbf{x} = [x \ y \ z]^T$: minuscule pour coordonnée spatiale (volume déformé)

$u_{,x} = \frac{\partial u}{\partial x}$: dérivée par rapport à x d'une fonction scalaire u

$\mathbf{u}_{,u} = \frac{\partial \mathbf{u}}{\partial u}$: dérivée par rapport à u d'une fonction vectorielle \mathbf{u}

$\nabla u(\mathbf{x}) = [u_{,x} \ u_{,y} \ u_{,z}]^T$: gradient d'une fonction scalaire

$\nabla \mathbf{u} = \begin{bmatrix} u_{,x} & u_{,y} & u_{,z} \\ v_{,x} & v_{,y} & v_{,z} \\ w_{,x} & w_{,y} & w_{,z} \end{bmatrix}$: gradient d'un vecteur \mathbf{u}

$\nabla \cdot \mathbf{u} = \frac{\partial u}{\partial u} + \frac{\partial v}{\partial u} + \frac{\partial w}{\partial u}$: divergence d'un vecteur \mathbf{u}

\dot{x} : dérivée première de x

\ddot{x} : dérivée seconde de x

$tr(\mathbf{A}) = \sum_i \mathbf{A}_{ii}$: trace d'une matrice carrée \mathbf{A}

$\boldsymbol{\sigma} : \boldsymbol{\epsilon} = \sum_{ij} \sigma_{ij} \epsilon_{ij}$: où $\boldsymbol{\sigma}$ et $\boldsymbol{\epsilon}$ sont des tenseurs de degré 2 et le résultat est un scalaire

$\mathbf{C} : \boldsymbol{\sigma} = \sum_{ij} C_{klij} \sigma_{ij}$: où \mathbf{C} est un tenseur du 4e ordre et $\boldsymbol{\sigma}$ est un tenseur du premier ordre.

Le résultat est un tenseur du deuxième ordre.

Particule, sommet ou degré de liberté : une position spécifique du domaine pour laquelle son déplacement est important pour la simulation (souvent une inconnue du système)